DISOCCLUSION FREE VIDEO INPAINTING APPLICATION TO OBJECT REMOVAL

Prof P. B. Sahane[1], Prof S. D. Kadam[2], Supriya Wadekar[3], Mahesh Yenare[4], Kishor Suryawanshi[5], Suresh Nayakawadi[6]

[1]Internal Guide, P K Technical Campus, Pune University, Maharashtra, India
[2]Internal Guide, P K Technical Campus, Pune University, Maharashtra, India
[3, 4, 5, 6] Student, P K Technical Campus, Pune University, Maharashtra, India
Email - kishorsuryawanshi569@gmail.com, wadekarsupriya2296@gmail.com, sureshnaykwadi@gmail.com, maheshyenare1995@gmail.com

Abstract: *Video inpainting is an important video enhancement technique used to facilitate the repair or editing of digital videos. The process of removing and reconstruction the specific Area in video known as video inpainting. It has been employed worldwide to transform cultural artifacts such as vintage videos/ films into digital formats. However, the quality of such videos is usually very poor and often contains unstable luminance and damaged content. In this project, we propose a video inpainting algorithm for repairing damaged content in digitized video films, focusing on automatic technique of video inpainting are computationally intensive and maintaining good spatio-temporal continuity. The proposed algorithm utilizes key techniques. Frame completion*

Key Words: *Image Inpainting, Depth Image based Rendering(DIBR), Disocclusion Filling, Frames Completion, Spatial, Temporal, Video Inpainting.*

1. INTRODUCTION:

Nowadays video inpainting, is a way of attracting a mind of researchers. Video Inpainting technique, in which we filling damaged or missing area in video sequence. The missing area is nothing but removal of one or more undesirable object in the scene. The region is inpainted is general: it may be still or moving, in the background for foreground. For example some object may not be wanted in a film shooting that unwanted object and we must have to remove it. It was not feasible to reshoot the drama because it was time consuming task and costly too, so to overcome this problem we are using video inpainting technique.

In system, user select the video for inpainting. Then system can extract the frames from video that contain unwanted objects. After frame extraction, System can select the frames automatically for inpainting from where frames are saved. System can automatically detect the unwanted object. System can select first frame from the selected frames as reference frame. By using the depth image based rendering, calculate the depth of the adjacent frame object and take it as reference frame for that next frame. Then object is removal easily. Masking of frames is also done system itself. System can rebuild the video and play the video. Video data is used in various fields.

Figure1. Before and after inpainting

Encoder-Driven Inpainting Strategy in Multiview Video Compression, where explicit instructions are transmitted minimally, and the decoder is left to independently recover remaining missing data via inpainting, resulting in lower coding overhead. Specifically, after pixels in a reference view are projected to a target view via depth image based rendering (DIBR) at the decoder. For blocks that are easy to inpaint, the decoder independently

completes missing pixels via template based inpainting. In free viewpoint video systems, color maps (RGB images) and depth maps (per-pixel distance between physical objects and capturing cameras) of the 3D scene as observed from multiple closely spaced cameras are captured at the encoder into a color plus-depth representation [4]. Armed with color and depth images of multiple views, images of intermediate virtual viewpoints can be synthesized at decoder via depth image based rendering (DIBR), enabling new applications such as free viewpoint TV, and immersive video conferencing.

In this technique Inpainting algorithm was proposed that can deal with removing the unwanted object in a video. Firstly the video converted into number of frames and then inpainting algorithm is applied to each frame by maintaining its spatial consistency and temporal continuity. Secondly includes inpainting of unwanted moving objects from the video which requires proper tracking of dynamic objects, for tracking RGB frames is converted into HSV (hue saturation value) format and then compared it with template matcher. With these two tasks tracking is done, after this applying inpainting algorithm to each frame the resultant video with desired removed unwanted object is obtained. This method automatically selects the parameter values and reduces the search region using region of interest (ROI).

Image Inpainting approach, Image Inpainting refers to the process of restoring missing or damaged areas in an image. This field of research has been very active over recent years, boosted by numerous applications: restoring image transmission, object removal in a context of editing or disoccclusion in image- based rendering (IBR) of viewpoints different from those captured by the cameras.

We propose to solve problem proceeding as follows: (i) finding a set of descriptors that encapsulate the information necessary to reconstruct a frame, (ii) finding an optimal estimate of the value of these descriptors for the missing corrupted frames, and (iii) using the estimated values to reconstruct the frames.

2. MOTIVATION:

Images and video are more popular for communication, entertainment and information retrieval purpose. We can build video in any occasion. When we capture the video, there is unwanted object is also captured. While removing that object inpainting technique is used. Firstly the image inpainting was used to remove the unwanted object. But problem is arising in video. So we develop a system for video inpainting for static and motion frames. We are motivated for video inpainting from image inpainting.

3. AIMS AND OBJECTIVE:

- To Remove Unwanted Object In Video.
- To use the super pixel for filling the hole those further improve the filling quality.
- To reduced complexity of exiting method.
- Nearest neighbor algorithm for select unwanted object & provides a good image or video quality.

4. SCOPE:

- Film restoration to reverse the deterioration cracks in video or scratches and dust spots in film.
- This technique can be used to replace the lost data in the coding and transmission of videos, for example, in a streaming video.
- It can also be used to remove logos in videos.
- Military can use the system in bad weather for finding enemy or solider.
- Investigation purpose, we use video inpainting

5. IMAGE AND VIDEO INPAINTING: RELATED WORK:

Inpainting is the process of reconstructing lost or deteriorated parts of image and video. Inpainting techniques are applied to object removal, text removal and other automatic modifications of images and videos.

A. Image Inpainting

Image inpainting means removing defect from paintings and photographs. Image inpainting refers to restoration methods used to remove damage or unwanted objects from an image in a natural manner. The result is same as original image. In the past, this has been addressed by two classes of algorithms: (i) "texture synthesis" for generating large image regions from sample textures, and (ii) "inpainting" techniques for filling in small image gaps

B. Video Inpainting

The process to remove the specific areas or repairing the damaged area in a video is known as video inpainting. It has been used in world to convert damaged or missing videos into proper video.

Video inpainting algorithm based on two categories-
1. Object based
2. Patched based

Especially, object removal is one of the key techniques in video inpainting. If an unwanted object is captured in a video sequences accident or if the object is decided to be unnecessary, we need to erase the object from the sequences. Therefore, video inpainting algorithms are used to remove the object and replace it appropriate object.

6. SYSTEM ARCHITECTURE:

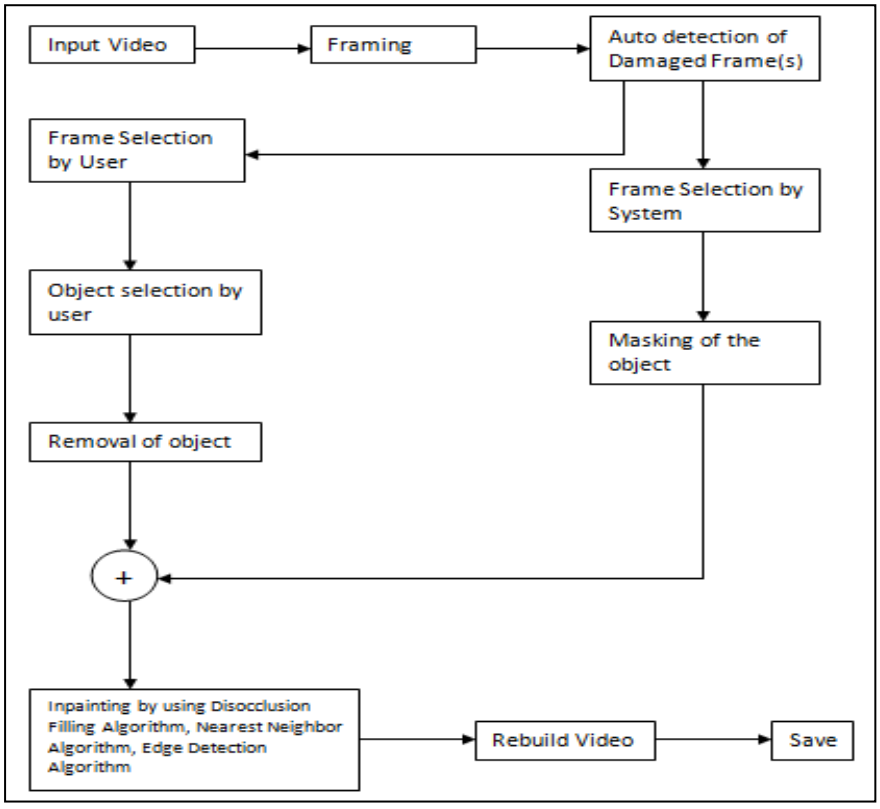


Fig.2 System Architecture

7. Algorithm:

7.1 K-nearest-neighbour algorithm:

We use Knn algorithm to finding the nearest pixel for filling the holes.
The concepts of Knn explain below.
K-nearest-neighbor algorithm Introduction:
The K-nearest-neighbor (KNN) algorithm measures the distance between a query scenarios using data set.
Distances
We can compute the distance between two scenarios
Using some distance function d(x, y).
Where,
x= f x1, x2 ...xn g
y= f y1, y2 ...yn g
Two distance functions are discussed in this summary:

1) Absolute distance measuring:

$$d_A(x,y) = \sum_{i=1}^{N} |x_i - y_i|$$

2) Euclidean distance measuring:

$$d_E(x,y) = \sum_{i=1}^{N} \sqrt{x_i^2 - y_i^2}$$

Because the distance between two scenarios is dependent of the intervals. This can be accomplished by replacing the scalars x, y with x0, y0 according to the following function:
Where,

$$x' = \frac{x - \bar{x}}{\sigma(x)}$$

Where,
x is the unscaled value, x is the arithmetic mean of feature across the data set, s(x) is its standard deviation, x0 is the resulting scaled value.
The arithmetic mean is defined as:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

We can then compute the standard deviation as follows:

$$\sigma(x) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2}$$

KNN steps:
1. Store the output values of the M nearest neighbors to query scenario q in vector
r = {r1, r2, … rm} by repeating the following loop M times:
(a) Go to the next scenario s^i in the data set, where is the current iteration within the domain { 1, ..., P}
(b) If q is not set or q <d (q, s^i): q d (q, s^i), t ◄ o^i
(c) Loop until we reach the end of the data set (i.e. i=P)
(d) Store into vector and into vector
2. Calculate the arithmetic mean output across as follows

$$\sigma(x) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2}$$

3. Return r as the output value for the query scenario q

7.2 Edge Detection Algorithm:
Four steps in edge detection algorithm:
1) Smoothing: Suppress the noise, without destroy the true edge.
2) Enhancement: Apply the filter to enhance the quality of frame.
3) Detection: Determine which edge pixel having actually noise and discarding that noise.
4) Localization: Determine the exact location of edge

Color Edge Detection Using Euclidean Distance and Vector Angle:

Most edge detectors work on the gray scale representation of the image. This cuts down the amount of data you have to work with (one channel instead of three), but you also lose some information about the scene. By including the color component of the image, the edge detector should be able to detect edges in regions with high color variation but low intensity variation. This edge detector uses two operators: Euclidean Distance and Vector Angle. The Euclidean Distance is a good operator for finding edges based on intensity and the Vector Angle is a good operator for finding edges based on hue and saturation. The detector applies both operators to the RGB color space of an image, and then combines the results from each based on the amount of color in a region. There is a difference vector and a vector gradient version. We chose to implement the vector gradient version.
The Euclidean Distance between two pixels is defined as:

$$D(\vec{v}_1 - \vec{v}_2) = \left\| \vec{v}_1 - \vec{v}_2 \right\|$$

Where v1 and v2 are RGB triplets (v = [R G B]).
The Vector Angle between two pixels is approximated by:

$$\sin\theta = \left(1 - \left(\frac{\vec{v}_1^{\,T}\vec{v}_2}{\|\vec{v}_1\|.\|\vec{v}_2\|}\right)^2\right)^{1/2}$$

Because sin θ ≈θ for small angles. Again, v1 and v2 are RGB triplets (v = [R G B]).The Euclidean Distance and Vector Angle are combined using a saturation-based combination method. This combination is defined as:

$$C_{GV} = \rho(S_1,S_2)\sqrt{1-\left(\frac{\vec{v}_i^{\,T}(x,y)\vec{v}_0(x,y)}{\|\vec{v}_i(x,y)\|.\|\vec{v}_0(x,y)\|}\right)^2} + (1-\rho(S_1,S_2)).\|\vec{v}_i(x,y)-\vec{v}_0(x,y)\|$$

Where,

$$\rho(S_1,S_2) = \sqrt{\alpha(S_1).\alpha(S_2)}$$

And:

$$\alpha(S) = \frac{1}{1+e^{-slope\,(S-offset)}}$$

The "slope" and "offset" values in the sigmoid α(S) are set experimentally, and S1 and S2 are the saturation values of each pixel. This combination weights the Vector Angle more heavily in areas of high color saturation and the Euclidean Distance more heavily in areas of low saturation.

The algorithm for finding edges in the image is as follows:

1. For each pixel in the image, take the 3x3 window of pixels surrounding that pixel.

2. Calculate the saturation-based combination of the Euclidean Distance and Vector Angle between the center point and each of the eight points around it.

3. Assign the largest value obtained to the center pixel.

4. When each pixel has had a value assigned to it, run the results through a threshold to eliminate false edges.

Color Edge Detection using the Canny Operator:

Another approach to edge detection using color information is simply to extend a traditional intensity based edge detector into the color space. This method seeks to take advantage of the known strengths of the traditional edge detector and tries to overcome its weaknesses by providing more information in the form of three color channels rather than a single intensity channel. As the Canny edge detector is the current standard for intensity based edge detection, it seemed logical to use this operator as the basis for color edge detection. The algorithm we used for applying colors to the canny edge detector was a very simple one:

1. Read in a color image and divide it into its three separate color channels.

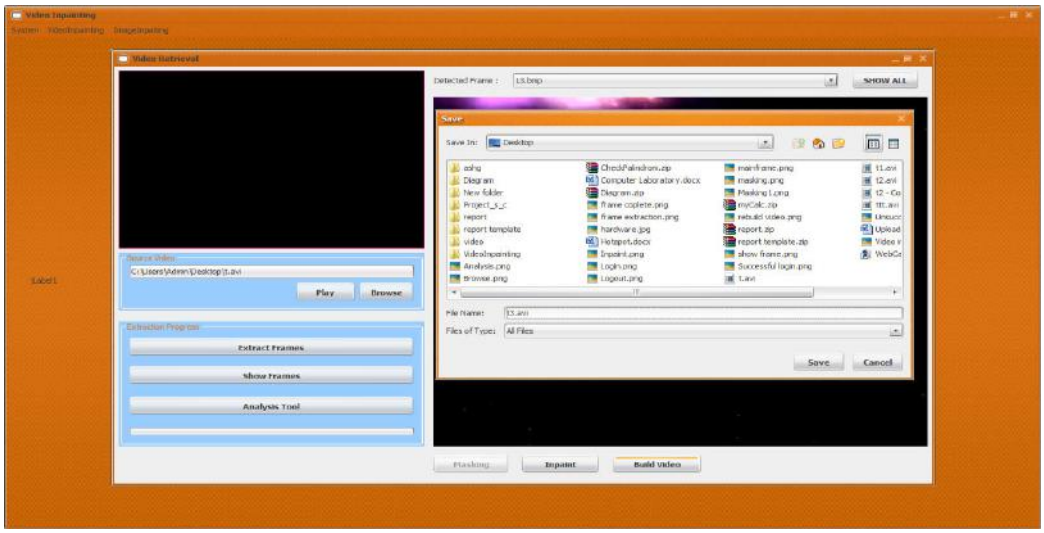
2. Run each color channel through the canny edge detector separately to find a resulting colored edge map.

3. Combine the resulting edge maps from each of the three color channels into one complete edge map. For this step there are a variety of ways you can combine the edges found for each different color, but we found that a simple additive approach provided the best results. So if there was an edge in any of the three colored edge maps, we added it to the general edge map.

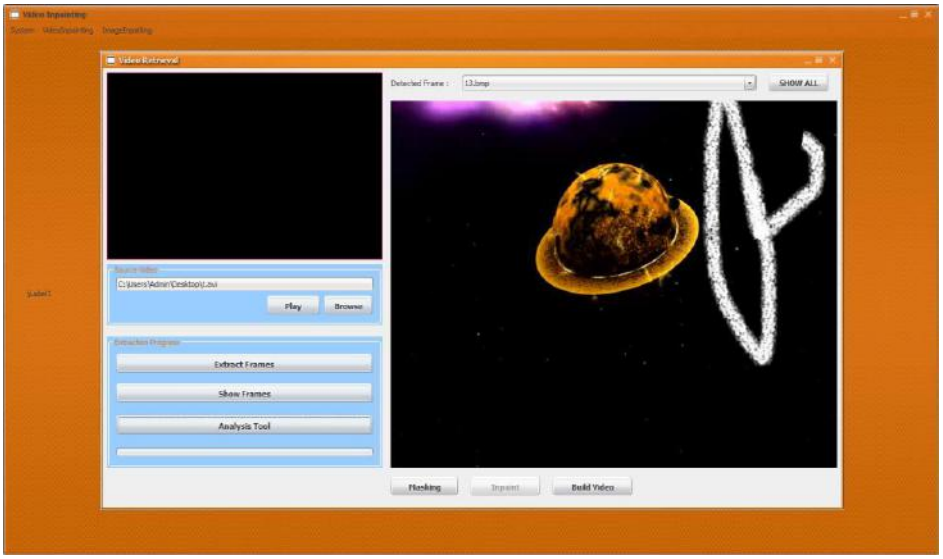
8. EXPERIMENTAL RESULTS:

The output of the implemented modules of the system is as follow:

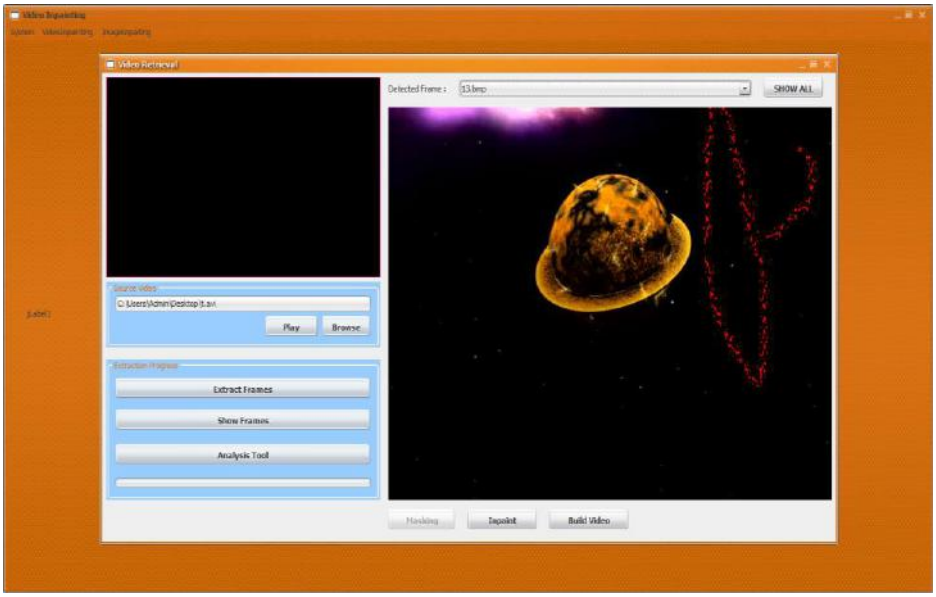
A. Input video selection:



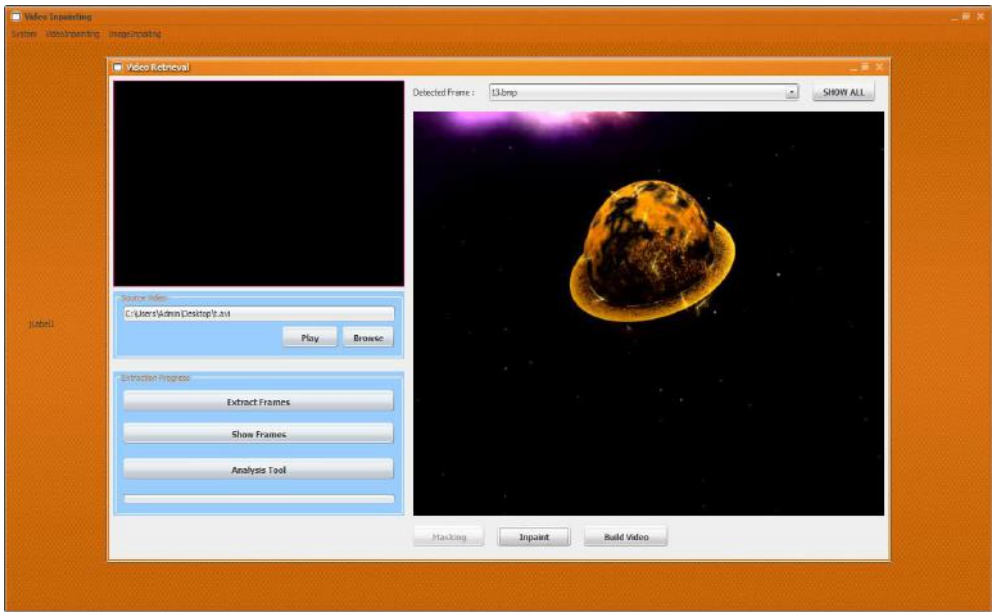
B. Detection of damaged Frames*:*



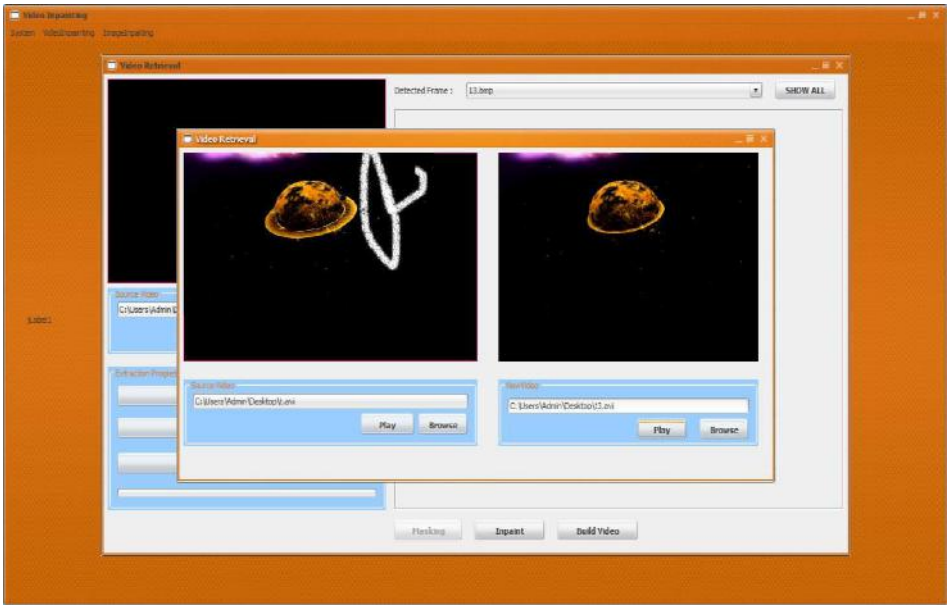
C. Masking Frame:



D. Inpainted Frame:



E. Comparison of Source and New Video:



9. CONCLUSION:

This paper has presented a novel algorithm for removing large objects from digital photographs. As per experimental results, we can conclude that this strategy can make more proficient output than other existing procedures. It fills the missing region accurately to avoid over-shooting artifacts. The major improvement over existing systems is that the given system does inpainting of the videos with underlying image inpainting technique and it can result in better and efficient output because we are combining multiple inpainted images to get the totally inpainted video. We have also does automatic object detection in frame which is to be removed. Object will be detected and removed automatically from all the frames. Rebuild the video properly.

10. ACKNOWLEDGMENT:

It gives us great pleasure in presenting the final project report on ‘disocclusion free video inpainting with short-term windows: application to object removal and error concealment’. We might want to thank the analysts and also distributers for making their assets accessible. We additionally appreciative to commentator for their significant recommendations furthermore thank the collage powers for giving the obliged base and backing.

REFERENCES:

1. N. Neelima, M. Arulvan, "Object Removal by Region Based Filling Inpainting", IEEE Transaction, 2013.
2. Yu Gao, Gene Cheung, Thomas Maugey, Pascal Frossard, Jie Liang, "Encoder- Driven Inpainting Strategy in Multiview Video Compression", IEEE 2015.
3. Martin Kppel, Karsten Miller, Thomas Wiegand, Fellow, "Filling Disocclusion Extrapolated Virtual Views Using Hybrid Texture Synthesis", IEEE 2016
4. Daniel Glasner, Shaibagon, Michel Irani, "Super-Resolution from a single image", IEEE 2009
5. Zihan Zhou, hallin Jin, Yi Ma, "Plane-Based Content preserving wraps for video stabilization", IEEE 2013
6. Prof. R.K. Sarawale , Nitin L. Tajane, Ashwini R. Makhare , Neha R. Reddy, "Inpainting on Static Object of Video and Maintaining Spatial Secular Stability", international Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 3, March 2015.
7. Mohammad Sufyan, Sagar Badnerkar, "Unwanted Object Removal In A Video By Using Video Inpainting Technique", International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 4, Issue 2, February 2015.
8. Vaishali U. Gaikwad P. V. Kulkarni, "Exemplar-based Video Inpainting for Occluded Objects", International Journal of Computer Applications (0975 8887) Volume 81 No.13, November 2013
9. Mounira Ebdelli, Olivier Le Meur, and Christine Guillemot, "Video Inpainting with short-term windows: application to object removal and error concealment", IEEE Transaction, Jan 2015.
10. P. Buyssens, M. Daisy, D. Tschumperle, O. Lezoray, "Exemplar-based Inpainting: Technical Review and new Heuristics for better Geometric Reconstructions", IEEE 2013.