International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-7 Issue-5, January 2019

A Novel Framework for Fast Video Inpainting

Md. Salman Bombaywala, Chirag Paunwala

Abstract*: *Video inpainting is considered as a complex problem in the current literature. This paper proposes a fast, efficient and automatic method of video inpainting to inpaint moving objects in the video. The presented algorithm employs the spatiotemporal coherency present in the video frames for inpainting while considering the fact that the background either has periodic motion or it remains stationary. The algorithm does not require any manual generation of the mask. Batch frame based inpainting is proposed to maintain motion information in case of background having periodic motion. A new dissimilarity measure; 3D N-SSD is introduced to find similar frames for frame-based video inpainting algorithms. The proposed algorithm is tested for different background and illumination conditions. We have done speed and quality test analysis by inpainting videos of different backgrounds. Quick execution times and high PSNR values for inpainted videos show effectiveness of our algorithm.

Index Terms: 3D N-SSD, inpainting mask, periodic background, reference frame, temporal data, video inpainting.

I. INTRODUCTION

Inpainting is a technique to modify the images and videos in such a manner that the modifications are undetectable. For maintaining this, inpainting algorithms fill in the missing part of image/video by using the information from surrounding parts within the image/video. Inpainting algorithms have been studied well in case of images. However, video brings an added dimension to the subject which leads to increasing complexities.Videos are found to be an essential medium of information used in many areas like home, film industry, surveillance and so on. Most of the post-processing is done manually on the video which requires an enormous amount of time and expertise. The post-processing fields like object removal, old film restoration, etc. are active areas of research in inpainting. Videos consist of a series of images with rapid update rates. However, it does not imply that the image inpainting algorithms directly apply to videos. Image inpainting algorithms aim to maintain the spatial coherency, while for video inpainting algorithms temporal domain must be considered to avoid artefacts in the reconstructed video. Our algorithm employs the spatiotemporal coherency within the video frames to obtain the desired results.In existing literature, the user determines the area to be inpainted by manually creating an inpainting mask and then the algorithm takes over. This makes the area to be inpainted in a video, a very subjective choice and hence its application dependent. Our algorithm is designed to detect moving objects and generate the inpainting mask automatically. However, for specific applications like scratch removal, text removal, burst error correction, etc. a user can modify the mask generation sub-module. Many algorithms and methods are proposed for video inpainting, each having their uniqueness. Bertalmio et al [1]. developed Navier Stokes video inpainting algorithm in which fluid dynamics principles were used to propagate the isophote lines towards the interior of the target region. It had very limited applications as it considered the spatial aspects of video only and ignored its temporal coherency. J.Jia et al. [2], developed a new method which inferred the missing static background and moving foreground in a video. For recovering background pixels, homography blending and layer segmentation were used to prevent flickers and maintain coherency. To inpaint foreground, sampling and alignment strategy was used by assuming the periodic motion of the object. Patwardhan et al. [3] inpainted the occluded objects in a video taken from a stationary camera. They used an optical flow based mask to determine whether a pixel is moving or stationary. The static background in a frame was filled first by copying the information from other frames. The remaining hole gets filled by priority based spatial filling-in scheme which gives a best matching patch for the highest priority location. The foreground moving object was then inpainted by using a priority based method and then the remaining area was inpainted as the background inpainting method. Cheung et al. [4] presented a method of video inpainting captured by stationary and moving cameras. The background was filled using adaptive background replacement. A dissimilarity measure was proposed to improve the continuity of the frames. Ghanbari and Soryani [5] presented a technique to restore video frames which has periodic motion. The algorithm separated foreground and background using a simple thresholding method. Background inpainting was done using Criminisi et al. [6] approach. For foreground inpainting, a mosaic was created based on user input of first and last damaged frame. The algorithm only considers those frames which have a complete object. The algorithm provides excellent results for video with periodic motion and a static background. Wexler et al. [7] reproduced the missing region by using patches obtained from different parts of the video. Since the moving object occupies only a small portion of the video frame, searching the data in the entire video is impractical. Ghorai et al. [8] used 5 features of a video box to find the similarities between the target block and source video box. The proposed algorithm used optimization on multi-scale using EM algorithm. The method produced comparable result at considerably better execution speed. Jia et al. [9] proposed a method inspired by [6] to fill the target region fragment by fragment rather than pixel by pixel. To increase the speed of execution, search space was reduced based on mean shift tracking. To maintain the consistency of the inpainted region, graph cut minimization approach was used. Jia et al. [10] presented a method to repair video under

Revised Manuscript Received on 30 January 2019.
***** Correspondence Author
Md.Salman R. Bombaywala*, Uka Tarsadia University, Bardoli, Gujarat, India.
Chirag N. Paunwala, Department of Electronics and Communication Engineering, Sarvajanik College of Enggineering and Technology, Surat, Gujarat, India.

the influence of variable illumination. They made use of a sample movel which consisted of one periodic motion of the object. The inpainting task was then reduced to warping and aligning the damaged movel with the sample movel and blending the target region. The method uses tensor voting scheme to warp and regularize the movel. The method simultaneously completes the static background and finally the movel is merged with the completed background. The results show the robustness of the algorithm. Ling et al. [11] converted the 3D video volume to 2D slices and used [6] to inpaint the slices. A virtual contour of the object was then obtained from the combined 2D slices. The matching posture is then searched from the undamaged postures in the video volume. The paper also provides a synthetic posture generation to be used if a matching posture in not available.

Most of the algorithms discussed above have large complexities associated with it and hence are not much fast. The algorithm proposed in present work is fast as compared to existing state of the art techniques as it utilises the coherent temporal data amongst the video frames. Further, in our algorithm, data is searched in a local vicinity of spatial volume which eliminates the complexities of exhaustive search. All the methods mentioned above require user intervention to obtain inpainting mask for every frame. S. Grover et al. [12] proposed a method in which user has to provide a mask in the initial frame and object tracking be used to obtain the mask in remaining frames. Our algorithm eliminates the need for tracking and generates the inpainting mask automatically. Our algorithm determines the nature of background by computing changes in the background features of the video frames. Depending upon the nature of background, an appropriate video inpainting algorithm is used. The paper is organised as follows: Section 2 describes the process of automatic mask generation and background determination. Section 3 presents two novel video inpainting algorithms with their advantages and limitations. In Section 4, we discuss the need for new similarity measure for frame-based inpainting and introduce 3D N-SSD. Section 5 shows tested results of the proposed algorithm. We summarise and conclude in section 6.

II. MASK GENERATION AND BACKGROUND DETERMINATION

The objective of an inpainting algorithm is to fill in the missing information in the hole as selected by a user in the form of a mask. We present a method to generate an inpainting mask for moving object automatically and accurately by detecting the moving object in a video sequence using a fast and accurate technique. The basic idea is to segment the image into moving foreground and a static background. The implemented method is simple and accommodates illumination variations and small motions in the background. On successful segmentation, the algorithm creates the mask by considering moving foreground object as an area for inpainting. We also present a method of inpainting for videos having the static background. The algorithm computes local and global features over the frame region belonging to the background to determine the nature of background. If the values of the feature exceed a predefined threshold, the algorithm classifies the video as having a background motion. Our algorithm is designed to perform well for both the above cases – static background and background with small periodic motion. However, we also present a simpler and effective method to inpaint a part or whole of video with a static background.

A. Automatic Mask Generation

The video first undergoes background modelling by calculating the deviation of pixels from the mean value. The algorithm thus classifies the pixels in these frames as stationary or non-stationary pixels. Few frames are considered initially to generate a background model by using the stationary pixels. The algorithm starts by considering 'n' initial frames and a window of size 'W'. After classification of pixels, the algorithm calculates the minimum and maximum values for stationary frames and further classifies the pixels as background or foreground depending upon the local threshold. The steps of the implemented method are as shown in below algorithm 1.

Algorithm 1: Video Segmentation

```
Data: W,fk.
Result: Segmented Video S
for k= 1: (n-W+1) do
 for height of frame do
  for width of frame do
        V = [f_k,f_{k+1},.....f_{k+(W-1)}];
   σ = stddeviation(V);
   D(p) = |D(k+W/2)-V(p)|; for each p=k+l
   L=sum of lowest W/2 values in D ;
     if L< (W/2) then
        f_{k+(W/2)}(x,y) ← stationary;
     else
        f_{k+(W/2)}(x,y) ← non stationary;
     end
  end
 end
end
for height of frame do
 for width of frame do
   Min(x,y) = minimum(f_s(x,y));
   Max(x,y) = maximum(f_s(x,y));
   Threshold (x,y) = (1/C)(Min(x,y) + Max(x,y));
   T_L = Threshold – Min;
   T_U = Threshold – Max;
      if T_L ≤ f ≤ T_U then
         S_f(x,y) = 0 ← Background pixel;
      else
         S_f(x,y) = 1 ← Foreground pixel;
      end
 end
end
```

Figure 1 shows the video segmentation result obtained by algorithm 1.

As can be seen the results contains some outliers and discontinuities. Morphological operation of opening and closing and blob analysis are used to obtain the required inpainting mask.

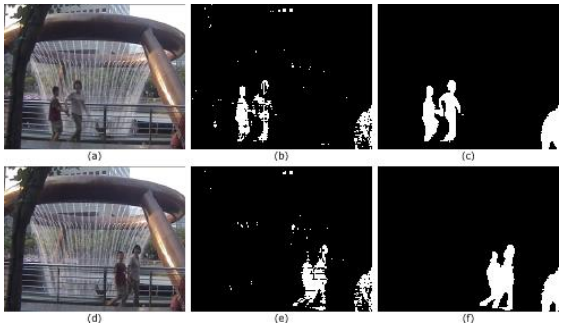

Figure 1. Result of mask generation using algorithm 1. (a) and (d) are original video frames. (b) and (e) represents the output of algorithm 1. (c) and (f) are the mask generated after morphological operations.

B. Background Determination

For the background regions obtained after mask generation, the background features need to be determine if the background is static or not. We divide the background region into non-overlapping blocks and calculate the differences in between the successive frames. The HSV colour model is utilized for calculating the differences. If the successive difference between the backgrounds is above a predefined threshold, the algorithm classifies that video as the one having motion. After the classification, one can decide to use the most efficient algorithm for the problem at hand.

III. PROPOSED ALGORITHM

Figure 2 summarises the notations used in the paper. 'Ω' denotes the occluded part of the video frame, while 'D' denotes the un-occluded part. 'V' is the spatio temporal volume representing the video frames. V=(ΩUD) and (Ω∩D)=Ø. 'R' is the reference frame and 'l' is the number of frames in the video sequence. Position of a pixel in the video is given by p(x,y,t) and d(p) ∈ R^3 is the data at position 'p'.

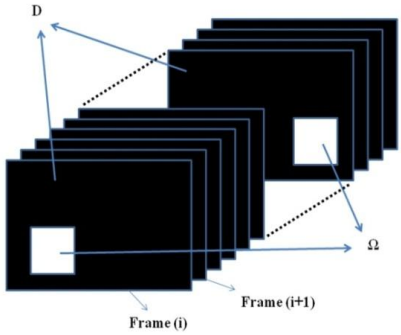

Figure 2. Notations used in the present paper.

As shown in fig.3, the missing data in the current frame is available in the previous frame and hence it can be used directly for the purpose of inpainting. The first column in the figure shows the mask of frames and the second column shows the image frame itself. One can observe that the data at coordinates (76, 21) in the current frame (frame x) is missing as indicated by a '0'. The data at the same coordinates are present in the previous frame (frame x-1) as shown by a '1'. Hence the newly occluded data can be inpainted from a previous frame without any discontinuity particularly in case of a steady background or if the frame rate of a video is very high. This availability of temporal data simplifies the process of inpainting. However, in case of small periodic motions in the background, consideration of only one reference frame will lead to visual discontinuity. Here batch frame based inpainting algorithm is proposed to overcome this problem.

The algorithm deals with the video inpainting problem under the following constraints: the background is considered to be static or having a small periodic motion and the camera used to capture the video is stationary. The algorithm does not require any manual input.

A. Video Inpainting for Static Backgrounds

As discussed earlier, the availability of occluded data in the temporal direction makes the process of video inpainting much simpler. For videos with a static background, previous or subsequent frames provide sufficient data to inpaint the occluded portion. Searching for data in every subsequent frame introduces unnecessary computations for the case of a static background. Hence, we find a reference frame from the video sequence. A reference frame is a frame which does not consist of any foreground object. If no such frame is available in the video, the algorithm searches for the frame having a minimum occluded region and inpaints that frame using any image inpainting algorithm [6],[13]-[16]. The target frames in the video are inpainted using the generated reference frame. Direct substitution of data from the reference frame to target frame gives good results. However, to avoid visual discontinuities and sharp edges, a weighted patch based technique can be utilised to get better continuity. The algorithm only substitutes the data corresponding to Ω to maintain any minor illumination changes in the rest of the frame. This is a crucial step as without this; the inpainted video will only be a replica of the reference frame resulting in loss of all the minor details in D.

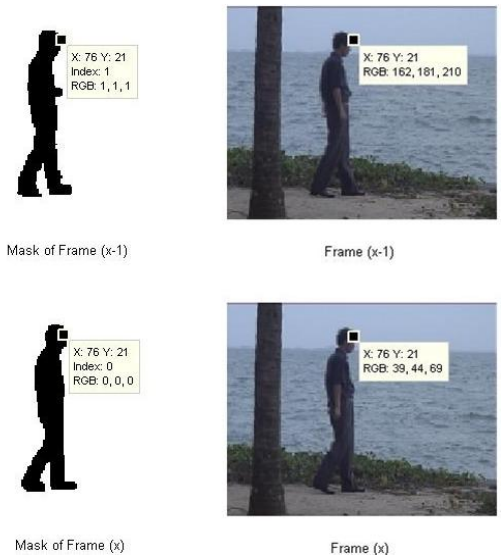

Figure 3. Frame comparison. The data missing in current frame is available in previous frame.

Once, the target frame is inpainted; we consider it to be the reference frame for the successive target frame. This approach maintains enough continuity among the video frames to make it plausible.

All the video frames are inpainted by repeating the above process. Algorithm 2 summarises the proposed method. Since successive reference frames are used, any error in the generation of initial reference frame will propagate throughout the video. Hence, the process of reference frame generation should be very accurate.

Algorithm 2: Succesive frame based video inpainting

Data: Ω,D,V,R,l.
Result: Inpainted video V_i
for t=1:l **do**
  if Ω ϵ frame(t) **then**
    while p(x,y,t) ϵ Ω **do**
      d(p) ← d(R(x,y))*σ ;
    end
    R← frame(t);
  else
    R← R;
  end
end
V_i = combine(frames)

B. Video Inpainting for Background with Periodic Motion

For background having periodic motions, inpainting based on a successive frame will lead to blocking effects in the reconstructed video. (see fig.4). The video inpainted for such case will be void of motion information in the reconstructed region. To overcome the above inconsistency, we designed a batch frame based inpainting algorithm where instead of using a single frame, batch or group of frames are used to inpaint the hole. This method provides consistency in background motion in the inpainted region.

Figure 4. Blocking effect observed in successive frame based method for background having periodic motion.

For every frame to be inpainted, the algorithm searches for the best matching frame in the entire video. The search is limited to the unmasked region of the frame to provide robustness even in the absence of a reference frame (a frame without Ω). For a target frame, the algorithm searches for valid data around the same spatial coordinates and different temporal coordinates. A 3D N-SSD measure is developed to find the similarity amongst the frame. Similarity search is done by converting the video in HSV color model and finding the similarity for each plane. The distance between frames fx and fy is calculated as

$$d(f_x, f_y) = \sqrt{\left(f_x^H - f_y^H\right)^2 + \left(f_x^S - f_y^S\right)^2 + \left(f_x^V - f_y^V\right)^2} \quad (1)$$

Calculation in HSV provides robustness to change in illumination.On finding a matching frame in the video, a group of frames including the matched frame and its following few are used to inpaint the group of target frames and its following frames. The number of frames considered for inpainting depends upon the velocity of the moving object and the frame rate of the video. Since the motion is constrained to be periodic and it is desired to fill the entire occluded data in one search, we avoid immediate temporal neighbourhoods for finding the best-matched frames. For objects with less velocity, the results obtained will not be as desired. Hence, care must be taken in choosing the matched frame group. Two constraints are imposed on the search space

1. The search should not be done in the immediate neighbourhood of the target frame. This constraint is necessary as the frames will be much similar to it immediate neighbourhood. If such frames are chosen for inpainting, it will result in visual replication of background motion and can even result in blocking effect in the inpainted region.
2. There should be no overlap between the masks of target batch and matched batch. This ensures enough object movement between the frame batches.

The batch frame based inpainting method is summarized in algorithm 3.

Algorithm 3: Batch frame based video inpainting

Data: Ω,D,V,R,l.
Result: Inpainted video V_i
for t=1:l **do**
 if Ω ϵ frame(t) **then**
  for frame(t_i) such that 't_i' ∉ neighbourhood to t
   while mask(p(x,y,t)) ∩ mask(p(x,y,t_i)) = Ø **do**
    dist = 3D N-SSD(frame(t_i),frame(t));
    if dist is smaller → matched(t) **end**
   end
  end
  end
  for k=t:t+n
   d(frame(k)) ← d(matched(k));
  end
end
V_i = combine(frames)

IV. RESULTS AND DISCUSSION

A. 3D N-SSD

We propose a new dissimilarity measure for frame-based inpainting. Since only unmasked parts of frames are considered for matching; in a framework of exhaustive search, it is possible that for some frame there are large numbers of pixels participating in the similarity match while for some pairs, the number of participating pixels is less. The former case will lead to a more significant similarity score as compared to later for obvious reasons.To avoid this, we divide the score by the total number of participating pixels.

This gives a justifiable similarity measurement between different frames.

We select the pair with best similarity score as the initial frame of the group.

$$NSSD = \frac{\sum_{p=1}^{3}\left(tar\,get_p - query_p\right)^2}{\sum_{p=1}^{3} numel_p} \quad (2)$$

where, query = frame being matched
target = frame being compared in search space
numel = number of participating pixels
The best matching pair is the pair which minimises the 3D N-SSD.

B. Discussion

The algorithm is tested and presented for different scenes having a static background as well as background with motion. The batch size is considered as 8 frames per batch which gives effective results for varieties of background conditions. It gives excellent results for both the cases and the execution is also speedy. However, a user can change the batch size depending upon the velocity of moving object in the video. For objects with less velocity, a small batch size will lead to improper results. The presented algorithm accurately creates a mask by separating out moving objects in an image. The threshold is adjusted such that significant changes in the video are segmented. This facilitates the mask generation in case of periodic small motions in the video. The algorithm is tested to create a mask for single and multiple moving objects. Figure 5 and 6 are the result of automatic mask creation for moving objects in the video. It is observed that the algorithm gives excellent results for mask generation. Since the mask gets generated automatically, the presented algorithm does not require any manual inputs or expertise to create the mask which reduces the overall time for inpainting a video.Figure 7 and 8 are results of video inpainting. The results demonstrate the effectiveness and accuracy of the presented inpainting algorithm for video with the static background (fig 7) and having periodic background motion (fig 8). We have purposefully edited the detected mask in fig. 7 to leave the shadow on the ground as it is. This helps in visualising the visual aspect of inpainting. We have compared the speed of our algorithm with Newson et al.'s [17] method. Table 1. shows the speed comparison of our proposed algorithm with Newson et al.'s [17] method. Rather than going for an exhaustive search in the entire video, we utilise the spatiotemporal coherency between the video frames to inpaint the occluded data rapidly. The occluded data covered by the object at a specific location within a frame is obtained by searching in the vicinity of the same spatial dimension but along the temporal direction of the video. Our algorithm is performing much faster for both the types of videos.

Table 1. Window Sizes at Different Scale

Sr. No.	Video	Number of Frames	Execution Time of [17]	Execution Time for proposed algorithm
1	Static Background (Ref. fig 7)	138	29.66 minutes	62 seconds
2	Background with periodic motion (Ref. fig 8)	100	27 minutes	53 seconds

Since inpainting is a subjective problem, it is difficult to quantify the visual result of inpainting. However, to test the effectiveness of our designed algorithm, we generated synthetic videos and inpainted them by using our algorithm. Figure 9 and 10 show the result of inpainting a floating text over a static background and a moving object against a background having periodic motion. To evaluate the quality of inpainting, PSNR of reconstructed video is calculated. As shown in Table 2. Maximum PSNR is achieved for video with a static background. However, the PSNR value for videos having a motion is also towards a larger side; indicating the accuracy of proposed inpainting algorithm.

Table 2. PSNR calculated for reconstruction of synthetic video.

Sr. No.	Video	Number of Frames	PSNR [17]	PSNR of our algorithm
1	Static Background (Ref. fig 9)	40	45.2746	66.2307
2	Background with periodic motion (Ref. fig 10)	35	37.0467	59.4048

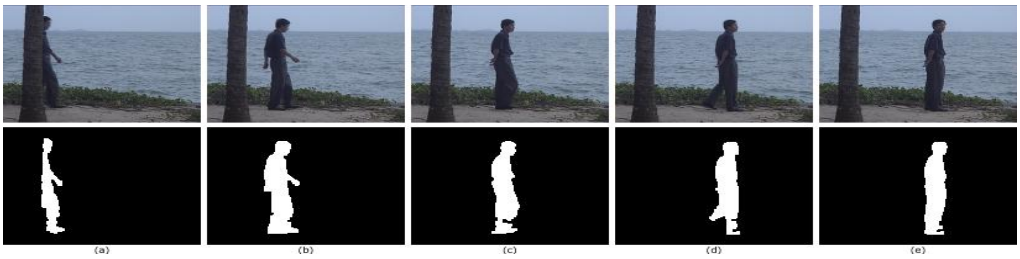

Figure 5. Automatic mask generation for moving object in the video.(Background with periodic motion).

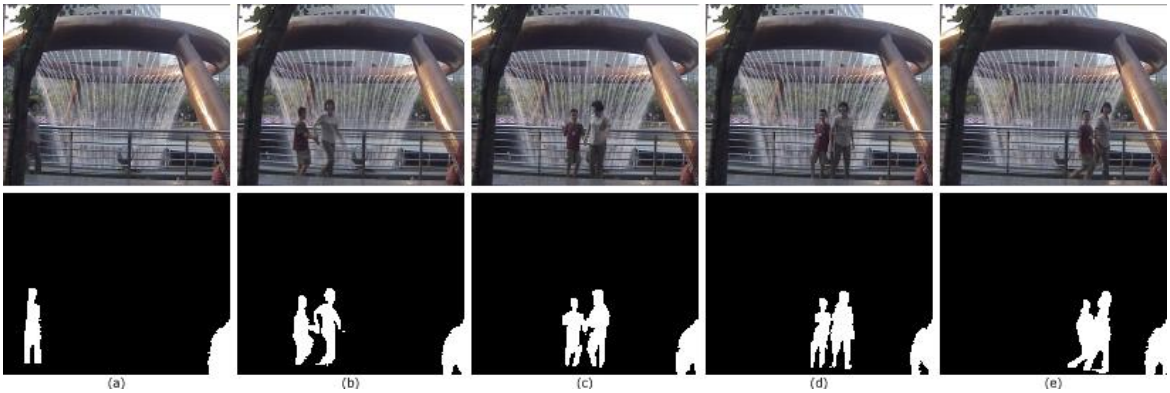

Figure 6. Automatic mask generation for moving object in the video.

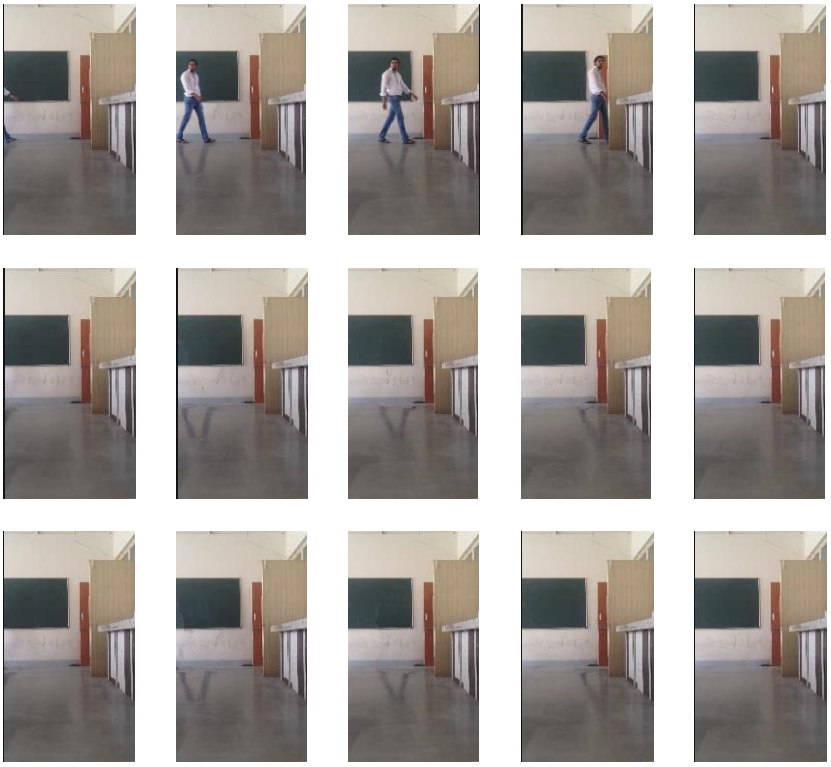

Figure 7. Inpainted Frames for static background. Video frames (top). Result of [17] (middle). Proposed Algorithm (bottom).

Figure 8. Batch frames based inpainting. Top row – Original video frame. Bottom row – Result of batch frame based inpainting.

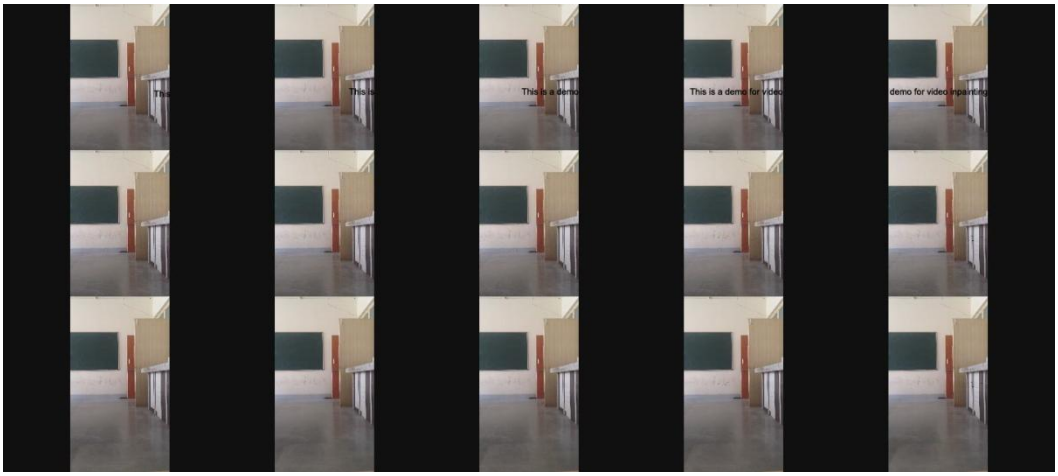

Figure 9. Batch frames based inpainting on synthetic video. Top row – Synthesised video frame. Middle row – Result of –[17]. Bottom row – Result of our algorithm.

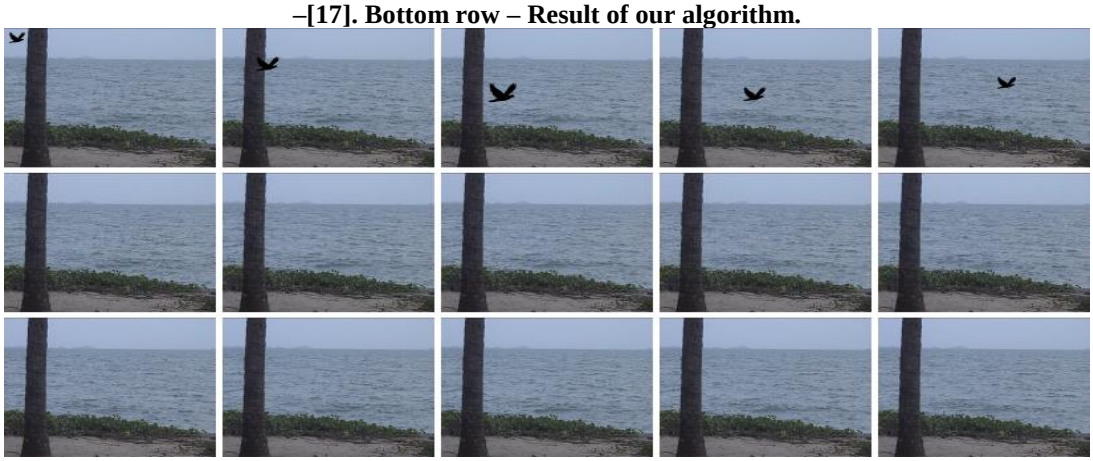

Figure 10. Batch frames based inpainting on synthetic video. Top row – Synthesised video frame. Middle row – Result of –[17]. Bottom row – Result of our algorithm.

V. CONCLUSION AND FUTURE SCOPE

The proposed algorithm introduces a simple, fast and novel approach to video inpainting. The simplicity of the algorithm lies in using the temporal coherency between the video frames for inpainting. We present a unique concept of batch frame inpainting and automatic mask generation for inpainting moving objects. The algorithm performs well in different situations having periodic background motion or a static background. Our algorithm improves the execution times and achieves better video quality. The paper also introduces a new dissimilarity measure for finding the similar frames for the problem of frame-based video inpainting. A user may change the batch size to achieve faster speeds at the cost of quality.The designed algorithm can be applied to several applications by using appropriate mask generation algorithm. Given a video with number or text patterns appearing randomly; an automatic text detection algorithm can be used to generate a precise mask and text removal can be done. Similar to text removal, scratch detection algorithm can be used to remove scratches appearing in the video sequence automatically. Burst error during transmission can cause some pixels to blank out within few frames. These pixels can be identified and inpainted to get a plausible video. section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

REFERENCES

1. M. Bertalmio, A.L. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," CVPR 2001: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, vol.1, pp.355-362, December 2001.
2. J. Jia, W. Tai-Pang, Y.W. Tai, and C.K. Tang, "Video repairing: inference of foreground and background under severe occlusion," CVPR 2004: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, vol.1, pp.364-371, June-July 2004.
3. K.A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting of occluding and occluded objects," ICIP 2005: IEEE International Conference on Image Processing, Genoa, Italy, pp.69-72, September 2005.
4. S.C.S. Cheung, J. Zhao, M.V. Venkatesh, "Efficient Object-Based Video Inpainting," ICIP 2006: IEEE International Conference on Image Processing, Atlanta, Georgia, USA, pp.705-708, October 2006.
5. A. Ghanbari, and M. Soryani, "Contour-Based Video Inpainting," MVIP 2011: 7th Iranian Conference on Machine Vision and Image Processing, Tehran, Iran, pp.1-5, November 2011.
6. A. Criminisi, P. Pérez, K. Toyama, "Region filling and object removal by exemplar-based image inpainting", IEEE Transactions on Image Processing, vol. 13, no. 9, pp.1200-1212, 2004.
7. Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.29, pp.463-476, 2007.
8. M. Ghorai, P. Purkait, and B. Chanda, "A fast video inpainting technique," International Conference on Pattern Recognition and Machine Intelligence, Berlin, Heidelberg, pp.430-436, December 2013.

9. Y.T. Jia, S.M. Hu, and R.R. Martin, "Video completion using tracking and fragment merging" The Visual Computer, vol. 21, no. 8, pp.601-610, 2005.
10. J. Jia, Y.W. Tai, T.P. Wu, and C.K. Tang, "Video repairing under variable illumination using cyclic motions," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp.832-839, 2006.
11. C.H. Ling, C.W. Lin, C.W. Su, Y.S. Chen, and H.Y.M. Liao, "Virtual contour guided video object inpainting using posture mapping and retrieval," IEEE Trans. Multimedia, vol. 13, no. 2, pp.292-302, 2011.
12. S. Grover, A. Mittal, S. Gupta, A.K. Sarje, and N. Kaushik, "A robust framework for automated digital video inpainting," International Journal of Signal and Imaging Systems Engineering, vol.1, no. 3/4, pp.185-196, 2008.
13. J. Wu, and Q. Ruan, "Object removal by cross isophotes exemplar-based inpainting," ICPR 2006: 18th IEEE International Conference on Pattern Recognition, Hong Kong, China, vol. 3, pp.810-813, August 2006.
14. P. Goswami, and C. Paunwala, C. "Exemplar-based image inpainting using ISEF for priority computation," CSCITA 2014: IEEE International Conference on Circuits, Systems, Communication and Information Technology Applications, Mumbai, India, pp.75-80, April 2014.
15. S. Bombaywala, P. Goswami, and C. Paunwala, "Comparative analysis of exemplar based image inpainting techniques," ET2ECN 2014: 2nd IEEE International Conference on Emerging Technology Trends in Electronics, Communication and Networking, Surat, India, pp.1-6, December 2014.
16. K.B. Desai, S.R. Bombaywala, and C.N. Paunwala, "Sparsity based image inpainting using optimisation techniques," TENCON 2015 IEEE Region 10 Conference, Macau, SAR, pp.1-6, November 2015.
17. A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, "Video inpainting of complex scenes," SIAM Journal on Imaging Sciences, vol.7, no.4, pp.1993-2019, 2014.