Video Inpainting on Digitized Vintage Films via Maintaining Spatiotemporal Continuity

Nick C. Tang, Chiou-Ting Hsu, *Member, IEEE*, Chih-Wen Su, Timothy K. Shih, *Senior Member, IEEE*, and Hong-Yuan Mark Liao, *Senior Member, IEEE*

***Abstract*—Video inpainting is an important video enhancement technique used to facilitate the repair or editing of digital videos. It has been employed worldwide to transform cultural artifacts such as vintage videos/films into digital formats. However, the quality of such videos is usually very poor and often contain unstable luminance and damaged content. In this paper, we propose a video inpainting algorithm for repairing damaged content in digitized vintage films, focusing on maintaining good spatiotemporal continuity. The proposed algorithm utilizes two key techniques. *Motion completion* recovers missing motion information in damaged areas to maintain good temporal continuity. *Frame completion* repairs damaged frames to produce a visually pleasing video with good spatial continuity and stabilized luminance. We demonstrate the efficacy of the algorithm on different types of video clips.**

***Index Terms*—Frame completion, motion completion, motion estimation, video inpainting.**

I. Introduction

In recent years, transforming cultural and historical artifacts such as photographs and vintage films/videos into digital format has become an important trend. However, because of their age, the visual quality of such images and videos after digitization is usually very poor and often contain unstable luminance and damaged content. Video enhancement techniques widely used to restore the visual content of vintage films include video denoising [27], video stabilization [23], and video inpainting [9], [17], [18], [24], [28], [31], [35], [36]. Video inpainting, one of the most challenging techniques, helps users remove undesirable objects and repair areas where content is missing or damaged.

Manuscript received December 24, 2009; revised June 07, 2010 and September 27, 2010; accepted January 24, 2011. Date of publication February 10, 2011; date of current version July 20, 2011. This work was supported in part by Taiwan E-learning and Digital Archives Programs (TELDAP) sponsored by the National Science Council of Taiwan under NSC Grant: NSC99-2631-H-001-020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pascal Frossard.

N. C. Tang is with the Institute of Information Science, Academia Sinica, Taipei, Taiwan (e-mail: nickctang@iis.sinica.edu.tw).

C.-T. Hsu is with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan (e-mail: cthsu@cs.nthu.edu.tw).

C.-W. Su is with the Department of Information and Computer Engineering, Chung Yuan Christian University, Taipei, Taiwan (e-mail: lucas@cycu.edu.tw).

T. K. Shih is with the Department of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan (e-mail: timothykshih@gmail.com).

H.-Y. M. Liao is with the Institute of Information Science, Academia Sinica and National Chiao Tung University, Taipei, Taiwan (e-mail: liao@iis.sinica.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2011.2112642

To deal with image inpainting problems, researchers initially focused on removing or repairing small regions of an image using interpolation or smoothing techniques. Subsequently, more powerful methods were developed to perform image inpainting on large continuous areas [11], [14], [32]. For example, Criminisi *et al.* [11] proposed an exemplar-based approach for repairing large continuous areas and obtained a reasonably good quality image inpainting effect. The approach takes a block as the basic unit and utilizes the concepts of priority maps and confidence levels to guide the inpainting process. A block with a higher confidence value indicates a lower degree of damage, so the block has a higher priority in the inpainting process. Textural information is then propagated from the surrounding areas to repair damaged regions. In [32], Sun *et al.* posited that most natural or artificial objects can be described approximately by some representative curves. In other words, the salient regions of an image can be sketched before their textural characteristics are introduced. The algorithm produces excellent inpainting results by drawing a few simple representative curves.

In recent years, researchers have extended these well-developed image inpainting techniques to the repair of videos. An intuitive approach involves applying image inpainting techniques to each video frame so that the completed frames are visually pleasing when viewed individually. However, this approach neglects the issue of continuity across consecutive frames, so the quality of the resulting video is usually unsatisfactory. To resolve the problem, both spatial (intra-frame) and temporal (inter-frame) continuity must be considered in a video inpainting process. Video inpainting has become more popular because of its potential applications in our daily life [9], [17], [18], [20], [24], [28], [31], [35], [36]. Venkatesh *et al.* [9] proposed an efficient video object inpainting algorithm to inpaint partially and completely occluded objects and their algorithm can maintain motion consistency by using sliding window registration and dynamic programming. In [24], Patwardhan *et al.* extended the image inpainting concept proposed in [11] to deal with digital videos. This approach first separates the background and foreground of a video and then generates the corresponding optical-flow mosaics. After inpainting the background of the video sequence, holes in the foreground are filled with patches extracted from adjacent frames directly by a texture synthesis process. The method has limited applicability because it only works well under certain types of constrained camera motion. Zhang *et al.* [36] used a graph cut algorithm to divide a video sequence into multiple layers based on the motion in each layer. Each layer is then inpainted by applying the proposed image inpainting algorithms. The drawback of this approach is that temporal consistency is not addressed. In [20], Kokaram and Godisll employed a 3-D autoregressive model to detect and reconstruct missing video data. The method uses an interpolation technique instead of patch duplication. Only small missing regions can be repaired and the issue of maintaining temporal consistency is not addressed. Jia *et al.* [17] proposed a two-phase sampling and alignment video inpainting approach that predicts motion in the foreground before repairing damaged foreground areas and adopted an image inpainting technique to repair damaged areas of separated background. Subsequently, they extended their algorithm to handle situations with varying illumination [18]. The illumination mask used in [18] regulates the intensity of inpainted frames until it is similar to the original video. However, intensity flickers are viewed as visual defects in vintage films. Therefore, when we do inpainting on damaged vintage films, we not only recover the missing content but also stabilize the intensity change across consecutive frames. The objective of the above-mentioned moves is to guarantee the recovery of visually pleasing results.

Most of the above-mentioned algorithms discussed the use of image inpainting techniques to repair damaged background areas in videos. However, if the damaged areas are too large, visual defects are still evident in the resulting videos. To address this problem, Wexler *et al.* [35] proposed optimizing the patch search process at different resolution levels. Holes in a frame are filled using different portions of the same video. Although the inpainting process yields high-quality results in small-sized videos, the method is time consuming and computationally expensive. Moreover, information about the missing content in every video frame must be provided in advance. In [28], Shen *et al.* proposed to construct motion manifolds of space-time volume and apply structure propagation methods to recover the missing portions of foreground object and background and maintain spatiotemporal continuity. Although the output is acceptable, the method does not work well when the missing portions of a space-time volume are large. Shiratori *et al.* [31] proposed to complete a damaged video by transferring motion fields sampled from other portions of the video. The limitation of this method is that it works only on stationary video and may easily cause over-smoothing artifacts. In a previous work [30], we proposed a video inpainting algorithm that segments a video into an intrinsic motion layer (created by the video camera) and an extrinsic motion layer (created by the moving object) and then removes the selected areas from different layers. The limitation of this method is that it can only handle videos that have consistent luminance and are recorded under stable camera motions such as panning.

Because restoration of digitized vintage films is an important application area, researchers have also developed video enhancement techniques especially for vintage films. For example, in [19], Joyeux *et al.* proposed a line scratch detection and removal algorithm. Although the line scratch method is very efficient, the authors only use an image interpolation method to repair damaged content. In [15], Gullu *et al.* used temporal coherence analysis to detect scratches in video images. Both methods [15] and [19] can only deal with small regions around defects. Machi and Collura [22] proposed using spatiotemporal analysis techniques to repair single frame defects, but they neglected the issue of maintaining temporal continuity.

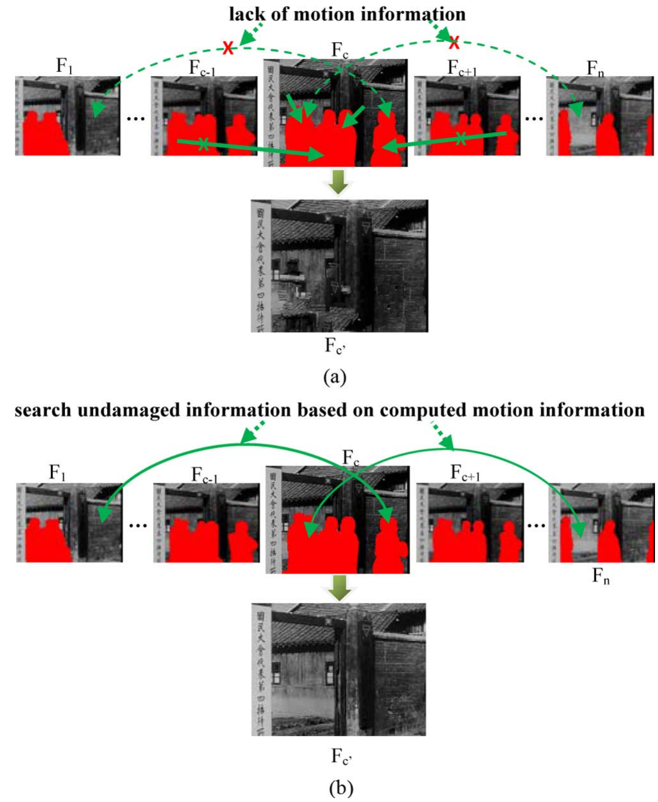


Fig. 1. Inpainting results produced by previous methods and our proposed method. (a) Existing video inpainting methods applying image inpainting-related techniques to inpaint a severely damaged video sequence tend to produce poor inpainting results due to a lack of accurate motion information. (b) Our proposed method can extract undamaged information from the entire video sequence based on computed motion information to produce good inpainting results.

Current inpainting methods can be divided into two approaches. The first approach inpaints the damaged areas of an image using only data from the same frame. The second approach searches both the current and neighboring frames to find reference data for use in inpainting. To maintain temporal continuity, both approaches use a simple motion estimation process to compute motion information, then propagate the inpainted information along the motion to neighboring frames. However, when dealing with severely damaged frames, both approaches usually yield poor inpainting results. In such cases, the undamaged areas in the current frame or neighboring frames are comparatively small, so the inpainting process has difficulty finding undamaged reference information due to the lack of accurate motion information. As a result, the inpainting process must use the same reference block repeatedly to inpaint missing areas and propagate the information to other frames. In addition, if the motion information is inaccurate, the inpainting process may also have to adjust the target block to maintain spatial continuity.

In our experiments, we found that when using existing video inpainting techniques to repair old films or remove undesirable objects, the unstable luminance and poor quality of the original film frequently cause visible defects in the resulting video. As a result, we believe a new approach is needed to tackle the challenges presented by old films as well as by modern digital videos. To this end, we propose a video inpainting algorithm

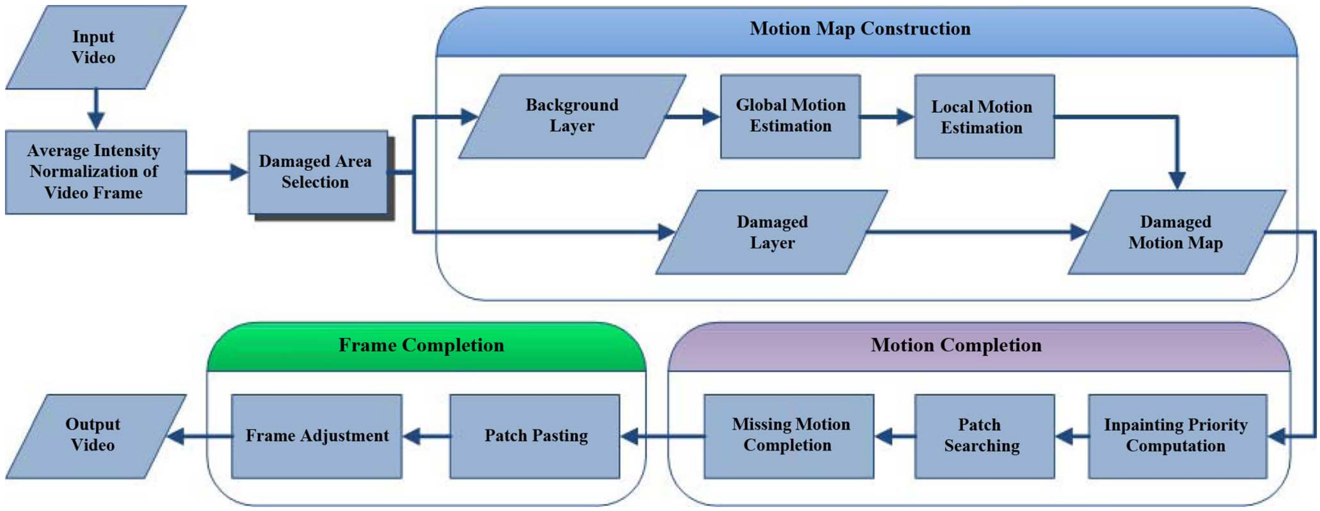


Fig. 2. Proposed framework.

that can address those challenges and produce visually pleasing results. When inpainting severely damaged videos, we begin by filling gaps in the temporal information to help the inpainting process obtain more reference data from the whole video sequence. Our proposed video inpainting algorithm involves two key steps: motion completion and frame completion. The first step, *motion completion*, tries to replace missing motion information to help the inpainting process obtain reliable reference data. The second step, *frame completion*, maintains the spatial continuity of the referenced content before it is pasted onto the corresponding missing area. This step is especially important when the luminance in old films is unstable.

Fig. 1 shows some examples of using existing video inpainting methods [17], [18], [24] applying image inpainting-related techniques to inpaint a severely damaged video sequence. In Fig. 1(a), the first and last frames contain undamaged reference information. However, without accurate motion information, the inpainting process can only use information derived from the current and/or neighboring frames to repair missing areas. The example shows how relying on spatial information from a single frame may result in poor inpainting results. Fig. 1(b) shows how, with complete motion information, the inpainting process can extract undamaged information from the entire video sequence and find reliable reference data to repair missing areas. In addition, experiment results demonstrate motion completion also significantly improves the temporal continuity of the final result.

Fig. 2 presents our proposed framework, which is comprised of three procedures: motion map construction, motion completion, and frame completion. Motion map construction is a preprocessing procedure. We begin by manually labeling damaged areas in vintage films to divide each succeeding video frame into a damaged layer and a background layer. The former shows the missing area and the latter shows the rest of the video content. Next, we estimate the motion information located in the background layer to construct a motion map for each frame. These maps form the basis of our video inpainting process and are used to replace the missing motion information in the motion completion procedure. Finally, the frame completion procedure uses a patch adjustment mechanism to paste data from neighboring or current frame onto the missing areas indicated in the damaged layer.

The remainder of this paper is organized as follows. Section II describes the construction of the motion map. Section III presents the proposed video inpainting algorithm. Section IV details the experiment results and Section V contains concluding remarks.

II. Motion Map Construction

Accurate motion information is the key to achieving good video inpainting results. Assuming that true motion information for the damaged layer and the background layer are available, the video inpainting process can easily search the background layer by tracking the true motion to find the best reference data to complete the missing areas in the foreground. Based on this assumption, we begin by constructing a motion map to track the motion information in the background layer and then repair the missing motion information in the motion completion step. We employ a two-step preprocessing procedure to construct a motion map. First, since unstable luminance tends to degrade motion estimation, we adopt an intensity normalization procedure to stabilize the video content. Second, we use global and local motion estimation to construct a motion map to preserve the motion flow of each block. Though using only intensity value as a criterion to judge similarity in vintage films may cause error matching problems, the two-step motion estimation procedure can really help improve estimation accuracy.

A. Intensity Normalization

Similarity comparison or template matching is the most commonly used search procedure in block-based motion estimation algorithms. However, since the procedure is very sensitive to changes in luminance, we need to normalize the average intensity of every frame before performing motion estimation. The normalization procedure first computes the difference between the average intensity, I_{diff}, of two consecutive frames, F_t and F_{t+1}, by

$$I_{diff} = \frac{\sum_{p} I(p)}{|F_t|} - \frac{\sum_{p'} I(p')}{|F_{t+1}|}, \ \forall p \in F_t, \ \forall p' \in F_{t+1} \qquad (1)$$

(a)

(b)

Fig. 3. Normalization of the average intensity of video frames. (a) Original video sequence. (b) Result of average intensity normalization.

where p and p' denote undamaged pixels in frame F_t and frame F_{t+1}, respectively; $I(p)$ represents the intensity of pixel p and $|F_t|$ and $|F_{t+1}|$ denote the total number of pixels in frame F_t and frame F_{t+1}, respectively. Based on I_{diff} , we adjust the intensity of each pixel in frame F_t to reduce the difference between the average intensity of F_t and F_{t+1}. Fig. 3 shows an original video sequence [Fig. 3(a)] and the sequence after intensity normalization [Fig. 3(b)].

B. Construction of Motion Map

After stabilizing the intensity of every frame, we estimate the motion flow for the remaining background layer. First, we apply a global motion estimation (GME) process to estimate the global motion between two adjacent frames. The result is then used by the proposed local motion estimation (LME) process to reduce the computational complexity and mismatch rate.

1) Global Motion Estimation: Two types of GME have been well studied, namely, feature-based methods [3], [5], [6], [38] and gradient-based methods [1], [7], [37]. Since feature-based GME techniques are generally more efficient, we use the Lucas-Kanade optical flow computation technique [21] in our algorithm. This well-known technique estimates the optical flow between two consecutive frames efficiently; however, it only works well when the pixel replacement task is small. To improve the estimation capability, Bouguet [2] developed an iterative Lucas-Kanade optical flow algorithm for each resolution. We employ Bouguet's pyramidal version to estimate global motion. In addition, we implement a feature extraction preprocess [29] to select features with good texture properties for our GME procedure. Fig. 4 shows an example of global motion estimation.

2) Local Motion Estimation: We propose a correlation-based motion estimation algorithm with a correction mechanism that computes block motion vectors and removes undesirable motion simultaneously. We use a larger block (16*16) to compute a motion vector and then refine the result to a smaller-sized block (8*8). Our local motion estimation (LME) procedure, which is

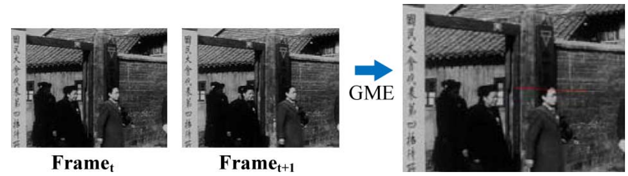


Fig. 4. Results of global motion estimation on an old film.

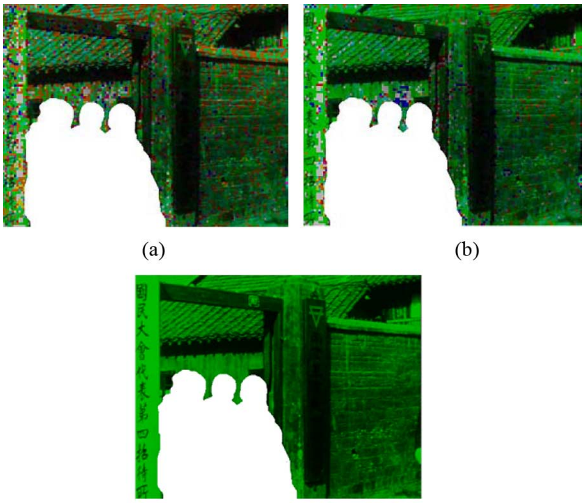
5	4	3	4	5
4	2	1	2	4
3	1		1	3
4	2	1	2	4
5	4	3	4	5

Fig. 5. Mask for initial motion selection during re-estimation.

based on the modified motion estimation procedure proposed by Cheung and Po [8], uses the initial motion obtained in the previous GME step. Although Cheung and Po's algorithm is efficient and robust in most cases, there may be some unreliable motion vectors in the first estimation phase. Hence, we propose using a re-estimation technique to identify such vectors and improve the estimation accuracy. The steps of the proposed local motion estimation algorithm are as follows.

1) Use the modified CDHS algorithm proposed in [8] with the initial motion computed in global motion estimation step to compute the block motion vectors for frame F_t. The distance between a source block B_i and a target block B_j in the HSI color space is calculated by
$$dis(B_i, B_j) = HSI_{SSD_{(i,j)}} \tag{2}$$
where the distance in each HSI color component is calculated individually by the sum of squared difference (SSD) shown in (3) at the bottom of the page.
2) Copy all blocks from F_t to generate a pseudo frame F'_{t+1} based on the motion vectors calculated in step 1. Frame F'_{t+1} is used to identify poorly estimated blocks by comparing its contents with those of frame F_{t+1}.
3) Compare the differences between F_{t+1} and F'_{t+1}. The differences can be viewed as a map of poorly estimated blocks.
4) For each poorly estimated block indicated in step 3, re-estimate the motion vectors by the CDHS algorithm with the initial motion obtained from surrounding valid motion vectors. Fig. 5 shows a mask used to perform initial motion selection. One can search the surrounding blocks (ordered by digits from 1 to 5) to identify a valid motion vector and then use it in the search process. However, if all surrounding

$$HSI_{SSD_{(i,j)}} = \sum_{a=0}^{p=1} \sum_{b=0}^{q=1} \sqrt{(HSI_n(i+a, j+b))^2 - (HSI_{n+1}(i+a+dx, j+b+dy))^2} \tag{3}$$



(a) (b)

(c)

Fig. 6. Motion maps obtained with different motion estimation methods. (a) By the local motion estimation method 4SS [26]. (b) By the proposed local motion estimation method. (c) By the proposed global motion estimation and local motion estimation method.

blocks are poorly estimated, one can randomly choose a direction that has a smaller estimation error to proceed.

5) Construct a motion map based on the final motion vectors calculated for each block.

Fig. 6 shows the results of implementing different motion estimation techniques. We use different colors to indicate different motion directions (e.g., green indicates "moving down" and red indicates "moving up"). In this example, all areas where motion information is missing are shown in white. As shown in Fig. 6(a), a lot of undesirable motions are produced when we only use a normal local motion estimation algorithm [26]. Fig. 6(b) and (c) shows that the proposed motion estimation algorithm reduces the number of undesirable motions effectively and estimates the motion vectors for each block precisely.

The constructed motion map of a frame reflects the motion flow of every block in that frame. This information can be used as a guideline when seeking available information from other frames and to maintain temporal continuity. After constructing the motion maps of each frame, we apply our motion completion procedure to recover missing motion information.

III. Video Inpainting by Maintaining Spatiotemporal Continuity

Most video inpainting methods use a search process to find the most suitable patch in the current frame or neighboring frames. After the best-matched result has been found, the target patch is pasted directly onto the missing area (to maintain good spatial continuity) and then propagated to other frames (to maintain temporal continuity). However, three problems often arise with this type of video inpainting process. First, as mentioned earlier, the process may fail to find reliable spatial information to inpaint missing areas when motion information is not available. Second, propagation of patches based on inaccurate motion degrades the quality of the resulting video. If a reliable patch cannot be found, some video inpainting processes employ a one-step motion segmentation procedure to trace the motion flow of

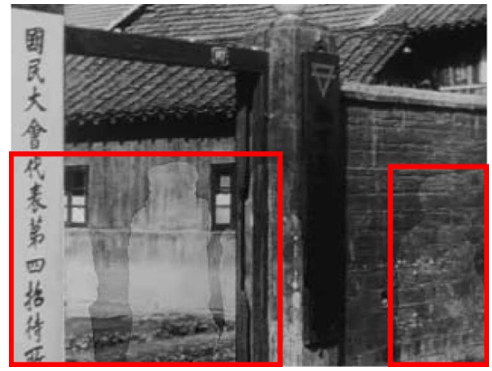


Fig. 7. Visual defects generated by pasting image data into onto a missing area directly.

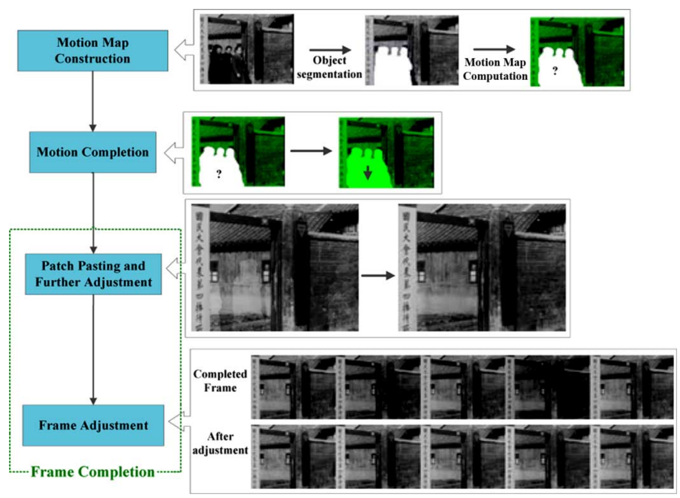


Fig. 8. Flowchart of the proposed motion and frame completion procedure.

each block. However, since there are usually noisy or irregular motions in the segmentation results, such processes may propagate target patches derived from inaccurate motion information to other frames. Inaccurate propagation usually results in mismatched boundaries and also destroys the temporal continuity. Third, pasting the most similar patch onto the missing area may result in visual defects, especially in old videos with unstable luminance. Because the patch search procedure only looks for the "most similar patch" in terms of the observed luminance, there is no guarantee that the selected patch will maintain good spatial continuity in the inpainted video. Fig. 7 shows an example of visual defects generated when best-matched patches are pasted onto a damaged area directly.

From these reasons, we conclude that spatiotemporal continuity is the key to guaranteeing visually pleasing video inpainting results. As previously discussed, our proposed video inpainting algorithm implements two key procedures, a motion completion procedure and a frame completion procedure, which try to inpaint damaged video content while maintaining good temporal continuity of the inpainted video. The flow chart of the algorithm is shown in Fig. 8. After constructing a motion map of each frame (as described in Section II), we apply the motion completion procedure to recover areas where motion information is damaged (e.g., the white area in Fig. 6). The motion map

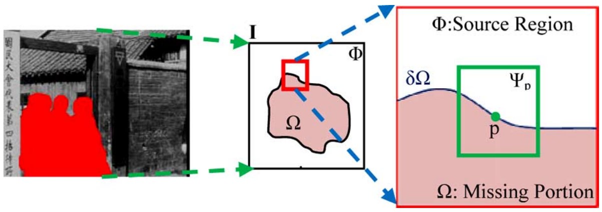


Fig. 9. Notations used in our the motion completion procedure.

is updated when the missing motion vectors are obtained. Then, we use the updated map to find the best-matched patch from either the current frame or other video frames based on the motion vectors in the map. Furthermore, to enhance the visual quality, we employ a patch adjustment process to regulate the content of the target patch, followed by a luminance stabilization process to stabilize the luminance of each frame.

A. Motion Completion

The objective of motion completion is to recover the motion information for missing areas by propagating the motion vectors obtained in the motion map construction step. In our video inpainting algorithm, the basic unit is a patch, which we define as a block. Motion completion involves two procedures: priority computation and patch searching. First, the priority computation process determines the completion order for each missing block. Second, the patch search process tries to find the most similar patch in the spatial and temporal domains to recover the missing information.

1) Priority Computation: Priority computation, a concept proposed by Criminis *et al.* [11], determines the inpainting priority of each patch. In [23], Patwardhan *et al.* first extends this concept into video painting procedure to repair missing portions on motion mosaics. In our previous work [33], we introduced the priority computation procedure with modified data term into a video inpainting procedure to repair missing portions of aged films. Although the results are acceptable, there are still some visual defects generated by high patch reusing rate during the inpainting process. To incorporate the priority computation concept into our video inpainting process properly, we extend the original 2-D spatial domain to a 3-D video space (i.e., the 2-D spatial domain plus the temporal domain) to ensure that the patch with the least damage along the spatial and temporal dimensions is inpainted first. We also include a weighting factor to ensure that the inpainting process will propagate unused information to missing areas, rather than reuse information already used for other patches. Finally, we modify the data term defined in [11] to speed up the computation and better maintain the information structure.

We now describe the motion completion procedure in detail. As shown in Fig. 9, let $\boldsymbol{I}$ be a video frame that comprises a missing portion $\boldsymbol{\Omega}$ and a source region (i.e., the remaining areas) $\boldsymbol{\phi}$, that is, $\boldsymbol{I} = \phi \cup \Omega$. The notation $\delta\Omega$ denotes the front contour on Ω and Ψ_p is an arbitrary patch centered at a pixel $p \in \delta\Omega$.

We define the priority term $P(p,t)$ for the patch centered at pixel p in frame F_t by

$$P(p,t) = C(p,t) * D(p,t) * W(F_t) \tag{4}$$

where $C(p,t)$, $D(p,t)$, and $W(F_t)$ denote the confidence term, data term, and weighting factor, respectively.

The weighting factor $W(F_t)$ measures the percentage of the source area available in each frame. A higher $W(F_t)$ value indicates that the frame F_t contains more source data; thus, it has a higher inpainting order. The definition of $W(f_t)$ is as follows:

$$W(F_t) = \frac{|\Phi_t|}{|I_t|} \tag{5}$$

where $|I_t|$ is the size of frame F_t and $|\phi_t|$ is the size of the source area ϕ_t in frame F_t.

The confidence term $C(p,t)$ measures the percentage of source data in a target patch $\Psi_{(p,t)}$. Before we compute the term, we initialize the confidence value of each pixel in $\boldsymbol{I}$ by

$$C_0(p,t) = \begin{cases} 1.0, & \text{if and only if } p \in \Phi \\ 0.0, & \text{if and only if } p \in \Omega. \end{cases} \tag{6}$$

As shown in Fig. 9, Ψ_p is a patch centered at pixel p. We define the confidence $C(p,t)$ of an arbitrary patch Ψ_p as follows:

$$C(p,t) = \frac{\sum_{q \in (\psi_{(p,t)} \cap \Phi)} C_0(q,t)}{\left|\psi_{(p,t)}\right|}, \quad \forall p,\ q \in F_t,\ p \in \partial\Omega_t,\ q \in \psi_{(p,t)} \tag{7}$$

where $|\Psi_{(p,t)}|$ is the size of $\Psi_{(p,t)}$.

A patch with a higher confidence value means its missing area is smaller and its priority in the video inpainting process is higher. However, the confidence term only indicates the degree of damage in a single patch; in other words, it does not provide higher level structural information about all patches. Therefore, the term can only play an auxiliary role by providing a rough guideline on where to start the inpainting process. Usually, a number of patches have high confidence values. To determine which one has the highest priority, we introduce a data term to strengthen the confidence term and solve the above-mentioned problem.

We use the data term to assess the importance of a target patch. Criminisi *et al.* [11] used the strength of isophotes to judge whether a target patch is important. Since the time complexity of computing isophotes is high, we propose computing the data term based on an edge map. Our modification of the data term defined in [11] is based on the observation of continuous structure derivation. First, we apply the Canny edge detector on $\boldsymbol{I}$ to obtain a binary edge map $\boldsymbol{BI}$. Let $\Phi\varepsilon \subset \boldsymbol{BI}$ be the area corresponding to $\Phi \subset \boldsymbol{I}$; that is, $\Phi\varepsilon$ is the edge map of the source area Φ. We then compute the percentage of the edge and the complexity of a patch's constituent colors to decide whether the patch is important. Our data term is defined as follows:

$$D(p,t) = \frac{\max\left(1, \sum_{q \in (\psi_{(p,t)} \cap \boldsymbol{\Phi}_\varepsilon)} c\right) \times \operatorname{var}\left(\psi_{(p,t)}\right)}{\left|\psi_{(p,t)}\right|}, \quad \forall p,\ q \in F_t,\ p \in \delta\Omega_t,\ q \in \Psi_{(p,t)} \tag{8}$$

$$\operatorname{var}(\psi_{(p,t)}) = \frac{\sum_{\forall i}\sum_{\forall j} c_{ij}}{i \times j}. \tag{9}$$

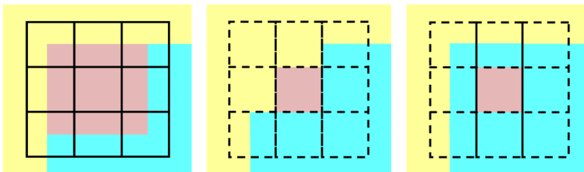


Fig. 10. Inpainting results derived by different strategies. (a) Damaged frame. (b) Incorrect inpainting result when only the confidence term is used. (c) Inpainting result when both the confidence term and the data term are considered.

The $\max$ function ensures a nonzero summation of the pixel count in the edge map and $var(\Psi_{(p,t)})$ is used to compute the degree of intensity variation in the patch $\Psi_{(p,t)}$. The use of intensity variation is to ensure the miss-detection of edge will not degrade the quality of inpainting result.

One may use the confidence term or the data term to determine the patch with the highest priority. If only the confidence term is used, the patches at the outer boundary would be processed first; however, the inpainting result may be incorrect, as shown in Fig. 10(b). On the other hand, if only the data term is used, the inpainting process would consider the available information contained in a patch, but the proportion of missing data in the patch will be neglected. As a result, the inpainting process would be biased. Fig. 10(c) shows an inpainting result when both terms are considered.

2) Patch Searching: After determining the inpainting priority of the patches in each frame, we search the best-matched patch for each damaged patch in order of priority. If a patch has the highest priority in the spatial and temporal domains, there is a better chance of finding a patch with the best quality. This is because our proposed approach considers the degree of damage, the relations with surrounding patches, and the degree of complexity when determining the priority of patches. Hence, we can ensure the quality of the search process. Let $\Psi_{(p^\wedge,t)}$ be the patch with the highest priority; that is

$$\Psi_{(p^\wedge,t)} = \arg\max(P(p,t)),\ p \in \delta\Omega_t. \tag{10}$$

An arbitrary patch template, $\Gamma_{(p^\wedge,t)}$, of $\Psi_{(p^\wedge,t)}$ is expressed as follows:

$$\Gamma_{(p^\wedge,t)} = \cup_{\pm k\Psi(p^\wedge,t)}(\Psi_{(p^\wedge,t)} \cap \Phi) \neq \phi \tag{11}$$

where ϕ is an empty set and $\pm k\Psi_{p^\wedge}$ represents the domain that covers the patch $\Psi_{(p^\wedge,t)}$ and its surrounding pixels outside $\Psi_{(p^\wedge,t)}$ within a distance k. The value of k is set as half the width of the patch. In our experiments, the size of a patch is $3*3$. The top two rows of Fig. 11(a) and (b) illustrate the steps of the proposed patch search algorithm. In the previous stage, we constructed the motion map of the background layer. We now follow the motion vectors obtained from that motion map to search for a similar patch. We start from the neighboring frames to see if there are any possible candidate patches [the green blocks in Fig. 11(a)]. The search process begins with the neighboring blocks and then moves to blocks that are farther away. If this step fails to find similar blocks among the neighboring frames, we perform an intra-frame search to locate a similar patch in the current frame [frame f_i in Fig. 11(b)].

Let $\Gamma_{(p^\wedge,t)}$ be the best matched patch among all the candidate patches; that is

$$\Gamma_{(q^\wedge,t')} = \min_{\Gamma(q^\wedge,t)\in\Phi(q,t')^r}(d(\Gamma_{(p^\wedge,t)},\Gamma_{(q^\wedge,t)})),\ \forall q \in P_s \tag{12}$$

where $\Phi^r_{(q,t)} \subset \Phi_t$ represents a square region of size $r \times r$ centered at q and $d(\Gamma_{(p^\wedge,t)},\Gamma_{(q^\wedge,t)})$ is a distance function that measures the difference between $\Gamma_{(p^\wedge,t)}$ and $\Gamma_{(q^\wedge,t)}$ by

$$\begin{aligned} d(\Gamma_{(p^\wedge,t)},\Gamma_{(q^\wedge,t)}) = {} & SSD(\Gamma_{(p^\wedge,t)},\Gamma_{(q^\wedge,t)}) \\ & * \max\left(1, \sum{}_{(q,t')\in(\Gamma(q^\wedge,t)\cap\Phi\varepsilon)} c\right) \\ & + fd(t,t') \end{aligned} \tag{13}$$

$$fd(t,t') = \begin{cases} \frac{1}{|t'-t|}, & \text{if } t' \neq t \\ \frac{1}{N}, & \text{if } t' = t. \end{cases} \tag{14}$$

In (13), SSD denotes the sum of the squared distance. The distance function defined in the equation considers three factors: the SSD, the number of undamaged pixels in the patch template, and the temporal distance $fd(t,t')$. The constant, $c = 1$, indicates the weight of an undamaged pixel and $fd(t,t')$ in the distance function ensures that the most similar patch within the distance range will be selected.

After selecting the target patch, we copy the motion vector of $\Psi_{q^\wedge}$ to $\Psi_{q^\wedge}$ in order to maintain temporal continuity. Fig. 11 illustrates the steps of the motion completion procedure. As shown in Fig. 11(a), the procedure starts with an inter-frame search to find the most similar patch in the neighboring frames. After an undamaged patch $\Gamma_{(q,j)}$ is identified, the missing motion information in p is updated with a motion vector (x_j, y_j) to reference the targeted patch in frame f_j. Fig. 11(b) shows that if a reference patch cannot be found in the neighboring frames, the process performs an intra-frame search to locate the most similar patch in the current frame. After the best-matched undamaged patch $\Gamma_{(q,i)}$ is identified, the missing motion information in p is updated with a motion vector (x_i, y_i) to reference the targeted patch in the current frame.

B. Frame Completion

After recovering the missing motion information from the motion maps, we implement the frame completion process to repair the missing areas and regulate the luminance of the video frames. The process involves two steps: patch pasting and frame adjustment. The former reduces the number of undesirable artifacts caused by unstable illumination and mismatched edges conditions and the latter uses a panoramic mosaic of the video to regulate the luminance of each frame and reduce the intensity of flicker in the video.

Fig. 12 illustrates the patch pasting process. After the missing motions in motion maps are completed, we stack the patches in the temporal domain according to the computed motion map and construct the space-time volume. A video can be viewed as a composition of several space-time volumes. Then, the patch pasting process will select similar undamaged patches from the same space-time volume and paste them into damaged areas. Fig. 13 shows an example that we apply the patch pasting process on an aged film. Fig. 13(a) shows a space-time volume produced by stacking all the patches in the temporal domain

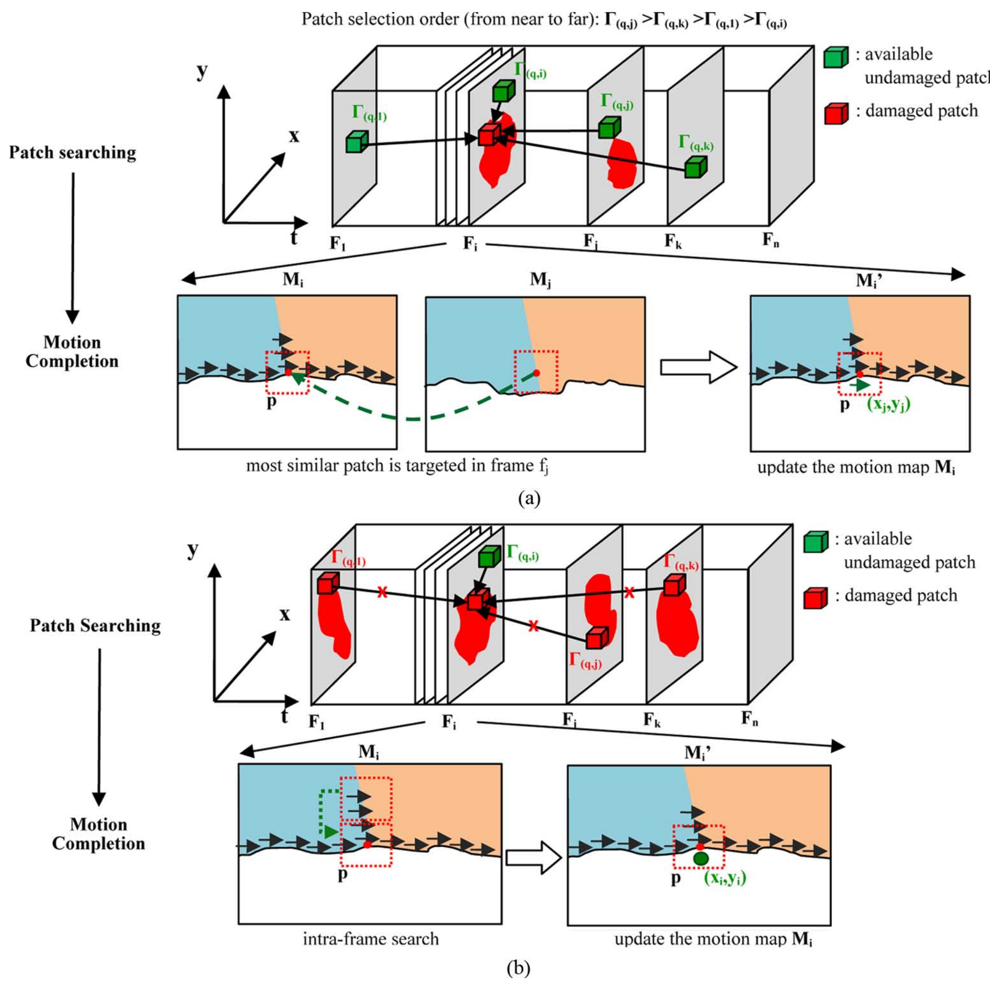


Fig. 11. Illustration of the proposed motion completion algorithm. (a) Inter-frame search: the search process tries to find the best-matched patch in the neighboring frames f_j. (b) Intra-frame search: if there is no undamaged patch in the neighboring frames, the procedure performs an intra-frame search to find the best-matched patch in the current image. After the best-matched undamaged patch is identified, the corresponding motion map is updated with a motion vector (x, y) to the targeted patch.

based on the computed motion maps. Fig. 13(b) shows the result that we apply the Poisson image editing technique proposed by Perez *et al.* [25] to seamlessly blend the reference patch into the damaged areas. The technique assumes that the source and target images have similar textures and colors across the boundary. Although this assumption could be a limitation for some applications, our case suffers no such limitation because the referenced patch is selected from the same space-time volume. Some examples of our patch pasting process are shown in Fig. 14. Fig. 14(a) and (c) shows the artifact derived by pasting a path directly onto a missing area; while Fig. 14(b) and (d) is the results of using the proposed patch pasting method.

After repairing all the missing areas, we use the multi-resolution spline technique proposed by Burt and Adelson [4] to composite all the repaired frames and generate a panoramic mosaic. The technique is widely used to combine two or more images in order to generate a large image mosaic. The objective is to blend low frequencies over a large spatial range and high frequencies over a small spatial range. During the blending process, the spatial information in the images, i.e., the luminance and color, is regulated so that the images match. We employ this technique in the frame adjustment process to stabilize the luminance of consecutive frames and then use a simple 3-band scheme to composite all the video frames and generate a panoramic mosaic P:

$$P = \bigcup_{t=1toN} B(f'_t) \tag{15}$$

where N is the number of frames in a video, f'_i is a repaired frame at time t, and B is the frame blending function. Fig. 15 present a composition result, which shows that the luminance of video frames can be regulated successfully during the blending process.

During the frame blending process, we also record the location of each frame in the mosaic P. Hence, P can be decomposed into several pseudo frames p_t according to the relative locations of the frames in P. A pseudo frame p_t is defined as follows:

$$\forall t \in N, p_t = P(f_{t_x}, f_{t_y}, f_W, f_H) \tag{16}$$

where N is the number of frames; (f_{t_x}, f_{t_y}) is the relative location of frame F_t in P, and f_W and f_H are the width and height of

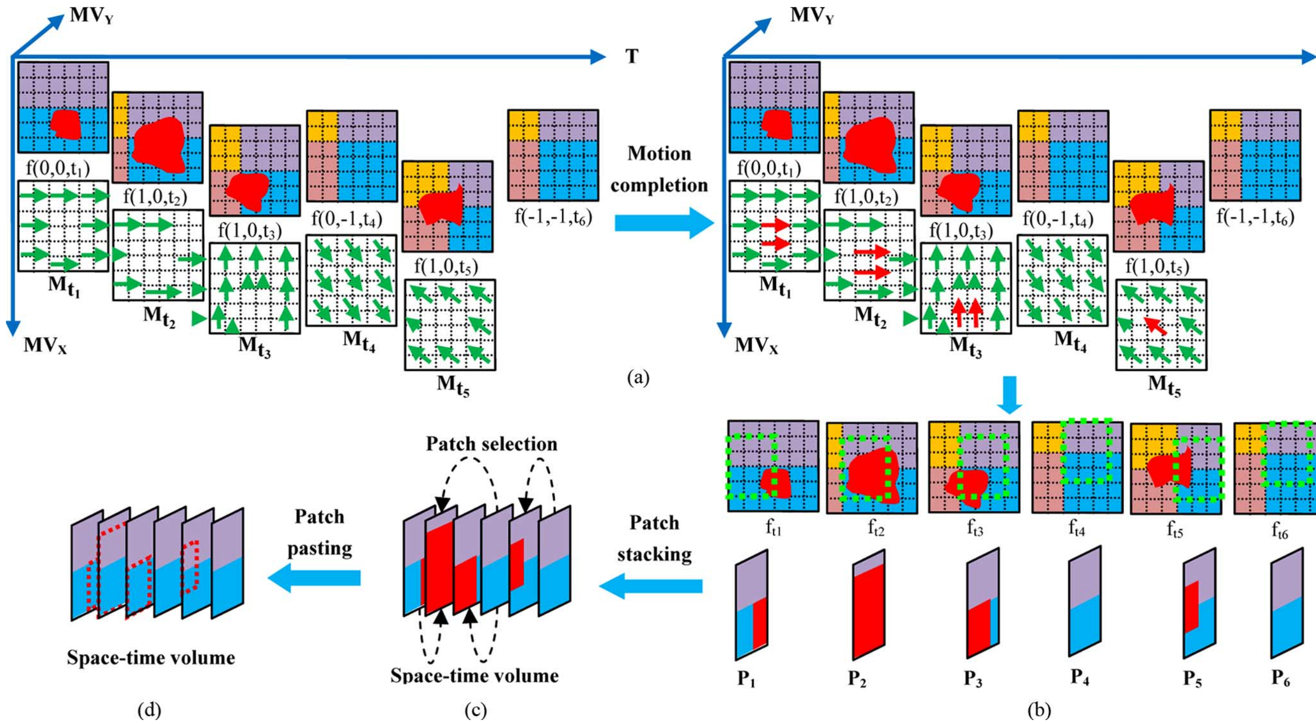


Fig. 12. Illustration of the proposed patch pasting process. After we building motion maps of each frame and complete missing motions [as shown in (a)], patches p_i Pi in each frame f_{ti} which have similar content can be tracked [as shown in (b)] and stacked in the temporal domain according to computed motion information [as shown in (c)]. Finally, undamaged patches in whole space-time volume can be used to inpaint the damaged areas [as shown in (d)].

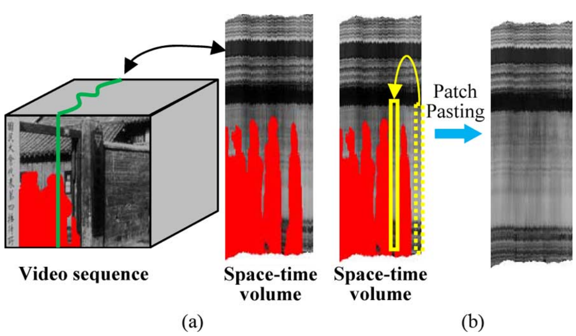


Fig. 13. Proposed patch pasting process, which selects undamaged data from the same space-time volume and pastes it onto the missing areas by applying a Poisson equation.

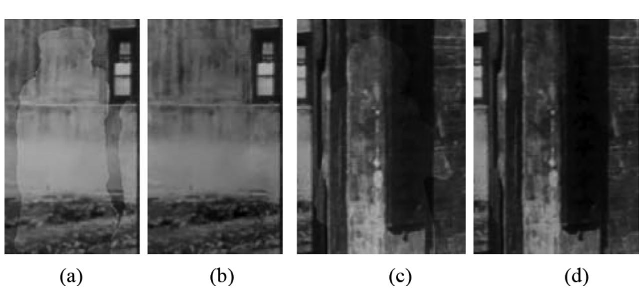


Fig. 14. Differences between pasting a patch directly and using our patch pasting method. (a) and (c) are the results of pasting a patch directly onto a missing area. (b) and (d) are the results derived by the proposed patch pasting method.

frame F_t, respectively. An example of p_t is shown in Fig. 15(b). Pseudo frames are used to regulate the global and local intensity of completed frames f'_t. Fig. 16 shows an example of intensity regulation.

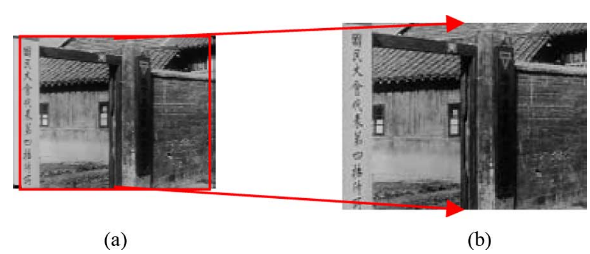


Fig. 15. Example of frame composition. (a) Camera motion Shaky. (b) Pseudo frame of P_c.

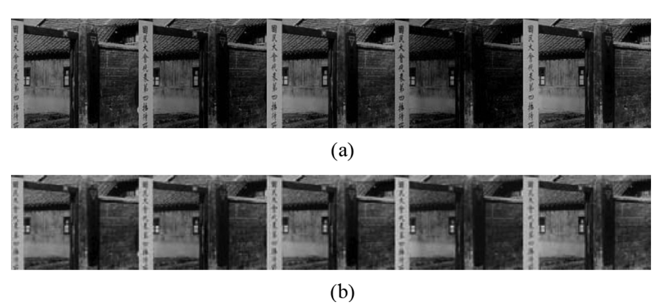


Fig. 16. Example of intensity regulation. (a) Completed frames. (b) Results of intensity regulation.

IV. Experiment Results

To evaluate the effectiveness of the proposed algorithm, we tested on different types of videos, including home videos, video

(a)
(b)
(c)
(d)

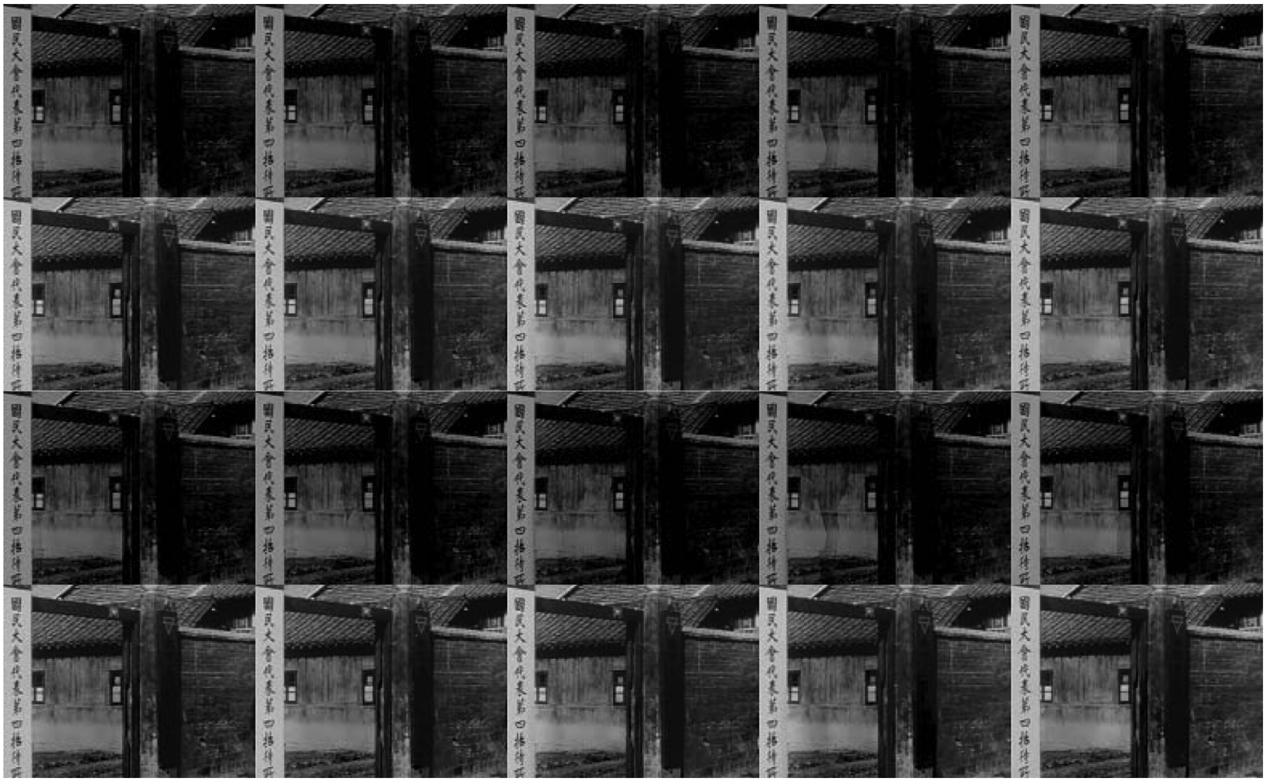


Fig. 17. Old film with unstable luminance camera motion and several moving objects. (a) Original video frames. (b) Inpainting results derived by [24]. (c) Inpainting results derived by [30]. (d) Inpainting results derived by the proposed algorithm.

TABLE I
DETAIL INFORMATION OF THE TEST VIDEOS

Fig.	Resolution	Frame number	Damaged Pixels (per frame)	Inpainting Time (per frame)
17	640 x 480	173	79003	2212.09 sec
18(a) and (b)	800 x 608	56	48538	1208.95 sec
18(c) and (d)	800 x 608	97	20797	478.34 sec
18(e) and (f)	800 x 608	100	36818	736.36 sec

games, and old films in the experiments. The proposed video inpainting algorithm was implemented on a P-4 2.6-GHz machine. The detail information of the test videos used in the inpainting process are summarized in Table I. All experiment results related to this work are available at http://research.twnct.net/VInpainting/.

In the following experiments, we compare the results derived by our algorithm with those of the methods in [24] and [30] for old films preserved by the National Archives Administration R.O.C. In Fig. 17, we first try to remove several large objects contained in an aged film to evaluate the efficacy of the proposed method when the video is severely damaged. This video also contains unstable luminance and camera motion. As previously mentioned, when large missing areas are present, existing inpainting methods usually fail to find sufficient reference information to repair the missing area. As a result, many visual defects can be seen in the videos produced by the two methods [24], [30]. In comparison, our proposed algorithm first applies motion completion to recover the missing motion, then conducts frame completion to stabilize the luminance while inpainting the missing area. Our result in Fig. 17(d) shows that the proposed algorithm is capable of repairing large missing areas and also achieves good inpainting results with stabilized luminance.

Fig. 18 shows the results of using the proposed algorithm to restore three old films. The three video sequences, which were preserved by the Chinese Taipei Film Archive, contain several types of defects caused by poor storage conditions. Since we have not yet developed a method for automatically detecting these defects, for this experiment, we manually labeled these defects. The results in Fig. 18 show that our proposed inpainting algorithm can repair the severely damaged old films effectively.

V. DISCUSSION AND CONCLUSION

We have proposed a novel video inpainting algorithm for digitized aged films. The algorithm consists of two procedures: motion completion and frame completion. In addition, a preprocessing procedure constructs a motion map to record the motion information in undamaged source areas. The motion completion procedure restores the motion in each missing area based on the completion order determined by the priority computation step. The completed motion map is used to improve the temporal continuity and find the best-matched result for inpainting damaged areas. The frame completion procedure seamlessly repairs all the damaged areas and reduces the intensity of video flicker. During the frame completion phase, we use a panoramic mosaic to help stabilize the global and local luminance and thereby obtain better restored videos.

We conducted experiments on several damaged aged films to demonstrate the efficacy of our proposed algorithm. Our results show that our proposed algorithm is better than previous algorithms in dealing with videos containing unstable motion

Fig. 18. Three old films with severely damaged contents and their corresponding repaired results. (a), (c), and (e) are the damaged video sequences, and (b), (d), and (f) are the respective results derived by the proposed algorithm.

and luminance, and it can produce visually pleasing results. In addition, the proposed inpainting process recovers missing parts of videos by using original undamaged information from all video frames whenever possible. Our approach reduces the block reuse rate and the restored videos are close to original state.

From a technical perspective, our contribution is twofold.

- We have shown that the motion map construction algorithm and the proposed motion completion procedure are critical when inpainting videos with large damaged areas.
- The frame completion procedure, which implements patch pasting and frame adjustment steps, yields visually pleasing results with seamless boundaries and stable luminance.

Our proposed algorithm has some limitations. First, our video inpainting method relies heavily on the results of motion completion and frame completion. If the damaged content covers a large area in every frame, visual defects may appear in the resultant video, as most of the useful reference data can only be obtained from neighboring areas nearby. Second, since the algorithm uses fixed-sized blocks in the local motion estimation and frame completion procedure, it may not be able to handle videos with large amounts of zoom-in/out camera motion. Third, shadows of undesired objects cannot be removed completely because they cannot be extracted precisely in our target segmentation procedure. In future work, we will try to improve results by combining some scale-invariant features and employing different block sizes during motion estimation and frame completion.

REFERENCES

[1] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Proc. 2nd Eur. Conf. Computer Vision*, 1992, pp. 237–252.

[2] J. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker: Description of the algorithm," *OpenCV Document, Intel, Microprocessor Research Labs*, 2000.

[3] M. Brown and D. Lowe, "Recognizing panoramas," in *Proc. Int. Conf. Computer Vision*, 2003, pp. 1218–1225.
[4] P. J. Burt and A. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Trans. Graph.*, vol. 2, pp. 217–236, 1983.
[5] D. Capel and A. Zisserman, "Automated mosaicing with super-resolution zoom," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1998, pp. 885–891.
[6] A. Cenci, A. Fusiello, and V. Roberto, "Image stabilization by feature tracking," in *Proc. 10th Int. Conf. Image Analysis and Processing*, 1999, pp. 665–670.
[7] T. -J. Cham and R. Cipolla, "A statistical framework for longrange feature matching in uncalibrated image mosaicing," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1998, pp. 442–447.
[8] C. H. Cheung and L. M. Po, "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 16–22, Feb. 2005.
[9] M. V. Venkatesh, S. -C. S. Cheung, and J. Zhao, "Efficient object based video inpainting," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 168–179, Jan. 2009.
[10] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[11] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.
[12] E. Cuevas, D. Zaldivar, and R. Rojas, Kalman Filter for Vision Tracking, Freie Univ. Berlin, 2005, Tech. Rep. B 05-12.
[13] E. Cuevas, D. Zaldivar, and R. Rojas, Particle Filter in Vision Tracking, Instituet fur Informatik, Freie Univ. Berlin, Tech. Rep. B-05-13.
[14] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," *ACM Trans. Graph.*, vol. 22, pp. 303–312, 2003.
[15] K. M. Gullu, O. Urhan, and S. Erturk, "Scratch detection via temporal coherency analysis," in *Proc. 2006 IEEE Int. Symp. Circuits and Systems*, 2006.
[16] A. M. Huang and T. Nguyen, "Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation," *IEEE Trans. Image Process.*, vol. 18, no. 4, pp. 740–752, Apr. 2009.
[17] J. Jia, T. -P. Wu, Y. -W. Tai, and C. -K. Tang, "Video repairing: Inference of foreground and background under severe occlusion," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun.-Jul. 2004, pp. I: 364–371.
[18] J. Jia, Y. -W. Tai, T. -P. Wu, and C. -K. Tang, "Video repairing under variable illumination using cyclic motions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 832–839, May 2006.
[19] L. Joyeux, O. Buisson, B. Besserer, and S. Boukir, "Detection and removal of line scratches in motion picture films," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 1999.
[20] A. C. Kokaram and S. J. Godsill, "Joint detection, interpolation, motion and parameter estimation for image sequences with missing data," in *Proc. Int. Conf. Image Processing*, Oct. 26–29, 1997, vol. 2, pp. 191–194.
[21] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artificial Intelligence*, 1981, pp. 674–679.
[22] A. Machi and F. Collura, "Accurate spatio-temporal restoration of compact single frame defects in aged motion pictures," in *Proc. 12th Int. Conf. Image Analysis and Processing*, 2003, pp. 454–459.
[23] Y. Matsushita, E. Ofek, X. Tang, and H. Y. Shum, "Full-frame video stabilization," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2005, vol. 1, pp. 50–57.
[24] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting under constrained camera motion," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 545–553, Feb. 2007.
[25] P. Perez, M. Gangnet, and A. Blake, "Poisson image editing," in *Proc. ACM SIGGRAPH 2003*, 2003, pp. 313–318.
[26] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst.Video Technol.*, vol. 6, no. 3, pp. 313–317, Jun. 1996.
[27] I. W. Selensnick and K. Y. Li, "Video denoising using 2D and 3D dual-tree complex wavelet transforms," in *Proc. SPIE Wavelets X*, San Diego, CA, Aug. 2003.
[28] Y. Shen, F. Lu, X. -C. Cao, and H. Foroosh, "Video completion for perspective camera under constrained motion," in *Proc. Int. Conf. Pattern Recognition*, 2006, pp. 63–66.
[29] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1994, pp. 593–600.
[30] T. K. Shih, N. C. Tang, and J. -N. Hwang, "Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 347–360, Mar. 2009.
[31] T. Shiratori, Y. Matsushita, S. -B. Kang, and X. Tang, "Video completion by motion field transfer," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006, pp. 411–418.
[32] J. Sun, L. Yuan, J. Jia, and H. Y. Shum, "Image completion with structure propagation," in *Proc. ACM SIGGRAPH 2005*, 2005, vol. 24, pp. 861–868.
[33] N. C. Tang, H. Y. M. Liao, C. W. Su, F. Huang, and T. K. Shih, "Video inpainting on digitized old films," in *Proc. Int. Conf. Knowledge-Based and Intelligent Information and Engineering Systems: Part II*, 2009, pp. 421–430.
[34] D. Turaga, M. Alkanhal, and T. Chen, Correlation Based Search Algorithms for Motion Estimation, Carnegie Mellon Univ., 1999, Tech. Rep.
[35] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.
[36] Y. Zhang, J. Xiao, and M. Shah, "Motion layer based object removal in videos," in *Proc. 7th IEEE Workshop Application of Computer Vision*, 2005, pp. 516–521.
[37] Z. Zhu, G. Xu, Y. Yang, and J. Jin, "Camera stabilization based on 2.5D motion estimation and inertial motion filtering," in *Proc. IEEE Int. Conf. Intelligent Vehicles*, 1998, vol. 2, pp. 329–334.
[38] I. Zoghlami, O. Faugeras, and R. Deriche, "Using geometric corners to build a 2D mosaic from a set of images," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997, pp. 420–425.

Nick C. Tang received the B.S., M.S., and Ph.D. degrees from Tamkang University, Taiwan, in 2003, 2005, and 2008, respectively.

Currently, he is a Postdoctoral Fellow with the Institute of Information Science, Academia Sinica, Taipei, Taiwan. His research interests include image and video analysis, computer vision, as well as computer graphics and their applications.

Chiou-Ting Hsu (M'98) received the B.S. degree in computer and information science from National Chiao Tung University, Hsinchu, Taiwan, in 1991 and the Ph.D. degree in computer science and information engineering from National Taiwan University (NTU), Taipei, Taiwan, in 1997.

She was a post-doctoral researcher with the Communication and Multimedia Laboratory, NTU, from 1997 to 1998. From 1998 to 1999, she was with Philips Innovation Center Taipei, Philips Research, as a senior research engineer. She joined the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, as an Assistant Professor in 1999 and is currently an Associate Professor. Her research interests include multimedia signal processing, video analysis, and content-based retrieval.

Prof. Hsu received the Citation Classic Award from Thomson ISI in 2001 for her paper "Hidden digital watermarks in images." She is currently an Associate Editor of *Advances in Multimedia*.

Chih-Wen Su received the B.S. degree in mathematics and the M.S. degree in computer science, both from Fu-Jen University, Hsinchuang, Taiwan, in 1999 and 2001 respectively, and the Ph.D. degree in computer science and information engineering from National Central University, Chung-Li, Taiwan, in 2006.

He is currently an Assistant Professor in the Department of Information and Computer Engineering, Chung Yuan Christian University, Taiwan. His research interests are in image and video analysis and content based indexing and retrieval.

Timothy K. Shih (SM'01) is a Professor at the National Central University, Chung-Li, Taiwan. He was the Dean of the College of Computer Science, Asia University, Taiwan, and the Department Chair of the CSIE Department at Tamkang University, Taiwan. His current research interests include multimedia computing and distance learning. He has edited many books and published over 460 papers and book chapters.

Dr. Shih is a Fellow of the Institution of Engineering and Technology (IET). In addition, he is a senior member of ACM. He participated in many international academic activities, including the organization of more than 60 international conferences He also joined the Educational Activities Board of the Computer Society. He was the founder and co-editor-in-chief of the *International Journal of Distance Education Technologies*, published by the Idea Group Publishing, USA. He is an associate editor of the IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES. He was an associate editor of the *ACM Transactions on Internet Technology* and an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA. He has received many research awards, including research awards from National Science Council of Taiwan, IIAS research award from Germany, HSSS award from Greece, Brandon Hall award from the USA, and several best paper awards from international conferences. He has been invited to give more than 30 keynote speeches and plenary talks in international conferences, as well as tutorials in IEEE ICME 2001 and 2006 and ACM Multimedia 2002 and 2007.

Hong-Yuan Mark Liao (SM'01) received the B.S. degree in physics from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1981 and the M.S. and Ph.D. degrees in electrical engineering from Northwestern University, Evanston, IL, in 1985 and 1990, respectively.

From August 1997 to July 2000, he served as the Deputy Director of the Institute of Information Science, Academia Sinica, Taipei, Taiwan. From February 2001 to January 2004, he served as the Acting Director of the Institute of Applied Science and Engineering Research, Academia Sinica. He has been jointly appointed as a Professor of the Computer Science Department of National Chiao Tung University, Hsin-Chu, since 2003.

Dr. Liao was the recipient of the Young Investigators' award from Academia Sinica in 1998. He received a Distinguished Research Award from the National Science Council of Taiwan in 2003 and 2010, a National Invention Award of Taiwan in 2004, a Distinguished Scholar Research Project Award from the National Science Council of Taiwan in 2008, and an Academia Sinica Investigator Award in 2010. In June 2004, he served as the conference co-chair of the 5th International Conference on Multimedia and Exposition (ICME 2004), technical co-chair of 2007 ICME, and General co-chair of the 17th International Conference on Multimedia Modeling. He also served as a committee member of some prestigious conferences such as ACM Multimedia conference (2005–2007) and World Wide Web conference (2007). During 2006–2008, he served as the president of the Image Processing and Pattern Recognition society of Taiwan. He is now the program director of Computer Science Division 2, National Science Council of Taiwan. He is on the editorial boards of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and the *IEEE Signal Processing Magazine*. He was an associate editor of the IEEE TRANSACTIONS ON MULTIMEDIA during 1998–2001.