Spatio-Temporal Video Completion in Spherical Image Sequences

Binbin Xu, Sarthak Pathak, Hiromitsu Fujii, Atsushi Yamashita, and Hajime Asama

***Abstract*—Spherical cameras are widely used due to their full 360° fields of view. However, a common but severe problem is that anything carrying the camera is always included in the view, occluding visual information. In this letter, we propose a novel method to remove such occlusions in videos taken from a freely moving spherical camera. Our method can recover the occluded background accurately in distorted spherical videos by inpainting the color and motion information of pixels. The missing color and motion information inside the occluded region is iteratively recovered in a coarse-to-fine optimization. Spatial and temporal coherence of color and motion information is enforced, considering spherical image geometry. Initially, feature-point matching is used to remove the effect of camera rotation in order to deal with large pixel displacements. Following this, the iterative optimization process is bootstrapped using a reliable estimate of motion information obtained by interpolating it from surrounding regions. We demonstrate its effectiveness by successfully completing videos and recovering occluded regions recorded in various practical situations and by quantifying it against other state-of-the-art methods.**

***Index Terms*—Computer vision for other robotic applications, omnidirectional vision.**

I. Introduction

SPHERICAL cameras can capture a full 360-degree field of view and thus are popular for immersive photography/videography, virtual reality [1], etc. They are also very useful for robotics, as they can capture more visual information, aiding environment perception and camera motion estimation [2]. However, a common, but severe problem in using spherical cameras is that the entity carrying the camera, such as a human hand or a robot body, is always included in its field of view, causing undesired occlusions. For example, even in the simplest application of a human holding a spherical camera to record a picture or a video, the hand occludes a large part of the view, as shown in Fig. 1(a). This creates a visually unsatisfactory image, as shown in Fig. 1(b). Such a 'giant hand' occlusion is caused by the fact any object carrying the camera is often attached very close to its camera lenses. In addition to this visually unsatisfactory effect, the occlusion also implies a loss of visual information. This can affect using many existing vision algorithms, as also noted in [3]. Therefore, completing these undesired occlusions by recovering the true background information is often necessary and desired for applications using spherical cameras.

(a) (b)

Fig. 1. Example of occlusions occurred in using spherical cameras (a) Holding a spherical camera to take a picture. (b) 'Giant hand' occlusion in the captured spherical image.

The basic concept of such video completion is that the camera moves and previously occluded information appears in other frames. It can be found and copied back to its appropriate location. There are many ways to do this. One class of methods involves tracing the missing pixels via their motion information, i.e., optical flow.

Since motion and color information are both missing in occluded regions, You *et al.* [4] assumed a spatial continuity of the motion field and attempted to interpolate the missing motion information from surrounding regions. The interpolated motion can be traced to fill in the missing color information. We further extended this approach to handle spherical videos [5]. However, usually, optical flow is estimated from known color information. In this case, the reverse phenomenon occurs. Optical flow is first estimated by motion interpolation under a spatial smoothness assumption, and is used to estimate color information. However, it is actually a chicken-and-egg problem and simply estimating motion information, i.e., optical flow using interpolation and using it to update color information in the occluded region is a sub-optimal solution. Moreover, interpolation leads to an overly smoothed output. Hence, the true color information and the motion information remain incoherent with each other.

In this letter, we attempt to solve these problems via an iterative estimation and updating of motion and color, while con-

Manuscript received February 15, 2017; accepted June 13, 2017. Date of publication June 21, 2017; date of current version July 7, 2017. This letter was recommended for publication by Associate Editor H. Araujo Castellanos and Editor F. Chaumette upon evaluation of the reviewers' comments. This work was supported in part by the Council for Science, Technology, and Innovation, Cross-ministerial Strategic Innovation Promotion Program (SIP), Infrastructure Maintenance, Renovation, and Management (funding agency: NEDO). *(Corresponding author: Binbin Xu.)*

The authors are with the Department of Precision Engineering, Graduate School of Engineering, University of Tokyo, Tokyo 113-8656, Japan (e-mail: xubinbin@robot.t.u-tokyo.ac.jp; pathak@robot.t.u-tokyo.ac.jp; fujii@robot.t.u-tokyo.ac.jp; yamashita@robot.t.u-tokyo.ac.jp; asama@robot.t.u-tokyo.ac.jp).

This paper has supplemental material available at http://ieeexplore.ieee.org.

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

Digital Object Identifier 10.1109/LRA.2017.2718106

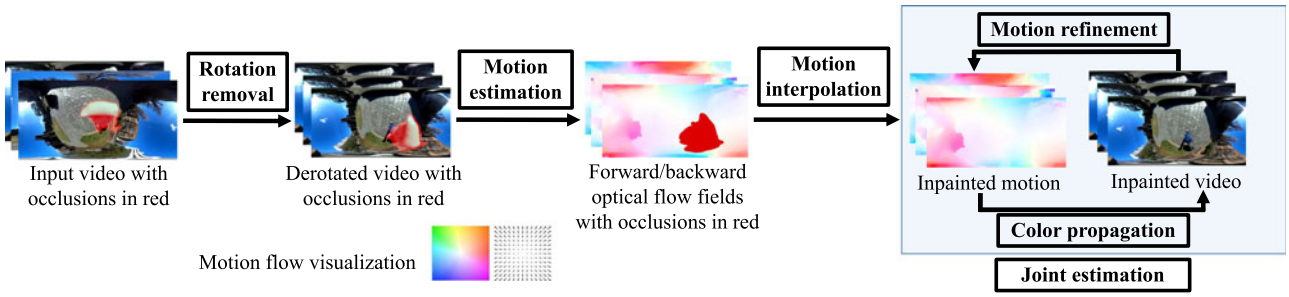


Fig. 2. The pipeline of the proposed method.

sidering the geometric properties of spherical images. We also introduce a way to deal with large displacements which cannot be estimated by common optical flow techniques. In this research, we assume a static background, where dynamic objects are away from occlusions. Given videos and masks, our procedure is summarized as follows. First, rotational motions inside spherical videos are estimated, and the video is 'derotated' in order to nullify large pixel displacements. Next, temporally forward and backward optical flow fields (i.e., flow fields w.r.t the previous and next frame, respectively) are estimated from these derotated videos and then initialized via interpolation. Following this, an iterative optimization method inpaints motion and color information alternatively in a coarse-to-fine way. Motion is inpainted and refined to enforce spatial and temporal coherence in occluded regions. Color information is inpainted by iteratively propagating corresponding pixels from other frames based on inpainted motion, in a temporally coherent manner. The pipeline of this method is illustrated in Fig. 2.

The main contributions in this letter are divided into three parts.

1) Rotation estimation is combined with spherical video completion work for the first time. It can improve the accuracy of motion estimation without losing any visual information for color propagation.
2) A reliable initialization for the color and motion iteration based on motion interpolation.
3) Temporal coherence of pixel trajectories on a spherical manifold for motion estimation.

The remainder of this letter is organized as follows. Section II discusses the related work in greater detail, highlighting our originality. Section III explains the derotation approach for dealing with large displacements. Section IV describes the iterative optimization of color and motion estimation in occluded regions, and the spatio-temporal coherence constraints. Section V shows the experiments and the comparisons with prior methods, followed by the conclusion in the last section.

II. Related Work

Several methods have been proposed for removing occlusions in spherical videos before. There is a class of methods that use information related to environment geometry. Kawai *et al.* search for similar exemplars from other frames on projected planes [6] and by aligning multiple frames based on a pre-reconstructed 3D model [7]. However, these cannot be used in general situations.

In addition to spherical videos, video completion has been more widely studied in perspective videos from a freely moving camera. They can be classified into two main categories. In the first category are methods search for the most similar exemplars, which are spatio-temporal cubes, in all video frames [8]. While these methods can produce plausible outputs, the occluded background cannot always be found as the most similar information due to the large search space. Another limitation is that the rigid cubes make it difficult to capture appearance changes caused by the distortion of spherical images. As a result, these methods cannot ensure true backgrounds recovered in spherical videos.

In the second category are approaches estimate pixel-wise motion fields in occluded regions, and use the estimated motion to find the corresponding pixels of occlusions in other frames. You *et al.* interpolate motion fields in occluded regions by interpolating the motion from surrounding regions [4]. However, their method relies on a linear approximation of motion flow in adjacent frames and thus is not suitable for the distorted motion pattern in spherical videos. As explained earlier, this method and our previously proposed improvement for spherical videos [5] directly use the interpolated motion for color propagation and thus suffer from incoherency of color and motion information.

Several recent approaches handle this incoherency by an iterative refinement procedure of color and motion [9], [10]. However, the former [9] uses a constant velocity assumption for motion trajectory estimation, which is not suitable for distorted spherical videos. Instead, we propose to project pixel movements to spherical manifolds in order to enforce temporal coherency of motion information. Besides, their method initializes the motion optimization as zero motion, which can lead to a local minimum. We propose a more robust initialization based on motion interpolation. The latter [10] initializes color information based on similarity searching and refines motion trajectory using initialized color information. Such initialization encourages the similarity in local neighborhoods on each frame separately, similar to the searching based video inpainting method [8]. However, it cannot ensure true background recovered in complex texture background since this nearest neighborhood searching usually leads to similarly textured contents. Instead, we initialize both color and motion based on motion interpolation and connect occluded pixels with their correspondences on other frames us-

ing motion trajectories, enforcing temporal coherence of color information.

In addition, all approaches that use optical flow suffer under large camera rotation, which induces large pixel displacements. We propose a feature point-based derotation approach to solve this problem.

III. Rotation Removal

Our method relies on finding the motion of each pixel by means of forward and backward dense optical flow fields. However, optical flow estimation is known for suffering from the large displacements problem [11] because most optical flow approaches constrain their search space to small neighbourhoods. Therefore, before the motion estimation step, we use an approach based on rotation estimation in order to deal with large movements in spherical videos.

Any spherical camera motion is a combination of pure rotation and translation, so it can also be decomposed into these two kinds of motion [12]. Instead of computationally intensive searching for both translations and rotations [12], rotations are the main interest in this work for two reasons. First, since pixel displacements in spherical images are mainly dominated by rotations [12], removing rotations would make optical flow estimation more accurate. Second, and more importantly, spherical images can be rotated to any orientation without any information loss. In this part, we only need to approximately estimate rotation in order to minimize pixel displacements. Hence, and in the interest of saving computational time, we use feature point matching instead of optical flow to estimate and stabilize rotation.

The proposed method first estimates rotations between consecutive pairs of frames based on feature point matching and then 'derotates' each frame to remove displacements caused by rotations. For each consecutive pair of frames, feature point correspondences are first calculated using A-KAZE features [13], which are well known for its robustness to distortions. The extracted point correspondences are then projected back to the spherical manifold and filtered to satisfy the epipolar geometry constraint:

$$\mathbf{X}'^{T}\mathbf{E}\mathbf{X} = 0, \tag{1}$$

where $\mathbf{E}$ is the essential matrix, composed of the rotation matrix and translation vector, and $\mathbf{X}$ and $\mathbf{X}'$ are the corresponding points projected on the spherical manifold. The essential matrix $\mathbf{E}$ is estimated by the eight-point algorithm [14] and outliers in correspondences are filtered using RANSAC [15]. Following this, Singular Value Decomposition (SVD) is performed to decompose $\mathbf{E}$ into the rotation matrix and the translation vector. Based on the estimated rotation matrix, the pair of frames can be derotated to the same approximate orientation. We derotate both the temporally forward and temporally backward frames to the same orientation as the target frame. Using sparse point correspondences, even very large displacements can be removed. An example of this effect can be seen in Fig. 3. In Fig. 3(a) and (b), two frames have a high degree of rotation with each other. After derotating Fig. 3(b), Fig. 3(c) is now at a similar orientation and have smaller pixel displacements to Fig. 3(a). Although this is a drastic example and consecutive frames will not be rotated to this extent, this derotation has a sizable contribution in reducing the estimation error, as will be shown later in the experimental section.

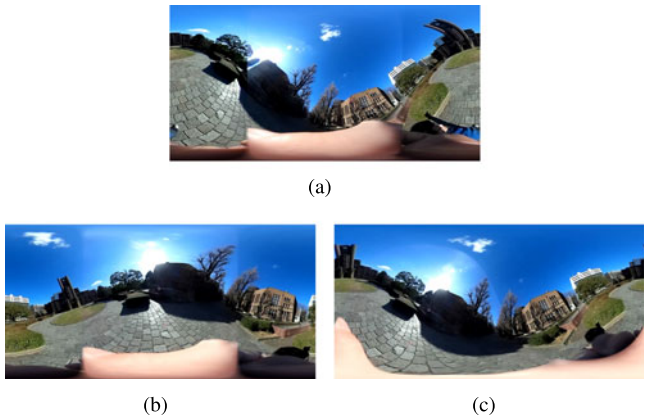


Fig. 3. Rotation removal to handle large displacements. (a) and (b) shows two frames at their original orientations, involving both translations and rotations. After removing estimated rotations, (a) and (c) show a more similar orientation and smaller displacements. (a) Image 1 at original orientation. (b) Image 2 at original orientation. (c) Image 2 after rotation removal.

IV. Joint Estimation of Color and Motion

After removing the estimated rotations between pairs of frames, the next step is to estimate the forward/backward motion fields inside the occluded regions so that the occluded pixels can be found by tracing the motion fields. We jointly estimate motion and color information in an iterative manner to enforce their coherence. In addition, we try to enforce their spatial and temporal coherence.

A. Notations

Let $\mathbf{I}_i$ be the target frame number *i*, and $\mathbf{U}_f$, $\mathbf{U}_b$ be the forward and backward optical flow fields, respectively. The forward flow $\mathbf{U}_f(x, y)$ denotes the flow vector (u_f, v_f) from a pixel located at (x, y) on Image $\mathbf{I}_i$ to a pixel located at $(x + u_f, y + v_f)$ on Image $\mathbf{I}_{i+1}$. Defining from the same reference frame, the backward flow $\mathbf{U}_b(x, y)$ denotes the flow vector (u_b, v_b) from a pixel (x, y) on Image $\mathbf{I}_i$ to a pixel $(x + u_b, y + v_b)$ on Image $\mathbf{I}_{i-1}$.

B. Objective Function

The joint estimation of motion and color is conducted by minimizing the following objective function:

$$\arg\min_{\mathbf{I},\mathbf{U_b},\mathbf{U_f}} (E_{\text{color:temporal}} + E_{\text{flow:spatial}} + E_{\text{flow:temporal}}), \tag{2}$$

where $E_{\text{color:temporal}}$ encourages temporal color coherence between consecutive frames, $E_{\text{flow:spatial}}$ encourages piecewise smoothness for the forward and backward flow fields, and $E_{\text{flow:temporal}}$ encourages temporal flow coherence between forward and backward flow fields. Different from the previous work [10], spatial color cost is not considered since it encour-

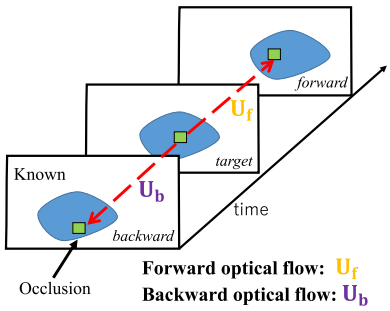


Fig. 4. Temporal color cost encourages temporal coherence between each three consecutive frames along forward and backward optical flow fields.

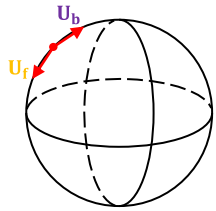


Fig. 5. Temporal flow cost encourages temporal coherence between forward and backward flow fields for the same reference frame. Notice that flow vectors are projected back to spherical manifolds to avoid distortions on 2D projection planes.

ages the similarity in local neighborhoods on each target frame. This may fill in plausible color information rather than true background and unnecessarily smooth the output. Instead, we use motion interpolation to initialize the solution and a temporal flow cost to encourage more coherent motion trajectories on spherical manifolds.

1) Temporal color cost: This cost encourages temporal color coherence between each three consecutive frames along forward and backward optical flow fields as visualized in Fig. 4. For pixel t on the image $\mathbf{I}_i$, which is denoted as $\mathbf{I}_i(t)$, we use the brightness constancy constraint [11] to model this cost:

$$E_{\text{color:temporal}} = \lambda_c \sum_{t\in i} \phi(|\mathbf{I}_i(t) - \mathbf{I}_{i+1}(t+\mathbf{U}_f)|^2) + \lambda_c \sum_{t\in i} \phi(|\mathbf{I}_i(t) - \mathbf{I}_{i-1}(t+\mathbf{U}_b)|^2), \quad (3)$$

where $\phi(x^2) = \sqrt{x^2+\varepsilon}$ is the Charbonnier penalty function (a convex and differentiable L1 norm) [16], $\varepsilon = 10^{-6}$ is a small constant, and λ_c is the weight coefficient for this cost.

2) Spatial flow cost: This cost is a common assumption on flow fields, namely, the estimated and inpainted flow fields should be piecewise smooth. Here the total variation (TV) model [17] is adopted to capture this prior information:

$$\phi_{TV}(\nabla \mathbf{U}) = \sqrt{|\nabla \mathbf{u}|^2 + |\nabla \mathbf{v}|^2 + \varepsilon}. \quad (4)$$

where ∇ denotes the gradient operator. Using the TV-L1 norm, spatial flow cost can be defined as:

$$E_{\text{flow:spatial}} = \lambda_s \sum_{t\in i} \left(\phi_{TV}(\nabla \mathbf{U}_b(t)) + \phi_{TV}(\nabla \mathbf{U}_f(t))\right), \quad (5)$$

where λ_s is the weight coefficient for this cost. It is used to penalize large magnitudes in the gradients of flow fields.

3) Temporal flow cost: This cost encourages temporal coherence between forward and backward optical flow fields for the same reference frame. It penalizes large changes of motion directions between forward and backward flows. Though a similar penalty term has been exploited before [9], the constant velocity assumption does not hold any more in spherical videos. On 2D image planes, motion trajectories are distorted and projected as curves, not straight lines. To address the distorted motions, the motion direction on spherical manifolds is preferred when considering flow trajectory coherence. Each pixel's movement on the spherical manifold is denoted as $(\mathbf{U}_X, \mathbf{U}_Y, \mathbf{U}_Z)$, where each component is the spherical motion in the X, Y, and Z directions, respectively. This movement can be obtained from its 2D equirectangular motion field $(\mathbf{u}, \mathbf{v})$ by projecting it using the Jacobian of the transformation as described in [18]. The direction of forward flow vectors $(\hat{\mathbf{U}}_{fX}, \hat{\mathbf{U}}_{fY}, \hat{\mathbf{U}}_{fZ})$ and the direction of backward flow vectors $(\hat{\mathbf{U}}_{bX}, \hat{\mathbf{U}}_{bY}, \hat{\mathbf{U}}_{bZ})$ on spherical manifolds can then be calculated by normalizing the magnitudes.

Once motion directions are known, the temporal flow cost is used to encourage forward and backward flow to be smooth along motion trajectories on spherical planes:

$$E_{\text{flow:temporal}} = \lambda_t \sum_{t\in i} \phi(|\hat{\mathbf{U}}_{fX} + \hat{\mathbf{U}}_{bX}|^2 + |\hat{\mathbf{U}}_{fY} + \hat{\mathbf{U}}_{bY}|^2 + |\hat{\mathbf{U}}_{fZ} + \hat{\mathbf{U}}_{bZ}|^2), \quad (6)$$

where λ_t is the weight coefficient for this cost and the sign "+" is used to compensate for the opposite direction of forward and backward flows regarding the same reference frame. Fig. 5 shows a visualization of this cost.

The weight coefficients λ_c, λ_s, and λ_t of the three cost functions are tuned manually. In all the experimental tests in this letter, they are set as $\lambda_c = 0.5$, $\lambda_s = 0.2$, and $\lambda_t = 0.1$. All pixels on image frames will be considered during optimization and thus the pixel notation t will be simplified unless otherwise specified.

C. Optimization

We use an iterative optimization method to solve (2) by alternating between motion refinement and color propagation. We first fix color I and refine forward/backward optical flow $\mathbf{U}_f$, $\mathbf{U}_b$ in motion refinement and then propagate color information I while fixing $\mathbf{U}_f$, $\mathbf{U}_b$.

It can be noticed that $\mathbf{U}_b$ and $\mathbf{U}_f$ are independent in (3) and (5) yet combined in (6). To decouple the two flow fields, the split Bregman method [19] is used in (6). The new temporal cost function then becomes:

$$E_{\text{flow:temporal}} = \lambda_t \sum_{t\in i} \Big(\phi\big((\hat{\mathbf{U}}_{fX} + \hat{\mathbf{U}}_{bX})^2 + (\hat{\mathbf{U}}_{fY} + \hat{\mathbf{U}}_{bY})^2 + (\hat{\mathbf{U}}_{fZ} + \hat{\mathbf{U}}_{bZ})^2\big) - b^{(k)}\Big)^2, \quad (7)$$

where $b^{(k)}$ is an auxiliary iterative variable and k is the iteration number. Then motion refinement of forward/backward flow fields can be iteratively updated by:

1) Fixing $\mathbf{U}_f^{(k)}$ and $b^{(k)}$, then updating $\mathbf{U}_b^{(k+1)}$;
2) Fixing $\mathbf{U}_b^{(k)}$ and $b^{(k)}$, then updating $\mathbf{U}_f^{(k+1)}$;
3) Updating $b^{(k+1)}$.

In the first and second step, $\mathbf{U}_b$ and $\mathbf{U}_f$ are updated by minimizing (2) via solving corresponding Euler-Lagrange equations. Specifically, since $\mathbf{U}_f$ is fixed in the first step, the independent term regarding $\mathbf{U}_f$ in the objective function can be ignored. Thus, the objective function in this step is:

$$\arg\min_{\mathbf{U_b}} \sum_{t\in i} \lambda_c \phi(|\mathbf{I}_i(t) - \mathbf{I}_{i-1}(t+\mathbf{U}_b)|^2) + \sum_{t\in i} \lambda_s \phi_{TV}(\nabla \mathbf{U}_b(t)) + E_{\text{flow:temporal}}. \quad (8)$$

We linearize the first term using a first order Taylor expansion and discretely approximate the second term for computational convenience. The implementation details can be found in [16], [17]. Then, the corresponding Euler-Lagrange equations for the approximated objective function can be solved by the iterative optimization method based on warping [20]. $\mathbf{U}_f$ can also be solved in the same way. After $\mathbf{U}_b$ and $\mathbf{U}_f$ are updated, then $b^{(k+1)}$ is updated by

$$b^{(k+1)} = b^{(k)} - \phi\big((\hat{\mathbf{U}}_{fX} + \hat{\mathbf{U}}_{bX})^2 + (\hat{\mathbf{U}}_{fY} + \hat{\mathbf{U}}_{bY})^2 + (\hat{\mathbf{U}}_{fZ} + \hat{\mathbf{U}}_{bZ})^2\big). \quad (9)$$

The above three steps are iteratively performed. The flow fields estimated in the last iteration are used to warp images to solve the optimal increment in the next iteration for motion refinement. We find 5 to 10 iterations are usually enough for convergence.

After motion refinement, color information on other frames is propagated to the occluded regions on target frame by tracking along forward/backward flow fields. We use the same color propagation strategy in [5], which finds corresponding pixels on other frames by tracing along the flow fields till an un-occluded region is reached. In this process, the continuity property of spherical images' field of view is also considered. When pixels are warped outside a lateral border of the equirectangular image, its corresponding location is found by continuing from opposite border.

V. Initialization and Multi-Scale Solution

One of the key stages in this work is the initialization of motion refinement since the objective function (2) is highly nonlinear. To avoid getting stuck at a local minimum, the iterative optimization is performed in a multi-scale, coarse-to-fine manner.

The frames are repeatedly downsampled (halved) from their full-resolution to set up an image pyramid. Initialization is conducted at the coarsest level i.e., the smallest image size. We first estimate the forward and backward optical flow fields on the target frame using FlowFields algorithm [21] (other state-of-the-art optical flow methods can also be used). Since the motion in the occluded regions is almost zero, directly using this estimate for initialization is not a good approach since it can lead to a local minimum. Roxas *et al.* used this principle and initialized backward/forward motion as zero [9]. Later in the experimental section, we show the effect of using this initialization process on the accuracy of the output. Instead, we use motion interpolation on the coarsest level for initialization. Motion in occluded regions is initialized by referring from surrounding motions on the target frame based on spherical polynomial interpolation, as used in our previous work [5]. Although the polynomial interpolation leads to an overly smooth motion estimate, it serves as a good estimate for the color and motion joint estimation process. The auxiliary variable $b^{(k)}$ is initialized as zero on the coarsest level.

After convergence on the coarsest level, the refined forward/backward optical flow fields are up-sampled to one finer level. This process is repeated until the convergence on the finest level, i.e., the largest image size of the image pyramid. To fasten optimizations, images on all levels of image pyramid are converted to gray images, except on the finest level. When the solution converges on the finest pyramid level, occluded regions can be filled in by image warping through the estimated motion trajectories to generate colored results. The proposed method is summarized in Algorithm 1.

Algorithm 1: The proposed method

Input: Image sequence $\mathbf{I}$, mask sequences $\mathbf{M}$
Output: Completed image sequences $\mathbf{I}$

```
Rotation estimation and removal in input videos;
Construct image and mask pyramids;
Initialization: interpolate backward/forward optical flow
  fields U_f and U_b on the coarsest level, b^(k) ← 0;
for Level l = 1 to L do
    for Iteration k = 1 to K do
        Update U_b^(k+1) while fixing U_f^(k) and b^(k);
        Update U_f^(k+1) while fixing U_b^(k) and b^(k);
        Update b^(k+1);
    Color propagation;
    Upsample U_b and U_f;
```

VI. Experiments

To test the robustness and effectiveness of the proposed method, we conducted experiments in various scenarios, under different background scenes and lightning conditions. Spherical videos used in the experiments were recorded using a RICOH THETA spherical camera, which directly provides equirectangular videos. The camera was held by hand (Fig. 1(a)), mounted on a moving AR Parrot Drone 2.0 (Fig. 7(a)), and carried by a selfie stick (Fig. 8(a)). The masks for specifying the rigid occlusions were manually extracted. In the case of non-rigid dynamic occlusions, such as hand, masks were generated using the video segmentation method [22] based on motion difference and manually modified in case of inaccurate segmentation. The

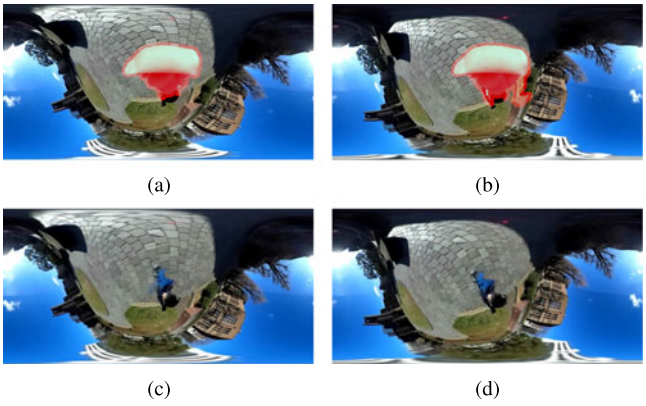


Fig. 6. 'Giant hand' occlusion removal results (two representative frames). The experimental setup can be seen in Fig. 1(a). We show two representative input images in image (a) and (b), with masks masked in red, and their corresponding recovered background in image (c) and (d).

mask regions are in red for visualization. The input data was at a resolution of 960 × 480 pixels. A MATLAB-based implementation without any parallel processing took around 3 minutes for each frame on a laptop with 3.0 GHz Intel i7 CPU with 8 GB memory.

We also conducted quantitative experiments to measure the accuracy of our proposed method. The motivation of this work was to recover real background behind occlusions. Therefore, it is important to check whether our approach could recover an accurate background, instead of just filling in some plausible information. In quantitative evaluations, we used the Root Mean Squared Error (RMSE) metric to calculate the RGB error between the ground truth and the completed video frames.

A. *Occlusion Removal Results*

We tested our proposed method in various scenarios. Here we show two representative input images and corresponding occlusion removed outputs for each occlusion situation. More results can be seen in the video attachment.

Fig. 6 shows a very common situation when a spherical camera is held by hand (as shown in Fig. 1(a)) to record a video in an outdoor environment. Input images with occlusion masks are shown in red in Fig. 6(a) and (b) and the occlusion removal results are shown in Fig. 6(c) and (d). It can be seen that the proposed method can recover the occluded stone brick floor behind the hand occlusion, even if some parts of background is falsely chosen as the occlusion (Fig. 6).

Fig. 7 shows a robotic application where a spherical camera is mounted on a drone (Fig. 7) to perceive the surrounding indoor environment. The robot body severely occludes around a fourth of the visual information in the captured video. From the occlusion removal results, it can be seen that our proposed method can also successfully recover the occluded background behind the robot body occlusions.

Fig. 8 shows another common situation where a man is recording a video using a selfie stick, as shown in Fig. 8(a). Both the

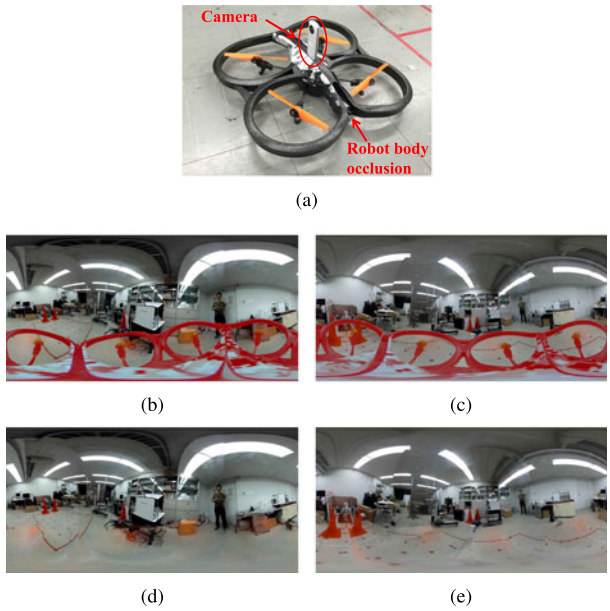


Fig. 7. 'Robot body' occlusion removal results (two representative frames). Images were taken through a spherical camera mounted on a flying drone as shown in (a). We show two representative input images in image (b) and (c), with occlusion masks in red, and their corresponding recovered background in image (d) and (e). (a) Experimemtal setup.

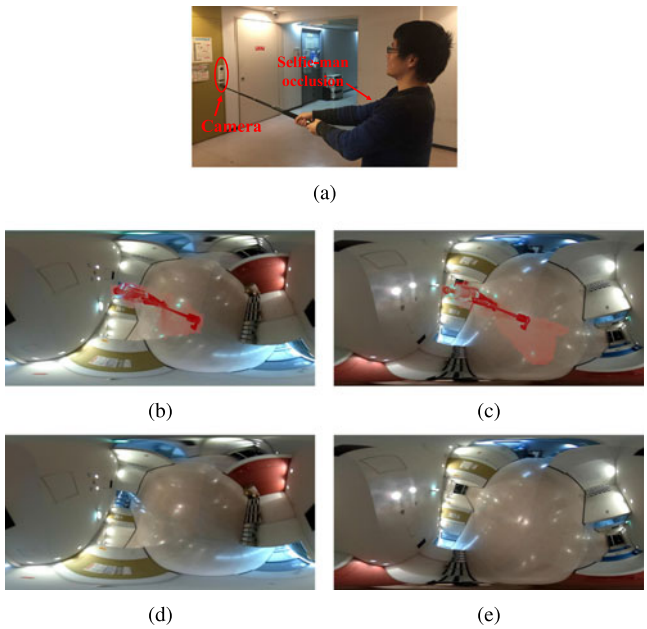
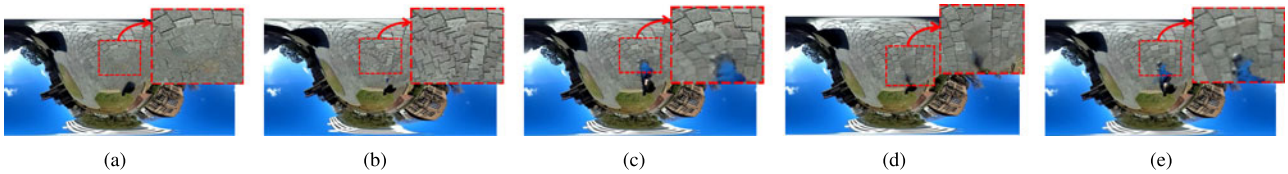


Fig. 8. 'Selfie-man' experimental results (two representative frames). Images were taken through a spherical camera held by a selfie stick as shown in (a). We show two representative input images in image (b) and (c), with masks masked in red. The corresponding backgrounds, such as corridor and window, have been recovered in image (d) and (e), respectively. (a) Experimemtal setup

man and stick are included in all captured images. A common requirement in such selfie applications is that the user may want to have a clean image, without the selfie stick or/and the photographer. The proposed method can realize this requirement as shown in the outputs. We masked the selfie stick and the photographer, with his shadows. The occluded backgrounds,



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

Fig. 9. Comparison with other state-of-the-art video completion methods on 'giant hand' occlusions. (a) Exemplar-based searching [8]. (b) Linear motion interpolation [4]. (c) Spherical motion interpolation [5]. (d) Searching-based motion coherence [10]. (e) Proposed.

such as the corridor (Fig. 8) and window (Fig. 8), all have been recovered.

B. Comparisons

Under the same conditions, we compared our method with other state-of-the-art video completion methods: Newson *et al.*'s exemplar-based search method [8], You *et al.*'s linear motion interpolation-based method [4], Huang *et al.*'s searching-based motion coherence method [10], and our previous spherical motion interpolation-based method [5]. We used the code from the authors of the work [8], [10] with the default parameters recommended by the authors.

Fig. 9 shows a representative frame to compare these methods on the 'giant hand' occlusion. The exemplar-based searching method [8] replaces occlusions with the most similar patches in image sequences. It can be seen more clearly in the red box from Fig. 9(a) that this method fills in plausible contents, which are similar texture patterns, but fails to recover the true background. Therefore, without the motion to guide the completion process, it is difficult to recover the true background. The linear motion interpolation-based method [4] estimates motion trajectories using linear motion approximation, which does not hold in spherical videos and causes noticeable zigzag artifacts in the recovered results, as shown in Fig. 9(b). Our previous spherical motion interpolation-based [5] avoids this approximation and corrects motion trajectories to conform to spherical videos. However, as explained, motion and color are yet incoherent and recovered background does not aligned well in Fig. 9(c). Searching-based motion coherence method [10] encourages the coherence of motion and color and thus solves the plausible problem in [8]. However, as Fig. 9(d) shows, their method still generates some artifact contents, similar to [8], because of the searching-based initialization. By comparison, the result from the proposed method in Fig. 9(e) is free of these errors and aligns properly with the rest of the image. More details can be seen in the video attachment.

We also quantitatively compared with these methods. To obtain ground truth videos, we used a thin string to hang a spherical camera from the ceiling to capture image sequences as also done in [5]. Both rotations and translations are involved in the videos. The captured videos, with almost zero occlusions, are used as the ground truth. Then, we put a uniformly distributed mask. Visual information inside the mask regions was manually deleted and five different methods were implemented on the impaired image sequences to recover the deleted information. An example ground truth input image with mask regions in red is shown in Fig. 10(a). The graph in Fig. 11 shows the RMSE comparison of five methods on each frame. The results show that the proposed method outperforms others on almost all frames.

(a) (b)

Fig. 10. Ground truth videos with mask overlays in red. Color information in masks was manually deleted before video completion process. (a) Rotations and translations. (b) Dominant translations.

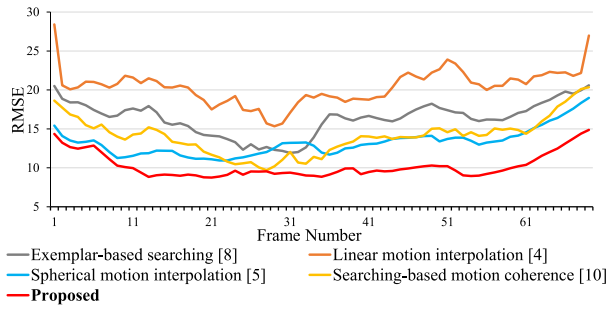


Fig. 11. Comparison of RMSE on all frames of rotation-dominated videos. The proposed method shows the best accuracy in all image frames, as compared to other four methods.

C. Quantitative Analysis of Each Contribution

On the same ground truth videos, we decided to evaluate the effect of each contribution proposed by us, as was mentioned in Section I: (i) rotation removal, (ii) motion interpolation for initialization and (iii) spherical temporal coherence. We evaluated RMSE values against the groundtruth by disabling or replacing each of them. The results are summarized in Fig. 12. As can be noticed, the RMSE is the lowest when all are enabled. The effect of each contribution in ensuring the accuracy of the result can be seen.

D. Discussion on the Effects of the Type of Camera Motion

During the experiments, we noticed that, in general, color information used to fill in occlusions from other frames is obtained via camera rotation. However, in cases where translations are dominant, cameras need to move further to find the information behind occlusions due to motion parallax. As a result,

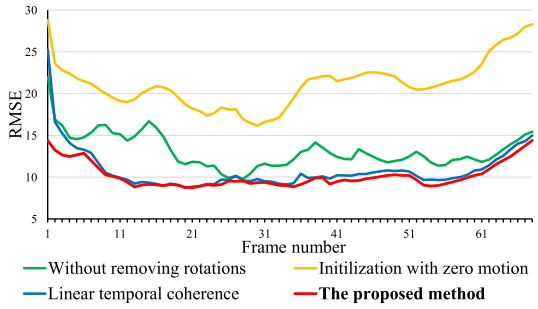


Fig. 12. The RMSE evaluations of each contribution in the proposed video completion method.

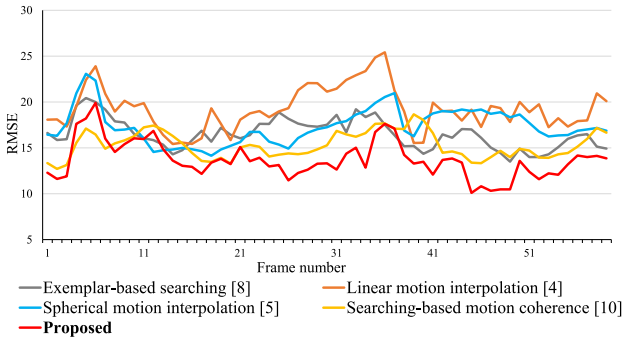


Fig. 13. Comparison of RMSE on all frames of a translation-dominated video. The proposed method shows the best accuracy in almost all image frames, as compared to other four methods.

the error in the iterative warping between temporally far frames can accumulate.

To evaluate the robustness under dominant translations, we evaluated the proposed method and compared it with other methods on the Park 1 sequence collected by [23], which involves almost only translations. Fig 10(b) shows one frame of the testing sequences. The RMSE comparisons of five methods in Fig. 13 show that the proposed method can give still best occlusions removal results in most frames, though it fails when the translations are too large on some frames.

VII. Conclusion

In this letter, we proposed a novel method to remove occlusions in spherical videos by jointly optimizing of color and motion for spatial and temporal coherence in spherical videos. We also proposed a reliable initialization for motion refinement and a method to remove large displacements in spherical videos. Experimental results in various scenarios demonstrate the effectiveness of our proposed approach and the improvement compared with prior research.

In this work, masks are manually given or modified based on segmentations, which is exhausting for users. Incomplete masks will cause errors to the initial interpolation for motion estimation. Future work will concentrate on how to automatically or interactively segment foreground occlusions in spherical videos. We will also considerer directly computing optical flow fields on spherical manifolds in future work.

Acknowledgment

The authors would like to thank A. Newson and J.-B. Huang for providing their codes for comparison of experimental results.

References

[1] S. Kasahara, S. Nagai, and J. Rekimoto, "Livesphere: Immersive experience sharing with 360 degrees head-mounted cameras," in *Proc. Adjunct Publ. 27th Annu. ACM Symp. User Interface Softw. Technol.*, 2014, pp. 61–62.

[2] Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza, "Benefit of large field-of-view cameras for visual odometry," in *Proc. 2016 IEEE Int. Conf. Robot. Autom.*, 2016, pp. 801–808.

[3] S. Kavanagh, A. Luxton-Reilly, B. Wüensche, and B. Plimmer, "Creating 360 educational video: A case study," in *Proc. 28th Aust. Conf. Comput.-Human Int.*, 2016, pp. 34–39.

[4] S. You, R. T. Tan, R. Kawakami, and K. Ikeuchi, "Robust and fast motion estimation for video completion," in *Proc. IAPR Int. Conf. Mach. Vis. Appl.*, 2013, pp. 181–184.

[5] B. Xu, S. Pathak, H. Fujii, A. Yamashita, and H. Asama, "Optical flow-based video completion in spherical image sequences," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2016, pp. 388–395.

[6] N. Kawai, K. Machikita, T. Sato, and N. Yokoya, "Video completion for generating omnidirectional video without invisible areas," *IPSJ Trans. Comput. Vis. Appl.*, vol. 2, pp. 200–213, 2010.

[7] N. Kawai, N. Inoue, T. Sato, F. Okura, Y. Nakashima, and N. Yokoya, "Background estimation for a single omnidirectional image sequence captured with a moving camera," *Inf. Media Technol.*, vol. 9, no. 3, pp. 361–365, 2014.

[8] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, "Video inpainting of complex scenes," *SIAM J. Imag. Sci.*, vol. 7, no. 4, pp. 1993–2019, 2014.

[9] M. Roxas, T. Shiratori, and K. Ikeuchi, "Video completion via spatio-temporally consistent motion inpainting," *Inf. Media Technol.*, vol. 9, no. 4, pp. 500–504, 2014.

[10] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Temporally coherent completion of dynamic video," in *Proc. ACM Trans. Graph.*, vol. 35, no. 6, 2016, Art. no. 196.

[11] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 43–77, 1994.

[12] R. C. Nelson and J. Aloimonos, "Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head)," *Biol. Cybern.*, vol. 58, no. 4, pp. 261–273, 1988.

[13] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *Proc. British Mach. Vision Conf.*, vol. 34, no. 7, pp. 1281–1298, 2013.

[14] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 580–593, Jun. 1997.

[15] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[16] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *Int. J. Comput. Vis.*, vol. 61, no. 3, pp. 211–231, 2005.

[17] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Proc. Joint Pattern Recognit. Symp.*, 2007, pp. 214–223.

[18] J. Gluckman and S. K. Nayar, "Ego-motion and omnidirectional cameras," in *Proc. Int. Conf. Comput. Vis.*, 1998, pp. 999–1005.

[19] T. Goldstein and S. Osher, "The split bregman method for l1-regularized problems," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

[20] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 25–36.

[21] C. Bailer, B. Taetz, and D. Stricker, "Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4015–4023.

[22] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1777–1784.

[23] W.-S. Lai, Y. Huang, N. Joshi, C. Buehler, M.-H. Yang, and S. B. Kang, "Semantic-driven generation of hyperlapse from 360 video," Ricoh Dataset, Park 1 sequence, 2017. [Online]. Available: http://vllab1.ucmerced.edu/wlai24/360hyperlapse/ricoh/