Background Inpainting for Videos with Dynamic Objects and a Free-moving Camera

Miguel Granados[1], Kwang In Kim[1], James Tompkin[1,2,3],
Jan Kautz[2], and Christian Theobalt[1]

[1]Max-Planck-Institut für Informatik, Campus E1 4, 66123 Saarbrücken, Germany
[2]University College London, Malet Place, WC1E 6BT London, UK
[3]Intel Visual Computing Institute, Campus E2 1, 66123 Saarbrücken, Germany

Abstract. We propose a method for removing marked dynamic objects from videos captured with a free-moving camera, so long as the objects occlude parts of the scene with a static background. Our approach takes as input a video, a mask marking the object to be removed, and a mask marking the dynamic objects to remain in the scene. To inpaint a frame, we align other candidate frames in which parts of the missing region are visible. Among these candidates, a single source is chosen to fill each pixel so that the final arrangement is color-consistent. Intensity differences between sources are smoothed using gradient domain fusion. Our frame alignment process assumes that the scene can be approximated using piecewise planar geometry: A set of homographies is estimated for each frame pair, and one each is selected for aligning pixels such that the color-discrepancy is minimized and the epipolar constraints are maintained. We provide experimental validation with several real-world video sequences to demonstrate that, unlike in previous work, inpainting videos shot with free-moving cameras does not necessarily require estimation of absolute camera positions and per-frame per-pixel depth maps.

1 Introduction

Imagining automatic object removal from videos conjures up many powerful applications: removing all other tourists from your holiday videos, removing unavoidably visible film crews from movie footage, or removing anachronistic elements from period pieces. While some progress has been made towards this goal in recent years, there are often many restrictions upon the input footage which need to be overcome. We present an algorithm for relieving the requirement for cameras to be static, enabling inpainting on footage captured with free-moving cameras without the use of dense per-frame geometry reconstruction.

Object removal requires inpainting the hole left by an object with background. With a static or scene-parallel-moving camera, the background stays mostly constant or moves only linearly (ignoring illumination changes). Existing image and video inpainting algorithms exist to solve these problems (Sec. 2), and they assume that the occluded region is *visible* in other frames, e.g., the background occluded by a person at a given frame might be revealed in another

frame as he or she moves. However, the notion of visibility in these algorithms is restrictive as the perspective of an occluded object or background region cannot change throughout the video. This is not the case for free-moving cameras, where often the camera motion is highly dynamic and nonlinear and, as such, the appearance (i.e., the projected image) of objects varies with perspective changes.

In this paper, we relax the notion of visibility: An occluded region is *visible* if it appears in other frames of the video, even under projective distortions. In general, the occluded part of the background appears in the video under different camera perspectives and accordingly, during the selection and blending, we compensate for the corresponding distortions.

If accurate 3D geometry of the scene and accurate 3D camera registration are available, then there are existing approaches to solve this problem [1–3]. However, manually constructing accurate 3D models for arbitrary scenes is costly and time consuming, and current automatic methods for geometry reconstruction, either in hardware or in software, are not always applicable and often require manual intervention. The dual problem of finding accurate camera positions and extracting depth maps suffers similar problems and is not applicable to all footage, especially footage with a narrow baseline or a rotating-only camera [4], or footage with a single dominant plane [5].

Our new method takes advantage of the geometric properties of the scene; however, it bypasses the dense depth estimation or 3D reconstruction steps. We perform frame-to-frame alignments using a set of homographies which are chosen such that they conform with the epipolar geometry whenever possible (Sec. 3.1). We use the subset of aligned frames in which the missing region is visible as candidates for a MRF-based inpainting process (Sec. 3.2). In a post-processing step, we remove luminance discrepancies between frames by performing gradient domain fusion (Sec. 3.3). We present experimental validation of our pipeline on several real-world videos from which objects are successfully removed (Sec. 4).

2 Previous work

Most existing non-parametric video inpainting algorithms can be regarded as extensions of relatively well-established image inpainting methods. These algorithms synthesize the pixel values in the hole as a combination of video patches (localized 3D windows in video volumes) sampled from visible regions of the video. Each algorithm is characterized by the energy functional that measures the compatibility of candidate inpainting patches at the boundary of the hole and, at the same time, the compatibility of different candidate inpainting patches with each other. The energy functional is either defined explicitly (globally) or implicitly (locally) and the video inpainting is formulated as a global or local energy minimization accordingly. Patwardhan et al. [6] proposed a local method that assigns a priority to every hole pixel based on the existence and direction of motion across the hole boundary. Proceeding by highest priority, the method copies those patches that best match the context of the pixel of interest. This algorithm was later improved to handle parallel-to-image-plane camera motions [7,

8]. Wexler et al. [9] proposed an explicit formulation defined over spatio-temporal patches sampled from every visible pixel in the input video. Completion is performed by solving a global energy functional through iteratively assigning to each missing pixel the most likely color among those patches that are closest to the current solution. Related approaches can be found in [10–12]. Spatio-temporal patches vary significantly with camera motion and thus they can no longer be matched to each other, even when perspective distortions are not severe. Non-parametric methods do not generalize well to free-moving cameras (see Sec. 4).

Several video completion techniques rely on specific knowledge of the region to be inpainted. For instance, if the hole contains specific objects (e.g., humans) then model-based approaches can be used [13, 14]. Assumptions on the specific motion (e.g., cyclic motion) can also be exploited [13, 15]. The algorithm of Venkatesh et al. [13] tracks and segments an occluded human and constructs a database of segmented frames where the occluded object is fully visible. The holes are filled by aligning frames in the database to the partially or fully occluded frames in the hole. This idea was further extended in [16]. Our method belongs to the model-based category, but we do not rely on modeling nor learning the structure of specific types of objects. Instead, we make geometric assumptions about the region to be inpainted: the occluded object is approximately piecewise planar and static, so that variation is mainly caused by camera motion and illumination changes. However, unlike previous approaches, we do not assume that the occluded object is visible from the same viewpoint in another frame. This is possible since our algorithm rectifies the distortion from viewpoint changes.

On free camera videos, the most related algorithm was proposed by Bhat et al. [1]. They provide a framework for performing several video editing operations on videos of static scenes, including object removal. They obtain a dense depth map for every frame using multi-view stereo (MVS), which they use to reconstruct the video with the desired edits. Camera parameters are estimated using structure-from-motion (SfM) [17]. Their video reconstruction algorithm assumes initial depth and color estimates for every frame. Following an MRF framework, they compute a composite of coherent patches from warped neighboring frames (or additional photographs). For object removal, they compute initial depth and color estimates as the median of reprojected non-hole regions of nearby frames. Sources with large depth discrepancies are discarded, and the remaining sources are penalized according to their color difference. Each frame is reconstructed independently, and temporal incoherences are removed by a gradient domain fusion algorithm. Our method has similarities with this pipeline but contains important differences. We reconstruct the occluded background using reprojected frames by compositing a few compatible, unoccluded sources, and we reduce temporal incoherences through gradient domain fusion. The fundamental difference is that our method does not require dense depth estimates for every frame, therefore bypassing the need for camera calibration and multi-view-stereo, which are both non-trivial to estimate and can fail on certain content (see Sec. 4). Instead, we construct a set of hole-filling hypotheses and choose the most suitable one. This provides inpaintings using plausible information about scene geometry.

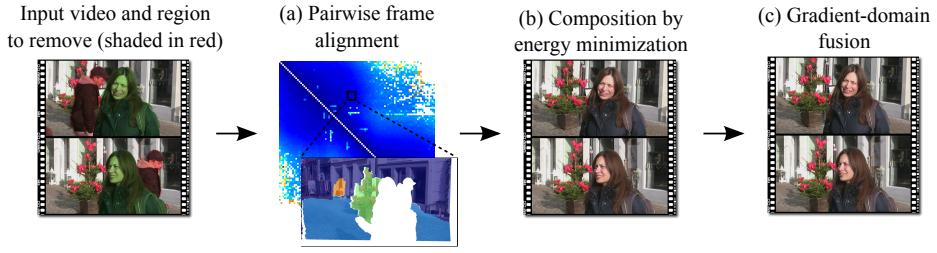

Fig. 1. Our inpainting pipeline proceeds as follows: (a) the input frames are pairwise aligned based on a set of local homographies; (b) the inpainting result is composited by minimizing a global energy functional which trades between the compatibility among aligned local image regions and the deviation from a guide image (a weighted average per pixel of the aligned pixels); and (c) in the post-processing stage, gradient-domain fusion is performed to remove potential illumination discrepancies.

3 Video Inpainting Method

The general problem of inpainting occluded regions within dynamic scenes is interesting and has wide applicability. However, contemporary technology is far from able to achieve plausible general video inpainting even with a fair amount of user interaction. As a step toward this goal, we focus on a specific set of applications which are of practical interest. First, we assume that the object to be removed occludes static background. This does not imply that the appearance of the background in the hole is static. Changes in both camera viewpoint and scene illumination (frequent in outdoor scenes) cause significant background appearance changes, as shown in our experiments (Sec. 4). Second, our algorithm assumes that the region behind the hole is unoccluded in at least one frame.

The inpainting problem is cast into one of identifying potential *source* frames and aligning and compositing them into the current *target* frame (Sec. 3.1). We do not assume that a single source frame covers the entire hole. Accordingly, one has to composite different sources in a coherent manner. For instance, it is better to copy pixels from a frame which has a similar camera viewpoint and illumination. We solve the compositing problem with a global energy functional, which we describe in Sec. 3.2. The rest of this section details the method of identifying and aligning source frames into target frames and dealing with global illumination changes (Sec. 3.3). The overall pipeline is illustrated in Fig. 1.

The input to our inpainting method is a video sequence, and a user-provided mask for the hole region to be filled (containing the object to be removed). Optionally, the user can provide a mask for other dynamic objects that should not be used as sources during the inpainting. The video of interest will be represented as a 3D volume $V : \Omega \otimes \{1, \ldots, T\} \mapsto \mathbb{R}^3$, where Ω is the discrete domain of all pixels in a frame (i.e., $\Omega = \{1, \ldots, m\} \otimes \{1, \ldots, n\}$ with m and n being the height and width of a frame), T is the number of total frames in the video, and each pixel in the video has 3-tuple color values. The hole $\mathcal{H}$ is represented as an index set on V. The optional dynamic region $\mathcal{F}$ is defined identically as

$\mathcal{H}$. The t-th frame in V and the corresponding hole therein will be denoted as V_t and $\mathcal{H}_t$, respectively. The (i, j)-th pixel in the t-th video is then represented with $V_t(i, j) := V(i, j, t)$ and the corresponding pixel in the hole is denoted in the same way: $\mathcal{H}_t(i, j) := \mathcal{H}(i, j, t)$.

3.1 Frame alignment

The first step of our algorithm is to generate candidates for each pixel in the hole. These candidates are generated independently for each frame t containing the hole $\mathcal{H}_t$. A candidate pixel originates from a source frame V_s, $s \in \{1, \ldots, T\} \setminus \{t\}$, which is transformed to compensate for the viewpoint difference F_{st} between V_s and V_t. We refer to transforming a frame V_s into $F_{st}(V_s) \approx V_t$ as *alignment*.

While there are various different methods for estimating F_{st}, we choose the homography as the basic element. In general, a single homography between a pair of frames does not provide a reasonable estimate of F_{st} since the homography relies on linear geometric assumptions which are infrequently the case in practice for most scenes. Instead, we approximate the scene by piecewise linear geometry, i.e., by an arrangement of planes. This allows us to align a frame pair by decomposing them into regions which each can be placed into correspondence by a homography, and thus concur with a local planarity assumption.

To obtain an alignment F_{st}, we first compute a set of candidate homographies between V_s and V_t, and then for each pixel we decide on a single homography that minimized the resulting alignment error. The result of this process is illustrated in Fig. 2. Generating candidate homographies starts with establishing geometrically consistent feature correspondences. First, we identify potential feature correspondences and discard outliers by estimating fundamental matrices using RANSAC. For consecutive frames, potential feature correspondences are obtained by KLT tracking [18, 19]; for more distant frame pairs, we perform approximate nearest neighbor matching on SURF features [20]. Once geometrically consistent inlying feature correspondences are obtained between the source and target frame, a set of homographies are adaptively estimated in an incremental scheme: At step 1, a homography is estimated from the current set of feature correspondences after outliers are identified with RANSAC. At step n, the set of input feature correspondences are replaced by the outliers determined at step $n - 1$ and a homography is estimated again. The process iterates until k_{max}-homographies are determined.

Once we have the set of candidates, we use the *expansion-move algorithm* [21–23] to assign a homography for each pixel. The algorithm finds an assignment that is a trade-off between the alignment error (i.e., color differences between the corresponding pixels, identified with the homography of interest) and the mismatch at the boundary between two adjacent regions aligned by different homographies. Let $H_{st} = \{H_{st}^1, \ldots, H_{st}^k\}$ be the set of candidate homography matrices that align V_s to V_t. The homography best aligning each pixel is found by minimizing the energy functional:

$$\mathcal{E}(K) = \sum_{p \in \Omega} E_p^1(K(p)) + \beta \sum_{(p,q) \in \mathcal{N}(\Omega)} E_{p,q}^2(K(p), K(q)), \tag{1}$$

Fig. 2. Homography-based frame alignment. *Top-left:* Target frame with the region to be inpainted shaded in red; *Top-right:* A source frame where the region to be inpainted is partially visible (the remaining parts would need to come from other sources); *Bottom-left:* The target frame is partially filled using the aligned source; *Bottom-right:* Overlay between the aligned source and the mapping K that selects the homography used to align each region. The linearity of homographies allows the algorithm to effectively extrapolate into the hole, for which no reference colors are known.

where $\mathcal{N}$ denotes the spatial neighborhood system (4-neighbors in the current algorithm), and $K : \Omega \to [1 \dots k]$ is the variable corresponding to the assignment of a homography $H_{st}^{K(p)}$ to a pixel $p \in \Omega$. The factor β balances the importance of the two terms; we set $\beta = 10$ in all our experiments. The alignment F_{st} is then given as $F_{st}(p) := H_{st}^{K^*(p)} p_h$, where p_h is p defined in homogeneous coordinates, and K^* is the labeling that minimizes Eq. 1. The data term

$$E_p^1(k) = C_{st}^k(p) \cdot ||V_t(p) - V_s(H_{st}^k p_h)||_2 \tag{2}$$

measures the color differences between the source and target frames if aligned by H_{st}^k. The smoothness term

$$E_{p,q}^2(u,v) = \mathbf{1}_{\{u \neq v\}} \left(\|V_s(H_{st}^u p_h) - V_s(H_{st}^v p_h)\|_2 + \|V_s(H_{st}^u q_h) - V_s(H_{st}^v q_h)\|_2 \right) \tag{3}$$

penalizes the pairwise color discrepancies between two adjacent pixels p, q in the source frame when they are aligned using distinct homographies H_{st}^u, H_{st}^v [24].

We include the factor $C_{st}^k(p)$ to represent the *compatibility* between the candidate homography H_{st}^k at pixel p and the fundamental matrix f_{st} between frames s, t (Fig. 3). This factor follows from the rationale that a single homography is very unlikely to provide a good alignment of the whole scene (unless it is composed of a single plane), and thus, each homography should only be used to align regions where it is compatible with the scene geometry. We approximate this using the epipolar constraints between the views. Therefore, we define the compatibility as the distance $(p_h f_{st})(H_{st}^k p_h)$ between the epipolar line of p in frame t and its location predicted by the homography:

$$C_{st}^k(p) = 1 - \left[\exp\left(-\frac{1}{2} \frac{\left(p_h f_{st} H_{st}^k p_h\right)^2}{\mathrm{r}^2} \right) - \frac{1}{2} \right], \tag{4}$$

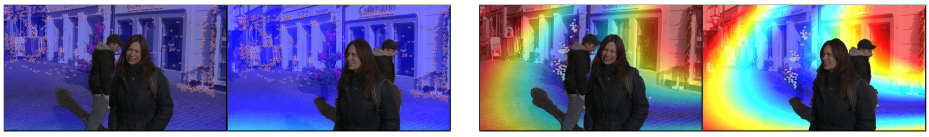

Fig. 3. Compatibility between fundamental matrix and homographies: Color-coded score for the first (left pair) and second (right pair) homographies (out of four) aligning the left and right frames shown in Fig. 2, and the key-points used to estimate them. This score encodes the spatially varying compatibility between each candidate homography and the fundamental matrix between the frames (blue: compatible, red: incompatible).

where r is the reprojection error threshold for homography/F-matrix estimation. Unlike SfM, a unique estimate of the fundamental matrix is not critical to our algorithm. For instance, when the camera motion is degenerate, there is a class of fundamental matrices in which only one is correct. However, we use the estimated fundamental matrix only for calculating the weight in Eq. 4. For the given class of degenerate fundamental matrices, the weights are all uniform. Accordingly, we can obtain the correct weighting in this degenerate case.

When assigning homographies from the source to the target frame, it is possible for a source pixel to be mapped to multiple target locations. Repetitiveness is not an issue in regions of uniform appearance; however, in highly structured regions, repetitiveness leads to artifacts in structures that should be unique. We prevent this situation by inverting the aligning process, i.e., we align the target to the source frame, find an optimal homography assignment, and then transform the region supporting each homography back to the domain of the target.

The energy (Eq. 1) is minimized with graph cuts. Before optimization, we detect and remove some spurious homographies that (a) do not preserve orientation, (b) show disproportionate scaling along only one axis (i.e., high ratios between the first two eigenvalues of the homography matrix; threshold set at 0.1), or (c) produce an area scaling that varies too much with position (i.e., the norm of the homography's projectivity vector is larger than a threshold set at 0.1), so the appearance of the target view cannot be properly reconstructed from the source due to discretization. Although the latter two cases can occur in practice, such projections are unlikely to be correctly detected since most interest point detectors are only invariant up to affine transformations. The two threshold parameters were fixed at these values for all experiments.

3.2 Scene composition

Frame alignment provides inpainting of the target frame V_t with parts of the source frames V_s. In general, there are several source frames in a video that partially or completely cover the hole $\mathcal{H}_t$. For each pixel in $\mathcal{H}_t$, a single frame must be chosen from all candidates found during frame alignment to produce a color value that is spatially consistent with its neighbors inside and outside the hole. Let $S_t : \mathcal{H} \rightarrow \{1 \dots T\}$ be the mapping specifying a source frame for every

hole pixel $p \in \mathcal{H}_t$. We obtain S_t by minimizing the energy functional:

$$\mathcal{E}'(S_t) = \sum_{p \in \mathcal{H}_t} {E'}^1_p(S_t(p)) + \gamma \sum_{(p,q) \in \mathcal{N}(\mathcal{H}_t \cup \partial\mathcal{H}_t)} {E'}^2_{p,q}(S_t(p), S_t(q)), \tag{5}$$

where $\mathcal{H}_t \cup \partial\mathcal{H}_t$ denotes the hole pixels and the non-hole pixels at its boundary, and γ balances the importance of the two terms ($\gamma = 10$ in our experiments).

The *smoothness term* ${E'}^2_{p,q}(u, v)$ measures the local color discrepancies [24] between two distinct source frames u, v had they been chosen to fill two adjacent hole pixels p, q, respectively:

$${E'}^2_{p,q}(u, v) = \mathbf{1}_{\{u \neq v\}} \Big(\|W^t_u(p) - W^t_v(p))\|_2 + \|W^t_u(q) - W^t_v(q)\|_2 \Big), \tag{6}$$

where W^t_u denotes frame $V_u(F_{ut})$ aligned to the current target V_t.

To guide the source frame selection process, we compute an initial inpainting:

$$R_t(p) = \frac{\sum_{l=1}^{T} a^t_l W^t_l(p)}{\sum_{u=1}^{T} a^t_l}, \tag{7}$$

where each pixel is filled with the weighted average of the candidate pixel values (see Fig. 4). The *alignment score* a^t_l represents the quality of the alignment between frames l and t. Let $e^t_l(q) = \|W^t_l(q) - V_t(q)\|_2$ be the color difference between two aligned non-hole pixels; the score a^t_l is given by

$$a^t_l = \exp\left(-\frac{A^t_l}{\sigma_{A^t}}\right), \text{ with } A^t_l = \frac{\sum_{p \in \Omega \setminus \mathcal{H}_t} d_{\mathcal{H}_t}(p) e^t_l(p)}{\sum_{p \in \Omega \setminus \mathcal{H}_t} d_{\mathcal{H}_t}(p)}, \tag{8}$$

and σ_{A^t} is the standard deviation of $\{A^t_l\}_{l=1\ldots T}$. Here, the weight $d_{\mathcal{H}_t}(p) = \exp\left(-\frac{1}{2}\frac{\mathrm{D}(p,\mathcal{H}_t)}{\sigma_d}\right)$ penalizes misalignments located close to the boundary of the hole, D is the distance transform, and σ_d represents the score fall-off, which is set to $\sigma_d = 8$ in our experiments.

We use a weighted mean as a guide for the optimization process instead of a mode. If we use the mode, one color is picked per pixel as a guide, and this color is likely to be selected during the graph-cut refinement since the cost of other colors is large. If we use the mean color, the costs of all potential source colors will be larger but similar. Hence, the optimization will emphasize more the opinion of the neighbors via the smoothness term. This can be seen as a label propagation from the neighbors whenever the data term is deemed uncertain. As such, tested empirically, our weighted mean gave better results than the mode.

Frame alignments and their corresponding scores are computed for every pair of images (T^2 scores, see Fig. 1-a), or for a sliding window of n frames around each target frame (nT scores). Other sampling strategies to reduce the number of candidate frames could be devised, such as randomized sampling and region growing [25]; we leave this issue for future work.

For any given pixel, the blending *data term* (the first term in (5)) is:

$${E'}^1_p(u) = \|W^t_u(p) - R_t(p)\|_2, \tag{9}$$

Fig. 4. The weighted average of aligned source frames is taken as guide for the optimization process: (left) input; (middle) guiding average; (right) resulting inpainting composite.

which penalizes the color difference between an aligned source frame W_u^t and the corresponding reference color obtained in Eq. 7.

We evaluate the energy functional only for cases where a candidate color $W_u^t(p)$ is properly defined, i.e., (a) when a correct alignment was found between the target and source frames, (b) when the corresponding source pixel is not a hole or dynamic pixel at the source, and (c) when the source pixel's projection lies within the target frame. Ill-defined candidates are excluded during optimization.

Discussion. An alternative to the first part of our method, i.e. aligning source frames to the target, is to perform structure-from motion (SfM) and multi-view-stereo (MVS), and use the estimated data (camera projection matrices and depth maps) to render the occluded regions from the target's viewpoint. However, this approach introduces two limitations: First, performing camera calibration using SfM implies that the camera translation needs to be sufficiently large as to properly triangulate the location of interest points, thus limiting the space of input videos. Additionally, since SfM is sensitive to the initial estimate of focal length, it might fail to provide correct camera locations on sequences where the focal length is variable. Our algorithm does not suffer from this restriction on camera movement. Also, it does not require camera intrinsics (e.g., focal length), thus it is independent of errors in their estimation.

Furthermore, MVS methods require that either the scene is completely static [1], or that the scene is simultaneously captured from different viewpoints using many cameras. In our method, we recover only as much geometrical information about the scene (homographies and fundamental matrices) as required to perform a plausible inpainting, thus avoiding the restrictions of recovering complete dense depth maps. In the limit, our method corresponds to pairwise stereo if we allow arbitrarily many homographies.

3.3 Handling illumination changes

Outdoor scenes typically exhibit changes in illumination. Since our energy functional enforces color value consistency, our algorithm tends to reconstruct the scene from illumination consistent source regions. However, when the illumination of the target frame is different from all sources, the result contains artificial boundary artifacts (Fig. 5). To resolve this, we perform gradient-domain fusion by solving the Poisson equation with Dirichlet boundary conditions to remove

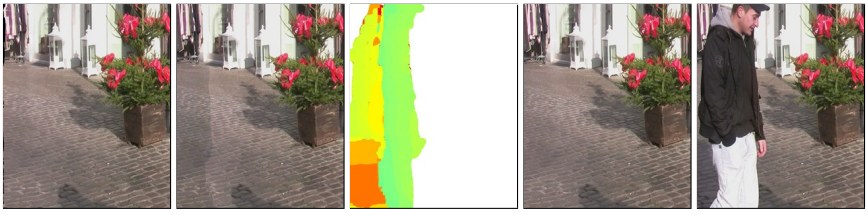

Fig. 5. Gradient-domain fusion. From left to right: previous frame (inpainted, blended) aligned using optical flow; current inpainted frame; color-coded labeling of the time-stamp of the chosen sources for the current frame; resulting gradient-domain fusion; original current frame for reference.

potential lighting differences [26]. To maintain temporal consistency, we introduce a regularizer which penalizes discrepancies between the reconstructed colors and their corresponding colors in the (optical-flow-aligned) previous frame [27]. Given the colors of the current and previous inpainted frames $\{f_p^*\}, \{g_p^*\}$, respectively, the Poisson-blended colors $\{f_p\}$ can be obtained by minimizing the discretized energy functional:

$$\min_f \sum_{(p,q)\in\mathcal{N}(\Omega)} [(f_p - f_q) - v_{pq}]^2 + \lambda \sum_{p\in\Omega} (f_p - g_p^*)^2, \tag{10}$$

where λ is the weight balancing the importance between the spatial and temporal boundary conditions. In our experiments, we set this value to half of the ratio of the total number of boundary conditions. To cope with gradient mismatches at the boundary between source regions, we set the corresponding guiding gradients to zero, i.e., $v_{pq} = \mathbf{1}_{\{S(p)=S(q)\}}(f_p^* - f_q^*)$. The result is obtained by solving the linear system

$$(|\mathcal{N}_p| + \lambda) f_p - \sum_{q\in\mathcal{N}_p^{\mathcal{H}}} f_q = \sum_{q\in\mathcal{N}_p^{\mathcal{H}}} v_{pq} + \sum_{q\in\mathcal{N}_p^{\partial\mathcal{H}}} f_q^* + \lambda g_p^*, \tag{11}$$

where $\mathcal{N}_p \equiv \mathcal{N}(p)$ and $\mathcal{N}_p^{\mathcal{H}} \equiv \mathcal{N}(p) \cap \mathcal{H}$, which in our implementation is computed by the conjugate gradients method.

4 Experiments and Discussion

To experimentally validate our algorithm, we use seven real-world sequences in four different scenes (S1-S7 in Fig. 6; all available on our project website). The sequences were captured with a hand-held Canon HV20 digital camcorder in anamorphic Full HD resolution at 25fps. Rolling shutter artifacts are present, particularly in S2, as this camera has a CMOS sensor. S1, S2 and S7 have 95, 100, and 80 frames respectively, and were processed at 1440x1080 resolution; S3-S6 have 180, 270, 225, and 220 frames, respectively, and were down-sampled to 960x720 pixels. The short sequences have every frame aligned to every other,

and the remaining sequences were aligned using sliding windows of sizes $n = 50$ to 100. All alignments were performed using the method presented in Sec. 3.1. The running times ranged from one hour ($n = 50$) to four hours ($n = 100$) running in parallel on a frame server with 64 logical processors.

Each sequence features two or more people moving in front of a static background; we remove one person from each. The mask of the object to be removed and the mask of the remaining foreground objects were created semi-automatically using the implementation of [28] available in Adobe After Effects CS5 (Fig. 6-a). S1, S2, S5, and S7 have small view point variations and narrow baselines, and S3, S5, and S6 were captured with a view point span of 10–20 degrees around the object of interest. S3 and S7 have varying focal lengths caused by zooming. All inpaintings were computed using identical parameters.

In S1-S2, the scene has three dominant planes: ground, facade, and tree. The facade contains non-trivial geometry (lamps, doorways, and showcases). Our algorithm produced perceptually plausible inpaintings despite the presence of small objects (tree in S1 and lamp in S2). This is because only one correct alignment per hole region is required to perform a reconstruction. In S3, the scene has a relatively uniform background under motion blur. This is a challenging sequence for our algorithm due to the lack of distinctive features. Feature matching performs poorly, leading to less suitable homography candidates.

S4-S6 have two feature rich, dominant planes (ground and facade); hence, our algorithm produces plausible inpaintings. However, part of the shadow of the removed person is still visible. This is caused by inaccuracies in the user-provided mask, which could be remedied by additional manual correction. In general, this is a challenging issue since the shadow boundary is difficult to mask correctly. Future work on simultaneous alignment and object segmentation that takes into account these luminance differences could address this issue.

As we use binary masks, semi-transparent objects (such as hair) are either inpainted with background or kept as is, causing temporal inconsistencies. Layer separation could be applied to inpaint individual layers separately [29].

Lastly, temporal inconsistencies are visible in some cases, especially in S3, since our algorithm does not directly enforce it (cf. [30]). In general, enforcing temporal consistency should improve the inpainting result. However, it is not straightforward to implement, since testing temporal coherence requires three pairs of correctly aligned frames, which does not always occur in practice. When we imposed this restriction, the number of possible sources was restricted by unavailable alignments, leading to less favorable results. We leave more sophisticated methods for temporal consistency as future work.

Nevertheless, our algorithm generates high quality results for sequences which do not significantly deviate from our assumptions (piecewise-planar geometry, non-flat textures); these cover a large range of scene and object configurations.

To illustrate the importance of perspective distortion correction, we perform a comparison with a non-parametric inpainting method [12] (see Fig. 7-b). As identical perspective distortions of the same object are unlikely to appear in videos with free-moving cameras, the result is geometry-inconsistent. Addition-

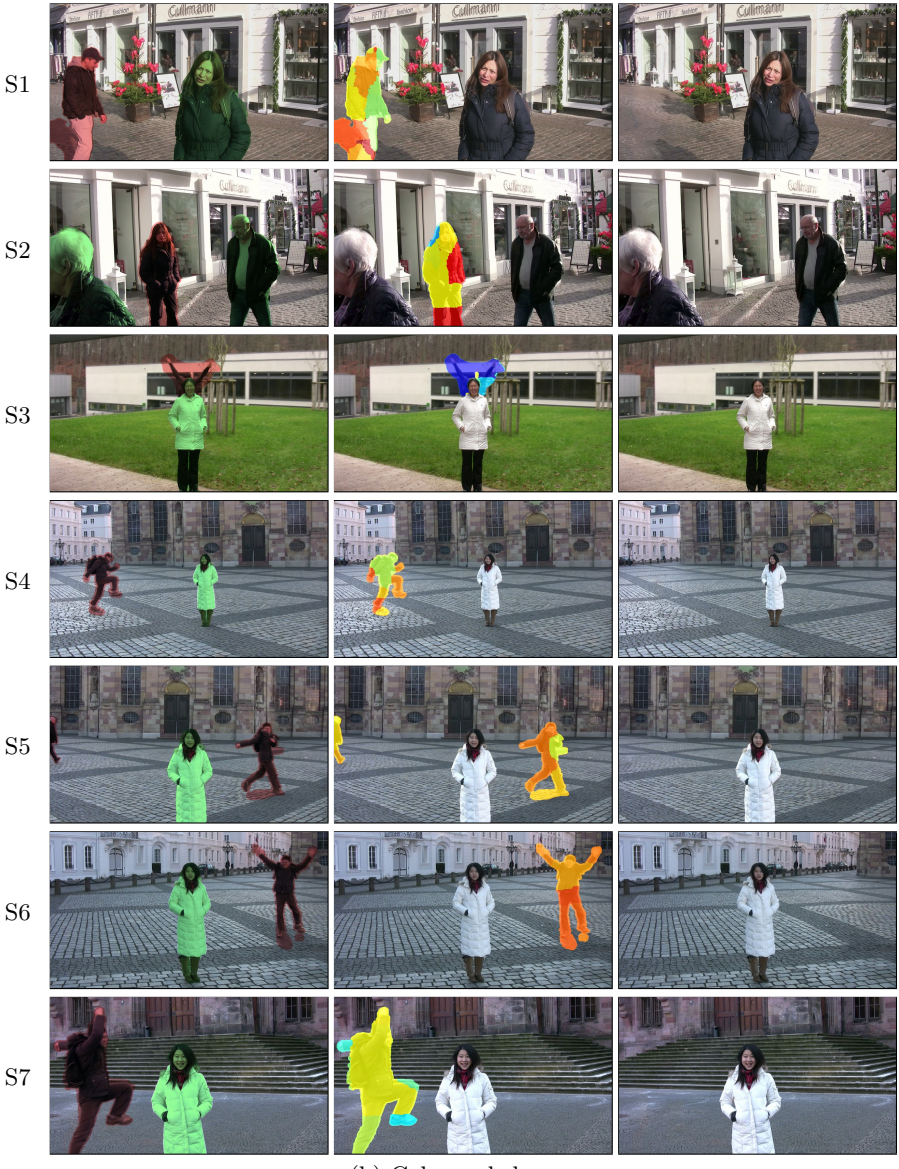

Fig. 6. Results of inpainting in our seven test sequences: (a) an input frame and overlayed masks for the hole and dynamic objects (shaded in red and green, respectively); (b) visualization of the source frames obtained by minimizing Eq. (5); (c) inpainting after compositing and gradient domain fusion. Subjects in S3-S7 were kindly provided by the Ministry of Silly Walks.

(a) Input (b) Our result (c) Result by [12] (d) SfM failure

Fig. 7. Non-parametric inpainting methods [12, 9, 7] do not support general camera motion and can produce inconsistent results ((c), in red). SfM also fails in this sequence: the reconstructed geometry is almost planar even though there is an obvious ground plane, and some of the cameras are estimated to be behind the scene ((d), upper-left).

ally, the SfM method of Snavely et al. [17] (used in [1]) fails to produce correct calibration in all our sequences (see Fig. 7-d , and supplementary material). This prevented a subsequent application of MVS as performed in [1].

5 Conclusions

We proposed a method for removing objects from free-camera videos. We inpaint the region behind them by compositing background regions visible in other frames. By assuming the scene can be decomposed into piece-wise planar regions, we correct perspective distortions caused by camera motion by estimating a composite of homography-aligned image sub-regions. Since there may be multiple candidate frames to fill the hole, the selection of pixel sources is formulated as a global energy minimization. Spurious illumination mismatches are removed using Poisson blending. In contrast to methods based in multi-view-stereo, our algorithm does not require recovering camera locations and depth map estimates.

References

1. Bhat, P., Zitnick, C.L., Snavely, N., Agarwala, A., Agrawala, M., Cohen, M.F., Curless, B., Kang, S.B.: Using photographs to enhance videos of a static scene. In: Rendering Techniques. (2007) 327–338
2. Shum, H., Kang, S.B.: Review of image-based rendering techniques. In: VCIP. (2000) 2–13
3. Debevec, P.E., Yu, Y., Borshukov, G.: Efficient view-dependent image-based rendering with projective texture-mapping. In: Rendering Techniques. (1998) 105–116
4. Torr, P.H.S., Fitzgibbon, A.W., Zisserman, A.: The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. IJCV **32** (1999) 27–44
5. Pollefeys, M., Verbiest, F., Gool, L.J.V.: Surviving dominant planes in uncalibrated structure and motion recovery. In: Proc. ECCV. (2002) 837–851
6. Patwardhan, K.A., Sapiro, G., Bertalmio, M.: Video inpainting of occluding and occluded objects. In: Proc. ICIP. (2005) 69–72
7. Patwardhan, K., Sapiro, G., Bertalmio, M.: Video inpainting under constrained camera motion. IEEE TIP **16** (2007) 545–553

8. Shih, T.K., Tang, N.C., Hwang, J.N.: Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity. IEEE Trans. Circuits Syst. Video Techn. **19** (2009) 347–360
9. Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. IEEE TPAMI **29** (2007) 463–476
10. Shen, Y., Lu, F., Cao, X., Foroosh, H.: Video completion for perspective camera under constrained motion. In: Proc. ICIP. Volume 3. (2006) 63–66
11. Hu, Y., Rajan, D.: Hybrid shift map for video retargeting. In: Proc. IEEE CVPR. (2010) 577–584
12. Granados, M., Tompkin, J., Kim, K.I., Grau, O., Kautz, J., Theobalt, C.: How not to be seen - object removal from videos of crowded scenes. Computer Graphics Forum (2012)
13. Venkatesh, M.V., Cheung, S.S., Zhao, J.: Efficient object-based video inpainting. Pattern Recognition Letters **30** (2009) 168–179
14. Shih, T.K., Tan, N.C., Tsai, J.C., H.-Y, Z.: Video falsifying by motion interpolation and inpainting. In: Proc. IEEE CVPR. (2008) 1–8
15. Jia, J., Tai, Y.W., Wu, T.P., Tang, C.K.: Video repairing under variable illumination using cyclic motions. IEEE TPAMI **28** (2006) 832–839
16. Ling, C.H., Lin, C.W., Su, C.W., Liao, H.Y.M., Chen, Y.S.: Video object inpainting using posture mapping. In: Proc. ICIP. (2009) 2785–2788
17. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. ACM Trans. Graph. **25** (2006) 835–846
18. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI. (1981) 674–679
19. Shi, J., Tomasi, C.: Good features to track. In: Proc. IEEE CVPR. (1994) 593 – 600
20. Bay, H., Ess, A., Tuytelaars, T., Gool, L.J.V.: Speeded-up robust features (surf). Computer Vision and Image Understanding **110** (2008) 346–359
21. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE TPAMI **26** (2004) 1124–1137
22. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE TPAMI **23** (2001) 1222–1239
23. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE TPAMI **26** (2004) 147–159
24. Kwatra, V., Schödl, A., Essa, I.A., Turk, G., Bobick, A.F.: Graphcut textures: image and video synthesis using graph cuts. ACM Trans. Graphics **22** (2003) 277–286
25. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: a randomized correspondence algorithm for structural image editing. ACM Trans. Graphics **28** (2009) 24:1–24:11
26. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Trans. Graphics **22** (2003) 313–318
27. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: Proc. IEEE CVPR, IEEE (2010) 2432–2439
28. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapcut: robust video object cutout using localized classifiers. ACM Trans. Graphics **28** (2009)
29. Yin, P., Criminisi, A., Winn, J.M., Essa, I.A.: Tree-based classifiers for bilayer video segmentation. In: Proc. IEEE CVPR. (2007)
30. Zelnik-Manor, L., Irani, M.: Multiview constraints on homographies. IEEE Trans. Pattern Anal. Mach. Intell. **24** (2002) 214–223