Video Background Completion Using Motion-guided Pixels Assignment Optimization

Zhan Xu, Qing Zhang, Zhe Cao, and Chunxia Xiao

***Abstract*—Background completion for consumer videos captured by free-moving cameras is a challenging problem. In this paper, we present a new approach to complete the holes left by removing objects with motion-guided pixels assignment optimization. We first estimate the motion field in the holes by applying a two-step motion propagation method. Then, using estimated motion field as guidance, the missing parts of the video are completed by performing pixels assignment optimization based on Markov Random Field (MRF), which optimally assigns available pixels from other neighboring video frames to the missing regions. Finally, we present an illumination adjusting approach to eliminate the illumination inconsistency in the completed holes. We validate our method on a variety of videos captured by free-moving cameras. Compared with previous methods, our method works better to keep the completed background spatio-temporally coherent, to complete video background with much depth discontinuity, and to make the illumination consistent in the completed region.**

***Index Terms*—video completion; motion field; Markov Random Field; illumination transfer.**

I. Introduction

VIDEO background completion is an important problem in computer vision and computer graphics communities, and has a variety of applications. For example, in movie production, unwanted people such as the staff need to be removed from the movie footage. In video-based street view construction, we hope that the videos record background scenes without being occluded by walking people and moving vehicles. Many video completion methods have been proposed in recent years, refer to [1] [2] for a survey. However, as it is an extremely challenging problem, although impressive results have been produced, these methods usually work well under some special conditions. For instance, the videos are recorded by static or parallel-to-scene moving cameras.

Completing the static background in the videos with free-moving camera suffers from the following difficulties. Firstly, for a video captured by free-moving camera, the observed background usually exhibits some perspective distortion, especially for camera with large-scale compound movement. Thus, it is difficult to recover the occluded background consistent with the corresponding camera perspective and the surroundings during completion processing. Secondly, as human's vision system is sensitive to spatio-temporal discontinuity of the video, the recovered background should be spatio-temporally coherent within the hole and consistent with the regions around the hole. Finally, outdoor scenes usually exhibit illumination variation. To achieve visually pleasing results, the illumination of the completed background should be consistent with that of the surrounding scenes. The illumination inconsistency artifacts occurring around the hole boundary and within the hole should be avoided.

In this paper, we propose a novel video background completion approach for videos captured by free-moving cameras. Similar to methods [3] and [4], our method is based on the following assumption: the missing region in one frame can be visible in its neighboring frames, despite some projective distortion. Thus, we can collect appropriate information from the available video content to complete the occluded background. By performing pixels assignment optimization, we can reduce the background distortion under the corresponding perspective, and achieve desirable completion results.

Our algorithm consists of three steps: motion field completion, background completion, and illumination adjustment. For the missing part in video volume, we first present an effective motion field completion approach to estimate the motion field in the hole, making the completed motion field consistent with that around the hole's boundary. After that, we complete the missing parts with the guidance of the completed motion field. We consider the completion process as a MRF-based optimization problem, and find the best assignment of pixels from other neighboring frames to fill the missing part. Finally, as filled pixels may come from different neighboring frames, we use adaptive illumination transfer method to address the illumination inconsistency occurred in the filled region.

In summary, our approach has the following two main contributions.

(1) Present a system to complete video background with large-scale compound camera movement and severe depth discontinuity, which is demonstrated difficult for previous methods based on plane-wise perspective transformation.

(2) Propose an illumination adjusting method to effectively eliminate the illumination gaps existing within the completed holes.

We present a variety of results in this paper to show that our system can complete the static background well in many aforementioned complex and challenging situations.

II. Related Works

Since Bertalmio et al. [5] extended their image inpainting algorithm to video, many video completion methods have been

Zhan Xu, Qing Zhang, Zhe Cao and Chunxia Xiao are with the Computer School, Wuhan University, Wuhan, Hubei, China, 430072. E-mail: xuzhan2012@whu.edu.cn, qingzhang@whu.edu.cn, zhecao@whu.edu.cn, cxxiao@whu.edu.cn.

Corresponding to Chunxia Xiao: cxxiao@whu.edu.cn

proposed, see [1], [2] for a survey. Some of them have implicit constraint for camera movement ([1], [6]–[10]), while other methods are not limited to such conditions and can be applied to more video types ([2]–[4], [11]–[16]).

The first category mainly aims at completing both static background and occluded moving objects. The input videos are always captured by static camera or camera moving parallel to the scene. Wexler et al. [6] formulated the completion as a non-parametric optimization and used an iterative method to fill the hole. Jia et al. [7] used tracking technique to restrict the searching space, and applies a fragment-by-fragment completion method. These two methods only works well for objects with repetition pattern such as periodically walking people. Patwardhan et al. [9] built image mosaics to produce temporally consistent results and reduced the search space. As they used block matching to align sequential frames, and considered the median shift of all the blocks as the camera shift, this work was limited to the constrained camera motion such as moving parallel to the image plane. Granados et al. [1] proposed a non-parametric algorithm that can deal with complex scenes containing dynamic background and non-periodical moving objects. Their completion process was performed by searching for an optimal pattern of pixel offsets from missing region to the accessible regions. The proposed energy function implicitly assumed that the repetitive pieces could be found in the video, thus it is mainly appropriate for statics cameras.

Different from the above methods, other methods are not limited to static or parallel-to-scene moving cameras, and can be applied to more complex video data. These methods can be roughly classified into two categories: methods based on motion field [4], [11]–[13], and methods based on perspective transformation [2], [3], [14]–[17].

Among the motion based methods, [4], [13] aim at completing both the occluded moving objects and the background, while [11] only aims at completing background. Shiratori et al. [4] completed video using motion field transferring. They calculated the Lucas-Kanade optical flow [18] for the unoccluded video regions, and synthesized the missing motion field using the available optical flow. Since only the nearest available pixels were used to fill the hole, the final result sometimes has misalignment and discontinuity artifacts. Moreover, this method did not handle illumination consistency during completion, so the illumination in the filled regions might be inconsistent. Tang et al. [11] proposed an motion-field based algorithm to complete vintage films via maintaining spatio-temporal continuity. They used patch averaging as the final result, leading to obvious blurring artifacts for video with rich texture detail. In [13], objects in the video were first separated, and holes are then completed according to corresponding regions. This approach needs complex parameter adjustment to obtain desirable results.

Perspective transformation is also a useful video completion technique. Some of the methods [14]–[16] in this category aimed at completing both static background and moving foreground at the same time. Jia et al. [14] separated the background region of some key frames into layers, and propagated the segmentation results throughout the video using mean-shift tracking. Homography blending was applied between specified homography matrices of all layers in order to avoid artifacts near the layers' boundaries. In [15], the authors proposed a method to deal with the illumination variation in the input video. Reference mosaics for each layer were constructed, and the intrinsic image separation of these mosaics was projected onto original frames. Because of the essential drawback in perspective-based registration (the scene has to be composed of a few major planes), this method sometimes cannot segment the video accurately when the background has depth discontinuity. Besides, their illumination completion method only applied the image repairing method [14] to illumination content, without considering the texture information.

Method proposed by Newson et al. [2] aims mainly at completing dynamic video texture. This method extended Patch-Match algorithm [19] to 3D domain to search ANN (approximate nearest neighbor) for spatio-temporal patches, and used a novel similarity metric accounting for texture features. For the moving camera cases, they first warped each frame to a reference frame with only one homography per frame and then performed the completion.

Granados et al. [3] proposed a method to inpaint the static background of videos, which is the most related method to ours. This method applied optimization in 3D space to segment all the frames into subregions (layers) with different homography. Then the video was completed frame-by-frame by picking several candidates for each missing pixel through subregion-level perspective transformation. A MRF optimization was performed to ensure the coherence within single image. Finally, they applied a spatio-temporal gradient fusion to handle the illumination inconsistency. Herling et al. [17] proposed a real-time video inpainting method which drastically reduced the time consumption of completion. However, to reduce the computation cost, they completed each frame just based on itself and the preceding frame, thus this method is not applicable for more complex cases.

As these perspective transformation based methods [2], [3], [14]–[17] rely on the planar assumption of homography, they usually work well for background consists of several major planes. When the depth of the scene changes abruptly and severely, the background cannot be approximated using a few planes, and unreasonable division of the background would lead to artifacts in the results. Another problem occurs during the feature extraction processing. It is difficult to extract feature points in some textureless regions because of rare texture and uniform appearance, such as the sky. These methods are thus seriously constrained under certain circumstances.

Recently, Ilan et al. [20] utilized data-driven strategy to obtain plausible results, which required large amount of additional data.

III. Problem Formulation and System overview

The content of video can be divided into foreground region and background region. Typically, objects are treated as foreground if they have intrinsic movement, such as walking people and running vehicles. Other objects are treated as background, whose movement is merely caused by the

Fig. 1: System overview. (a) The input: original video and specified masks. (b) Motion field completion. The original motion field and the completed result are visualized. (c) In each frame, the missing pixels search for their possible candidates in the neighboring frames. (d) The label distribution of our MRF-based optimization and the background completion result for one frame. (e) The final output frame after performing the illumination adjustment strategy.

camera's motion. One exception is object like waterfall or river with dynamic video texture. We also treat it as background since it is usually not the dominant object of the video. When foreground objects' movement is not parallel to the background, they will occlude the background, sometimes even occlude each other.

In this paper, we are primarily concerned about the static background completion (without dynamic video texture). In other words, we want to process the following two situations: (A) One or several foreground objects occlude the background, and we remove some or all of these foreground objects; (B) Foreground objects occlude each other, and we maintain the front one of these objects while removing others to reveal the background.

The input of our algorithm is a video clip $V(x, y, t)$ which has one or more moving foreground objects. We first make masks for the objects to be removed. We use rotoscoping technology such as [21], [22] to track and extract these objects. Removing these objects leaves blank holes Ω in the video volume, and the pixels in Ω are the missing pixels. The region in $V(x, y, t)$ that used to complete Ω as prior information is denoted as Φ (reference regions). Most of the time, $\Phi = V - \Omega$. However, when some parts of the video are not appropriate to be used as prior information for background completion, they should not be included in Φ. For example, in situation (B) when foreground objects occlude each other, we want to maintain the foremost one and remove others, then the front one is marked and excluded from Φ. Our goal is to fill the holes Ω using pixels in Φ to get a completed video with spatio-temporally coherent appearance.

Fig.1 shows the pipeline of our system. We first compute the motion field of Φ using optical flow. Based on it, we use a two-step iterative method to complete the motion field in Ω, ensuring that the complete motion field is reasonable in the interior and consistent with hole exterior (Section 4). Then, with the guidance of the motion field, for the missing part in each frame, we find corresponding part in Φ among neighboring frames, and apply a pixel-assignment process based on MRF to determine the final value for each missing pixel (Section 5). Finally, to handle illumination gaps in the filled region, we present an illumination adjustment strategy to obtain results with illumination consistence (Section 6).

IV. Motion Field Completion

Motion field completion has broad applications in video manipulation, such as [23], [24]. In our work, completing the missing motion field in Ω depends on the known parts. We begin with calculating the motion field of Φ using optical flow. The dense optical flow is computed by applying the method in [25]. Compared with other methods ([18], [26]), the method [25] suits our work better, because it can produce more reliable optical flow around the foreground objects' boundaries.

For each pixel p in Ω, in step I we first estimate its initial optical flow by adopting progressive motion propagation. The completion of the motion field starts from the hole boundary, and progressively advances inwards. Then in step II, we refine the results obtained in the first step by motion field summarization.

Step I: To propagate the motion field inwards and create a reasonable initialization for subsequent optimization, we minimize the following energy function:

$$\sum_{Q \subset \Omega \cup \partial\Omega} \min_{P \subset \Phi} D(P, Q), \tag{1}$$

where P and Q are spatio-temporal cubic patches. We call Q the target cuboid and P the source cuboid. Q denotes any cuboid which has at least one missing pixel, thus it belongs to $\Omega \cup \partial\Omega$, where $\partial\Omega$ denotes the non-hole pixels near the boundary of Ω. We utilize angular difference as [4] to measure the distance $D(P, Q)$ between two spatio-temporal cuboids in the motion field. Note that we use the homogeneous form of optical flow to account for both direction and magnitude. The typical size of the cuboid is $7 \times 7 \times 5$. Each target cuboid Q overlaps with neighboring cuboids, which helps to make the completed result spatio-temporally coherent.

As no data exists in missing hole, to optimize Eq.1, we apply a progressive strategy which approaches the greedy patch comparison in western order. We set a threshold that when 60% or more of the pixels in a target cuboid are known, by then we search for its most similar source cuboid. At this time, $D(P, Q)$ only measures the distance for known pixels in both cuboids. In our experiments, we find 60% is a appropriate threshold. If the threshold too high, the progressive motion propagation would not be efficient, since only a few patches could be matched each time. If the threshold too low, the propagation result would be inaccurate and unreliable.

We constrain the searching process in a certain region $\Phi' \in \Phi$ near the hole, since the motion field in the hole is most relevant to that around the hole's boundary. Φ' is obtained by dilating the mask of the hole. Once the number of pixels in the dilated area reaches twice of that in the hole area, we stop. If the hole area is extremely large (more than 1/4 of the pixels in the frame are missing), we stop when the dilated area and hole area have the same number of pixels. Such constraint can significantly increase efficiency.

The source cuboid we have found is then copied to the location of the target cuboid. As the source cuboid is complete, it is able to fill the missing pixels in the target cuboid with optical flows at the corresponding positions. Once the missing pixels are assigned with optical flow, they are then treated as known pixels in the following propagation process. Eq.1 ensures that the motion field propagates smoothly from Φ to Ω according to the motion field distribution of Φ. With this method, each missing pixel will be filled.

Eq.1 can be further optimized by an iterative method similar to [6]. Each target cuboid Q will be matched an approximate nearest neighbor cuboid P from Φ'. As each cuboid Q overlaps with neighboring cuboids, a single pixel in Q may be occupied by many cuboids containing it. Each cuboid overlapping this pixel would contribute one possible value to it. Intuitively, this pixel should take all these possible values into consideration. The final optical flow for such missing pixel is calculated as follows:

$$\mathbf{m}_p = \frac{\Sigma_i \omega_i \mathbf{m}_p^i}{\Sigma_i \omega_i}, \tag{2}$$

which means that the final optical flow vector is the weighted average of all contributing flows $\mathbf{m}_p^i$ at p by the occupied cuboids Q^i. The weight $\omega_i = s_i \lambda_i$, where $s_i = exp(-D(Q^i, P^i)/2\sigma_d^2)$ measures the similarity between target cuboid Q^i and source cuboid P^i (σ_d is the 75-percentile of $D(Q^i, P^i)$ for all i). λ_i defines the closeness level between the central pixel of cuboid Q^i to boundary of the hole. Target cuboids closer to the boundary should have higher confidence as they are more related to the known pixels. To accomplish this, we first calculate the distance $L(Q^i)$ between the central pixel of Q^i and the hole boundary, and then determine this weight as $\lambda_i = exp(-L(Q^i)/2\sigma_l^2)$, where σ_l is also the 75-percentile of $L(Q^i)$ for all i.

Step II: In the second pass, we further refine the results received in the first step using motion field summarizing method. Inspired by data summarizing method [27], [28], we use the results of step I as the initial value, and apply the following bidirectional similarity to refine the motion field in Ω:

$$\begin{aligned} &\beta \frac{1}{N_\Omega} \sum_{P \subset \Omega} \min_{Q \subset \Phi'} D(P, Q) + \\ &(1-\beta) \frac{1}{N_{\Phi'}} \sum_{Q \subset \Phi'} \min_{P \subset \Omega} D(Q, P), \end{aligned} \tag{3}$$

where Φ' is the regions near the hole in Φ as mentioned above. $N_{\Phi'}$ and N_Ω denote the number of cuboids in Φ' and Ω. Eq.3 means that for each cuboid $Q \subset \Omega$, we search for its most similar cuboid $P \subset \Phi'$, measure their distance $D(P, Q)$, and vice-versa. Using above similarity measure, Ω will contain as much as possible optical flow from Φ', and introduce as few as possible new optical flow artifacts that were not in the Φ'. Thus, the motion field in Φ' can be propagated into Ω more smoothly and accurately. In all the experiments, we set β to 0.8.

Similar to [27], Eq.3 can be optimized using an iterative updating rule. We need to iteratively search and vote the nearest cuboid to minimize Eq.3, and obtain progressively improved motion field for Ω. Please refer to [27] for more technical details.

To accelerate the convergence of iterative process, we build a space-time Gaussian pyramid for the motion field of the video, and adopt a multi-scale completion scheme. The completion begins from the level with coarsest scale. The result of a coarser level is propagated to a finer level as a new initiation for iteration optimization. In this case, Step I (Eq.1) is only performed on the coarsest level to provide initialization for Step II (Eq.3). In our experiments, we build a pyramid with 3-5 levels. Iteration on each level is terminated if (1) the number of iteration times reaches a pre-defined parameter (specified in Tab.II), or (2) the difference between two passes of iteration is less than 15. In each iteration, we use the ANN method [29] to accelerate the nearest cuboid search, which constructs a kd-tree to contain the source cuboids. Such method greatly improves search efficiency.

Our motion completion method is based on the observation that for most videos, the movement of static background is only caused by camera movement, thus its movement is uniform, and the completed motion field should be smooth. However, when the background consists of parts with definitely different movements, users can apply interactive rotoscoping [21] to segment the background into different regions according to respective movements. Different parts of Ω can find corresponding regions in Φ for motion field completion. The manual efforts for such interaction is acceptable. The average time to segment one frame is about half a minute. Note that such extra background segmentation is only necessary for limited video types, and none of the sequences shown in Section VII requires such manual segmentation. Fig.2 shows some motion field completion results. The motion field propagates smoothly from the hole's exterior to interior, and the completed part is consistent with the surroundings. Our motion field completion algorithm is summarized in algorithm 1.

V. Motion-Guided Background Completion

Our method is based on the assumption that the missing region in one frame is visible in other frames. With the completed motion field, we can use each of the neighboring frames to fill partial or all of the missing holes in the target frame, getting a *local solution* for the target frame. However, the *local solution* usually fills certain parts of the hole; furthermore, these naive *local solutions* may exhibit some distortion due to inaccurate optical flow correspondence. Thus, to receive a complete and plausible solution for the target hole, we need to design a strategy to appropriately rearrange these *local solutions*. Simple arrangement with nearest-prior strategy as

Fig. 2: Motion field completion. **First row:** one frame from the input videos. **Second row:** original motion field calculated by [25]. **Third row:** completed motion field. The motion field is visualized using the visualization tool presented at http://hci.iwr.uni-heidelberg.de/Static/correspondenceVisualization/

Algorithm 1 MotionCompletion()

Input: Video V, holes Ω, reference regions Φ
Output: V_m: motion field of V.

1: Compute the motion field of Φ.
2: Construct pyramids Ω^l and Φ^l $(l = l_1, ..., l_{co})$ (l_{co} is the coarsest scale).
3: Complete the motion field of $\Omega^{l_{co}}$ with that of $\Phi^{l_{co}}$ by minimizing Eq.1.
4: $\Omega^{l_{co-1}} \leftarrow \Omega^{l_{co}}$
5: **for** $l = l_{co-1}$ to l_1 **do**
6: Complete the motion field of Ω^l with that of Φ^l by minimizing Eq.3.
7: **if** $l \neq l_1$ **then**
8: Propagate Ω^l to finer level.
9: **end if**
10: **end for**

(a) (b)

Fig. 3: (a) Completion results using our completed motion field combined with the color propagation method (nearest-prior strategy) of [4]. Note that without pixels assignment optimization, artifacts exist on the wall and the window. (b) Completion results using motion-guided pixels assignment optimization.

in [4] always leads to misalignment in the result, as illustrated in Fig.3(a). Instead, we establish a MRF-based optimization method to compute a *global solution* based on these *local solutions*, and assign optimal pixels to the missing region to make the completed result spatio-temporally coherent.

To compute the *local solutions*, we copy corresponding information from neighboring frames into target frame. We use the motion completion method introduced in Section 4 to compute both forward and backward motion field for the video. Suppose f_t is the target frame to be completed, f_{t+i} $(i \in [-r, r])$ is a neighboring frame of f_t, and r is a user-specified parameter to define the radius of neighboring frame interval. According to motion field correspondence, pixel in f_t can find its corresponding pixel in f_{t+i} (suppose $i > 0$) through forward optical flow. Similarly, pixel in f_{t+i} can find its corresponding pixel in f_t through backward optical flow. We obtain two *local solutions* from each neighboring frame f_{t+i} for f_t though both forward and backward motion field in this way, which is showed in Fig.4. For all neighboring frames f_{t+i} $(i \in [-r, r])$, we receive in total $4r$ *local solutions*.

When applying pixel-to-pixel copy strategy to images with integer coordinates, the precision loss of optical flow correspondence may lead to unfilled seams. As 2D patch enjoys the advantage to maintain the local structures and features, instead of pixel-to-pixel copy strategy, we apply patch-by-patch copy strategy to avoid unfilled seams and fluctuation in structures. We first decompose f_t into overlapping 2D patches. For each target patch T in f_t containing missing pixels, we find its corresponding 2D patch S in f_{t+i}, according to the patch center correspondence in terms of optical flow correspondence. Then we copy the color of S to T. The final color value for pixel p in the *local solution* is the weighted average of all overlaps. The weight ω_k is a Gaussian term with a large σ parameter to avoid blurring artifacts. It means that we give dominant weight to the central pixels and smaller weights to other pixels in the patches. In our experiments, we set the size of S and T as 5×5.

With the computed *local solutions*, we develop an optimizing strategy to select the most proper pixels from these *local solutions* and get a globally optimal result for target frame f_t. We formulate this problem as a labeling problem in a MRF framework. MRF model builds up an undirected graph

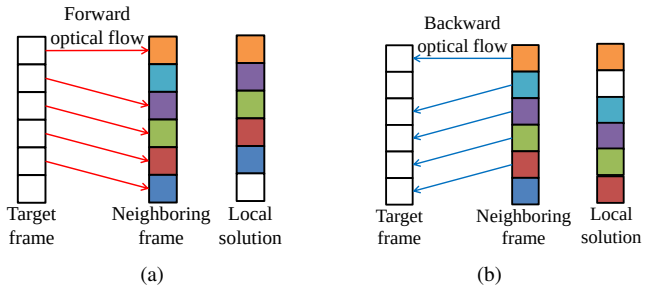


Fig. 4: (a) By using one neighboring frame and forward optical flow, we obtain a *local solution* for the target frame. (b) Using the same neighboring frame and the backward optical flow, we obtain another *local solution*.

Fig. 5: The white region is the hole left by the removed person. Yellow region is the regions we use for evaluating the accuracy of frame alignment as $EV(f_t)$.

to optimally choose a label for each node and obtains minimal global cost [30]. For each missing pixel p in f_t, we choose an optimal pixel with the same space coordinates as p in the *local solution* set $\{f_t^{t+i}\}$, and assign it as the final color of p. First we pick n $(n \leq 4r)$ candidates $\{p^1, p^2, ..., p^n\}$ with the same space coordinates as p from $\{f_t^{t+i}\}$ in a symmetrical frame-pair order, giving priority to the candidates in nearer neighboring frames. Note that in some *local solutions*, there may be no corresponding pixel at the position of p, since p may be invisible in the neighboring frame. Once the number of candidates reaches n, we stop further candidate selection for p. In our experiments, n is set to 8.

Each node in the MRF corresponds to a pixel in or surrounding the hole in the target frame f_t. We add edges between any nodes with their 4-neighborhood nodes. Our goal is to assign a label to each missing pixel, indicating its final choice from all its candidates. Suppose that function $L(p)$ maps p to its final selected candidate's label, and the range of $L(p)$ is $\{1, 2, ..., n\}$. Then in the *global solution*, the final color of p is set to the color of $p^{L(p)}$. Let pixel q be the 4-connected neighboring pixel of p, we minimize the following cost function:

$$\sum_{p} E_p(L(p)) + \alpha \sum_{p,q} E_{p,q}(L(p), L(q)) \tag{4}$$

The parameter α balances the contribution of two terms, and we set α from 6 to 10 in our experiments according to the different video content. Typically, videos with more detailed texture have larger α value.

We start by setting each pixel p an initial value $I(p)$, which is computed as the weighted average of all the corresponding pixels at the position of p from *local solutions*. The weight is based on the accuracy of optical flow correspondence between target frame and its neighboring frame. By dilating the mask of the hole, we select some area $EV(f_t)$ in the target frame (the yellow region in Fig.5), which is the surrounding region around the missing hole, as the sample for measuring the accuracy of correspondence. For each pixel p in $EV(f_t)$, we calculate the color difference between p and its corresponding pixel p' (according to optical flow alignment) in frame f_{t+i}. The similarity ω_i between frame f_t and f_{t+i} is defined as :

$$\omega_i = exp(-\sum_{p \in EV(f_t)} \|f_t(p) - f_{t+i}(p')\|_2) \tag{5}$$

The initial color of missing pixel p in $I(p)$ is then defined as:

$$I(p) = \frac{\sum_{i \in [-r,r]} \omega_i f_{t+i}(p')}{\sum_{i \in [-r,r]} \omega_i} \tag{6}$$

This initialization makes the following optimization process prefer the candidate coming from better-aligned frames.

Based on this initialization, the data term $E_p(L(p))$ measures the cost of choosing $p^{L(p)}$ for p. We define $E_p(L(p))$ as the color difference between the chosen candidate $p^{L(p)}$ in each iteration and $I(p)$:

$$E_p(L(p)) = \|p^{L(p)} - I(p)\|_2 \tag{7}$$

The consistency term $E_{p,q}(L(p), L(q))$ requires adjacent pixels to have coherent appearance and structure. Intuitively, pixels coming from the same frame are more coherent. As a result, we devised the following penalty term:

$$\begin{aligned} &E_{p,q}(L(p), L(q)) = \\ &\quad (\|p^{L(p)} - p^{L(q)}\|_2 + \|q^{L(p)} - q^{L(q)}\|_2) + \\ &\quad \lambda(\|\nabla p^{L(p)} - \nabla p^{L(q)}\|_2 + \|\nabla q^{L(p)} - \nabla q^{L(q)}\|_2) \end{aligned} \tag{8}$$

This term punishes the color and gradient difference for adjacent pixels in holes of f_t when choosing candidates from different neighboring frames. As lines or other structural features always cause strong response in gradient field, in Eq.8, we maintain the gradient information of the structure. The balance parameter λ is set as 0.5 in our experiments.

We use graph cuts [31] to optimize the MRF energy function Eq.4, similar to [32] and [3]. Note that the consistency term Eq.8 does not integrate the temporal neighbors explicitly. However, the temporal coherence can be guaranteed to some extent as we apply motion field to find corresponding locations for the missing pixels. Motion field has been used for video completion as an intrinsic presentation of a video to maintain temporal consistency in previous works such as [4], [12]. With the motion field used in building temporal correspondence, our method can complete the missing pixels better.

Similar to the mean-shift approach in [6], our pixels assignment optimization also tries to accommodate potential errors

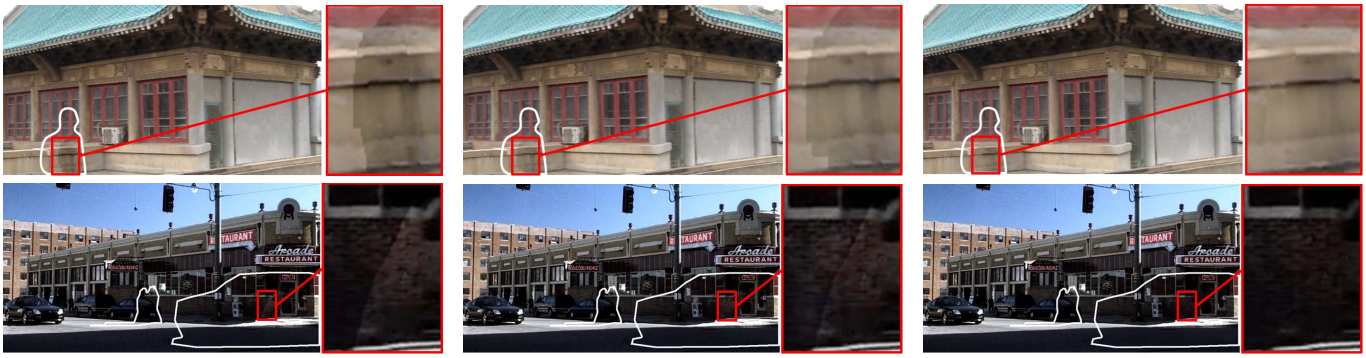


Fig. 6: **First column:** results without illumination adjustment. **Second column:** results using gradient fusion. **Third column:** results with our illumination adjustment method.

of ANN, but in a different way. One advantage is, results of [6] sometimes exhibit blurring artifacts, because they perform averaging operation to get the final appearance, while our approach stitches pixels from different frames, thus maintains texture details better.

When we perform pixels assignment optimization in a single pass, sometimes the video may exhibit a small amount of ghost shadow. This happens because the final assigned pixel may come from any neighboring frame f_{t+i} ($i \in [-r, r]$). When the neighboring frame threshold r is large, corresponding pixels in two sequential frames may be assigned with pixels from two distant frames. For example, pixel p_t in frame f_t may come from frame f_{t-r}, while pixel p_{t+1}, which is the corresponding pixel of p_t in frame f_{t+1}, may come from frame f_{t+1+r}.

To alleviate this problem, we repeat the background completion processing 2-4 times. Each time we complete every target frame with fewer neighboring frames (less r), as well as the number of candidates n. That means we rely more on the smooth motion field and less on the MRF-based optimization for individual frames. Corresponding pixels in sequential frames would be assigned with pixels from closer frames. Especially, in the last pass, we set the parameter r to 1, only considering the nearest neighbor frames on both sides of each target frame. The final value of each filled pixel in f_t comes either from f_{t-1} or from f_{t+1}. In this way, the relationship between adjacent frames becomes stronger, and the temporal coherence of the completed result is consolidated. Please see the accompanying video. This process is described in Algorithm 2.

Algorithm 2 VideoCompletion()

Input: Video V with n frames $f_1 ... f_{end}$
Holes Ω
r: search radius of the neighboring frames
r_s: step size of r
Output: Completed video V_c

Initialization: $V_r \leftarrow V$-Ω, $r \leftarrow 10$ (initial value of r)
MotionCompletion (V_r)
repeat
  for $t = 1$ to end **do**
    for all $p \in \Omega$ in $f_t \in V_r$ **do**
      Find corresponding pixels within neighboring frames f_{t+i} $(-r \leq i \leq r)$ guided by motion field
      Assign one pixel for p by minimizing Eq.4
    end for
  end for
  $V_{r-r_s} \leftarrow V_r$
  $r \leftarrow r - r_s$
until $r = 1$
for each key frame **do**
  Adjust illumination of the key frame
end for
Propagate the adjusted illumination to the non-key frames
$V_c \leftarrow V_r$

VI. Illumination Adjustment

The video completion process mentioned above picks final pixels from different frames to complete the missing background, without considering the variance of illumination among these frames in a global way. For outdoor scenes with varying illumination and camera rotation movement, the illumination of the completed background may be non-uniform and exhibits illumination differences, as illustrated in the first column of Fig.6.

The illumination gaps may exist not only around the boundaries of the hole, but also in the interior of the completed regions, which may be more obvious. This is because during MRF-based optimization, we stitch together pixels from different frames to maintain texture detail. The location where pixels with different labels meet are very likely to generate illumination gaps. Pixels near the hole center always come from frames far apart from each other, so they may have quite different illumination information.

To solve this similar problem, [14] applied the gradient fusion method [33]. However, this method cannot solve our problem, because it only modifies the gradient at the boundary of the hole and fails to handle nonuniform inner illumination. The method [3] extended [33] and presented a spatial-temporal fusion. This method tends to smooth the illumination gaps rather than remove them, as illustrated in the middle row of Fig.6.

In an alternative way, we propose a novel method to address

the illumination inconsistency problem. The basic idea of our method is to transfer the illumination from Φ to the filled regions Ω, so that Ω has similar illumination condition as that of Φ. We first pick one key frame from every five frames in the completed video, and adjust the illumination of these key frames using illumination transferring method. After that, based on the motion field correspondence, we propagate the illumination adjustment results from key frames to other non-key frames and obtain final illumination corrected results for Ω.

Illumination adjustment strategy for key frames is inspired by [34]. It consists of three main steps: (1) we first decompose the key frame into fragments using mean-shift method [35]. (2) For each fragment T in Ω, we find a fragment S in Φ with similar texture based on the texture feature matching [36]. (3) Finally, we transfer the illumination from S to T using adaptive illumination transfer method [37], obtaining the illumination recovered result for a certain key frame. We use Gabor filter [38] to extract texture features from original frames. Note that regions near the hole usually have similar texture as the hole region, so we restrict the texture matching for T in a small part of Φ near the hole. To make the transition between fragments more smooth and natural, we dilate each fragment to generate some overlaps in between, and perform crossfading in the overlapped regions. The pixel's intensity value in the overlapped region of two fragments T_1 and T_2 is determined as $\rho \times lum_{T_1} + (1-\rho) \times lum_{T_2}$, where lum_{T_1} and lum_{T_2} are illumination values from T_1 and T_2 at this pixel, and ρ decreases linearly from 1 on T_1 side to 0 on T_2 side.

After performing illumination adjustment for the key frames, we adopt the following strategy to propagate the adjusted illumination from a key frame to its neighboring non-key frame. Suppose that in a key frame f_t, fragments $S_t \in \Phi$ and $T_t \in \Omega$. S_t is the most similar fragment (in texture similarity) of T_t. According to the motion field correspondence, when S_t moves to S_{t+i} in f_{t+i}, and T_t and T_{t+i}, then we transfer the illumination from S_{t+i} to T_{t+i}. If S_{t+i} moves out the frame, or contains more than 20% missing pixels, we search for a new similar fragment in f_{t+i} for T_{t+i}, and transfer the illumination from the new fragment to T_{t+i}.

To make the illumination change naturally and coherently through the whole video, we introduce a bidirectional illumination propagation method. For two key frames f_{t1} and f_{t2} (assume f_{t1} comes before f_{t2} in video playing order), with the forward optical flow from f_{t1} to f_{t2}, and the inverse optical flow from f_{t2} to f_{t1}, we undertake the forward and backward illumination propagation as mentioned above, respectively. For any intermediate frame between f_{t1} and f_{t2}, its final illumination result is the weighted average of both the forward and backward illumination propagation results, and the nearer key frame has larger weight. Using such illumination propagation method, we can obtain temporally coherent illumination results. The bottom row in Fig.6 shows the results of our illumination adjustment. Bidirectional process technique is also used in other fields such as video watercolorization [39].

Our video completion method is summarized in algorithm 2.

VII. Experimental Results and Discussion

We apply our system on a variety of videos with different types of scenes to validate the proposed algorithms. Some of these videos are captured by Canon EOS 60D (EF-S 18-135 IS) camera. All the results are obtained from single PC, with 64-bit Window7 system, Intel Xeon 3.3GHz CPU and 16G RAM.

The running time is determined by the size of the video as well as the size of holes Ω. Optical flow computation is the most time-consuming step, because the motion field method [25] calculates relatively accurate dense optical flow for each frame using iterative estimation. Although motion field completion has two steps and works in 3D video volume, it is not so time-consuming since we constrain our search space near the hole. Running times for background completion and illumination adjustment are also acceptable. In the MRF-based optimization, only pixels in and around the hole are treated as nodes in the graph, and each missing pixel has fixed number of labels. During illumination adjustment, most searching and transferring operations are only performed on key frames. The total running time is about 1.5 to 6 hours using single PC, excluding the time for manual interaction (30 seconds for each frame typically).

We notice that most video completion methods are also time-consuming. The running time of our method and them can be found in Table I. Note that besides [2], [4], running time for [3] is stated in their paper as $\sim$60 mins to $\sim$240 mins computed in parallel on a 64-core server, which varies as temporal window size varies between 50 and 100 frames([3], Sec.4).

Some parameters are discussed below. During motion field completion, we build Gaussian space-time pyramid to accelerate the convergence of iteration. We denotes the total number of levels in the pyramid as n_{pyr}, and number of iterations on the coarsest level l_{co} as itr_{co}. For the l-th level ($l \neq l_{co}$), we usually perform $itr_{co} - itr_{bias} \times |l - l_{co}|$ times of iteration, where itr_{bias} is the difference of iterative number between two sequential levels. Another important parameter is the α in Eq.4, which balances the contribution of data term and consistency term. We provide the values for all these parameters in Table II.

Experiment 1 (Fig.7): We compare our algorithm with [4] and demonstrate the importance of each step of our algorithm. The video size is $714 \times 372 \times 122$, and we remove the walking person ranging from the 1^{st} frame to the 122^{nd} frame. The camera has translation and rotation movement (nearly 90 degree), leading to obvious illumination change. Results of [4] are illustrated in the second row of Fig.7, which use Lucas-Kanade optical flow [18] and single-pass progressive completion to estimate the motion field. Illumination inconsistency artifacts (illumination gaps) are not addressed, and the texture structure is not well preserved. In the third row, we show the results produced by the combination of our motion field completion method and color propagation in [4]. Misalignment is alleviated, showing the merit of our motion field completion method. In the fourth row, we show the results using our motion completion method and MRF-based pixels assignment

TABLE I: Running time for each experiment
For each experiment, we give the time for motion field calculation using [25] (Time 1), motion field completion (Time 2), background completion (Time 3), illumination adjustment (Time 4), total time and accessible running rime for [2], [4].

Exp	Video size	Hole size(pixels)	Time 1	Time 2	Time 3	Time 4	Total time	Time for [4]	Time for [2]
Exp.1	$714 \times 372 \times 122$	3,636,030	144min	67min	55min	51min	317min	96min	—
Exp.2	$634 \times 334 \times 46$	159,282	47min	31min	23min	20min	121min	77min	—
Exp.3	$960 \times 530 \times 161$	3,599,330	154min	62min	81min	61min	358min	—	—
Exp.4	$960 \times 540 \times 96$	4,709,410	136min	49min	62min	40min	287min	—	300min
Exp.5	$960 \times 720 \times 93$	2,348,832	147min	42min	52min	33min	174min	—	291min
Exp.6	$960 \times 520 \times 27$	6,006	37min	29min	19min	12min	97min	—	—
Exp.7	$960 \times 516 \times 54$	3,387,520	60min	45min	37min	24min	166min	—	—
Exp.8	$960 \times 540 \times 72$	2,887,760	85min	46min	44min	35min	210min	—	—

Fig. 7: Experiment 1 (Comparison with [4]). **First row:** three frames from the input video. **Second row:** results of [4]. **Third row:** results using our motion field completion method and color propagation method in [4]. **Fourth row:** results using our motion field completion method and MRF-based pixels assignment optimization. **Fifth row:** final results of our method.

optimization. Color and structure inconsistency in each frame is removed. Although our method does not rely on perspective transform alignment, no perspective distortion is generated when completing the building consisting of several planes. In the bottom row, we show our final results with illumination adjustment. The illumination gaps are effectively removed.

Experiment 2 (Fig.8): We perform another comparison with [4]. The size of this video is $634 \times 334 \times 46$, and we remove the telegraph pole ranging from the 1^{st} frame to the 46^{th} frame. The facade of background building exhibits strong structure details. The camera's nonlinear trajectory contains translation and rotation movement. Our motion-guided pixels arrangement process picks optimal pixels, and effectively maintains the structures of the background, while the color propagation method in [4] leads to serious accumulation error, and brings about structure misalignment artifacts.

TABLE II: Parameters in our experiments

Experiments	n_{pyr}	itr_{co}	itr_{bias}	$alpha$
Exp.1	3	17	7	9.5
Exp.2	3	17	7	10
Exp.3	4	20	6	6
Exp.4	5	22	5	6
Exp.5	4	20	6	8
Exp.6	3	17	7	6
Exp.7	3	17	7	8
Exp.8	3	17	7	6

Experiment 3 (Fig.9): We run our algorithm on one video ($960 \times 530 \times 161$) captured at a fixed point with 90-degree rotation only, and compare with [3]. We aim at removing the

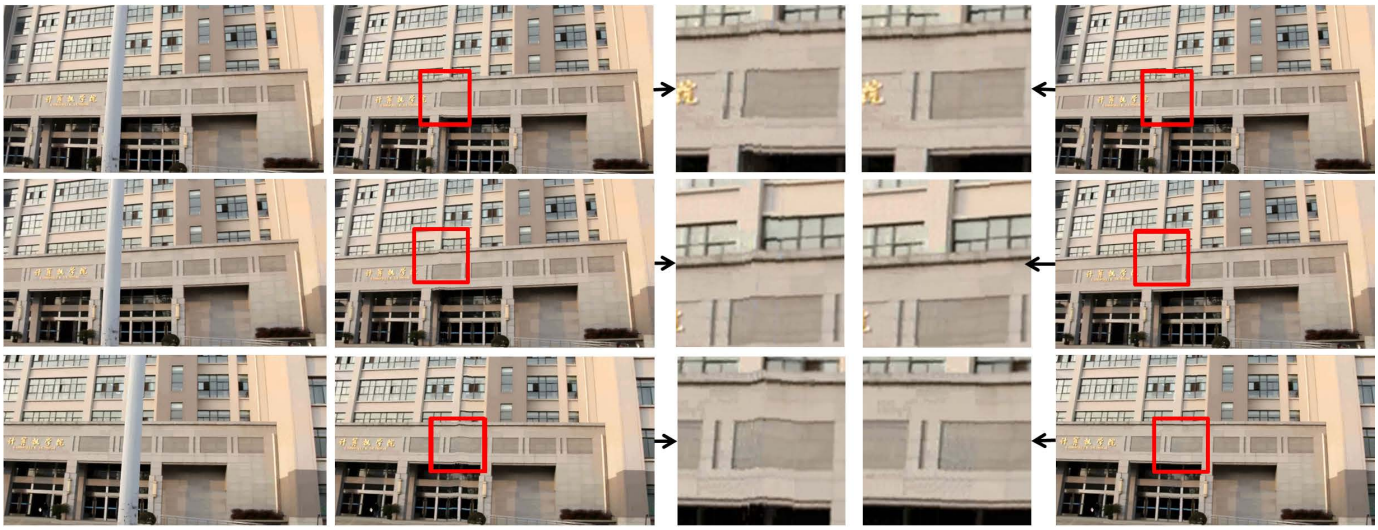
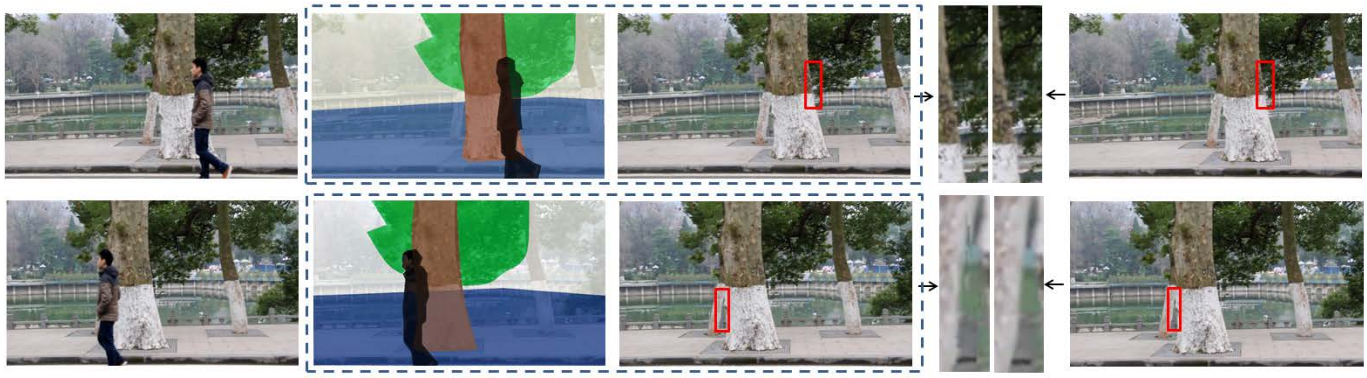


Fig. 8: Experiment 2 (Comparison with [4]). **First column:** original frames. **Second column:** results of [4]. **Third column:** close-up comparison. **Fourth column:** our results.



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

Fig. 9: Experiment 3 (Comparison with [3]). (a) Original input frames. (b) The divided planes and results of [3]. (c) Close-up comparisons. (d) Our results.

walking person ranging from the 2^{nd} frame to the 160^{th} frame. The background of this video contains much depth discontinuity, as it consists of different objects with different depth. We implement the method [3] with careful parameter setting. The automatic scene division of [3] is shown in Fig.9(b). As some small objects with various depth, such as the front trees, cannot be effectively divided, the results exhibit some artifacts near the boundaries. Moreover, temporal inconsistency occurs in their results (please refer to the accompanying demo). In contrast, our method relies on pixel-by-pixel correspondence without the need to divide the background into planes. The motion field guided method ensures temporal coherence in a more valid way, which leads to better output.

Experiment 4 (Fig.10): We present another comparison result with [3], as well as the latest video completion method [2], on a video ($960 \times 540 \times 96$) provided by [3] on their website (http://gvv.mpi-inf.mpg.de/projects/vidbginp/index.html). We focus on removing the jumping man ranging from the 6^{th} frame to the 92^{nd} frame. Results of [3] could be obtained on the same website. Spatio-temporal inconsistency occurs in the result of [3], and blurring artifacts are also generated. One main reason is that, [3] relies on plane-wise perspective transformation, so it is feasible only when the scene consists of several simple planes. When the depth of the scene exhibits abrupt discontinuity, it is difficult to divide the background into reasonable planes. For example, they cannot separate the wooden fences reasonably. Our method does not suffer from such a problem. The optical flow presents a pixel-to-pixel correspondence, and we get much better result than [3].

Methods in [2] works well for dynamic video texture, but cannot get satisfied result when the camera has large-scale rotation movement. Following the instruction in [2], we first transform all the frames in the video to the perspective of the middle frame with one homography for each frame, and run the program they released on their project page. Their results, as shown in the second row of Fig.10, also have obvious blurring and spatio-temporal inconsistency artifacts.

Experiment 5 (Fig.11 and Fig.12): We make one more comparison with [3] and [2] on an known video ($960 \times 720 \times 93$) provided by [3] on their website.

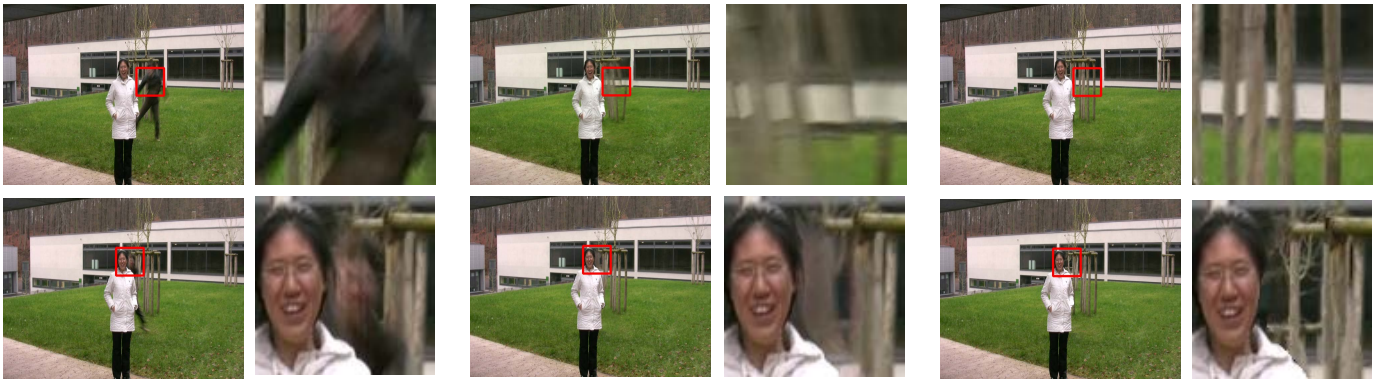


Fig. 10: Experiment 4. **First row:** comparison with [3]. **Second row:** comparison of [2]. From left to right, original frame, results of [3] (first row) and [2] (second row), our results.

Fig. 11: Experiment 5. **First row:** comparison with [3] in two frames. **Second row:** comparison with [2] in two frames. From left to right, each frame consists of the original frame, results of [3] (first row) and [2] (second row), close-up comparisons and our results.

In this video, we remove the man walking behind the woman, ranging from the 44^{th} frame to the 93^{rd} frame. As shown in the first row of Fig.11, [3] cannot recover the shape of the shadow and flower structure accurately, leading to jittering artifacts when the man occludes them. In the second row, we compare with the result of [2] (http://perso.telecom-paristech.fr/~gousseau/video_inpainting/Granados_comparisons/index_granados_comparisons.html). Their method does not suit well to videos with much perspective variation, and also fails to avoid illumination inconsistency in their result. Our method produces result with better recovered structure of background, and the illumination is better adjusted.

This outdoor video is captured by a camera with large-scale compound movement, which leads to severe illumination change. Method [3] alleviates the illumination discontinuity, but illumination gaps are still visible in their result, as shown in the second row of Fig.12. Our illumination adjustment strategy does not rely on the original gradient, and gets uniform illumination in the hole, as showed in the third row of Fig.12.

Experiment 6 (Fig.13): In this video clip ($960 \times 520 \times 27$), a weasel and its shadow are removed which ranges from the 1^{st} frame to the 27^{th} frame. The resolution of this video is low, and the sky and the wall in the background are hard for feature extraction because of uniform texture. However, our motion field guided method can effectively capture their movement according to smoothness assumption.

Experiment 7 (Fig.14): In this video ($960 \times 516 \times 54$), we remove three pedestrians (ranging from the 1^{st} frame to the 54^{th} frame) and the car (ranging from the 4^{th} frame to the 41^{st} frame), leaving only one pedestrian as reference. Due to serious information loss caused by the front car, our results exhibit a little flickering artifacts when we try to keep global spatio-temporal coherence. However, the holes are filled with little projective distortion, and the result is acceptable. In the accompanying demo, we also demonstrate the result of repeating background completion several times with fewer neighboring frames each time, which alleviates the ghost shadow artifacts.

Experiment 8 (Fig.15): We select a piece of video clip

Fig. 12: Experiment 5 (Comparison with [3]). **First row:** original frames. Full image for the 205^{th} frame, and close ups for the 201^{st}-208^{th} frames. **Second row:** corresponding results of [3]. **Third row:** corresponding results of our method.

Fig. 13: Experiment 6: **First row:** original frames. **Second row:** our results. The weasel is removed. Video credits: "Hotel Transylvania" (2012) ©Columbia Pictures.

($960 \times 540 \times 57$) from a movie, and try to remove one soldier and one woman at the same time, both ranging from the 1^{st} frame to the 57^{th} frame. We treat the other standing soldier as a part of the background. A little misalignment on his body will lead to noticeable completion error. The movement of the camera is not parallel to the scene. In the result, no misalignment artifacts occur on the soldier, which demonstrates that our algorithm is reliable in this example and similar examples.

Limitation: Our algorithm works well in most situations when the background has severe depth discontinuity, or the camera has large-scale compound movement. However, when the missing part of the video is extremely large and the distortion in the video is severe, the motion field we complete may not be so coherent with the non-hole region. In this case, if there are strong features missing in the hole, such as an obvious edge, our method may fail to recover the original structure, as illustrated in Fig.16.

Another limitation is the running time of our algorithm. It still fails to provide a fast feedback at present, which will be considered as a future work. [17] is a good attempt in this direction. They consider only temporal consistency from one frame before, thus provide real-time completion results. However, as they use limited data to fabricate the missing region, they cannot get reasonable results in complex situations, which need data from both preceding frames and subsequent ones.

Finally, it is still difficult for our motion field completion method to inpaint the motion field of articulated objects, especially for objects moving non-periodically, which few methods could handle well. We think this may be considered as a limitation for all the motion field based completion methods.

VIII. Conclusion

In this paper, we have presented a novel video background completion approach using motion-guided pixels assignment optimization. Our method aims at completing video background with complicated camera movement. We first complete the motion field of the holes by a two-step optimizing propagation. Then, with the completed motion field as guidance,

Fig. 14: Experiment 7: **First row:** original frames. **Second row:** our results. The car and three pedestrians are removed. Video credits: "21 Grams" (2003) ©Focus Features.

Fig. 15: Experiment 8: **First row:** original frames. **Second row:** our results. The walking woman and man are removed. Video credits: "The Tourist" (2010) ©Columbia Pictures.

we develop a MRF-based optimization to assign each missing pixel a value from the neighboring frames. Finally, illumination adjustment is performed to remove the illumination inconsistency artifacts in the completed background. Many video completion methods have been proposed, and every method has its advantages and disadvantages. Only a few methods have been proposed to complete video background with large-scale compound camera motion. Among these methods, our method works better at handling the background with much depth discontinuity. Thus, we believe that our work can be considered as a good complementary for the video completion community.

(a) (b)

Fig. 16: Failed example. (a) Input video with mask. (b) Background completion results.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and insightful suggestions, as well as Dr. Feng Tang for proofreading. This work was partly supported by the National Basic Research Program of China (No. 2012CB725303), the NSFC (No.41271431, No.61472288), NCET (NCET-13-0441) and the Key Grant Project of Hubei province (2013AAA02).

REFERENCES

[1] M. Granados, J. Tompkin, K. Kim, O. Grau, J. Kautz, and C. Theobalt, "How not to be seen—object removal from videos of crowded scenes," in *Computer Graphics Forum*, vol. 31, no. 2pt1. Wiley Online Library, 2012, pp. 219–228.

[2] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, "Video Inpainting of Complex Scenes," *Submitt. to SIIMS*, Jan. 2014.

[3] M. Granados, K. I. Kim, J. Tompkin, J. Kautz, and C. Theobalt, "Background inpainting for videos with dynamic objects and a free-moving camera," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part I*, ser. ECCV'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 682–695.

[4] T. Shiratori, Y. Matsushita, X. Tang, and S. B. Kang, "Video completion by motion field transfer," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 411–418.

[5] M. Bertalmio, A. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–355–I–362 vol.1.

[6] 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.

[7] Y.-T. Jia, S.-M. Hu, and R. R. Martin, "Video completion using tracking and fragment merging," *The Visual Computer*, vol. 21, no. 8-10, pp. 601–610, 2005.

[8] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting of occluding and occluded objects," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 2. IEEE, 2005, pp. II–69.

[9] K. A. Patwardhan, G. Sapiro, and M. Bertalmío, "Video inpainting under constrained camera motion," *Image Processing, IEEE Transactions on*, vol. 16, no. 2, pp. 545–553, 2007.

[10] M. Vijay Venkatesh, S.-c. S. Cheung, and J. Zhao, "Efficient object-based video inpainting," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 168–179, 2009.

[11] N. Tang, C.-T. Hsu, C.-W. Su, T. Shih, and H.-Y. M. Liao, "Video inpainting on digitized vintage films via maintaining spatiotemporal

continuity," *Multimedia, IEEE Transactions on*, vol. 13, no. 4, pp. 602–614, Aug 2011.
[12] M. Liu, S. Chen, J. Liu, and X. Tang, "Video completion via motion guided spatial-temporal global optimization," in *Proceedings of the 17th ACM International Conference on Multimedia*, ser. MM '09. New York, NY, USA: ACM, 2009, pp. 537–540.
[13] T. K. Shih, N. C. Tang, and J.-N. Hwang, "Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 3, pp. 347–360, 2009.
[14] J. Jia, W. Tai-Pang, Y.-W. Tai, and C.-K. Tang, "Video repairing: inference of foreground and background under severe occlusion," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, June 2004, pp. I–364–I–371 Vol.1.
[15] J. Jia, Y.-W. Tai, T.-P. Wu, and C.-K. Tang, "Video repairing under variable illumination using cyclic motions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 5, pp. 832–839, May 2006.
[16] Y. Shen, F. Lu, X. Cao, and H. Foroosh, "Video completion for perspective camera under constrained motion," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3, 2006, pp. 63–66.
[17] J. Herling and W. Broll, "High-quality real-time video inpaintingwith pixmix," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, no. 6, pp. 866–879, June 2014.
[18] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision." in *IJCAI*, vol. 81, 1981, pp. 674–679.
[19] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics-TOG*, vol. 28, no. 3, p. 24, 2009.
[20] S. Ilan and A. Shamir, "Data-driven video completion," in *State of the Art Report, Proceedings Eurographics'14*, 2014.
[21] A. Agarwala, A. Hertzmann, D. H. Salesin, and S. M. Seitz, "Keyframe-based tracking for rotoscoping and animation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 584–591, Aug. 2004.
[22] C.-Y. Chung and H. Chen, "Video object extraction via mrf-based contour tracking," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 1, pp. 149–155, Jan 2010.
[23] Y. Matsushita, E. Ofek, X. Tang, and H.-Y. Shum, "Full-frame video stabilization," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 50–57. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.166
[24] L. Chen, S. Chan, and H. Shum, "A joint motion-image inpainting method for error concealment in video coding," in *Image Processing, 2006 IEEE International Conference on*, Oct 2006, pp. 2241–2244.
[25] C. Liu, W. Freeman, E. Adelson, and Y. Weiss, "Human-assisted motion annotation," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.
[26] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping." Springer, 2004, pp. 25–36.
[27] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 1–8, 2008.
[28] Y. Nie, Q. Zhang, R. Wang, and C. Xiao, "Video retargeting combining warping and summarizing optimization," *The Visual Computer*, vol. 29, no. 6-8, pp. 785–794, 2013.
[29] D. M. Mount and S. Arya, "ANN: A library for approximate nearest neighbor searching," 1997. [Online]. Available: http://www.cs.umd.edu/mount/ANN/
[30] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*. AMS, 1980.
[31] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
[32] P. Bhat, C. L. Zitnick, N. Snavely, A. Agarwala, M. Agrawala, M. Cohen, B. Curless, and S. B. Kang, "Using photographs to enhance videos of a static scene," in *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, ser. EGSR'07. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007, pp. 327–338.
[33] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, Jul. 2003.
[34] C. Xiao, D. Xiao, L. Zhang, and L. Chen, "Efficient shadow removal using subregion matching illumination transfer," in *Computer Graphics Forum*, vol. 32, no. 7. Wiley Online Library, 2013, pp. 421–430.
[35] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, May 2002.
[36] C. Xiao, M. Liu, N. Yongwei, and Z. Dong, "Fast exact nearest patch matching for patch-based image editing and processing," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 8, pp. 1122–1134, 2011.
[37] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer graphics and applications*, vol. 21, no. 5, pp. 34–41, 2001.
[38] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 8, pp. 837–842, Aug 1996.
[39] A. Bousseau, F. Neyret, J. Thollot, and D. Salesin, "Video watercolorization using bidirectional texture advection," in *ACM SIGGRAPH 2007 Papers*, ser. SIGGRAPH '07. New York, NY, USA: ACM, 2007.

Zhan Xu received his BSc degree from the School of Digital Media, Jiangnan University in 2012. Currently, he is a master student at the School of Computer, Wuhan University. His research interests include image and video processing.

Qing Zhang received the BSc and MSc degrees from the School of Computer, Wuhan University, in 2011 and 2013, respectively. Currently, he is working toward the PHD degree at School of Computer, Wuhan University, China. His research interests include image and video processing and computational photography.

Zhe Cao is currently an undergraduate student from the School of Computer, Wuhan University. His research interests include image and video processing.

Chunxia Xiao received his BSc and MSc degrees from the Mathematics Department of Hunan Normal University in 1999 and 2002, respectively, and received his Ph.D. from the State Key Lab of CAD & CG of Zhejiang University in 2006, China. He is currently a professor at the Computer School, Wuhan University, China. His research interests include digital geometry processing, image and video processing, and computational photography.