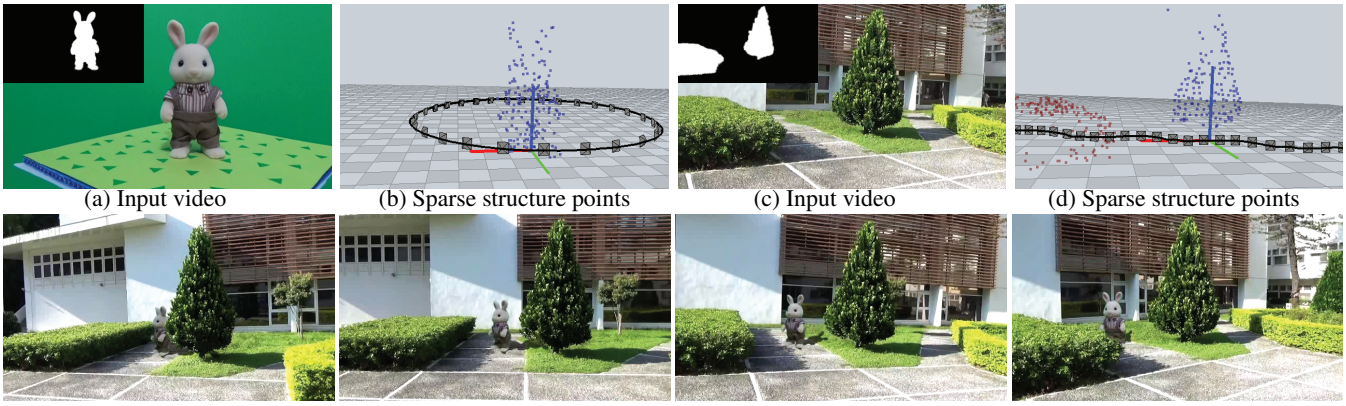
EUROGRAPHICS 2016 / J. Jorge and M. Lin
(Guest Editors)

Volume 35 (2016), Number 2

Interactive Videos: Plausible Video Editing using Sparse Structure Points

Chia-Sheng Chang[1,2] Hung-Kuo Chu[1] Niloy J. Mitra[2]

[1]National Tsing Hua University, Taiwan [2]University College London



(a) Input video (b) Sparse structure points (c) Input video (d) Sparse structure points

Figure 1: *Starting from raw video sequences (a,c), we extract corresponding sets of sparse structure points (b,d), which are then used to enable direct object level video edits (bottom row). (Please refer to the supplementary for the video sequences.)*

Abstract

Video remains the method of choice for capturing temporal events. However, without access to the underlying 3D scene models, it remains difficult to make object level edits in a single video or across multiple videos. While it may be possible to explicitly reconstruct the 3D geometries to facilitate these edits, such a workflow is cumbersome, expensive, and tedious. In this work, we present a much simpler workflow to create plausible editing and mixing of raw video footage using only sparse structure points (SSP) directly recovered from the raw sequences. First, we utilize user-scribbles to structure the point representations obtained using structure-from-motion on the input videos. The resultant structure points, even when noisy and sparse, are then used to enable various video edits in 3D, including view perturbation, keyframe animation, object duplication and transfer across videos, etc. Specifically, we describe how to synthesize object images from new views adopting a novel image-based rendering technique using the SSPs as proxy for the missing 3D scene information. We propose a structure-preserving image warping on multiple input frames adaptively selected from object video, followed by a spatio-temporally coherent image stitching to compose the final object image. Simple planar shadows and depth maps are synthesized for objects to generate plausible video sequence mimicking real-world interactions. We demonstrate our system on a variety of input videos to produce complex edits, which are otherwise difficult to achieve.

1. Introduction

RGB video is the most ubiquitous mode for capturing spatial and temporal events. With the rapid proliferation of videocams, digital cameras, and smart phones, capturing video is now easier than ever before. While decades of research have investigated denoising, deblurring, color/contrast enhancement, segmentation, etc. for such raw videos, performing object-level editing in a post processing video editing phase remains difficult.

The key difficulty in supporting such object-space edits is the lack of an underlying 3D model of the scene. This shortcoming makes the following tasks particularly challenging: ensuring correct perspective under view changes, handling occlusion effects under relative object movements, and updating shadows due to object/camera changes. Specialized equipments, e.g., recording events with carefully synced multiple cameras can address many of these challenges. However, the corresponding setup costs are high and often requires extensive rigging of the environments. Alternately, one

can create detailed 3D models of the underlying scenes. While the resultant editing framework can be very powerful and compelling, especially for special effects involving synthetically introduced geometry, such a workflow is again expensive and only justifiable in high budget scenarios (e.g., for a movie).

In this paper, we demonstrate that even coarse level 3D scene information in the form of sparse structure points (SSP), directly extracted from raw video footage, can be exploited to enable plausible yet non-trivial video edits. Our approach does not require any specialized acquisition setup and works under only a few assumptions. Specifically, we assume that the video objects are opaque and rest on planar grounds so that they can be easily moved around or transplanted onto other planes (in target videos) using SSPs. We do not, however, assume the objects to be box-like or expect access to accurate 3D models (e.g., retrieved from a database).

Starting from a raw video footage, we first use structure-from-motion to extract a sparse set of points along with corresponding camera information. User-scribbles are then utilized to organize the raw point sets by grouping them into foreground and background components. Finally, we propose how to make use of the structured point sets to support direct object-level video edits for single or across multiple videos. Technically, we enable this via a novel image-based rendering approach driven by the sparse structure points to warp and stitch patches from adaptively selected video frames. A key observation is that even with sparse 3D information, we can reliably recover meaningful information about camera perspectives and plausibly handle inter-object occlusions. Figure 1 shows a typical example achieved using our system (please refer to the supplementary video). We evaluated our system on a range of input videos and present comparisons with baseline methods as well as 3D reconstruction methods.

In summary, our main contributions are: (i) a video editing framework that directly supports object-level edits (e.g., shuffling, duplicating, manipulating objects); (ii) proposing a hybrid representation for video object using SSP extracted from raw video that enables direct object manipulation in 3D; and (iii) a novel image-based rendering algorithm using SSP to plausibly synthesize video objects in novel views via a structure-preserving image warping on the adaptively selected video frames and a spatio-temporally coherent image stitching.

2. Related Work

Video editing. The wide availability of portable and affordable video recorders enables the unrestricted access to the video contents in daily life. This has motivated a large body of research to develop advanced video editing tools in video compositing [SE02, XCF06, GGC*08, ZDJ*09, RWSG13, ZYQ*14] and producing after effects, such as bullet-time [ZDJ*09], action shots [KWB*15], etc. These works share a common goal of achieving plausible video editing without explicitly reconstructing the 3D geometries of the scenes. For example, in video composition, Schödl and Essa [SE02] optimize the sequence of source 2D video sprites to adapt to user specified motion trajectory. Xiao *et al.* [XCF06] propose a 3D camera trajectory alignment to seamlessly transfer a static video object across two video sequences of different scenes. Goldman *et al.* [GGC*08] exploit the tracked 2D object motion to enable interactive video annotations that associates graphical objects (e.g., text, scribbles) with moving objects, and support intuitive video navigation by direct manipulation. Zhong *et al.* [ZYQ*14] introduce a system that uses only estimated 2D transformations between consecutive frames to enable replacing the background in a video of a moving foreground subject without suffering from slippage artifacts. Zhang *et al.* [ZDJ*09] reconstruct from raw video the temporally consistent video depth maps, which is utilized to produce a variety of refilming effects.

With access to only the sparse 3D points and camera motion of input video, Liu *et al.* [LWCT14] is capable of generating realistic, 3D-aware tracking shots effect from videos, which is otherwise laborious even for professionals. Klose *et al.* [KWB*15] demonstrate how 3D information, when appropriately utilized, can enable many compelling scene-space video processing effects. With a similar motivation, we target to further support 3D object-level manipulations using coarse level 3D scene in the form of SSPs. Further, we handle complex interactions such as inter-object occlusions, which is beyond the capability of existing alternatives.

Video matting. A key component for realistic video composition is the extraction of foreground objects along with high quality alpha mattes. Various algorithms have been developed for natural video matting and segmentation [CAC*02, WBC*05, BWSS09, CHM*10]. We benefit from these existing algorithms and leverage them to extract foreground masks and alpha mattes from raw input videos.

Novel view video rendering. Our work is also closely related to the concept of image-based rendering from novel views. Works in a recent decade appears in various contexts, such as light fields [DLD12], video stabilization [LGJA09, KCS14], free-viewpoint navigation [CDSHD13], etc. In general, the problem is solved by variational image warps guided by estimated meshes [DLD12], proxy cuboids [ZCC*12], synthesized depth maps [CDSHD13], 3D proxy geometries [KCS14], or proxy planes [HM15]. Single or multiple input frames are warped and blended to synthesize the novel view. Compelling result is ensured as the change of camera view is within a controlled range. Instead of changing camera views in a static scene, we focus on objects with complex structures moving freely in a dynamic scene. We propose a novel image-based rendering tailored to our setting (see Section 6.1 for comparison with Liu et al. [LGJA09]).

Video-based modeling. The ultimate solution to render photorealistic video composition is probably to reconstruct high fidelity 3D models of the scene from videos. For example, van den Hengel *et al.* [vdHDT*07] develop an interactive modeling system to build 3D models from a video; Newcombe *et al.* [NFS15] propose a fully automatic system based on dense SLAM to reconstruct non-rigidly deforming scenes in real-time; and Wang et al. [WKM15] introduce dynamic SfM to simultaneously model moving objects and camera paths. Li et al. [LZS*11] fuse image and LiDAR information to abstract building models as textured cuboid structures. In addition, reconstructing 3D models of static and rigid objects captured using hand-held camera is now easy by using commercial tools such as Autodesk 123D Catch [Aut09] and Vi3Dim [Vi311]. However, these tools are still limited in its ability to reconstruct scenes

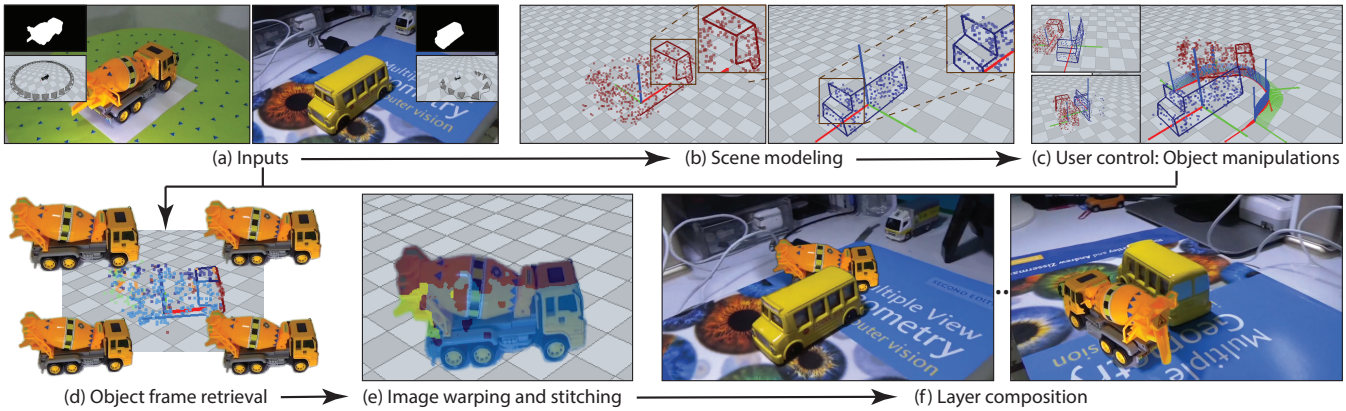


Figure 2: *Overview: (a) Given the input videos, the system, assisted by users, starts by (b) organizing the recovered 3D information to a set of sparse structure points (SSP). (c) Such SSPs, when organized properly, enable the user to perform various object-level edits. The system aims to re-render the edited objects for each of target frames using a novel image-based rendering technique that runs in three stages: (d) First, multiple input frames are adaptively retrieved from object video, which are further (e) warped and stitched to form the final object image. (f) Object images along with synthesized shadows and depth maps are blended to generate plausible video sequence.*

with complex geometries (e.g., tree). As a result, they may produce models of mediocre quality (see Figure 10(a,b)) and require significant post-processing before further edits. Recently, Xu *et al.* [XLS*11] present a data-driven approach for synthesize realistic video animation of humans merely according to user-defined body motions and viewpoints. Such system, although powerful and compelling, relies on high quality multi-view video sequences acquired in a setup studio, which quickly gets expensive. In contrast, we present a much simpler workflow to create plausible manipulation directly from raw RGB videos.

3. Overview

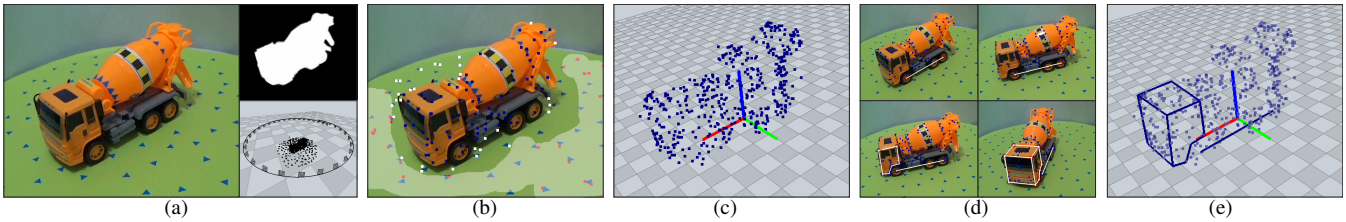
The system takes as input single or multiple video clips captured from different scenes under respective camera motions. Our goal is to create a video editing system that allows users to perform various 3D object-level edits, while the system automatically synthesizes plausible video sequence mimicking real-world interactions between video objects and scene (e.g., occlusions, shadows), without explicitly reconstructing the 3D geometry models. This is achieved by exploiting 3D proxy geometry for foreground video object in the form of *sparse structure points (SSP)* to enable a novel *image-based rendering* technique that warps, stitches and blends dynamically selected input frames to re-render object images from novel camera views and compose the final video sequence.

Preprocessing and assumptions. To acquire object-level information in both 2D and 3D, we pre-process the input videos using state-of-the-art tools. In the 2D domain, we segment video objects from the background in each frame using commercial tools (e.g., Rotobrush in Adobe After Effects [BWSS09]), and generate alpha mattes for foreground objects using [HRR*11]. We utilize the Voodoo Camera Tracker [TB], which is based on the structure-from-motion (SfM) algorithm [PVGV*04], to estimate 3D scene information, including camera motions and a set of sparse structure points. To guarantee the stable outputs of SfM, we make the following assumptions on input videos: (i) video objects are opaque, unoccluded and placed statically on some dominant horizontal reference surfaces (e.g., ground plane, tabletop, etc); (ii) both video objects and reference surfaces contain adequate texture features for extracting sufficient structure points; and (iii) videos are free of noticeable motion blur and temporal changes in lighting are prohibited. Note that although such preprocessing is time-consuming, it needs only to be executed once for each video (see Figure 2(a)).

User control. Given preprocessed input videos, the user can scribble a few 2D strokes and optionally traces edges or polygons on any frame of video in which the object is visible, while the system then automatically extracts a reference plane, object-level SSPs, and a scene hierarchy describing their mutual relations. We organize the hierarchical coordinate system in a way such that the object SSPs can be easily moved around or transplanted onto planes in other videos (see Figure 2(b)). This facilitates the following object-level 3D manipulations: (i) applying 3D transformations (i.e. translation, rotation and scaling) to objects, (ii) specifying keyframe animation on objects, (iii) duplicating objects, and (iv) transferring objects across videos. For instance, to transfer objects from a video to the other one, the user simply adds the object SSP to the reference plane of the other video. By default, the objects rest on the plane and are centered at the origin of world coordinate frame, while the user can further refine object placements (see Figure 2(c) and supplementary video).

Our framework consists of three stages: (i) Scene modeling, (ii) Image-based rendering using SSP, and (iii) Layer composition.

(i) Scene modeling. In this stage, the system leverages the recovered 3D scene information (i.e., camera parameters and sparse 3D points) from input videos and the user scribbles to model object-level SSPs and a scene hierarchy (see Section 4). Specifically, the system is based on the user-prescribed scribbles and line segments to automatically infer a reference plane and organize sparse 3D points as well as traced 3D edges to form the object-level SSP (see Figure 2(b)). To facilitate manipulation in 3D, the system re-estimates a new hierarchical coordinate system where the world



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

Figure 3: *With a few user-scribbles (b,d), the system analyzes the raw video (a) to extract a set of sparse structure points (SSP) as a set of 3D points (c) / edges (e), which we use as scene proxy.*

coordinate frame and local coordinate frame of object are aligned to the reference plane. Object SSP and camera poses are then normalized and transformed to the new world coordinate frame.

(ii) Image-based rendering using SSP. The input to this stage is a new 3D configuration of object SSP in the world coordinate frame of target video. We refer to the input video from which the object SSP is extracted as the *object video*, while the *target video* represents the video where the object is manipulated. Frames of object video and target video are called *object frames* and *target frames*, respectively. The system aims to re-render the objects from the camera view of each of target frames. In traditional image-based rendering from novel views, the goal is to alter original camera poses. In contrast, we allow users to perform direct 3D object manipulations that poses two challenges: (a) object and target videos are captured with very different camera motions, and (b) manipulated objects may introduce complex interactions with scene geometry, such as occlusion and casting shadows. We overcome these problems using a novel image-based rendering that exploits the object SSP and recovered camera motions from input videos to combine several input object frames to form the target object images.

The system first decomposes the object SSP into partially overlapping parts based on the local shape similarity and spatial proximity (see Section 5.1). Then it finds appropriate object frames for each object part based on a similarity metric measuring the distance between an object frame and a target frame from positions of camera and object part. We improve the temporal smoothness across the entire target sequences using a Markov Random Field (MRF) model (see Section 5.2 and Figure 2(d)). For each target frame, the system jointly warps the retrieved object frames of object parts along with associated alpha mattes using a *structure-preserving image warping*. Such image warping is guided by the projection of underlying SSP in corresponding object frames and target frame, and aims to preserve the structures within and across object parts in 2D domain (see Section 5.3). The system then performs a *spatio-temporally coherent image stitching* on warped object frames to compose the final object image (see Section 5.4 and Figure 2(e)).

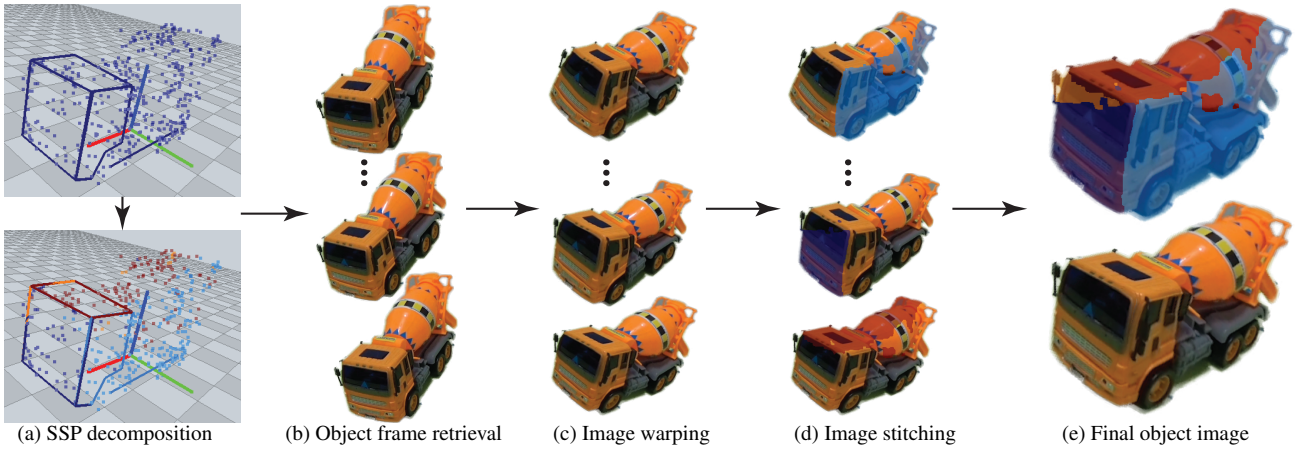
(iii) Layer composition. To composite plausible target sequence, we take a simple approach: our system synthesizes a simple planar shadow and an approximate depth map for each object. The former is done by applying aforementioned warping-based approach to the alpha mattes of object from a user-defined point light source. The depth map is generated by a smooth interpolation of the depth values sampled on the projection of SSP to target camera view. Finally, the object layer (i.e., object image, alpha matte and depth map) and shadow layer are blended appropriately with the target frame to compose the final video sequence (see Section 5.5 and Figure 2(f)).

4. Scene Modeling

Modeling SSP. The estimated camera parameters as well as a set of 3D points describe rough 3D information about the whole scene, but require further processing to be structured into object-level SSP. The system starts by finding structure points for each object. It tests the projection of visible structure points against the corresponding object masks in a per-frame basis and collects candidates across the entire video sequence. Such SSPs along with the approximate point normal constitute the baseline SSP of object. We then fit a reference plane to remaining structure points using RANSAC. In some cases, where the background scene may contain non-planar surface, our system allows users to guide the fitting process by specifying 2D scribbles on any of the frame where the reference surface shows sufficient visible structure points overlaid on it (see Figure 3(b)).

Modeling scene hierarchy. According to the assumption that objects are placed on a ground plane, we compute a local coordinate frame for each object by projecting the structure points to the reference plane as 2D points and finding an oriented minimum bounding box of the 2D points. The center of 2D points, the two dominant directions of bounding box and plane normal form respectively the origin, xy-plane and z-axis of local coordinate system. We further pick one of object coordinate frames as a new world coordinate frame to which the objects and camera poses are transformed (see Figure 3(c)). This provides a scene hierarchy, which helps users to intuitively manipulate objects (e.g., moving on the reference plane) and directly transfer objects from the reference plane of object video to that of target video.

Modeling edge primitives. Some objects in daily life present abundant edge structures, especially in the context of man-made objects such as vehicles, buildings, etc. Hence using the SSP with only structure points in our warping-based technique can not guarantee the preservation of these edge structures in the final results. Inspired by van den Hengel *et al.* [vdHDT*07], we support a similar user interface to trace polygons (see Figure 3(d,bottom)) or edges (see Figure 3(d,top)) on video frames, while the system exploits the structure points and multi-view geometry algorithms [HZ03] to automatically infer the 3D counterparts to structure the SSP (see Figure 3(e)). During tracing, the system automatically snaps the traced lines to nearby 2D edges. Similar to structure points, each edge primitive is also associated with a list of frames where it is visible. We determine the visibility of polygonal edges by the visi-



(a) SSP decomposition (b) Object frame retrieval (c) Image warping (d) Image stitching (e) Final object image

Figure 4: *Image-based rendering using SSP: (a) Given a SSP in a target view, the system first decompose the SSP into parts based on local shape similarity. (b) It then adaptively selects object frames for individual part. Note that the retrieved object frames could have very different perspective from novel view. (c) The system performs a structure-preserving warp to jointly warp the selected object images to target view. (d) The warped images are then stitched to form the final object image (e) in a spatio-temporally coherent manner to reduce temporal jitter.*

bility of associated polygon. If an edge is shared by two polygons, then it is visible as long as one of the polygons is visible. However, determining the visibility of a single edge primitive is tricky as it has no meaningful normal direction. Instead, we request the user to manually prescribe the keyframes where the visibility of edges toggles.

5. Image-based Rendering using SSP

As discussed in Section 3, the object manipulations performed by users result in a new 3D configuration of object SSP in the world coordinate frame of target scene. The next stage is to re-render the object images from the novel target camera views. A straightforward approach is to find an input frame in object video with closest camera view to the novel view, and warp the entire image using the point correspondence between the images of SSP in object frame and target frame. Nevertheless, such *global* warp would easily lead to severe distortions especially as the novel camera view substantially deviates from the original one [CDSHD13]. To alleviate visible artifacts incurred by the global warp, we adopt a *local* warp strategy that first decomposes the SSP into parts based on local shape similarity (see Figure 4(a)), selects carefully object frames for individual part (see Figure 4(b)), and performs a structure-preserving warp to jointly warp the selected object images to target view (see Figure 4(c)). The warped images are then stitched to form the final object image in a spatio-temporally coherent manner to reduce temporal jitter (see Figure 4(d,e)). These object images along with synthesized object-wise shadow and depth maps are blended with target frames to imitate complex real-world interactions in the final compositive video sequence.

5.1. Proxy Geometry Decomposition

Object parts corresponding to nearly planar surfaces will introduce less perspective distortion in image warp. Hence, we divide the structure points of SSP into disjoint parts based on a shape similarity metric using the estimated point normal and 3D position. We define the shape similarity measurement between two structure points, p_i and p_j, as the weighted norm $\|p_i - p_j\|^2 + \beta\|p_i^n \cdot p_j^n\|^2$, where p and p^n represent respectively the 3D position and normal of structure point. The weight β is used to adjust the relative influence of the positional and normal terms, and we used an empirical setting of β=1.0. We use non-parametric mean shift clustering algorithm with kernel and bandwidth functions as suggested in [CM02] to obtain disjoint clusters of structure points. Each cluster corresponds to an object part where the constituent structure points are used to guide a local warp (see Figure 4(a,bottom)). To enhance the spatial coherence between adjacent object parts in the final image warp, we relax the boundary of clusters and allow partial overlapping between clusters, which means structure points can be associated with multiple object parts based on its proximity to clusters using a distance threshold. Similarly, the edge primitives are labeled to belong to an object part if the average distance between endpoints and structure points of object part is within a threshold. Note that such decomposition needs to be executed only once for each object and can be subsequently reused.

5.2. Object Frame Retrieval

Given the decomposed SSP, the next step is to find appropriate object frames for object parts for the warping-based synthesis. We base our frame retrieval algorithm on the following observations: (i) for each object part, the camera view in the retrieved object frame should be close to the camera view in target frame in order to obtain good warping result; (ii) matching only individual object part frame-by-frame will obtain retrieved frames that are not temporally adjacent in object video. This may incur spatial inconsistency as well as temporal flickering artifacts in the appearance of object. Thus, for the first point, we define a frame-to-frame distance metric that measures the similarity between two camera views relative to an object part using camera positions and structure points. To feature a spatio-temporally coherent time-varying object appearance, we model the frame retrieval problem using multi-label Markov Random Fields (MRF) that take into account

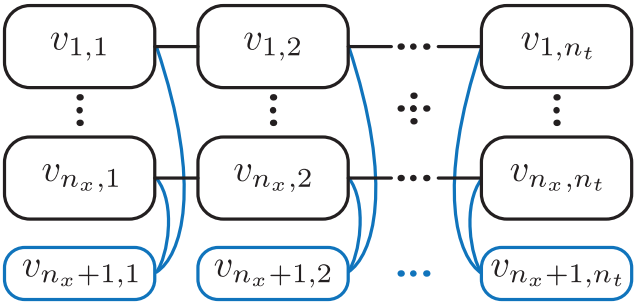


Figure 5: *An illustration of spatio-temporal graph. Temporal and spatial edges are colored black and blue, respectively.*

spatio-temporal relationships between object parts, and solve the problem using optimization.

Frame-to-frame distance metric. We denote each object part as $\{\mathbf{X} = (\mathbf{P}, \mathbf{E}, \tilde{\mathbf{P}}_k, \tilde{\mathbf{E}}_k)\}$, where $\mathbf{P}$ and $\mathbf{E}$ stand respectively for structure points and edge primitives of object part, while $\tilde{\mathbf{P}}_k$ and $\tilde{\mathbf{E}}_k$ return respectively structure points and edges that are visible in k-th frame of object video. Let $\{\mathbf{C}^t_k\}$ and $\{\mathbf{C}^o_k\}$ be respectively the camera positions of target video and object video with subscript k indicates frame index. Our distance metric is based on the angular difference of two cameras viewing at a 3D point in object coordinate frame. The system first registers the camera motions of target video (i.e., novel camera views) and object video to the the local coordinate frame of object. The distance between a target frame and an object frame with regard to an object part is defined as:

$$D_{\text{cam}}(\mathbf{C}^t, \mathbf{C}^o, \mathbf{P}) = \frac{1}{|\mathbf{P}|} \sum_{p \in \mathbf{P}} D_{\text{ang}}(\mathbf{C}^t, \mathbf{C}^o, p), \tag{1}$$

where

$$D_{\text{ang}}(\mathbf{C}^a, \mathbf{C}^b, p) = \frac{180}{\pi} \cdot \arccos\left(\frac{\mathbf{C}^a - p}{\|\mathbf{C}^a - p\|} \cdot \frac{\mathbf{C}^b - p}{\|\mathbf{C}^b - p\|}\right). \tag{2}$$

Multi-label MRF. To model the spatio-temporal relationships between object parts, we build a graph $G = (V, E)$, where the nodes $V = \{v_{i,k} | i = 1, ..., n_x, k = 1, ..., n_t\}$ represent object parts in temporal video sequence in which $v_{i,k}$ indicates the i-th object part in the k-th frame of target video with n_x and n_t denoting the total number of object parts and target frames, respectively. The edges E comprise of two types of edges, the temporal edge and the spatial edge. The former connects nodes $v_{i,k}$ and $v_{i,k+1}$, and the latter connects nodes $v_{i,k}$ and $v_{n_x+1,k}$, where $v_{n_x+1,k}$ indicates a dummy node at each temporal frame to ensure an acyclic graph. Figure 5 illustrates such a graph. Our goal is to assign each node $v_{i,k}$ a frame from object video that balances among three energy terms, namely *frame similarity*, *spatial smoothness* and *temporal smoothness*. Let the object frames assigned to the nodes be $F = \{\mathcal{F}_{i,k} | i = 1, ..., n_x, k = 1, ..., n_t\}$.

The *frame similarity term* aims to minimize the difference between target frame and selected object frame based on our frame-to-frame distance metric, and is defined as:

$$E_{fs}(F) = \sum_{k=1}^{n_t} \sum_{i=1}^{n_x} D_{\text{cam}}(\mathbf{C}^t_k, \mathbf{C}^o_{\mathcal{F}_{i,k}}, \mathbf{P}_i), \tag{3}$$

The *spatial smoothness term* prefers using the same or temporally adjacent object frames for object parts in the same target frame to enhance the coherence of object appearance, and is defined as:

$$E_{ss}(F) = \sum_{k=1}^{n_t} \sum_{i=1}^{n_x} D_{\text{ang}}(\mathbf{C}^o_{\mathcal{F}_{n_x+1,k}}, \mathbf{C}^o_{\mathcal{F}_{i,k}}, 0) \delta(\mathcal{F}_{n_x+1,k}, \mathcal{F}_{i,k}), \tag{4}$$

where

$$\delta(\mathcal{F}_i, \mathcal{F}_j) = \begin{cases} 1 & \text{if } |\mathcal{F}_i - \mathcal{F}_j| < 60 \\ \phi & \text{otherwise.} \end{cases}$$

We set ϕ to 100 to penalize the case where two selected object frames are not within a temporal window of 60 frames.

Similarly, we define the *temporal smoothness term* as:

$$E_{ts}(F) = \sum_{k=1}^{n_t-1} \sum_{i=1}^{n_x} D_{\text{ang}}(\mathbf{C}^o_{\mathcal{F}_{i,k}}, \mathbf{C}^o_{\mathcal{F}_{i,k+1}}, 0) \delta(\mathcal{F}_{i,k}, \mathcal{F}_{i,k+1}), \tag{5}$$

which favors using the same or temporally adjacent object frames for object parts in consecutive target frames to avoid temporal flickering artifacts.

Solving the object frame retrieval problem then amounts to minimizing the total energy function:

$$F^* = \arg\min_F [E_{fs}(F) + \lambda_s (E_{ss}(F) + E_{ts}(F))], \tag{6}$$

with λ_s controlling the relative importance among the energy terms. By carefully organizing unary and binary terms, the Equation 6 is equivalent to the typical formulation of multi-label Markov Random Fields, which can be solved efficiently using multi-label graph cut algorithm [BVZ01].

5.3. Structure-preserving Image Warping

Once we have retrieved the best matching frames from object video for object parts, the next step is to warp the retrieved object frames to the target view using the underlying SSPs. In one possible alternative, we can adopt existing state-of-the-art techniques in image warping [LGJA09, IMH05, SMW06] to warp each object frame *individually* to the target view based on the point correspondences derived from the projected structure points in object and target views. However, such direct approach will clearly introduce visual artifacts due to the non-linear nature of image warp. Especially in the context of man-made objects with abundant edge structures, individual warp will introduce distorted structures (e.g., bending edges) within object parts and discontinuous structures (e.g., broken edges) across object parts (see Figure 8). To address above issues, we propose a novel structure-preserving image warping that augments the existing system [LGJA09] with the terms tailored for structures preservation, and then warps *jointly*, rather than individually, the retrieved object frames to the target view.

Formulation. For each object part, the system first divides the retrieved object frame $\mathcal{F}^o$ into an $n \times m$ uniform grid mesh with $\hat{V}$ denoting the initial grid vertex. We render the image warp as computing new vertex position V for the grid mesh that minimizes an energy function comprised of following energy terms.

The *point alignment term* measures how well the projection of visible structure points in $\mathcal{F}^o$ aligns with corresponding projection in

target view after the warp. Since the projected structure point is typically not coincident with grid vertex, we represent the projected point as $\bar{p}_i = w_i^T \hat{V}_i$, where $\hat{V}_i$ is a vector of the four vertices enclosing the grid where the $\bar{p}_i$ lies in, and w_i contains the four bilinear interpolation coefficients. The energy term is defined as:

$$E_{pa}(V) = \frac{1}{|\tilde{\mathbf{P}}_{\mathcal{F}^o}|} \sum_{p_i \in \tilde{\mathbf{P}}_{\mathcal{F}^o}} \omega(p_i) \|w_i^T V_i - \bar{p}_i^t\|^2, \quad (7)$$

where $\bar{p}_i^t$ is the projected structure point in target view. Note that the structure points will turn visible and invisible over time instantly, leading to temporal jitter artifacts [LGJA09]. Therefore we introduce a piecewise-linear function, $\omega(p_i)$, that fades-in and fades-out the influence of each structure point over time to improve the temporal coherence, and is defined as:

$$\omega(p) = \begin{cases} 1 & \text{if } \theta_p \leq T_\theta \\ 1 - \frac{\theta_p - T_\theta}{\pi/2 - T_\theta} & \text{if } T_\theta < \theta_p < \pi/2 \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where θ_p is the angle between the point normal and target camera viewing direction, and we set $T_\theta = \pi/10$ in our experiments.

The *similarity transformation term* measures the deviation of each deformed mesh grid from a similarity transformation with regard to the initial mesh grid and is denoted as $E_{st}(V)$. We follow exactly the same formulations as in [LGJA09] and refer the readers to the Section 4.1.2 therein for details.

The *edge preservation term* captures how well the projection of visible edges in $\mathcal{F}^o$ coincide with the counterpart projection in target view after the warp. A 2D line can be parameterized using the line equation $l(x,y,\alpha,b) : \sin(\alpha)x - \cos(\alpha)y + b = 0$. We first estimate the parameters (α_e, b_e) for each visible edge e using its projected line segment in target view. Then we uniformly sample on its counterpart projection in object frame a set of 2D points, which is denoted as $\mathbf{P}_e = \{p = [S_x(p), S_y(p)]^T\}$, where the $S_x(p)$ and $S_y(p)$ are the x- and y-components of sample point represented using the bilinear interpolation of the grid mesh V. We define energy as:

$$E_{ep}(V) = \frac{1}{|\tilde{\mathbf{E}}_{\mathcal{F}^o}|} \sum_{e \in \tilde{\mathbf{E}}_{\mathcal{F}^o}} \sum_{p \in \mathbf{P}_e} l(S_x(p), S_y(p), \alpha_e, b_e)^2. \quad (9)$$

Note that the energy terms we have discussed so far are formulated based on the individual object part using independent grid mesh. This will incur misalignment among individually warped images and produce discontinuous appearance in the final object image. We alleviate such artifact by introducing a *structure preservation term* that models the spatial alignment of common structure points and edges shared by object parts as soft constraints. First, we compile the structure points and edges (i.e., two endpoints) shared by multiple object parts into a set of shared points $\mathbf{P}_s = \{p_{i,j}\}$, where $p_{i,j}$ indicates a 3D point shared by i- and j-th object part and is visible to corresponding retrieved object frames. We represent the projection of such shared points using bilinear interpolation of individual grid meshes V_i and V_j, and aim to align the projections among the warped images. Thus, the energy term is defined as:

$$E_{sp} = \frac{1}{|\mathbf{P}_s|} \sum_{p_{i,j} \in \mathbf{P}_s} \|w_i^T V_i - w_j^T V_j\|^2. \quad (10)$$

Influence map. To measure to relative importance of object parts in both image warp and stitch, we compute an influence map M_i for each object part based on the visible structure points and edges in target view. This is done by a simple approach that draws on a blank image a Gaussian kernel $G(x,y) = \exp(-x \cdot y / 2\sigma^2)$ centered at each projected structure point and along the projection of edges in target view with the kernel size of $30\omega(\bar{p}^t) + 1$ and 31 pixels, respectively. Pixel's value is accumulated during the drawing and then normalized to $[0,1]$.

Energy optimization. Now, at each target frame, our system jointly warps the retrieved object frames to target view by simultaneously computing the new vertex positions of all grid meshes such that the following combined warp energy is minimized:

$$\arg\min_{\{V_1,\ldots,V_{n_x}\}} [\lambda_{sp}E_{sp} + \sum_{i=1}^{n_x} \beta_i(\lambda_{pa}E_{pa}(V_i) + \lambda_{st}E_{st}(V_i) + \lambda_{ep}E_{ep}(V_i))], \quad (11)$$

where the four weighting coefficients $\{\lambda_i | i \in \{sp, pa, st, ep\}\}$ controlling the relative influence of each energy term, while the $\beta_i = \sum_{(x,y)} M_i(x,y)$ is used to control the relative importance of individual warps. The resultant formulation in Equation 11 is a linear least-square problem, which can be solved efficiently using standard sparse linear system solvers.

5.4. Spatio-temporally Coherent Image Stitching

Our goal in this step is to stitch all the warped object frames together to obtain the final object image for each of target frames. Such objective is identical to the work by Agarwala *et al.* [ADA*04] to combine seamlessly parts of a set of photographs into a single composite image. Therefore, we build upon their framework and design novel objective terms that pay attention to spatio-temporal coherence in the final stitched images.

Spatial domain stitch. For each target frame, the image stitching problem can be rendered as a discrete pixel labeling problem, where the system manages to choose a label L_p from one of the warped object frame for every pixel p in the object image such that the following objective energy is minimized:

$$\arg\min_{L_p} [\sum_p E_{data}(p, L_p) + \sum_{p,q \in N(p)} E_{smooth}(p, q, L_p, L_q)], \quad (12)$$

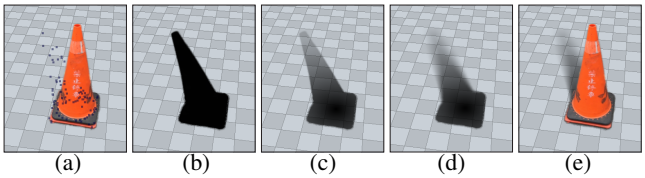
where the data term $E_{data}(p, L_p) = 1 - M_{L_p}(p)$ encourages selecting pixels with higher influence values, while the smoothness term $E_{smooth}(p, q, L_p, L_q)$ measures the spatial relationships to 4-neighbors $N(p)$ and favors invisible stitch seams. Specifically, E_{smooth} is a weighted combination of three energy terms:

$$E_{smooth} = \lambda_c E_c + \lambda_g E_g + \lambda_e E_e, \quad (13)$$

where the E_c and E_g follow the same formulations in [ADA*04] that match pixels respectively in color and gradient domains. We further introduce the 'edge' term E_e to prevent the seams from cutting through the edges, and define the energy term as:

$$E_e(p, q, L_p, L_q) = \|M^e_{L_p}(p) + M^e_{L_p}(q)\| + \|M^e_{L_q}(p) + M^e_{L_q}(q)\|. \quad (14)$$

Temporal coherence. There is, however, a factor that may cause temporal flickering in the interior appearance of object image due to the drastic changes of stitch seams in temporal frames (see Figure 9). We propose a greedy approach to improve the temporal coherence by encouraging the stitch seams in the current target frame



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

Figure 6: *Shadow map synthesis. (a) Projection of SSP on the plane from light view. (b) Warped alpha matte. (c) Applying shadow attenuation effect. (d) Gaussian blurred boundary. (e) Final result.*

to be as similar as possible to those in the previous frames. We first compute a 2D motion field for every two consecutive target frames. This is achieved by a smooth scatter data interpolation of a sparse set of 2D motion vectors derived from the projections of structure points in two consecutive target views. Thus, given a point p in i-th frame, we denote the corresponding pixel in j-th frame as $f_{i,j}(p)$. With such temporal correspondence, we introduce extra data and smoothness terms measured in temporal domain as follows:

$$E^t_{data}(p,L_p,k) = \sum_i^t \delta(L_p, L_{f_{k,k-i}(p)}), \text{ and}$$
$$E^t_{smooth}(p,q,L_p,L_q,k) = \sum_i^t \delta(L_p, L_{f_{k,k-i}(p)}) + \tag{15}$$
$$\delta(L_q, L_{f_{k,k-i}(q)}) + \delta(L_{f_{k,k-i}(p)}, L_{f_{k,k-i}(q)}) + \delta(L_p, L_q),$$

where k is the index of current frame, t is temporal window of size 2, and $\delta(i,j)$ is a delta function, which returns 1 if two parameters (frames) are equal and 0 otherwise. The Equation 12 is revised as:

$$\arg\min_{L_p}[\sum_p E_{data} + \lambda_t E^t_{data} + \sum_{p,q\in N(p)} E_{smooth} + \lambda_t E^t_{smooth}]. \tag{16}$$

The four weighting coefficients $\{\lambda_i | i \in \{c,g,e,t\}\}$ control the relative influence of each energy term, and we solve stitching problem in a greedy fashion by successively optimizing the Equation 16 for each target frame using multi-label graph cut algorithm [BVZ01]

5.5. Layer Composition

Shadow map synthesis. Shadows are important cues for creating realistic compositions. However, without the access to the 3D models of scene and objects, it is difficult to cast realistic shadows respecting the geometry of scene and objects. Moreover, the difference of camera motion as well as the lighting direction in the target and object videos render the conventional approaches that extract and transfer shadows from object video to target view infeasible [XCF06]. Instead, we exploit the SSPs and the estimated ground plane to synthesize plausible planar shadows using the proposed image-based rendering. First, the user places and manipulates a point light source in 3D, while the system computes so-called shadow points as the intersections of plane and rays casting from light source to every visible structure point (see Figure 6(a)). The projections of shadow points in target view are used to guide the image-based rendering that warps and stitches several alpha mattes of objects retrieved from the point of view of light source in the target view to obtain a shape mask (see Figure 6(b)). Then we unproject in target view every pixel of shadow mask to the plane, followed by modulating the pixel's intensity with a value $\alpha_{gain} \cdot \exp(-d)$, where d is the distance between unprojected point and bottom center of object, and α_{gain} controls the overall intensity of shadow. This mimics the shadow attenuation effect (see Figure 6(c)). Lastly, we apply Gaussian blur on the mask boundary to obtain the final shadow map (see Figure 6(d)).

Illumination adjustment. Since the lighting condition in the object video is generally different from the target video (e.g., indoor versus outdoor), this may introduce illumination changes between object image and background within and across frames. We propose a simple approach to alleviate such artifact. Specifically, we first utilize the point normal, viewing directions from camera and light source to compute an intensity for each structure point based on the Gouraud shading model. Then the object image is modulated with a shading map generated by interpolating the sampled intensity values at object's structure points.

Depth map estimation. Since our system naturally supports mixing of multiple objects across videos, the inter-object occlusions needs to be handled properly to generate plausible results. For this purpose, we follow the same approach in [KCS14] to generate a dense depth map for each object. In brief, we first obtain a sparse samples of depth values based on the projection of structure points in target view, and a dense depth map is obtained by solving an optimization problem with objectives of preferring the smoothness of pixel values while approximating the depth values at samples.

The final composition of each target frame is then rendered in two steps. First, we directly blend all the shadow maps with target image. Then, we superimpose all the object images onto the target image while blend the pixels of object images using the corresponding alpha values in an order of descending depth.

6. Results and Evaluation

We have tested our system on a number of input videos captured from a wide variety of contents, ranging from nature outdoor scenes to man-made objects of different scales (e.g., real vehicles, toy models). All the videos were with resolution of 640×360 pixels and captured either using a freely moving hand-held camera or in a set up environment where the object is rest on a rotary platform and recorded by a static camera. For evaluation, we mixed arbitrary video footage by transferring objects across videos, and performed various 3D manipulations on objects to generate 17 plausible video sequences mimicking real-world interactions. Some input videos along with reconstructed SSPs and 8 edited video sequences can be found in Figure 7. We show four output frames in each result to show the capability of our system to recover perspective distortions and plausibly handle shadows as well as inter-object occlusions. We refer the reader to supplementary video for complete video sequence and other editing results.

Parameters setting. Our image-based rendering algorithm runs in several stages each of which defines a set of weighting coefficients λ in the objective function. The empirical values for these coefficients are as follows: (i) $\lambda_s = 0.1$ in Equation 6; (ii) $\lambda_{sp} = 4$, $\lambda_{pa} = 1$, $\lambda_{st} = 4$, and $\lambda_{ep} = 4$ in Equation 11; and (iii) $\lambda_c = 1$, $\lambda_g = 1$, $\lambda_e = 10$, and $\lambda_t = 20$ in Equation 16. We used the same parameters setting for generating all the results. While in Section 5.3, we used a grid mesh of resolution 50×50 for image warp.

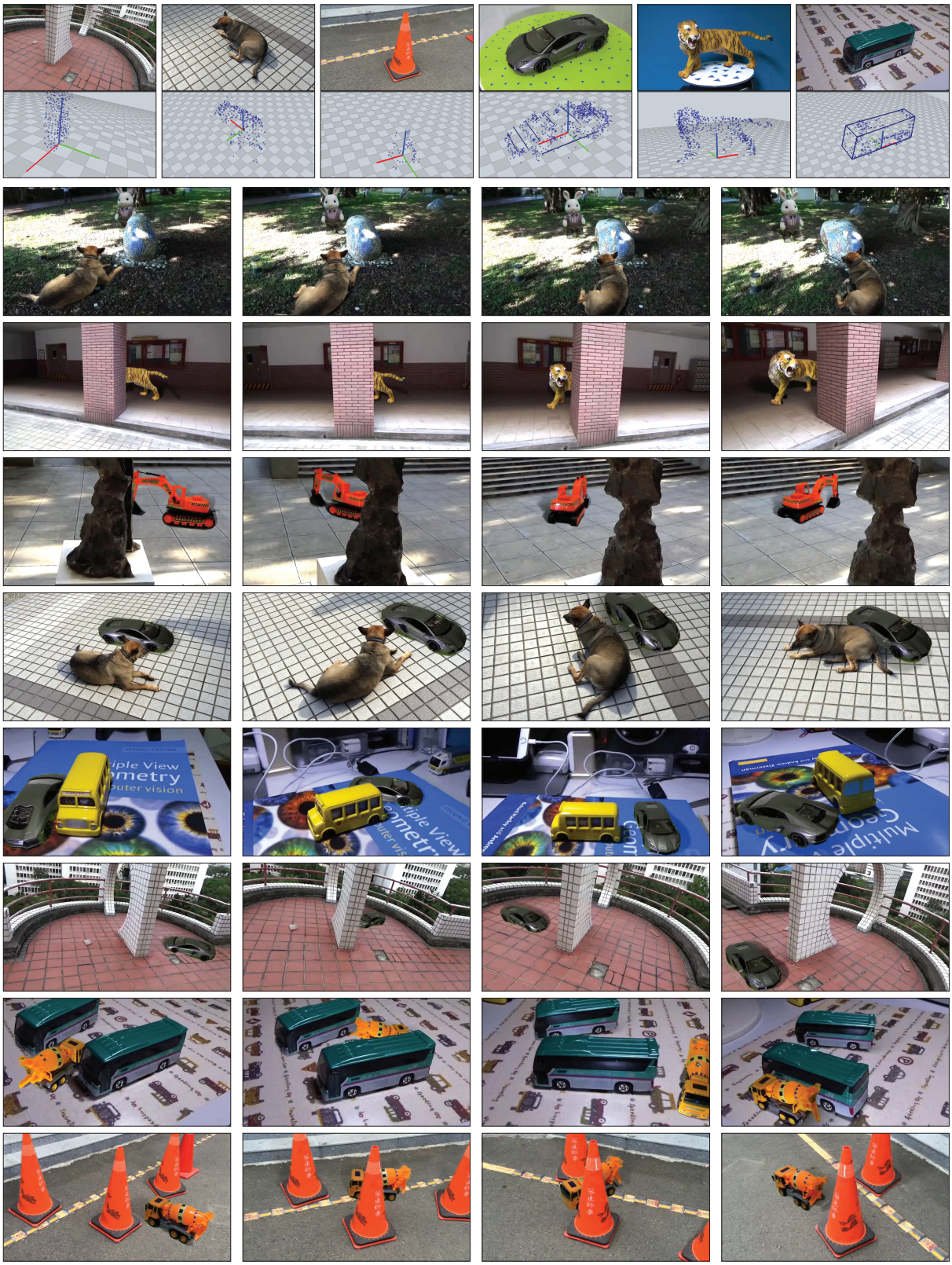
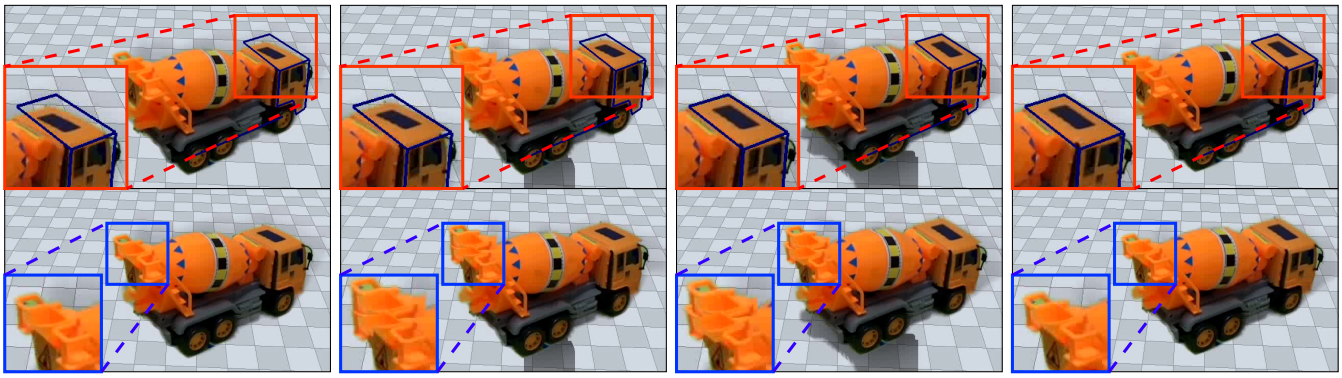


Figure 7: *(Top row) 6 input videos and their SSPs. Below are eight video sequences generated by our system using mixed 3D manipulations, e.g., shuffling, keyframe animation (4-9th rows), and duplicating (8-9th rows). In each result, we show four representative frames in which shadows and inter-object occlusions are handled plausibly by our system. See supplementary video for complete sequences.*



(a) Global warp (b) Local warp (c) Local warp + edge preserve (d) Our result

Figure 8: *Comparing results generated using different image warping strategies. Please refer to Section 6.1 for the detailed description. We show a frame in the video sequence generated by different methods and compare the visual quality in terms of edge preservation within object parts (top row) and structure preservation across object parts (bottom row).*

6.1. Evaluation

We extensively evaluated the performance of our system using several experiments including: (i) comparison with a baseline approach that globally warps the whole object frame retrieved via a nearest neighbor search; (ii) validating the effectiveness of edge and structure preservation terms in the image warping; (iii) comparing results with and without enabling the temporal-coherence in the image stitching; (iv) performing a stress test on the image-based rendering in terms of changing novel camera view; and (v) comparison with the 3D reconstruction methods.

Comparison with baseline method. We implemented a baseline approach that: (i) uses all structure points as a single SSP to drive the image-based rendering; (ii) retrieves a single object frame using simple nearest neighbor search; and (iii) warps globally the retrieved object frame to target view. For the sake of simplicity, we call such baseline approach the "global warp" method, while the method that warps multiple frames to form a final image is called the "local warp" method. The term 'warp' here stands for the image warping technique proposed in [LGJA09]. An example of using the baseline approach is shown in Figure 8(a). While such global warp guarantees the smoothness of texture appearance, it can not effectively recover the perspective distortion and produce visible misalignment near the edge structures (see Figure 8(a,top)). In contrast, our result suffers none of above artifacts (see Figure 8(d)). See the supplementary video for a clear presentation.

Performance of image warp. To validate the effectiveness of structure-preserved image warp, we conducted an experiment that compares the visual quality of video sequences generated using different settings as follows: (i) apply local warp to each object part individually, (ii) enable the edge preservation term (i.e., Equation 9) in local warp, and (iii) our approach. Experimental results show that incremental improvements in the visual quality are noticeable as we augmenting the conventional local warp (see Figure 8(b)) with edge preservation to maintain the edge structures within object parts (see Figure 8(c)). Our approach further considers preserving structures across object parts produces even better result (see Figure 8(d)).

Performance of image stitch. To show the effectiveness of temporal-coherent image stitch, we produced video sequences with and without enabling temporal coherence term in Equation 16. We can observe from Figure 9 that without accounting for temporal coherence, the image stitch will suffer from apparent changes of stitch seams in consecutive frames (see supplementary video).

Stress test. Lastly, we evaluated how far the target camera view can deviate from the input object video before artifacts are noticeable. We setup different virtual camera motions by incrementally increasing/decreasing the viewing angle of input camera and re-render the video sequence. Experimental results indicate that our approach can maintain the rendering quality within a range of $\pm 10^{\circ}$ (see supplementary video).

Comparison with 3D reconstruction methods. We compared the quality of our results with those generated by two commercial 3D reconstruction tools, Vi3Dim [Vi311] and Autodesk 123D Catch [Aut09]. Given an object video, the output of these tools is a textured 3D model reconstructed from video frames. However these tools, even powerful, may still lead to strong artifacts such as generating blurred textures and inaccurate geometries when dealing with complex objects. Moreover, the reconstructed 3D models may contain redundant information from background and thus require extra post-processing before further edits (see Figure 10(a)(b)). In

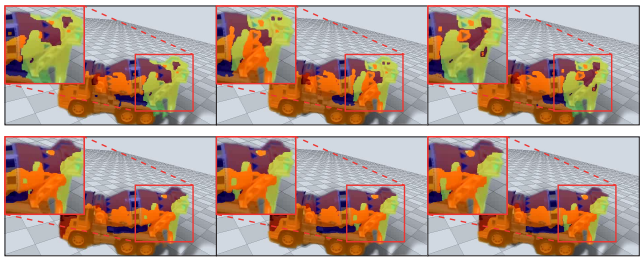


Figure 9: *Three consecutive frames (left to right) of a video sequence generated by our system without (top) and with (bottom) considering the temporal coherence in image stitch. Note that in the former case, the boundary of green patch in the rear of truck changes dramatically when comparing to our result where the shapes of stitch seams are stable in temporal frames.*

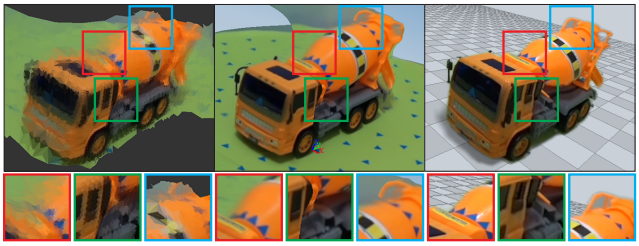


Figure 10: *Comparing our result with those rendered with 3D models reconstructed using commercial tools. Noticeable artifacts, such as blurred texture and inaccurate geometry, are introduced in conventional 3D reconstruction from video.*

contrast, our system generates superior results (see Figure 10(c)) without requiring full 3D models.

6.2. Performance

Preprocessing. For a video footage with around 10 seconds in length and 30 FPS in frame rate, it took $\sim$ 1 hour on average to prescribe the masks for 2 foreground objects using Adobe Roto-brush tool, 2 minutes to compute the alpha mattes, and 20 minutes to run structure-from-motion algorithm using Voodoo Camera Tracker. While the first and last operations are most computationally intensive, they can be executed in parallel.

Timing performance. Since our system requires only a small amount of edge primitives to produce plausible results, the manual effort in scene modeling stage took in average less than 3 minutes for a model with complexity similar to the one shown in Figure 3. After that, the automatic algorithm took 1.5 seconds in average to synthesize a target frame per object with the unoptimized codes. Table 1 reports the detailed running times at each stage in our framework to generate the video sequences in Figure 7. The major bottleneck lies in image warping and stitching in which the time complexity is proportional to the complexity of object SSP.

row	obj#	frame#	frame retrieval	warping	stitching	composition	total
2	3	153	0.03	0.55	1.85	0.14	2.57
3	2	168	0.03	1.22	1.39	0.13	2.77
4	2	96	0.05	0.52	1.30	0.23	2.1
5	2	250	0.02	2.21	1.28	0.09	3.6
6	2	217	0.02	0.47	1.13	0.1	1.72
7	2	148	0.04	1.08	1.33	0.15	2.6
8	3	121	0.04	0.88	2.21	0.18	3.31
9	4	169	0.03	0.9	2.34	0.13	3.39

Table 1: *Timing (sec./frame) for generating results in Figure 7.*

6.3. Limitations

Our approach has several limitations: (i) The plausibility of our results is subject to the quality of structure-from-motion algorithm, which may fail when the assumptions of input video are violated. For instance, videos with severe temporal lighting changes (see Figure 11(a)), textureless scene (see Figure 11(b)), transparent foreground object (see Figure 11(c)), and parts of video object are invisible across all frames will all lead inaccurate 3D scene information and misaligned/structureless object SSP. This will inevitably

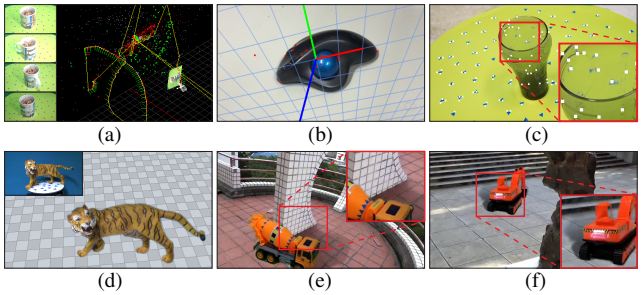


Figure 11: *Limitations. Structure-from-motion algorithm fails in the videos with (a) severe temporal lighting changes, (b) textureless scene, and (c) transparent foreground object. (d) Noticeable perspective distortion will appear as the novel camera view is deviated too much from original video. (e) Fake shadow artifacts are visible as two objects are close to each other (e.g., truck and arch). (f) Our system cannot properly handle object surfaces with strong specular highlights, which results in inharmonious appearance.*

cause perspective distortion in the image warp. (ii) The image-based rendering fails to synthesize object from a novel view deviated too much from original video (see Figure 11(d)). (iii) The system cannot produce complex inter-object shadows in our simple planar shadow assumption (see Figure 11(e)). (iv) Our illumination adjustment can not deal with strong specular highlight and reflection on the object surface (see Figure 11(f)).

7. Conclusion

We present a video editing system that enables object level edits in a single video or across multiple videos, and produces plausible video sequence without explicitly reconstructing the 3D geometries of the scenes. We demonstrate that by utilizing a small amount of user interaction to re-structure a set of sparse structure points recovered from input video, our system is able to achieve non-trivial video edits mimicking real-world interactions, such as shadows and inter-object occlusions. The technical contribution lies in a novel image-based rendering algorithm using the sparse structure points as proxy to guide a structure-preserving image warping on several input frames selected from object video, followed by a spatio-temporally coherent image stitching to synthesize the final object image from novel view. The effectiveness of our system is evaluated extensively on a variety of input videos.

In the future, we plan to explore the following directions: (i) accelerate the image-based rendering process by exploiting its intrinsic nature of parallelization; improve the quality of composition by incorporating sophisticated shadow creation, illumination adjustment [FL11], and appearance harmonization [SJMP10]; adaptively devote processing power based on model saliency [CMH*15]; and finally, extend our system to handle videos with dynamic foreground (e.g., human with articulated motions [NFS15], moving objects [WKM15]).

Acknowledgements

We thank the anonymous reviewers for their invaluable comments, suggestions, and additional references. The project was supported

in part by the Ministry of Science and Technology of Taiwan (102-2221-E-007-055-MY3 and 103-2221-E-007-065-MY3) and the ERC Starting Grant SmartGeometry (StG-2013-335373).

References

[ADA*04] AGARWALA A., DONTCHEVA M., AGRAWALA M., DRUCKER S., COLBURN A., CURLESS B., SALESIN D., COHEN M.: Interactive digital photomontage. *ACM Trans. Graph. (Proc. SIGGRAPH) 23*, 3 (2004), 294–302. 7

[Aut09] AUTODESK: 123D Catch. http://www.123dapp.com/catch, 2009. 2, 10, 11

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 11 (2001), 1222–1239. 6, 8

[BWSS09] BAI X., WANG J., SIMONS D., SAPIRO G.: Video snapcut: Robust video object cutout using localized classifiers. *ACM Trans. Graph. (Proc. SIGGRAPH) 28*, 3 (2009), 70:1–70:11. 2, 3

[CAC*02] CHUANG Y.-Y., AGARWALA A., CURLESS B., SALESIN D. H., SZELISKI R.: Video matting of complex scenes. *ACM Trans. Graph. (Proc. SIGGRAPH) 21*, 3 (2002), 243–248. 2

[CDSHD13] CHAURASIA G., DUCHENE S., SORKINE-HORNUNG O., DRETTAKIS G.: Depth synthesis and local warps for plausible image-based navigation. *ACM TOG 32*, 3 (2013), 30:1–30:12. 2, 5

[CHM*10] CHU H.-K., HSU W.-H., MITRA N. J., COHEN-OR D., WONG T.-T., LEE T.-Y.: Camouflage images. *ACM Trans. Graph. (Proc. SIGGRAPH) 29* (2010), 51:1–51:8. 2

[CM02] COMANICIU D., MEER P.: Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 5 (2002), 603–619. 5

[CMH*15] CHENG M.-M., MITRA N. J., HUANG X., TORR P. H. S., HU S.-M.: Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE TPAMI)* (2015). 11

[DLD12] DAVIS A., LEVOY M., DURAND F.: Unstructured light fields. *Comp. Graphics Forum (Proc. EUROGRAPHICS) 31*, 2pt1 (2012), 305–314. 2

[FL11] FARBMAN Z., LISCHINSKI D.: Tonal stabilization of video. *ACM Trans. Graph. (Proc. SIGGRAPH) 30*, 4 (2011), 89:1–89:10. 11

[GGC*08] GOLDMAN D. B., GONTERMAN C., CURLESS B., SALESIN D., SEITZ S. M.: Video object annotation, navigation, and composition. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology* (2008), UIST '08, pp. 3–12. 2

[HM15] HENNESSEY J. W., MITRA N. J.: An image degradation model for depth-augmented image editing. *Comp. Graphics Forum (Proc. SGP)* (2015). 2

[HRR*11] HE K., RHEMANN C., ROTHER C., TANG X., SUN J.: A global sampling method for alpha matting. In *IEEE CVPR* (2011), pp. 2049–2056. 3

[HZ03] HARTLEY R., ZISSERMAN A.: *Multiple View Geometry in Computer Vision*, 2 ed. Cambridge University Press, 2003. 4

[IMH05] IGARASHI T., MOSCOVICH T., HUGHES J. F.: As-rigid-as-possible shape manipulation. *ACM Trans. Graph. (Proc. SIGGRAPH) 24*, 3 (2005), 1134–1141. 6

[KCS14] KOPF J., COHEN M. F., SZELISKI R.: First-person hyper-lapse videos. *ACM Trans. Graph. (Proc. SIGGRAPH) 33*, 4 (2014), 78:1–78:10. 2, 8

[KWB*15] KLOSE F., WANG O., BAZIN J.-C., MAGNOR M., SORKINE-HORNUNG A.: Sampling based scene-space video processing. *ACM Trans. Graph. (Proc. SIGGRAPH) 34*, 4 (2015), 67:1–67:11. 2

[LGJA09] LIU F., GLEICHER M., JIN H., AGARWALA A.: Content-preserving warps for 3d video stabilization. *ACM Trans. Graph. (Proc. SIGGRAPH) 28*, 3 (2009), 44:1–44:9. 2, 6, 7, 10

[LWCT14] LIU S., WANG J., CHO S., TAN P.: Trackcam: 3d-aware tracking shots from consumer video. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 33*, 6 (2014), 198:1–198:11. 2

[LZS*11] LI Y., ZHENG Q., SHARF A., COHEN-OR D., CHEN B., MITRA N. J.: 2d-3d fusion for layer decomposition of urban facades. In *ICCV* (2011). 2

[NFS15] NEWCOMBE R., FOX D., SEITZ S.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. *IEEE CVPR* (2015). 2, 11

[PVGV*04] POLLEFEYS M., VAN GOOL L., VERGAUWEN M., VERBIEST F., CORNELIS K., TOPS J., KOCH R.: Visual modeling with a hand-held camera. *Int. J. Comput. Vision 59*, 3 (2004), 207–232. 3

[RWSG13] RÜEGG J., WANG O., SMOLIC A., GROSS M.: Ducttake: Spatiotemporal video compositing. In *Comp. Graphics Forum* (2013), vol. 32, pp. 51–61. 2

[SE02] SCHÖDL A., ESSA I. A.: Controlled animation of video sprites. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), SCA '02, pp. 121–127. 2

[SJMP10] SUNKAVALLI K., JOHNSON M. K., MATUSIK W., PFISTER H.: Multi-scale image harmonization. *ACM Trans. Graph. (Proc. SIGGRAPH) 29*, 4 (2010), 125:1–125:10. 11

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3 (2006), 533–540. 6

[TB] THORMÄHLEN T., BROSZIO H.: Voodoo camera tracker. *Freely available for download at http://www. digilab. uni hannover. de 2*. 3

[vdHDT*07] VAN DEN HENGEL A., DICK A., THORMÄHLEN T., WARD B., TORR P. H. S.: Videotrace: Rapid interactive scene modelling from video. *ACM Trans. Graph. (Proc. SIGGRAPH) 26*, 3 (2007). 2, 4

[Vi311] VI3DIM: Vi3dimv2. http://www.vi3dim.com, 2011. 2, 10, 11

[WBC*05] WANG J., BHAT P., COLBURN R. A., AGRAWALA M., COHEN M. F.: Interactive video cutout. *ACM Trans. Graph. (Proc. SIGGRAPH) 24*, 3 (2005), 585–594. 2

[WKM15] WANG T. Y., KOHLI P., MITRA N. J.: Dynamic sfm: Detecting scene changes from image pairs. *Comp. Graphics Forum (Proc. SGP)* (2015). 2, 11

[XCF06] XIAO J., CAO X., FOROOSH H.: 3d object transfer between non-overlapping videos. In *Virtual Reality Conference, 2006* (2006), IEEE, pp. 127–134. 2, 8

[XLS*11] XU F., LIU Y., STOLL C., TOMPKIN J., BHARAJ G., DAI Q., SEIDEL H.-P., KAUTZ J., THEOBALT C.: Video-based characters: Creating new human performances from a multi-view video database. *ACM Trans. Graph. (Proc. SIGGRAPH) 30*, 4 (2011), 32:1–32:10. 3

[ZCC*12] ZHENG Y., CHEN X., CHENG M.-M., ZHOU K., HU S.-M., MITRA N. J.: Interactive images: Cuboid proxies for smart image manipulation. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4 (2012), 99:1–99:11. 2

[ZDJ*09] ZHANG G., DONG Z., JIA J., WAN L., WONG T.-T., BAO H.: Refilming with depth-inferred videos. *IEEE Trans. on Vis. and Comp. Graphics 15*, 5 (2009), 828–840. 2

[ZYQ*14] ZHONG F., YANG S., QIN X., LISCHINSKI D., COHEN-OR D., CHEN B.: Slippage-free background replacement for hand-held video. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 33*, 6 (2014), 199:1–199:11. 2