Video Completion Using Gray Level Co-occurrence Matrix and Object Update

Sameh Zarif 1,2, Ibrahima Faye 1,3, and Dayang Rohaya 1,2
1Centre of Intelligent Signal and Imaging Research, Universiti Teknologi Petronas, Malaysia
2Department of Computer and Information Sciences, Universiti Teknologi Petronas, Malaysia
3Department of Fundamental and Applied Sciences, Universiti Teknologi Petronas, Malaysia

Abstract: Reconstructing and repairing damaged parts after object removal of digital video is an important trend in artwork restoration. Video completion is an active subject in video processing, which deals with the recovery of the original data. Most previous video completion approaches consume more time in extensive search to find the best patch to restore the damaged frames. In addition to that, visual artifacts appear when the damaged area is large. This paper presents a video completion method without the extensive search process. The proposed framework consists of a segmentation stage based on low resolution version and background subtraction, a tracking stage based on Gray Level Co-occurrence Matrix (GLCM), and a completion stage based on object prior position and object update. The proposed method reduces the completion time to a few seconds and maintains the spatial and temporal consistency. It works well when the background has clutter or fake motion, and it can handle changes in object size and in posture.

Keywords: Video completion, video inpainting, object removal, GLCM, background subtraction.

Received March 21, 2014; accepted February 4, 2015

1. Introduction

Video completion is an interesting new research topic in multimedia computing and video processing since 2000. In recent years, transforming cultural and historical artifacts such as photographs and old films into digital format has become an important trend. Most of old videos that have historical value after digitization are very poor and often contain unstable luminance and damaged content. Videos are usually subject to degradation due to bad environment, dust and dirt, scratches, water, noise, compression or transmission, and others have logos [9, 16, 24].

The purpose of video completion is to modify and repair the damaged structure and texture areas in a non-detectable way for an observer not familiar with the original video. The challenges of video completion compared with image completion are the huge number of data to be restored and the maintenance of spatial and temporal consistency between video frames. Most of image inpainting methods are based on Partial Differential Equations (PDE) to repair the damaged area. This approach performs well on images with only structure information but fails to inpaint areas containing fine detailed texture [20]. Texture synthesis approaches were proposed for repairing large unknown areas on images with pure textures. Texture synthesis approaches could be used to repair digitized images especially if the missing area needs to be filled with some texture. Most images are neither pure texture nor pure structure. Natural images usually contain both structure and texture information. Image completion approaches combine both texture synthesis and inpainting techniques to reconstruct large holes in the image by inpainting unknown areas on the image using both texture and structure information [13, 25].

In image completion methods that work on standalone images cannot be extended to sequences and video frames without taking into account the temporal continuity with a robust tracking method. The completed frames appear inconsistent with each other because these methods take care of the spatial information, it does not simultaneously handle the continuity across consecutive frames. To better design a video inpainting process, both the spatial and temporal continuity issues have to be considered since the human visual system is more sensitive to motion distortion [18].

There are numerous applications for video completion, including video editing and film post production, wireless video transmission (e.g., recovering lost blocks), special effects (e.g., removal of unwanted objects), video restoration (e.g., scratch removal), removal of occlusions such as text, subtitle, logos; removal of information from aerial video like orientation and location. Other applications include annotation removal from medical and military sequences and video [11, 16, 24].

This paper proposes a completion method based on three main steps. The first step is using low resolution and background subtraction to separate the background from the foreground. The second step is using Gray Level Co-occurrence Matrix (GLCM) to recognize the
object of interest from the other moving objects. The third step is using the object prior position and object update to complete the recognized object from prior frame.

The remaining of the paper is organized as follows. Section 2 gives an overview of the related video completion methods. In section 3 the proposed algorithm is presented. Section 4 illustrates some examples to show the effectiveness of the proposed method. Finally, the conclusions is drawn in section 5.

2. Related Works

In recent years, researchers started to address the problems of video completion using number of standalone image completion algorithms to each video frame as independent image. One of the first efforts for video inpainting is made by Bertalmio et al. [2]. The authors considered only the spatial information in video and performed the inpainting on a frame by frame base using the method of a PDE in [3]. It works well in small structured holes, but fails to inpaint large regions in a video and does not utilize the temporal information. Jia et al. [10] proposed a method that aims to challenging the presence of occlusions in static camera scene. The authors segment a video into a moving object layer and static background layer manually. Layer segmentation and holography techniques are used to recover static background. Moving object pixels are repaired by sampling motion data to maintain the temporal coherence. This method presents artifacts at the boundaries of the damaged region.

Zhang et al. [27] presented a method that divides video frames into different non-overlapping motion layers using graph cut algorithm. After removing the undesirable video objects in each layer, an image inpainting method repairs each layer separately. The inpainted information is propagated to other frames using the known motion parameters, and then all layers are combined to restore the completed video. This method does not take into account the temporal consistency between adjacent frames and is limited to rigid bodies. Wexler et al. [22] proposed a space-time video completion method for stationary camera as a global optimization problem. It solves the inpainting problem by sampling a set of spatial-temporal patches from other frames to fill in the missing data. It optimizes the patch search process at different resolution levels using spatio-temporal pyramids and nearest neighbor algorithms. Exhaustive searching strategy for finding appropriate patches leads to high computational load. This method handles only periodic moving object that does not significantly change in scale.

Patwardhan et al. [17] pioneered a video completion method for constrained camera motion. This method constructs three mosaics for background, foreground and the corresponding optical flow based on the motion vectors. The extended exemplar-based image inpainting of [5] is used to repair the background holes. The extracted texture patches from the neighbor frames are used to complete the remaining foreground hole. It works well under some constraints in the camera motion. It fails when the object changes in size. Shih et al. [19] proposed a video completion method that divides the video frames into an intrinsic motion layer and an extrinsic motion layer. Exemplar based method of [5] is used to inpaint the damaged area in each layer by improving the match strategy using modifications on the data term. It can only handle videos that have consistent luminance and stable camera motion.

Vijay et al. [21] presented an object-based video inpainting technique that separately completes the background and the foreground. Adaptive background replacement and the image inpainting method in [5] are used to complete the background. The foreground object is formatted as an energy minimization problem. Dynamic programming and window-based dissimilarity are introduced to select the optimal candidates by solving the minimization problem. This method reduced the complexity, but still takes more time. Xia et al. [23] proposed an exemplar-based video inpainting method based on Gaussian Mixture Model (GMM) to improve finding best patches. The method consists of a pre-processing stage that separates the background from the foreground by constructing GMMs for each video frame. A Video inpainting stage completes the missing background by copying information from other frames. After that, it repairs the remaining missing area by using the exemplar-based image inpainting algorithm. This method does not do well when the background and foreground are close to each other.

Nick et al. [6] proposed a video inpainting technique to repair old damaged films. This method starts with normalizing the poor quality video frames that cause visible defects in the resulting video by averaging the intensity. Local and global motion estimation is used to construct the motion map. A motion completion stage is used to repair the damaged motion information to obtain more reference data from the video frames. A frame completion stage is applied to the damaged areas in the video frames using patch pasting and frame adjustment. Visual defects appear if the damaged area is large and shadows of undesired objects cannot be removed. Mosleh et al. [5] proposed a video completion approach based on Bandlet transform and the exemplar-based image inpainting of [5]. The completion process starts with separating the moving objects from the background using the Bilayer video segmentation method of [4]. A precise optimization in Bandlet transform is used to complete the damaged background after removing the object of interest. The exemplar-based image
Inpainting method is performed to repair the occluded part in the moving foreground. This method produces satisfying results but still takes more time in the search strategy.

Most of the recent video completion techniques suffer from the following major limitations. First, searching the source patch in spatial and temporal frames is computationally expensive and easily leads to error match and such error will accumulate and propagate to other frames. Over 90 percent of the total computational time is spent on the search for match patches [6, 21]. Second, large damaged areas cause visual artifacts in the result. Third, most methods handle only periodic moving objects that do not significantly change in scale or pose.

3. Proposed Method

As in most state of the art works on video inpainting, the proposed method assumes that the scene is taken by a stationary camera. For static camera scenes, the missing information that appears in the background after removing the moving foreground can be recovered since it is present in most of the frames. The goal of the proposed method is to fill the damaged areas with the original information using the prior positions instead of searching and estimating data from the surrounding frames. Figure 1 presents the general block diagram of the proposed method. The proposed video completion method is based on three main stages that will be detailed in the following subsections.

3.1. Segmentation Stage

The object of interest in image completion is manually selected by the user. However, it is very difficult in video to manually select the object of interest in all video frames due to the huge number of video frames. Therefore, the first step in most video completion techniques is detecting and tracking the moving objects. Accurate video completion methods depend on the quality of the detection and tracking of moving objects. This subsection highlights the segmentation stage that is a critical part of the proposed work.

Background subtraction techniques are the most common for separating moving objects from the background when the camera is stationary. The proposed method starts by performing frames median to generate an accurate background model which will be used in the background subtraction process. Frames median or temporal median determines the median pixel based on the total number of frames. The main problem of background subtraction techniques is that it is more sensitive to small motion or change derived from fake motion like lighting and tree leaf between frames especially in the outdoor scenes. Such problem is known to be difficult and significant [14]. The fake motion that derived from the unstationary background can cause the failure of the segmentation stage. The proposed method handles the fake motion problem by using a low resolution version of the images to enhance the background subtraction results. Low resolution is achieved by decreasing the frame spatial resolution using 2×2 block average. The frame size after applying the low resolution will be one quarter of the original size as illustrated in Figure 2. There are two advantages of using low resolution process before doing the background subtraction. First, it removes the fake motion derived from unstationary background. Second, it reduces the computational time of the segmentation stage by using the new frame size.

The background subtraction is performed by taking the absolute difference between the low resolution of each frame \( F(x, y, t) \) and the low resolution of the computed background \( B(x, y) \) as defined in Equation 1. Motion mask between the background and the moving object is obtained by performing thresholding on the subtraction result as defined in Equation 2.

\[
D(x, y) = |F(x, y, t) - B(x, y)| \tag{1}
\]

\[
Mask(x, y) = \begin{cases} 1 & D(x, y) > Thr \\ 0 & Otherwise \end{cases} \tag{2}
\]

After background subtraction using low resolution, the moving object may appear with hole. A Morphological dilation operation is then performed to the mask image to fill in the gaps and merge the near isolated regions.
The initial resolution is then set back to start the completion stage. Figure 3 illustrates the results of the segmentation stage with and without low resolution.

Figure 3. Segmentation results.

### 3.2. Tracking Stage

A basic requirement for an efficient object completion is that, the object of interest must be consistently known over the time. Therefore, after the detection of the moving objects is achieved, a tracking stage is needed for identifying the object of interest among the moving objects in the scene. Several types of tracking methods have been proposed in previous works. One of the common types that are used is the feature based tracking. It consists in extracting features that will be used to recognize the object, for recognition [1, 7]. Before discussing the tracking stage, the paper describes the concept of GLCM, which is used in the tracking stage as a feature.

GLCM also called Gray Tone Spatial Dependency Matrix was introduced by Haralick [8]. GLCM is a statistical approach to analyze and determine the similarity direction of gray level tones in an image. It considers a second order texture calculation that measures the relationship between two neighboring pixels in the original image. GLCM defines the probability that gray level \( i \) called the reference pixel occurs at a distance \( d \) in direction \( \theta \) from gray level \( j \) called the neighboring pixel in the image. The neighbor pixel is chosen to be the one to the right or left for the horizontal direction (\( \theta=0 \) and 180), top or bottom for the vertical direction (\( \theta=90 \) and 270), top left or bottom left for the left diagonal (\( \theta=22.5 \) and 292.5, \( \theta=135 \) and 315, or \( \theta=112.5 \) and 247.5). The angle is specified by the relationship or the offset between the reference pixel and its neighbor. The offset is represented by the number of rows and columns between the reference pixel and its neighbor as shown in Figure 4. For each image, eight GLCM matrices, representing the above mentioned directions are computed in order to determine the main directional color distribution.

Figure 4. The directions used for computing GLCM.

The first GLCM is computed according to the horizontal direction (\( \theta=0 \) for the right neighbor pixel and \( \theta=180 \) for left neighbor pixel) with distance \( d=1 \). Let \( M \) be a small section of a gray level image. Each pixel within the matrix \( M \) becomes the reference pixel in turn, starting in the upper left corner and proceeding to the lower right. The number of times the gray level \( i \) occurs at a distance \( d=1 \) from the gray level \( j \) is counted. \( Glh \) and \( Grh \) represent the right and left horizontal GLCM matrices respectively.

\[
M = \begin{bmatrix}
0 & 0 & 0 & 3
0 & 5 & 0 & 0
0 & 1 & 3 & 2
0 & 0 & 0 & 3
\end{bmatrix}
\]

\[
Glh = \begin{bmatrix}
3 & 2 & 1 & 0
0 & 5 & 0 & 0
0 & 1 & 3 & 2
0 & 0 & 0 & 3
\end{bmatrix}
\]

\[
Grh = \begin{bmatrix}
3 & 0 & 0 & 0
2 & 5 & 1 & 0
1 & 0 & 3 & 0
0 & 0 & 2 & 3
\end{bmatrix}
\]

The top left cell in \( Grh \) represents the number of times the reference pixel with grey level 0 have a right neighbor pixel with grey level 0 occurs within the image area. The next cell (value 2) illustrates when the reference pixel is 0 and its right neighbor is 1. The cells in \( Glh \) represent how many times the reference pixel have left neighbor with the same grey value.

Texture similarity calculations require symmetrical GL CM. \( Glh \) is the transpose of \( Grh \). A symmetrical matrix is obtained by adding \( Grh \) to \( Glh \). Finally, for computing the probability of occurrence, the symmetrical matrix is normalized by dividing each element by the sum of all elements, as illustrated in Equation 3.

\[
GN(i,j) = \frac{GH(i,j)}{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} GH(i,j)}
\]

\[
GH = \begin{bmatrix}
6 & 2 & 1 & 0 \\
2 & 10 & 1 & 0 \\
1 & 1 & 6 & 2 \\
1 & 0 & 2 & 6 \\
\end{bmatrix}
\]

\[
GN = \begin{bmatrix}
1 & \frac{1}{40} & \frac{1}{40} \\
\frac{1}{40} & \frac{1}{40} & \frac{1}{40} \\
\frac{1}{40} & \frac{1}{40} & \frac{1}{40} \\
\frac{1}{40} & \frac{1}{40} & \frac{1}{40} \\
\end{bmatrix}
\]

The object of interest is initially manually selected from the first frame. Its eight normalized GLCMs corresponding to the eight predefined directions are computed. After the segmentation stage separating the background from the moving objects, the connected components are used to extract the smallest bounding box for each moving object detected in the motion mask. The eight normalized GLCMs are then
computed for each bounding box detected from the current frame.

The Mean Square Error (MSE) is used to check the degree of similarity between the eight GLCMs of the object of interest and the eight GLCMs of the detected moving objects in the current frame as illustrated in Equation 4. \( Goop(i,j) \) represents one of the eight GLCMs of the object of interest, \( Gd(i,j) \) represents the corresponding GLCM of the detected moving object, \( c = 1, 2, ..., 8 \) and \( n, m \) are the size of the GLCM matrix. The average MSE for the eight GLCMs is computed for each object as illustrated in Equation 5. The object that has minimum average from the detected moving objects is selected to be the object of interest.

\[
MSE_c = \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} \left[ Goop(i,j) - Gd(i,j) \right]^2
\]  
(4)

\[
\text{AverageMSE} = \frac{1}{8} \sum_{c=1}^{8} MSE_c
\]  
(5)

### 3.3. Completion Stage

Video frames have advantages compared to a standalone image. Damaged information can be recovered if it is present in at least one of the other frames. To avoid an extensive search process, the proposed completion stage uses this advantage to complete the entire damaged area from the object prior position rather than completing individual patches.

Figure 5 presents the steps of the completion stage. Removing a moving object is complicated compared to removing a static object from video scenes. Problems such as object size and pose change, occlusion, illumination change, very similar objects etc., can potentially affect the accuracy of the tracking and completion. Extracting features that will work in all different moving objects and handling these problems is very challenging. To overcome these problems a periodic update of the initial input of the object of interest is implemented. It consists in using the recognized object from the tracking stage to update the initial object of interest for every frame. The initial object of interest after update will be most similar to the object of interest in the next frame. Therefore, the accuracy of the tracking stage to recognize the next object of interest will increase. The missing data in the first frame is completed using the available information located in the corresponding bounding box from the background model. The computed background model is used only once for the first frame. To maintain the gradual and sudden changes, the proposed completion method completes the remaining frames using the object prior position from the previous completed frame instead of the background model. As illustrated in Figure 5, the quality of the proposed method does not depend on the quality of the computed background model. The solid box represents the recognized object of interest, the dashed box represents the available corresponding box in the previous completed frame, and the dotted box represents the inpainted area in the current frame.

![Figure 5. Completion stage. Sample frames in the first row, motion mask in the second row, and completed frames in the third row.](image)

The advantage of using the prior position in the completion is to maintain the inpainted video from using an extensive search process to find the best data from spatial and temporal frames. In addition to that, using the entire object completion instead of individual patch search avoids flickering and ensures the temporal consistency. The proposed method also does not suffer from merging the new data into the missing area that causes differences around the boundary. In consequence, the completed area is very smooth.

The stages of the proposed method can be summarized in Algorithm 1.

**Algorithm 1: Proposed completion method.**

- **Input** video scene, initial object of interest;
- **Generate** the background model;
- **for** each frame **do**
  - **Compute** GLCM for the initial object of interest;
  - **Background subtraction** using low resolution;
  - **Generate** the motion mask using thresholding;
  - **Perform** dilation;
  - **Separate** moving objects using object labeling;
- **for** each moving object detected **do**
  - **Compute** GLCM for the moving object;
  - **Calculate** MSE between steps 4 and 10;
  - **Select** the object that has minimum MSE as in Equation 5;
- **End for**
Update initial object of interest using the selected object; Complete the missing data using prior position. End for end

4. Discussion of Results

The proposed method has been implemented using MATLAB R2009a and tested on a PC with Pentium Dual Core 2.8 GHz CPU and 2 Gigabyte memory. Because of the changes in the inpainted frames in structure, texture, geometric attributes, and hole size, it is very difficult to evaluate the quality of the inpainting techniques by traditional objective evaluation such as Peak Signal to Noise Ratio (PSNR), MSE, and Structural Similarity (SSIM). The quality of image and video inpainting depends on the human visual perception system rather than mathematical measures [11, 12, 26].

The existing video completion methods are still few and the original video data are not available in most cases to make a comparison. To evaluate the proposed method, the following challenges, which constitute limitations for most of the related works, are considered:

1. Non-Stationary Background.
2. Object Change in Scale.
3. Object Change in Posture.
4. Static Object Occludes the Object of Interest.
5. Illumination Changes.
7. Processing Time.

The compressed version of the original and inpainted video scenes of the proposed method is uploaded as a supplementary material. Table 1 summarizes the details of the examples used in this work before compression.

<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Resolution</th>
<th>No. of Frames</th>
<th>Completion Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 and 7</td>
<td>480x270</td>
<td>405</td>
<td>32 Sec.</td>
</tr>
<tr>
<td>6</td>
<td>596x336</td>
<td>95</td>
<td>7 Sec.</td>
</tr>
<tr>
<td>8</td>
<td>480x270</td>
<td>360</td>
<td>25 Sec.</td>
</tr>
<tr>
<td>9</td>
<td>300x100</td>
<td>240</td>
<td>16 Sec.</td>
</tr>
<tr>
<td>10</td>
<td>480x270</td>
<td>90</td>
<td>6 Sec.</td>
</tr>
<tr>
<td>11</td>
<td>596x336</td>
<td>350</td>
<td>32 Sec.</td>
</tr>
</tbody>
</table>

4.1. Handling Non-Stationary Background

This subsection highlights the challenge of non-stationary background and how the proposed method overcomes it. Most of outdoor scenes have non-stationary background due to fake motion like waving trees and lighting. Such movements can cause the failure of the segmentation and tracking stages. Figures 3 and 5 illustrate the efficiency of the proposed method in handling an outdoor scene containing non-stationary background by using low resolution before background subtraction. Most previous video completion techniques have a weakness with non-stationary background.

4.2. Handling Object Change in Scale

Moving objects that significantly change in scale constitute a very challenging problem in most of video completion methods. The matching process fails to return the best patch for most of the methods that use patch searching and patch matching. The proposed method removes the search and match process by using object prior position and initial input update. Figure 6 demonstrates a real life scene having a moving bike that changes in size and a normal walking person. The red circle in the first row represents the unwanted object to be removed. The results in the second row show that the proposed method successfully removed the moving bike and completed the background.

4.3. Handling Object Change in Posture

Object changing in pose is always a big challenge in the field of pattern recognition. Red circle in figure 7 shows an example of removing foreground object that significantly changes in pose for a video sequence that contains multiple objects. The moving object changes in pose but also the movement is not periodic. The proposed method uses the updated version of the object of interest and GLCM to recognize the object that changes in pose. It also completes the hole after removing the object from the previous completed frame to maintain the smoothness. The second row in figure 7 demonstrates that the proposed method works well when the moving object changes in pose and moves in a non-periodic motion.

Figure 6. Bike removal original frames in the first row, completed frames in the second row.

Figure 7. Object change in posture example. original frames in the first row, completed frames in the second row.
4.4. Handling Static Occlusion

Occlusion is a very challenging problem in video inpainting. The object of interest, in most of real life videos, may be occluded by other static objects from the background or other moving objects. The proposed method works well when the object of interest is occluded by a static object or occludes a static object. It fails when the object is occluded by, or occludes, other moving objects. Figure 8 shows the efficiency of the proposed work in removing moving object occluded by long grass and reconstructs the grass. Figure 9 presents an example for moving jumping girl occluded by another girl who moves her hands. The second row in Figure 9 shows that the proposed method works well in the first two frames but have slight defects in the other frames. The challenges in this example are in the jumping motion, bad background model and the occlusion of the two moving girls. The proposed work overcomes the jumping motion and the bad background model by using the prior position from the previous completed frame as mentioned before and initial object update. The defects that appear in the girl’s hands in the last three frames are due to the occlusion between the hands motion and the moving object.

Figure 8. Static occlusion example. original frames in the first row, completed frames in the second row.

Figure 9. Dynamic occlusion example. original frames in the first row, completed frames in the second row.

4.5. Handling Illumination Changes

Light and environment changes during the day hours, sudden switch of light, and shadows during the day hours constitute a real challenge. Most techniques that depend on the background model fail to repair the illumination changes. Some methods need to update the background model every few frames. These techniques take more time. The proposed method depends on the previous completed frame, not on the background model. Therefore, it does not need to update the background model every time. The previous completed frame already contains any slight changes that happened in the background. Figure 10 illustrates an example for indoor video scene that has sudden changes due to switching light on. The results that appear in the second row of Figure 10 are satisfactory but not perfect due to the shadow that appears when switching on the light.

Figure 10. Sudden illumination change example. original frames in the first row, completed frames in the second row.

4.6. Loss of Tracking

Tracking and recognizing the object of interest from multiple moving objects is very important and difficult in video completion. All the above-mentioned challenges are the main reasons for the loss of tracking. Most of previous related works assume that the scene consists of stationary background and moving objects don’t change in size and pose. The quality of the completion process depends on the quality of the tracking of the object of interest. The proposed work uses an object update process to update the initial object of interest to be closer to the object of interest in the next frame. The object update process reduces the loss of tracking. Figure 11 illustrates the effectiveness of the proposed work in tracking and recognizing objects from a video scene which contains multiple objects.

Figure 11. Multiple moving objects scene. original frames in the first row, completed frames in the second row.

4.7. Processing Time

Processing time is the most challenging problem in all video completion techniques. The processing time in most of the previous video completion methods are very high compared to the proposed method. The
performance of the proposed work is near to be real time. The method in [11, 22] takes 17 minutes and 4 hours respectively to repair 100 frames at 320×240 resolution on P-4 machine. The method in [17] takes about 15 minutes to complete 50 frames video at 320×240 resolutions by using C++ and P-4 machine. The method in [21] takes 10 to 20 minutes to complete jumping girl example that mention in Figure 9. It uses MATLAB and P-4 with 4 Gigabyte memory machine. The complexity analysis and the processing time of the methods in [11, 17, 21, 22] are located in [11, 21].

Table 1 illustrates the processing time of the proposed method. The big difference in the processing time is mainly because of the removal of the extensive patch search process and the similarity matching between patches that take more time as mentioned before. Optimizing the code and porting it to C++ could increase the performance. Table 2 shows that, despite a bigger number of frames and a bigger resolution, the proposed method is still much faster than the others.

Table 2. Processing time comparison between the proposed work and the others.

<table>
<thead>
<tr>
<th>Method</th>
<th>Machine</th>
<th>No. of Frames</th>
<th>Resolution</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>In [11]</td>
<td>Pentium 4</td>
<td>100</td>
<td>320×240</td>
<td>17.5min</td>
</tr>
<tr>
<td>In [22]</td>
<td>Pentium 4</td>
<td>100</td>
<td>320×240</td>
<td>1Hr</td>
</tr>
<tr>
<td>In [17]</td>
<td>Pentium 4</td>
<td>50</td>
<td>320×240</td>
<td>15min</td>
</tr>
<tr>
<td>In [21]</td>
<td>Pentium 4</td>
<td>240</td>
<td>300×100</td>
<td>480sec</td>
</tr>
<tr>
<td>Proposed</td>
<td>Pentium 4</td>
<td>405</td>
<td>480×370</td>
<td>125sec</td>
</tr>
</tbody>
</table>

5. Conclusions

In this work, a novel fast and efficient video completion method based on GLCM, object update, and prior position completion has been proposed. The proposed method removes the extensive search process which is commonly used in previous techniques and which takes more time and leads to error accumulation. Also, it effectively maintains the spatial and temporal consistency by using the original data from the prior completed frame. Moreover, it works well when the object size and pose change, illumination change and with a non-stationary background. The proposed method has a weakness in separating the moving objects when occlusion happens. Also, it cannot handle object shadow. The future work consists in extending the proposed method to solve the occlusion problem, shadow, and handling moving camera scenes.

References


Sameh Zarif received his BSc and MSc degrees in information technology from Menofia University, Egypt, in 2005 and 2009 respectively. He is currently pursuing the PhD in computer and information sciences from centre of intelligent signal and imaging research (CISIR), UniversitiTeknologi PETRONAS (UTP), Malaysia. In addition to his current research into image inpainting, texture synthesis, and image completion, his interests lie in image processing, computer vision, and computer graphics.

Ibrahima Faye received his BSc, MSc, and PhDin Mathematics from University Paul Sabatier of Toulouse, France and specialized Master in in Engineering of Medical and Biotechnological Data, from EcoleCentrale Paris. He is currently associate professor at the department of Fundamental and Applied Sciences ofUniversitiTeknologi PETRONAS (UTP), Malaysia. His research interests lie in Engineering Mathematics, Signal and Image Processing, Multiresolution Analysis, Pattern Recognition, and CAD Systems for Breast and Lung Cancer detection. He is member of IEEE EMBS and French Mathematical Society.

Dayang Rohaya received her Bachelor Science degree in Computer Science from University of Nebraska, Omaha, USA in 1985 and Masters of Science degree inComputer Science from Western Michigan University in 1987. She completed her Doctor of Philosophy from Loughborough University 2005. Her research specializations are VirtualReality in Education & training, human factors in VR, Augmented Reality in Entertainment and Education. Currently she is an Associate Professor in the Department of Computer andInformation Sciences at University Technology Petronas, Malaysia.