Video Inpainting Using a Contour-based Method in Presence of More than One Moving Objects

A. Ghanbari Talouki, M. Majdi and S. A. Edalatpanah

Abstract
To restore corrupted images and video frames, a technique named inpainting or completion is used. This paper introduces an approach which is used to inpaint damaged objects in video frames. These objects are damaged whether partially or totally. In order to get to a more visually pleasant result, the moving objects and the background are separated from each other. To inpaint the stationary background, a patch-based method is used. Missing moving objects are inpainted using an object-based method with the help of a contour-based method. Presence of more than one moving object makes a problem; how to separate the goal object from other moving objects. This paper proposes a new method based on contour to solve this problem. Finally, the inpainted foreground and background are superimposed; the result is the inpainted video.

Keywords
Video inpainting, contour, moving objects separation, similarity measure.

1. INTRODUCTION
One of the interesting subjects in image and video processing is image and video inpainting or completion. Video inpainting means reconstruction of those parts of the video which are destroyed by any reason. To inpaint this video, a similarity measure is used to find the most suitable data in order to fill in pixels of hole which are generated because of an occluding object. Inpainting should be done somehow not to generate and place pixels in such a way that makes the result unpleasant to the viewer; because in inpainting, keeping visual consistency of the inpainted video is very important [1].

Image inpainting methods do not naturally extend to consider time [2]. To extend methods of image inpainting to video inpainting, an initial consideration is to define video data as some distinct images. By this definition, image inpainting methods can be used for any of these images individually. High temporal consistency is not considered here. Therefore, it would not give a pleasant result [3].

In order to preserve the temporal continuity of the moving object, none of image inpainting approaches will directly be used for video inpainting. The first attempt for this purpose was introduced in [4]. Partial Differential Equation (PDE) was used to preserve continuity between video frames [5].

Generally, video inpainting problems are divided to three different problems [6]: a) Stationary background with moving objects. b) No stationary background with stationary objects. c) No stationary background with moving objects [7]. In this paper condition “a” is considered.

According to either inpainting methods separate foreground objects from background or not, video inpainting methods are categorized to two subsets [8]: 1) Inpainting approaches which separate background and foreground. 2) Inpainting approaches which do not separate background and foreground.

The first approach has some advantages compared to the second approach [9]. As a result, the first approach is considered in this paper. Advantages are: i) separating foreground and background helps finding better matches. Besides, search space which should be searched for finding a suitable template, reduces. ii) Background is stationary; consequently, background is inpainted once. After that, the inpainted background is copied to other frames.

Video inpainting algorithms can be divided to five methods [10]: a) Patch-based algorithms. b) Object-based algorithms. c) Motion layer-based algorithms. d) PDE-based algorithms. e) Combinational algorithms.

- Patch-based algorithms: These algorithms consider a patch with the most priority at each iteration. Then, video information except the hole, which must be completed, is searched with the purpose of finding the most suitable patch which can be used for filling in the hole. A sample research in this area is presented in [2].
- Object-based algorithms: These algorithms consider bigger patches which are defined as objects; in simpler terms, these algorithms inpaint one frame every iteration. A sample research in this area is presented in [11] that is an extended version of [12].

Authors Details:
A. Ghanbari Talouki
M. Majdi
*S. A. Edalatpanah(Corresponding Author)
Dept. of Computer Engineering, Ayandegan Institute of Higher Education, Tonekabon, Mazandaran, Iran
e-mail: saedalatpanah@gmail.com
Subject Category: Engineering
Sub Category: Image Processing
Editor-in-Chief: Dr. Sahadev Roy
• Motion layer-based algorithms: Video sequence is divided into different motion layers. Therefore, layer order is defined in overlapping areas. To remove an object, layer of that object is removed. This process remains holes on other layers. By filling in these holes, the remaining layers are completed. Superimposing of these layers results in inpainted video. A sample research in this area is presented in [13].
• PDE-based algorithms: Holes in these criteria are small. PDE is used to inpaint these small holes. [14] introduced a PDE-based method.
• Combinational algorithms: These algorithms combine motion layer-based methods with patch-based methods. Motion layer-based methods restrict the area which is searched in order to find the best patch. A sample research in this area is presented in [15].

In this paper object-based video inpainting method is used; because it takes less time to complete the hole and since one object is inpainted at each iteration, continuity is preserved more than other methods. To improve object-based method, we changed the similarity measure. The other improvement is the method that is used to separate the goal object from other moving objects.

The framework that is used to fill in corrupted portions of video data considers more than one occluding objects that cause more missing data. The assumptions in this paper are stationary background, periodic motion and no scale changing for the foreground.

II. THE PROPOSED VIDEO INPAINTING METHOD
To begin with, the moving foreground and the stationary background are separated from each other by the use of a threshold-based mechanism. To inpaint the stationary background, if the occluding object is moving, we can copy correct places of the background. Yet, in the presence of a stationary occluding object, the hole is inpainted; because there is no information about behind that occluding object. To inpaint the moving foreground, in the presence of more than one moving foreground, separation of the goal object from other moving objects is needed. Then, the foreground is completed. By composing the inpainted foreground and background, the inpainted video is acquired. Schematic overview of the proposed algorithm is shown in Fig. 1.

A. Separation of the moving foreground from the stationary background
To differentiate between the moving foreground and the stationary background, a value named threshold is used. The absolute difference between the current frame and the modeled background is calculated. Then the threshold is applied to this difference. Equation (1) as described below is for the above purpose.

\[ f_{rg_t} = \begin{cases} 
1 & \text{if } |image\_t - Back\_{t-1}| > T \\
0 & \text{otherwise} 
\end{cases} \]  \hspace{1cm} (1)

where \(image\_t\) is the input frame at time \(t\), \(Back\_{t-1}\) is the background model, and \(f_{rg_t}\) is the foreground at frame \(t\). Gaussian Mixture Model (GMM), originally introduced in [16] and improved in [17] is used for modeling the background. In this approach, to model each pixel, a mixture of \(NumGauss\) Gaussian distributions is used. The parameters which are used to define the model, are constantly changed and refined. The number of components which are used for GMM is selected automatically per pixel. Suppose that the probability of observing a special pixel at previous time is defined. Then, the probability of observation of that special pixel at this time, \(t\), is

\[ P(X_t|Back) = \sum_{n=1}^{NumGauss} wght_{n,t} \cdot N(X_t, \mu_{n,t}, \Sigma_{n,t}) \]  \hspace{1cm} (2)

where \(NumGauss\) is the number of components of GMM and \(wght_{n,t}, \mu_{n,t}\) and \(\Sigma_{n,t}\) are introducing weight, mean value and covariance matrix for the \(n^{th}\) Gaussian in the mixture at time \(t\), respectively. Until now, each pixel is modelled. As a result, using \(NumLrg\) largest items of the mixture model, the background is estimated. \(NumLrg\) is acquired from equation (3).

\[ NumLrg = \arg\min_{NumLrg} \left( \sum_{n=1}^{NumLrg} w_{n} > (1 - frg) \right) \]  \hspace{1cm} (3)

where \(frg\) is the maximum portion of data that is defined as foreground. Subtraction of each frame from the model of the background gives the moving foreground. This process is done for each pixel. After calculating this difference, a median filter is applied to the frame.

B. Background Inpainting
Background inpainting algorithm that is used in this paper, is exemplar-based inpainting which was introduced in [18]. The target region which is going to be filled, is the region that its data is missed; this hole is completed by patches from the source region, a region with correct data. For any pixel that is placed on the boundary of the target region (hole), a value named priority is defined as equation (4), then, a patch centered at the pixel that has the most priority, is selected to be the first portion of that hole which is going to be filled in.

\[ Pri(p) = Con(p) \cdot Dat(p) \]  \hspace{1cm} (4)

where \(Con(p)\) and \(Dat(p)\) are confidence and data values, respectively. To define these two values, equations (5) and (6) are used.

\[ Con(p) = \frac{\sum_{q \in \Psi_{p}(t-\delta) \cdot Con(q)}}{|\Psi_{p}|} \]  \hspace{1cm} (5)

\[ Dat(p) = \frac{|\Psi_{p}^{\perp} \cdot n_{p}|}{\gamma} \]  \hspace{1cm} (6)

where \(\Psi_{p}\) defines a patch that is centred at pixel \(p\). \(|\Psi_{p}|\) defines the area of the patch \(\Psi_{p}\). \(\gamma\) is used as a normalization factor (e.g. for greyscale images, \(\gamma = 255\)). \(n_{p}\) depicts orthogonal unit vector to the boundary of the hole and \(\perp\) expresses the orthogonal operator.
Supposing the patch with the most priority value for its center is shown by \( \Psi_p \), the patch \( \Psi_q \) is found as the best match of \( \Psi_p \) in source region. In other word, \( \Psi_q \) has the minimum value for Sum Squared Difference (SSD) among all other patches. Now, it’s time to copy \( \Psi_q \) to \( \Psi_p \); only those pixels of \( \Psi_p \) that has not value accept the new value. After copying new values, confidence term of any pixel of the target patch that intersects with the hole, is updated. This process is iterated until the hole is completely filled. The exemplar-based method used to inpaint the hole in background is an appropriate method which preserves both structure and texture of the frame.

In presence of stationary occluding object, this inpainted hole is copied to other frames. Yet, in presence of moving occluding object, inpainting is not needed any more; because there is information about the area behind the occluding object.

A. Foreground Inpainting

We are going to improve the method which is used to fill holes of moving object. To inpaint the foreground, a contour-based foreground comparison and an object-based inpainting algorithm is used. Object-based method helps preserving both structure and temporal continuity of the goal moving object. To explain simpler, firstly, we consider having one moving object. The method is described in this situation and similarity measures which are used to find the most suitable patch are defined for four different conditions (these different conditions are stated following) with this assumption. Later, assumption changes and we consider more than one moving object. This assumption introduces the problem of differentiating between the goal moving foreground and other moving foregrounds.

II. VIDEO INPAINTING IN THE PRESENCE OF ONE MOVING OBJECT

Until now, the stationary background and the moving foreground are separated from each other. Now, it’s time to extract the template of foreground (moving object) in each frame and bring forward the databases resulted from moving object. One of the resulted databases is contour-database which is used to compare templates in order to find the best match.

From section II/A we have frames of video that are black for background and white for foreground. The first database we are going to construct is contour-database; to make it,
Supposing the patch with the most priority value for its center is shown by $Ψ^p_q$, the patch $Ψ^p_q$ is found as the best match of $Ψ_p^q$ in source region. In other word, $Ψ^p_q$ has the minimum value for Sum Squared Difference (SSD) among all other patches. Now, it’s time to copy $Ψ^p_q$ to $Ψ_p^q$; only those pixels of $Ψ_p^q$ that has not value accept the new value. After copying new values, confidence term of any pixel of the target patch that intersects with the hole, is updated. This process is iterated until the hole is completely filled. The exemplar-based method used to inpaint the hole in background is an appropriate method which preserves both structure and texture of the frame.

In presence of stationary occluding object, this inpainted hole is copied to other frames. Yet, in presence of moving occluding object, inpainting is not needed any more; because there is information about the area behind the occluding object.

A. Foreground Inpainting

We are going to improve the method which is used to fill in holes of moving object. To inpaint the foreground, a contour-based foreground comparison and an object-based inpainting algorithm is used. Object-based method helps preserving both structure and temporal continuity of the goal moving object. To explain simpler, firstly, we consider having one moving object. The method is described in this situation and similarity measures which are used to find the most suitable patch are defined for four different conditions (these different conditions are stated following) with this assumption. Later, assumption changes and we consider more than one moving object. This assumption introduces the problem of differentiating between the goal moving foreground and other moving foregrounds.

II. Video Inpainting in the Presence of One Moving Object

Until now, the stationary background and the moving foreground are separated from each other. Now, it’s time to extract the template of foreground (moving object) in each frame and bring forward the databases resulted from moving object. One of the resulted databases is contour-database which is used to compare templates in order to find the best match.

From section II/A we have frames of video that are black for background and white for foreground. The first database we are going to construct is contour-database; to make it, contour of foreground in each frame is acquired. Fig. 2 shows contour of the foreground in one frame.

Pixel coordinates of the contour in each frame is calculated. These coordinates are sorted ascending in both row and column for each frame. The minimum item in row, maximum item in row, minimum item in column and maximum item in column are introduces as minRow, maxRow, minCol and maxCol, respectively. Consequently, the bounding box is defined as $(minCol: maxCol) \times (minRow: maxRow)$. Cutting this box from frame gives contour of foreground per frame. Fig. 3 illustrates a portion of contour-database. The bounding box has different coordinates for each frame. Since the method proposed in this paper uses contours to compare templates in order to find the best match, equalization of contour bounding boxes should be done. To do this, the maximum and minimum lengths and widths of all bounding boxes are calculated and named as maxHeight, minHeight, maxWidth and minWidth. For frames which have lengths and widths smaller than maxHeight and maxWidth, zeros (black area) are added to the right and down part of the template, respectively.

To acquire database of foreground shown in Fig. 4, bounding box is used as before. Yet, frames are changed so that the black area (the background) is unchanged, but, the white area (the foreground) is colored like the original frame. These templates are used to be placed on hole.

To fill the foreground hole, four conditions are considered:

(i). Partially occluded objects are not considered, at least one complete period of object’s motion with the length equal to the number of frames of hole can be found somewhere in correct portions of video data.

(ii). Partially occluded objects are not considered, no complete period of object’s motion with the length equal to the number of frames of hole can be found in correct portions of video data.

(iii). Partially occluded objects are considered, at least one complete period of object’s motion with the length equal
to the number of frames of hole can be found somewhere in correct portions of video data.

(iv). Partially occluded objects are considered, no complete period of object’s motion with the length equal to the number of frames of hole can be found in correct portions of video data.

Condition number (i) will be explained in more detail, but other conditions will be stated just as equations.

A. No occluded object, complete period

Partially occluded objects is not considered here. As a result, only complete objects and completely destroyed objects are in database. Therefore, comparison can be started whether from left side of the hole or right side of it. Another assumption is that there is a complete period of object’s motion which has the length equal to the length of the hole. Suppose that video frames are depicted as \( f_1, f_2, \ldots, f_h, f_h+1, \ldots, f_{h+n-1}, f_{h+n} \ldots f_s \) to \( f_{s+n} \) are frames including hole. Therefore, the length of hole is \( n \). The moving objects in different frames are shown as \( O_1, O_2, \ldots, O_{h+1}, O_h, O_{h+1}, \ldots, O_{h+n}, O_{h+n}, \ldots, O_N \). \( O_h \) to \( O_{h+n} \) are objects in frames including hole. The aim in video inpainting is finding appropriate templates from database to fill in these \( n \) frames. If the period of the motion is \( \text{per} \), then \( O_i \equiv O_{i+\text{per}} \) \( (i \) stands for frame number). To find the most appropriate patch [19] calculates absolute difference of all complete patches’ contours \( (\text{con}.) \) and the object’s contour one before the hole \( (\text{con}_{b,i}) \) by equation (7). This study completes hole from left.

\[
d(h-1,i) = |\text{con}_{h-1} - \text{con}_{i}|, \quad i = 1,2,...,N, \quad i \neq h-1,h,...,h+n-1 \tag{1}
\]

where \( i \) is the frame number and \( N \) is the number of frames of video.

To preserve continuity, [19] extends the above approach so that five best templates (those which have the least value for \( d(h-1,i) \)) are chosen. These contours are named as \( \text{con}_{b1}, \text{con}_{b1}, \text{con}_{b3}, \text{con}_{b4} \) and \( \text{con}_{b5} \). Now, it’s time to apply the second comparison. It ensures that templates around which was used by equation (7) also match. Equation (8) is the final equation which is used to find the most suitable template.

\[
B = \arg \min_i (|\text{con}_{h-2} - \text{con}_{b_i-1}| + \sum_{k=1}^{2} (|\text{con}_{h+n+k-1} - \text{con}_{b+i+n+k} + d(h-1,b_i))), \quad i = 1, 2, 3, 4, 5 \tag{8}
\]

where \( B \) is the index of the best template. Consequently, the replacement is done as the following relation:

\[
f_h f_{h+1} \ldots f_{h+n-1} \leftarrow O_{b+1}, O_{b+2} \ldots O_{b+n} \tag{2}
\]

B. No occluded object, no complete period

The difference between this condition and the above condition is that we don’t have information about the most suitable template after the hole. In simpler terms, with assumption that \( B \) indicates the index of the best match, replacement is done as:

\[
f_h f_{h+1} \ldots f_{h+p} \leftarrow O_{b+1}, O_{b+2} \ldots O_{b+p+1} \quad B+p+1 < h+n-1 \tag{3}
\]

where \( p-1 \) is the length of the period of the best match found. To find the index of the starting best template, equation (11) is replaced with equation (8).

\[
B = \arg \min_i (\sum_{k=1}^{2} |\text{con}_{h-1-k} - \text{con}_{b_i-k}| + d(h-1,b_i)), \quad i = 1, 2, 3, 4, 5 \tag{4}
\]

After using relation (10), the new hole is defined as \( f_{h+p+1}, f_{h+p+2} \ldots O_{b+p+1} \). Again, equation (11) is applied in order to find \( B \), but it should be notified that now \( \text{con}_{b+p+1} \) is named as \( \text{con}_b \). To replace the found templates, relation (10) is used. It iterates until there is no hole to be filled in.

C. Occluded object, complete period

As described before, the object right one before the hole was considered to be compared with correct templates in contour-database; because no occluded object was considered there. Yet, the remaining two conditions, this condition and the next, consider partially occluded objects. Consequently, \( \text{con}_{b} \) is compared with other templates. To be specific, equations (7) and (8) change to (12) and (13), respectively.

\[
d(h, i) = |\text{con}_h - \text{con}_i|, \quad \text{where, } i = 1, 2, \ldots, N, \quad i \neq h, h+1, \ldots, h+n-1 \tag{12}
\]

\[
B = \arg \min_i (|\text{con}_{h-1} - \text{con}_{b_i-1}| + \sum_{k=1}^{2} |\text{con}_{h+n+k-1} - \text{con}_{b_i+n+k}| + d(h, b_i)), \quad \text{where, } i = 1, 2, 3, 4, 5 \tag{13}
\]

D. Occluded object, no complete period

Equation (12) is used as before, but equation (13) is replaced with (14) in order to exert not existence of a complete suitable period.

\[
B = \arg \min_i (\sum_{k=1}^{2} |\text{con}_{h-k} - \text{con}_{b_i-k}| + d(h, b_i)), \quad \text{where, } i = 1, 2, 3, 4, 5 \tag{14}
\]

E. Video Inpainting in the presence of more than one moving object

Considering more than one moving object introduces a new problem: “how to separate the goal object (the object which its motion is going to be preserved) from other moving objects?” As shown in Fig. 5, the goal object (object number 1) should be differentiated from other moving objects (objects number 2 and 3).

To do this separation, each time the object number 1 overlaps with other moving objects, we consider the other object (for example, object number 2) as the goal object and use the pre-described method to find its best match which is appropriate to be placed in destroyed frames. In another words, we temporally change the goal object with other moving objects and suppose that the new goal object is occluded by the old goal object. Therefore, we should find the most suitable template for this new occluded goal object. It is
the same for other moving occluding objects, too. Fig. 6 depicts the results of changing the goal object temporally; the green object is the best match of the new goal object.

Fig. 4. Numbering of different moving objects.

Firstly, the initial bounding box is defined as 
\((\text{minCol}: \text{minCol}+ \text{maxWidth}) \times (\text{min Row: min Row}+ \text{maxHeight})\).

These values are defined from noncorrupted frames. For destroyed frames, either partially or completely, these values are used, too. Fig. 7 shows portions of database acquired from the above mentioned method.

Fig. 5. Separation of the goal object from other moving objects. (a) interference of objects number 1 and 2. The green area is the best match for object number 2. (b) interference of objects number 1 and 3. The green area is the best match for object number 3.

To complete the hole, for white areas, the original data is retrieved. Retrieval of white areas is the same as explained before. Yet, for the green areas, only those parts of the best template are copied that have coordinates equal to those green areas. Therefore, the portions of goal object that is missed, will be completed and the remaining parts will be considered as background.

III. EXPERIMENTAL RESULTS

The videos used in our task are “jumping girl”, “three persons” and “seq2_ori”. Fig. 8 shows the results of the proposed algorithm on “jumping girl”.

Fig. 6. Two templates from partially occluded database. (a) interference of objects number 1 and 2. (b) interference of objects number 1 and 3.

The comparison between the two rows of Fig. 9 depicts that the proposed algorithm has better results. The proposed algorithm in this paper distinguishes between the static background and the moving foreground. As a result, there is no more over-smoothing for the background. Moreover, jumps that usually occur while the moving object enters the hole or exits from there, reduced.

Fig. 7. Results of the proposed video inpainting algorithm. (a) a portion of template database which include hole in it. (b) the inpainted template database.

Fig. 8. Comparison of the results of the proposed algorithm with the results of [2] (a) Results of algorithm proposed in [2]; (b) Results of the proposed algorithm. there is no more over-smoothing for the background.

The comparison between the two rows of Fig. 10 depicts that the proposed algorithm has better results. On the first row, the first and the second pictures from left have two right hands. Yet, the proposed algorithm solved this problem so that on the second row, the first and the second pictures from left have one right hand. On the first row, the third picture from left has not gotten a complete template. Yet, the proposed algorithm solved this problem so that on the second row, the third picture from left has gotten a complete template.
Fig. 9. Comparison of the results of the proposed algorithm with the results of [2] (a) Results of algorithm proposed in [2]; (b) Results of the proposed algorithm. On the first row there are incorrect numbers of hands or there is an incomplete template which is used for completion. These problems are solved by the proposed method on the second row.

The comparison between the two images of Fig. 11 depicts that the proposed algorithm has better results.

Fig. 10. Comparison of the result of the proposed algorithm with the result of [20] (a) Result of algorithm proposed in [20]; (b) Result of the proposed algorithm. The left side picture is with shadow of the occluding object. Yet, the proposed algorithm solved this problem so that for the right side picture there is no shadow of the occluding object.

The left side picture is with shadow of the occluding object. Yet, the proposed algorithm solved this problem so that for the right side picture there is no shadow of the occluding object.

IV. CONCLUSION AND FUTURE WORK.

This paper introduced a new approach for inpainting the destroyed videos. To begin with, separation of two components of video is done. Then, according to either the occluding object is stationary or moving, holes of background are completed by inpainting or correct portions copying, respectively. To fill in the corrupted portions of the foreground, an object-based inpainting approach is used. Two major problems that were considered in this study were definition of a practical similarity measure which consider different situations that may occur and the next consideration was about how to separate the goal object from other moving objects. Both of these two issues are improved based on a contour-based method.

The remaining limitations that can be improved are: consideration of non-periodic motion. This study is acceptable in the presence of periodic motion. In other case, the algorithm will not produce pleasant results. The problem of video inpainting can be more challenging by consideration of scale changing for foreground objects and dynamic background.

References

[1] S. B. Satpute, S. D. Gadge and G. T. Kadnar, "Super Resolution-based Image with Video Inpainting," International Journal of Engineering Science & Research Technology, vol. 4 no. 10, pp. 202-205, 2015.

[2] Y. Wexler, E. Shechtman, M. Irani, “Space-Time Completion of Video,” IEEE Trans. Pattern Analysis Machine Intelligence, vol. 29, no. 3, pp. 463-476, 2007. https://doi.org/10.1109/tpami.2007.60

[3] B. P. Tejasvee, R. Dhar, P. P. Rajasvee and P.B. Jawalkar, "Video Inpainting Using Image Inpainting, International" Journal of Computer Science and Mobile Computing, vol. 4, no.10, pp. 105-110, 2015.

[4] M. Bertalmio, A.L. Bertozzi, G. Sapiro, “Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting,” Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. 355-362, 2001. https://doi.org/10.1109/cvpr.2001.990497

[5] A. Pawar and A. P. Phatade, "A Effective Way to Modify Video for Deleting Foreground Object from Background Using Exemplar Based Inpainting Method," International Journal of Engineering Science & Research Technology, vol. 5, no. 2, pp. 346-356, 2016.

[6] T. K. Shih, N. C. Tang and J. N. Hwang, “Exemplar-Based Video Inpainting Without Ghost Shadow Artifacts by Maintaining Temporal Continuity,” IEEE Transactions on Circuits and Systems for Video Technology vol. 13, no. 3, pp. 347-360, 2009. https://doi.org/10.1109/tcsvt.2009.213519

[7] M. Soryani, A. Ghanbari and A. Koochahi "Dynamic Video Texture Inpainting Using Improving LDS," British Journal of Mathematics & Computer Science, vol. 4, no. 20, pp. 2872-2883, 2014. https://doi.org/10.9734/bjncs/2014/4751

[8] M. Singh and E. S. Baghla, "A Review on Object Removal Using Examplar Based Image Inpainting Technique," International Journal of Engineering Science & Research Technology, vol. 4, no. 6, pp. 472-475, 2015.

[9] Y. K. Zakir, R. Prajot, K. Krishna and K. Anish, "A Hierarchical Super Resolutioon based Video Inpainting Tool," Internation Journal of Advance Research And Innovative Ideas In Education, vol. 2, no. 2, pp. 236-240, 2016.

[10] R. T. Vinod and N. M Nitiket, "An Intelligent Video Repairing Approach Using Object Inpainting: A Review," Internation Journal of Research in Advent Technology, vol.4, no. 3, pp. 236-240, 2016.

[11] M. V. Venkatesh, S. S. Cheung and J. Zhao, “Efficient Object-Based Video Inpainting”, Journal of Pattern Recognition Letters, vol. 30, no. 2, pp. 168-179, 2009. https://doi.org/10.1016/j.patrec.2008.03.011

A. Ghanbari Talouki, M. Majdi and S. A. Edalatpanah, “Video Inpainting Using a Contour-based Method in Presence of More than One Moving Objects” International Journal of Advanced Engineering and Management, Vol. 2, No. 1, pp.37-44, 2017. 0i http://ijoem.org/00202-03
[12] S. Cheung, J. Zhao and M. Venkatesh, “Efficient Object-Based Video Inpainting,” International Conference on Image Processing, pp. 705–708, 2006. https://doi.org/10.1109/icip.2006.312432

[13] Y. Zhang, J. Xiao and M. Shah, “Motion Layer Based Object Removal in Videos,” Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION’05), vol. 1, pp. 516-521, 2005. https://doi.org/10.1109/acvmot.2005.75

[14] M. Bertalmio, G. Sapiro, C. Ballester and V. Caselles, “Image Inpainting,” Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’00, pp. 417-424, 2000. https://doi.org/10.1145/344779.344972

[15] T. Siratori, Y. Matsushita, S. B. Kang, X. Tang, “Video Completion by Motion Field Transfer,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), vol. 1, pp. 411-418, 2006. https://doi.org/10.1109/cvpr.2006.330

[16] C. R. Wren, A. Azarbayejani, T. Darrell and A. P. Pentland, “Pfinder: Real-time Tracking of the Human Body”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 780-785, 1997. https://doi.org/10.1109/34.598236

[17] Z. Zivkovic, “Improved Adaptive Gaussian Mixture Model for Background Subtraction”, Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, vol. 2, pp. 28-31, 2004. https://doi.org/10.1109/icpr.2004.1333992

[18] N. C. Tang, H. Y. M. Liao, C. W. Su, F. Huang and T. K. Shih, “Video Inpainting on Digitized Old Films”, Lecture Notes in Computer Science, vol. 5712, pp.421-430, 2009. https://doi.org/10.1109/tmm.2011.2112642

[19] A. Ghanbari and M. Soryani, “Contour-Based Video Inpainting,” 7th Iranian Conference on Machine Vision and Image Processing, pp. 1-5, 2011. https://doi.org/10.1109/iranianmvip.2011.6121586

[20] C. H. Ling, C. W. Lin, C. W. Su, H. Liao, Y. Chen, “Video Object Inpainting Using Posture Mapping”, 16th IEEE International Conference on Image Processing (ICIP), vol. 10, pp. 2785-2788, 2009. https://doi.org/10.1109/icip.2009.5414183