Plane-Based Content-Preserving Warps for Video Stabilization

Zihan Zhou
University of Illinois at Urbana-Champaign
zzhou7@illinois.edu

Hailin Jin
Adobe
hljin@adobe.com

Yi Ma
Microsoft Research Asia
mayi@microsoft.com

Abstract

Recently, a new image deformation technique called content-preserving warping (CPW) has been successfully employed to produce the state-of-the-art video stabilization results in many challenging cases. The key insight of CPW is that the true image deformation due to viewpoint change can be well approximated by a carefully constructed warp using a set of sparsely constructed 3D points only. However, since CPW solely relies on the tracked feature points to guide the warping, it works poorly in large textureless regions, such as ground and building interiors. To overcome this limitation, in this paper we present a hybrid approach for novel view synthesis, observing that the textureless regions often correspond to large planar surfaces in the scene. Particularly, given a jittery video, we first segment each frame into piecewise planar regions as well as regions labeled as non-planar using Markov random fields. Then, a new warp is computed by estimating a single homography for regions belong to the same plane, while inheriting results from CPW in the non-planar regions. We demonstrate how the segmentation information can be efficiently obtained and seamlessly integrated into the stabilization framework. Experimental results on a variety of real video sequences verify the effectiveness of our method.

1. Introduction

With the fast development of hand-held digital cameras, we have seen a dramatic increase in the amount of amateur videos shot over the past decade. However, very often people find their videos hard to watch, mainly due to the excessive amount of shake and undirected camera motions in the footage. Therefore, there has been an urgent demand in developing high-quality video stabilization algorithms, which are able to remove the undesirable jitters from amateur videos so that they look like to be taken under smooth, directed camera paths.

In general, there are two major steps in stabilizing a jittery input video, namely (1) designing new smooth camera paths, and (2) synthesizing stabilized video frames according to the new path. In this paper, we focus ourselves on the second step, which still remains a highly challenging problem nowadays. Most existing methods [19, 10, 6, 15, 13] apply a full-frame 2D transformation to each input frame to obtain the stabilized output frame. Despite its computational efficiency and robustness, this approach is well-known for its inability in handling the parallax effects of a non-degenarate scene and camera motion, as illustrated in Figure 1 (first row).

In fact, in the ideal case one will need the *dense* 3D structures of the scene in order to create a novel view of it. However, obtaining such a dense reconstruction from 2D images is extremely challenging in terms of both effectiveness and efficiency. Several attempts have been made along this direction [5, 7, 3], which rely on image-based rendering (IBR) to generate new images of a scene as seen along the smooth camera path. But these techniques are all limited to static scenes, among other issues. In a recent work [16], Liu et al. propose a novel method, namely content-preserving warping (CPW), which instead uses the *sparse* 3D points obtained by any structure from motion system for synthesis. The key idea of CPW is that the true dense deformation can be well approximated by diffusing the sparse displacements suggested by the reconstructed 3D points via a carefully chosen regularization term. This approximation is shown to be sufficient for stabilization, producing state-of-the-art results in many challenging cases, as long as there are enough feature tracks in each image region. In practice, however, large textureless regions often exist in the scene, such as ground, building facades, and indoor walls, where feature tracks are rare. It has been noticed that CPW performs poorly in these regions, as illustrated in Figure 1 (second row).

In this paper, we propose a new synthesizing scheme which aims to remedy this important issue of CPW. Our key observation is that real scenes often exhibit strong structural regularities, in the form of *one or more planar surfaces*, which are largely ignored so far by existing methods. More importantly, these planar surfaces typically correspond to the textureless regions in the scene, which are problematic to CPW as well as many other methods.

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.298

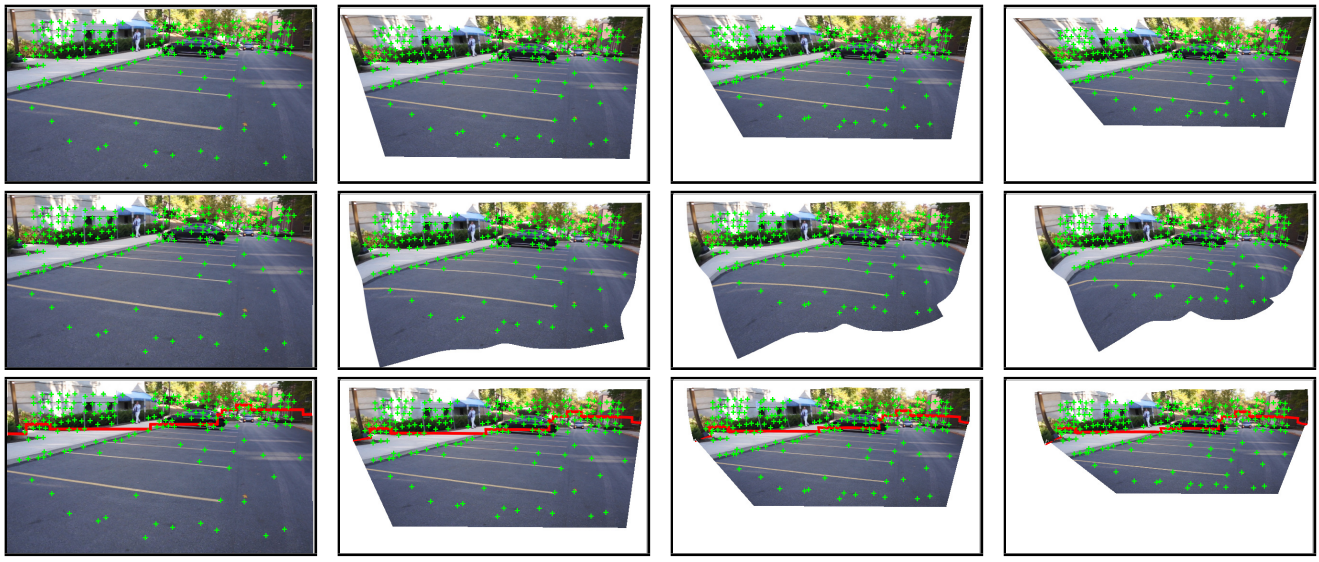


Figure 1. **Effects of various warping methods.** Each row shows a sequences of warps of a single input frame created by pulling the camera away from its original location. **First row:** Warping based on 2D transformation (e.g., homography) is too rigid to handle general motion and structures, resulting in large distortions in non-planar regions (e.g., buildings). **Second row:** Content-preserving warping preserves the non-planar structures well, but yields increasingly visible distortion in the textureless regions (i.e., the ground) where features are rare. **Third row:** Our plane-based warping is able to produce visually pleasing results by combining the strengths of both methods. Red line represents the boundary of planar and non-planar regions obtained by our video segmentation algorithm.

Therefore, our goal is to develop a novel 3D stabilization method that can explicitly take advantage of the presence of (relatively large) planar surfaces in the scene. To this end, we propose to automatically detect large planes in the scene, and partition each frame into regions associated with each plane, as well as regions that are "non-planar". Note that, since our ultimate goal is to improve the stabilization system and produce jitter-free videos, it is crucial for our segmentation algorithm to process the entire video in a short period of time, and obtain results which can be seamlessly integrated into the stabilization pipeline. To achieve this goal, we develop a novel algorithm which directly works on the same uniform grid mesh that is employed by CPW, and only uses geometric cues for fast processing. This is contrary to the existing piecewise planar scene segmentation algorithms, which operate at the per-pixel level and rely on multiple low-level and high-level photometric cues. These methods are generally too slow for stabilization purposes, taking hours to process a video with a few hundred frames. We demonstrate that our algorithm is capable of processing the entire video in about 30 seconds, and obtaining results that are sufficient for stabilization.

With the segmentation information, our new plane-based warping method computes a single homography for image regions that belong to the same plane, while borrowing the results of CPW for non-planar regions (Figure 1 third row). In this way, we not only seamlessly integrate the information about planar structures of the scene into the stabilization framework, but also provide an unified framework for 2D-3D stabilization. When the scene is dominated by complex non-planar or dynamic structures, our method becomes CPW which is known to work well in such cases, whereas on the other end, if the scene contains a single large plane, it reduces to the robust and efficient 2D method.

1.1. Related Work

In general, depending on the level of scene geometry one recovers, existing video stabilization techniques can be roughly divided into two categories. Methods in the first category [19, 10, 6, 15, 13] aim to estimate a single 2D transformation between each pair of frames. Stabilization is then obtained by smoothing the parameters of 2D transformations followed by synthesizing a new video using the smoothed parameters. It is well known that 2D stabilization can only achieve limited smoothing before introducing noticeable artifacts to the output video. Several ideas have been examined in recent years to alleviate this problem, including interpolating the homography matrices in a transformed space [10], considering user's capturing intention [6], directly smoothing a set of robust feature trajectories [15], and designing an ℓ_1-optimal camera path [13].

In order to fully handle general scene structure and camera motion, 3D stabilization methods [5, 7, 3, 16] attempt to recover true camera motion and scene structures via structure from motion (SFM) systems. Stabilization is subsequently done by smoothing the camera path in 3D and synthesizing a new video based on the smoothed path. To avoid the dependency on structure from motion techniques, [17] directly smoothes the 2D feature trajectories based on the observation that they approximately lie in a low-

dimensional subspace over any short period of time. Alternatively, [11] resorts to epipolar point transfer, which only requires projective reconstruction. However, all these methods except [11] solely rely on features that allow reliable tracking, and hence suffer from the presence of large textureless regions. In [11], epipolar constraints are used to search for additional matches along edges. But this approach is very sensitive to noise, and does not work if there is no strong edge in the scene. Recently, [18] proposed to use additional depth sensors to compensate for the lack of feature tracks, but access to depth data is unrealistic for the vast majority of amateur videos.

The problem of segmenting video into motion layers that admit parametric transformation models was first studied in [25], and remains an active research topic in computer vision today. Since its goal is to obtain simultaneous motion estimation and segmentation, it typically involves iterative schemes which are prone to local minima. Given camera motion and 3D point cloud, early works on piecewise-planar scene segmentation from multiple images [1, 26] are based on line grouping and plane sweeping, whose complexity is prohibitive beyond a few images. More recently, [2] and [24] both combine the idea of random sampling consensus (RANSAC) with photometric consistency check to obtain piecewise planar scene models. However, the experiment results in both papers only involve simple examples with little non-planar structure. In addition, their computational complexity is still too high for our purpose. For example, it is reported in [24] that it takes 14 hours to process a sequence consisting of 380 frames. Finally, planes extracted from 3D point clouds or depth maps have been recently explored to improve the performance of multi-view stereo (MVS) systems [21, 8, 9, 20]. But these methods are again too slow for more than a few images. In summary, none of the existing methods meets our goal of obtaining satisfactory segmentation results within a few seconds for long video sequences.

2. Overview of the Content-Preserving Warping Technique

Since our method is built upon the content-preserving warping (CPW) technique introduced in [16], in this section we give a brief review of it.

Generally speaking, CPW is an image warping technique specifically designed for 3D stabilization, which aims to deform an input frame according to a set of 2D sparse displacement constraints induced by the 3D viewpoint change, while minimizing the distortion of local shape and salient image content. In particular, it takes two sets of corresponding 2D points as input – $\hat{P}$ in the input frame, and P in the output frame – and create a dense warp guided by the displacements from $\hat{P}$ to P. For 3D stabilization, $\hat{P}$ and P are obtained by projecting the reconstructed 3D points into input and output (stabilized) cameras, respectively.

To create the dense warp, CPW first divides the original video frame $\hat{I}$ into an $m \times n$ uniform grid mesh, represented by a set of N vertices $\hat{V} = \{\hat{\boldsymbol{v}}_q\}_{q=1}^N$. Then, it estimates a warped version of the mesh, denoted by $V = \{\boldsymbol{v}_q\}_{q=1}^N$, for the output frame by minimizing the following objective function:

$$E(V) = E_d(V) + \alpha E_s(V), \tag{1}$$

where α is a scalar weight between the data term $E_d(V)$ and smoothness term $E_s(V)$.

Data term. The data term penalizes the difference in the output frame between the projected location of each point P_t and the location suggested by the estimated mesh V. For each point $\hat{P}_t$ in the input frame, a bilinear interpolation of the four corners of the enclosing grid cell, denoted by $\hat{V}_t$, is first computed so that $\hat{P}_t = w_t^T \hat{V}_t$. Here, the vector w_t contains the four coefficients that sum to 1. Then, the data term is defined as:

$$E_d(V) = \sum_t \|w_t^T V_t - P_t\|^2. \tag{2}$$

Smoothness Term. The smoothness term measures the deviation of the estimated 2D transformation of each grid cell from a *similarity transformation*. This is inspired by the work [14], which suggests that warps resembling a similarity transformation can effectively avoid noticeable distortions of image content due to shearing and non-uniform scaling, and hence should be preferred as long as the viewpoint change is not too large, which is indeed the case in video stabilization. [14] further shows that this constraint can be written in the form of every three vertices that form a triangle in a grid cell. Specifically, let $(\hat{V}_1^\Delta, \hat{V}_2^\Delta, \hat{V}_3^\Delta)$ and $(V_1^\Delta, V_2^\Delta, V_3^\Delta)$ denote the vertices of any triangle Δ in the input and output grid mesh, respectively. Then, its deviation from a similarity transformation can be written as

$$\boldsymbol{e}_s(\Delta) = \|V_1^\Delta - (V_2^\Delta + a_\Delta(V_3^\Delta - V_2^\Delta) + b_\Delta R_{90}(V_3^\Delta - V_2^\Delta))\|^2, \tag{3}$$

where a_Δ, b_Δ satisfy

$$\hat{V}_1^\Delta = \hat{V}_2^\Delta + a_\Delta(\hat{V}_3^\Delta - \hat{V}_2^\Delta) + b_\Delta R_{90}(\hat{V}_3^\Delta - \hat{V}_2^\Delta), \tag{4}$$

and $R_{90} = [0\ 1; -1\ 0]$ is a 2D rotation matrix.

Finally, the smoothness term $E_s(V)$ is the sum of $\boldsymbol{e}_s(\Delta)$ over all eight triangles of each vertex:

$$E_s(V) = \sum_\Delta \boldsymbol{e}_s(\Delta). \tag{5}$$

Since minimizing the energy $E(V)$ is a linear least-squares problem in the set of unknown V, it can be solved efficiently by any standard linear system solver. The output frame is then generated using standard texture mapping algorithm according to V.

Finally we note that, according to the above discussion, the warp obtained by CPW tends to be close to a *similarity transformation*, especially in regions where features are rare or non-existing. However, similarity transformation cannot faithfully represent the projective effects of the scene, and hence may cause serious wobble effects in the stabilized videos. Next, we show how this problem can be properly addressed by incorporating information about scene planes.

3. Fast Piecewise Planar and Non-Planar Scene Segmentation for Videos

In this section, we propose a fast two-step approach to automatically segment each video frame into piecewise planar and non-planar regions. First, we detect scene planes from 3D point cloud obtained by structure from motion using a robust multiple structure estimation algorithm called J-Linkage [23]. Second, we describe a novel video segmentation algorithm, which classifies each grid cell in the CPW framework into $K+1$ classes – one for each of the K detected planes, plus a "non-planar" class. For this problem, we lay out a MRF formulation for the entire sequence to simultaneously take into account the spatial coherence between neighboring cells within each frame, and improve the segmentation consistency across different frames. We now describe these two steps in detail.

3.1. Multiple Plane Detection

Since real scenes often contain multiple planes as well as non-planar structures, we adopt a robust multiple structure estimation method called J-Linkage [23] to detect planes from 3D point cloud. Similar to the popular RANSAC technique, this method is based on sampling consensus. Meanwhile, it has been shown in [23] that J-Linkage substantially outperforms other variants of RANSAC for multiple structure detection, such as sequential RANSAC and multi-RANSAC [29], in many real applications including 3D plane fitting.

Basically, J-Linkage works in the following way. It first generates a large number (typically a few thousands) of putative models by random sampling. Next, for each data point, a preference set (PS) of models is computed, which include all the models to which the distance from that data point is less than a threshold ϵ. J-Linkage then uses a bottom-up scheme to iteratively group data points that have similar PS. Here, the PS of a cluster is defined as the intersection of the preference sets of its members. Specifically, in each iteration, J-Linkage computes the Jaccard distance between any two clusters A and B:

$$d_J(A,B) = \frac{|A \bigcup B| - |A \bigcap B|}{|A \bigcup B|}, \tag{6}$$

and merge the two clusters with the smallest distance. As in RANSAC, the only free parameter of J-Linkage is the consensus threshold ϵ, which is set to 10 in our experiments. Also, since our goal is to detect large scene planes, we only keep those clusters with a support size larger than one sixth of the total number of points.

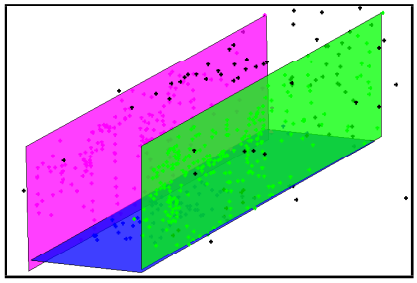


Figure 2. Three planes are detected by J-Linkage [23] on the video shown in FIgure 3.

Figure 2 shows the result of applying J-Linkage to the 3D point cloud for an indoor video sequence taken by a person walking down the corridor with a hand-held camera (see Figure 3 for some input frames). In this example, three planes are detected, namely the ground and two side-walls. Although J-Linkage fails to detect the other two planes, namely the ceiling and front door, due to their small support sizes, we still consider the result successful as these two planes only occupy a very small portion of the video frames.

3.2. A Markov Random Field Formulation for Video Segmentation

Once a set of dominant planes is detected, the next step is to perform piecewise planar and non-planar segmentation for each input frame. To take both spatial and temporal consistency into consideration, we define a Markov random field for the entire sequence. For each frame, $I_f, f = 1, \ldots, F$, we divide it into a 64×36 uniform grid mesh and build a graph $\mathcal{G}_f = (\mathcal{V}_f, \mathcal{E}_f)$ on it. Each vertex $p \in \mathcal{V}_f$ is a cell of the mesh, while the edges, $\mathcal{E}_f$, denote the neighboring relationship between cells. Then, the graphs $\{\mathcal{G}_f\}$ from all frames are merged into a large graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, by adding edges between the two cells at the same spatial location in two consecutive frames.

Given a set of K 3D planes, our goal is to assign a unique label l_i to each vertex $p_i \in \mathcal{V}$. That is, $l_i = k, k = 1, 2, \ldots, K$ if p_i belongs to the k-th plane, and $l_i = 0$ if p_i lies on any non-planar surface. The solution $L = \{l_i\}$ can be obtained by minimizing the energy function

$$E(L) = \sum_{p_i \in \mathcal{V}} \Psi_i(l_i) + \sum_{e_{ij} \in \mathcal{E}} \Psi_{ij}(l_i, l_j), \tag{7}$$

which involves a unary data function Ψ_i and a pairwise smoothness function Ψ_{ij}. In this paper, we adopt the popular multi-label graph-cut algorithm [4] to minimize $E(L)$. It is guaranteed to find a solution that is within a constant fac-

Figure 3. **Piecewise planar and non-planar scene segmentation. Top Row:** Results of classifying each 3D point (represented by its image in each frame) into the $K+1$ classes based on the proposed distance measure $\|\boldsymbol{x}-\boldsymbol{x}_k^*\|_2$. Each color represents a class, with black circles corresponding to the points labeled as "non-planar", i.e., $\|\boldsymbol{x}-\boldsymbol{x}_k^*\|_2 > \beta, \forall k$. **Bottom Row:** Segmentation results obtained by the proposed method.

tor of the global minimum, and has been shown to produce satisfactory results in many vision tasks [22].

Data term. For a vertex in the f-th frame, $p_i \in \mathcal{V}_f$, the function Ψ_i is defined as follows. Let $\mathcal{X}_i$ be the set of 3D points whose images in the f-th frame lie in the cell corresponding to p_i. Then, for each point $X \in \mathcal{X}_i$, we compute its projection to the k-th plane, denoted as X_k^*. We further denote $\boldsymbol{x}$ and $\boldsymbol{x}_k^*$ as the images of X and X_k^* in the f-th frame, respectively. The function Ψ_i then measures the image distance between $\boldsymbol{x}$ and $\boldsymbol{x}_k^*$:

$$\Psi_i(l_i) = \begin{cases} \sum_{X\in\mathcal{X}_i} \min\{\|\boldsymbol{x}-\boldsymbol{x}_k^*\|_2, d_{\max}\}, & \text{if } l_i = k > 0 \\ \beta|\mathcal{X}_i|, & \text{if } l_i = 0 \end{cases} \tag{8}$$

where β is a penalty assigned to each point $X \in \mathcal{X}_i$ if the corresponding cell is classified as "non-planar". Note that, geometrically, β can be viewed as a threshold that determines how far the images of X and its projection onto the k-th plane X_k^* may be before X is considered not belonging to that plane. On one hand, by comparing the image distance instead of the distance in 3D, β sets a uniform threshold across all 3D points which is irrelevant to their individual uncertainty in the 3D space. On the other hand, the value of β should depend on the overall accuracy of structure from motion, and is chosen to be 1.5 times the size of each cell in our paper. For example, for a 640×360 input frame, we have $\beta = 15$. In addition, the distance measure has been truncated in Eq. (8) to $d_{\max}$ in order to prevent it from being dominated by a small number of poorly reconstructed 3D points. We fix $d_{\max} = 2\beta$ for all the experiments.

In Figure 3 (first row) we show the results of classifying each 3D point (represented by its image in each frame) into the $K+1$ classes based on the proposed distance measure for an indoor scene. As one can see, the classification results indeed give us very strong cues for segmentation.

Smoothness term. For each edge $e_{ij} \in \mathcal{E}$ in the same image I_f, the smoothness function is defined as:

$$\Psi_{ij}(l_i, l_j) = \delta(l_i, l_j) \cdot g(i, j), \tag{9}$$

where $\delta(l_i, l_j)$ is the indicator function which takes value 0 if $l_i = l_j$, and 1 otherwise.

The function $g(i, j)$ is designed to improve the estimation of label boundaries by imposing geometric constraints derived from multiple planes in the scene. First, for each pair of planes in the scene (if one exists), we compute the 2D intersection line L between them in each frame I_f. Then, we find all pairs of neighboring cells (p_i, p_j) in I_f where the centers of p_i and p_j lie on different sides of L, and accumulate all such pairs for all intersection lines in a set $\mathcal{E}_f^L$. Finally, the function $g(i, j)$ is defined as

$$g(i,j) = \begin{cases} \lambda_1, & \text{if } (p_i, p_j) \notin \mathcal{E}_f^L \\ \lambda_2, & \text{otherwise} \end{cases} \tag{10}$$

For edges e_{ij} across two frames, the smooth cost is defined as

$$\Psi_{ij}(l_i, l_j) = \lambda_3 \delta(l_i, l_j). \tag{11}$$

In this paper, λ_1, λ_2 and λ_3 are empirically set to $\lambda_1 = \lambda_3 = 10$, $\lambda_2 = 2$ for all experiments.

In Figure 3 and Figure 4, we show some representative results of the proposed method. As one can see, our segmentation algorithm correctly identifies the large planar regions in a variety of indoor and outdoor scenes. However, since our algorithm purely relies on geometric cues, the label boundaries estimated by it may not be very accurate. This is mainly due to the uncertainty in 3D reconstruction, which decides the smallest possible threshold β one can choose to distinguish points on a plane from others. In addition, the facts that our algorithm only operates on a coarse spatial grid, and that feature points are not evenly distributed in the images, could also contribute to the errors. Nevertheless, we find that these errors have little effect on the final stabilization results, since the shifts in viewpoint are usually small for video stabilization.

In terms of speed, for a typical sequence such as the one shown in Figure 3 with 250 frames, the plane detection[1] and

[1]We use the Matlab code downloaded from the J-Linkage website: http://www.diegm.uniud.it/fusiello/demo/jlk/.

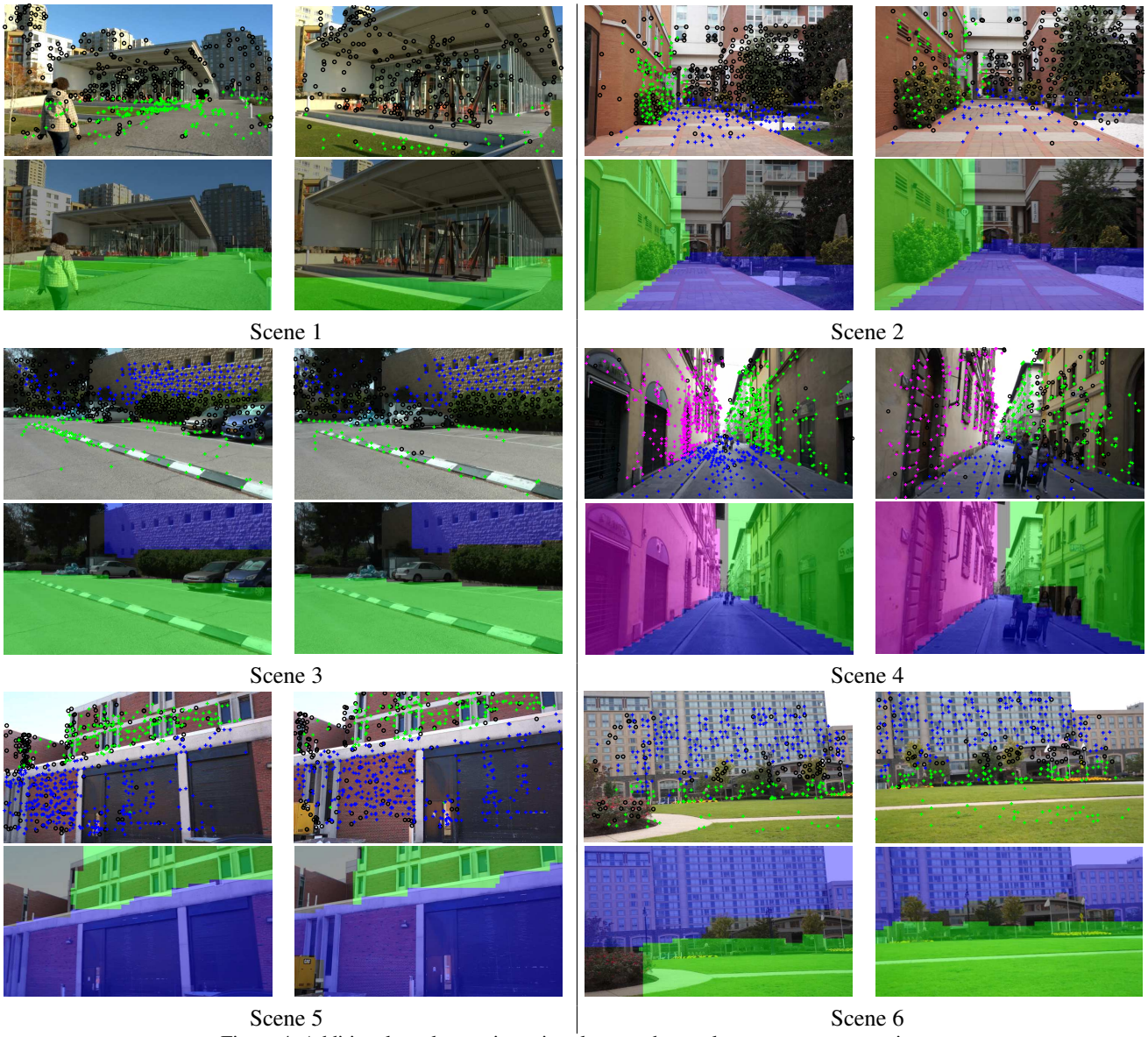


Figure 4. Additional results on piecewise planar and non-planar scene segmentation.

piecewise planar scene segmentation algorithms take about 10 and 15 seconds on a desktop PC with 3.40GHz CPU and 12GB memory, respectively.

4. Plane-Based Stabilization

As we have already discussed, this paper aims at leveraging the flexibility of CPW and the structural regularities (i.e., planar surfaces) of the scene to produce high-quality stabilization results, especially in the cases where CPW performs poorly because of large textureless regions. In this section, we describe our plane-based stabilization algorithm in detail.

Like other 3D stabilization methods, our plane-based method first applies structure from motion to recover the original camera motion and sparse 3D point cloud. In this paper, we use ACTS [27], a publicly available structure from motion system. To generate the stabilized camera path, we apply Gaussian filter to the original camera parameters. Since a camera can be modeled by a rotation matrix $R \in SO(3)$ and its center $C \in \mathbb{R}^3$, we apply a Gaussian filter to these two components separately. Note that, since the space of rotation matrices is not Euclidean, the filtering of the rotational component is done in a locally linearized space at each timestamp in the same way described in [16].

For novel view synthesis, we also follow the same idea of [16] by processing one input frame at a time to avoid ghosting effect caused by the moving objects. Each input frame is divided into a 64×36 grid mesh $\hat{V} = \{\hat{\boldsymbol{v}}_q\}_{q=1}^{N}$ and the content-preserving warp is then computed. We de-

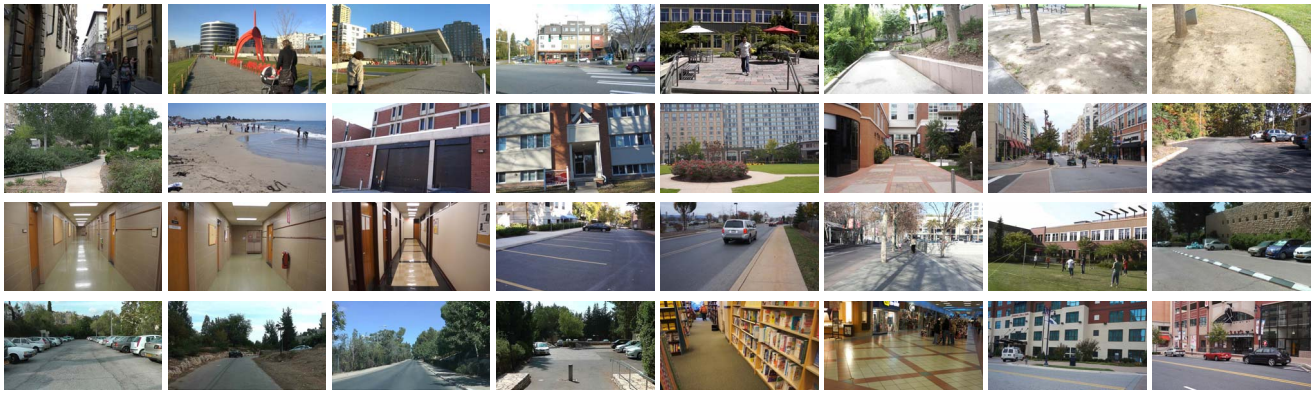


Figure 5. Snapshots of the videos used for evaluation.

note the output mesh by $V^0 = \{\boldsymbol{v}_q^0\}$. To incorporate information about the piecewise planar scene structures into stabilization, we give a label, l_q, to each vertex of the mesh according to the labels of its surrounding cells. For any vertex that lies on the segmentation boundary (hence the surrounding cells have more than one labels), we simply assign the smallest label to it. Based on the labels, a new mesh $V = \{\boldsymbol{v}_q\}$ is computed:

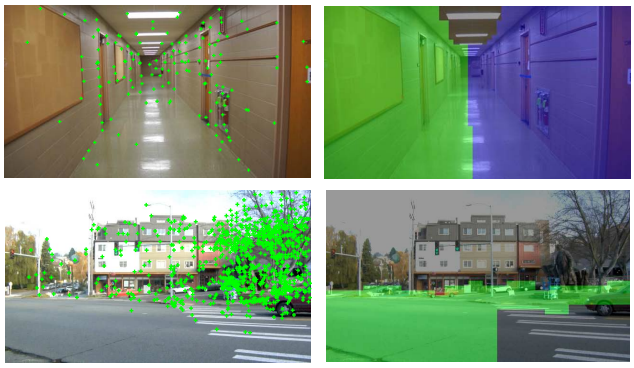
$$\boldsymbol{v}_q = \begin{cases} H_k \hat{\boldsymbol{v}}_q & \text{if } l_q = k, k = 1, \ldots, K \\ \boldsymbol{v}_q^0, & \text{if } l_q = 0 \end{cases} \tag{12}$$

where H_k is the homography induced by the k-th plane between the input and output frames. The output frame is then obtained using standard texture mapping algorithms.

5. Experiments

We have tested our algorithm on 32 video sequences (see Figure 5) which consist of one or more large scene planes, including 5 videos that are used in [16] to demonstrate the performance of CPW. These sequences cover a wide range of scenes from both natural and indoor/outdoor man-made environments. Among them, noticeable wobble effects can be seen in 18 results obtained by CPW, due to the lack of feature tracks in large planar regions. Meanwhile, our plane-based method succeeds in 30 of the 32 videos, generating satisfactory stabilization results. We show a number of results in our project website.[2]

Challenging cases. For the other two testing videos shown in Figure 6, our method is not able to completely remove the wobble effects, although it still produces better results than CPW. In the first video, only a very small number of points are reconstructed on the ground, with a large number of outliers due to reflection. Therefore, J-Linkage fails to detect the ground plane in the case. Consequently, our segmentation algorithm incorrectly assigns the ground regions to the planes corresponding to the walls, causing undesirable artifacts in the stabilized video. In the second video, the ground is slightly curved, which confuses our plane detection and segmentation algorithms. As a result, a portion of the ground region is labeled as non-planar, hence the wobble effects remain in the output video.



(a) Input frame with points (b) Segmentation result

Figure 6. **Challenging cases for our method. Top row:** In this case, only a very small number of points are detected on the ground. Some of them actually correspond to the reflection. **Bottom row:** In this case, the ground is slightly curved.

In fact, both cases reveal the dependency of our method's performance on a few free parameters in the plane detection and segmentation algorithms, for which a set of fixed values is certainly not enough to handle all cases. Nevertheless, we have shown in this paper that, by exploiting scene structures such as the planar surfaces, our method significantly outperforms CPW in many challenging cases.

6. Conclusion, Limitations, and Future Work

In this paper we have described a novel method for video stabilization, which outperforms the state-of-the-art methods by taking advantage of the presence of large planes in the scene. Our method is built upon the newly proposed CPW framework, but is able to avoid the difficulties of CPW in handling large textureless regions. In particular, we have proposed an efficient Markov random field formulation to segment each video frame into piecewise planar and non-

[2] http://perception.csl.illinois.edu/stabilization/

planar regions. This level of scene understanding is shown to be ideal for generating high-quality jitter-free videos in a variety of practical scenarios.

Like CPW and many other 3D methods, our algorithm relies on structure from motion to get accurate information about the 3D scene structures and camera motions. For this reason, all the videos tested in this paper are chosen to be friendly to SFM. Also, we do not address other common issues in video stabilization, including the smaller field of view, motion blur [19], and rolling shuttle effects [12].

Another bottleneck of our method is the plane detection part. Currently we use the robust model estimation package J-Linkage, but it leaves to the user to decide the minimum number of inliers for a valid model; hence it may fail when the number of reconstructed 3D points on the plane is extremely small. A different direction would be combining plane detection with 3D reconstruction, as studied in [28].

7. Acknowledgement

This work is supported in part by Adobe Systems Incorporated. Zihan Zhou and Yi Ma gratefully acknowledge support by ONR N00014-09-1-0230, NSF CCF 09-64215, and NSF IIS 11-16012.

References

[1] C. Baillard and A. Zisserman. Automatic reconstruction of piecewise planar models from multiple views. In *CVPR*, pages 2559–2565, 1999. 3

[2] A. Bartoli. A random sampling strategy for piecewise planar scene segmentation. *Computer Vision and Image Understanding*, 105(1):42–59, 2007. 3

[3] P. Bhat, C. L. Zitnick, N. Snavely, A. Agarwala, M. Agrawala, M. F. Cohen, B. Curless, and S. B. Kang. Using photographs to enhance videos of a static scene. In *Rendering Techniques*, pages 327–338, 2007. 1, 2

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001. 4

[5] C. Buehler, M. Bosse, and L. McMillan. Non-metric image-based rendering for video stabilization. In *CVPR (2)*, pages 609–614, 2001. 1, 2

[6] B.-Y. Chen, K.-Y. Lee, W.-T. Huang, and J.-S. Lin. Capturing intention-based full-frame video stabilization. *Comput. Graph. Forum*, 27(7):1805–1814, 2008. 1, 2

[7] A. W. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. *International Journal of Computer Vision*, 63(2):141–151, 2005. 1, 2

[8] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Manhattan-world stereo. In *CVPR*, pages 1422–1429, 2009. 3

[9] D. Gallup, J.-M. Frahm, and M. Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *CVPR*, pages 1418–1425, 2010. 3

[10] M. Gleicher and F. Liu. Re-cinematography: Improving the camerawork of casual video. *TOMCCAP*, 5(1), 2008. 1, 2

[11] A. Goldstein and R. Fattal. Video stabilization using epipolar geometry. *ACM Trans. Graph.*, 32(5), 2012. 3

[12] M. Grundmann, V. Kwatra, D. Castro, and I. Essa. Effective calibration free rolling shutter removal. *IEEE ICCP*, 2012. 8

[13] M. Grundmann, V. Kwatra, and I. A. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *CVPR*, pages 225–232, 2011. 1, 2

[14] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 24(3):1134–1141, 2005. 3

[15] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen, and M. Ouhyoung. Video stabilization using robust feature trajectories. In *ICCV*, pages 1397–1404, 2009. 1, 2

[16] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3d video stabilization. *ACM Trans. Graph.*, 28(3), 2009. 1, 2, 3, 6, 7

[17] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala. Subspace video stabilization. *ACM Trans. Graph.*, 30(1):4, 2011. 2

[18] S. Liu, Y. Wang, L. Yuan, J. Bu, P. Tan, and J. Sun. Video stabilization with a depth camera. In *CVPR*, pages 89–95, 2012. 3

[19] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum. Full-frame video stabilization with motion inpainting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1150–1163, 2006. 1, 2, 8

[20] B. Micusík and J. Kosecká. Multi-view superpixel stereo in urban environments. *International Journal of Computer Vision*, 89(1):106–119, 2010. 3

[21] S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *ICCV*, pages 1881–1888, 2009. 3

[22] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. F. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6):1068–1080, 2008. 5

[23] R. Toldo and A. Fusiello. Robust multiple structures estimation with j-linkage. In *ECCV (1)*, pages 537–547, 2008. 4

[24] R. Toldo and A. Fusiello. Photo-consistent planar patches from unstructured cloud of points. In *ECCV (5)*, pages 589–602, 2010. 3

[25] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, 1994. 3

[26] T. Werner and A. Zisserman. New techniques for automated architectural reconstruction from photographs. In *ECCV (2)*, pages 541–555, 2002. 3

[27] G. Zhang, X. Qin, W. Hua, T.-T. Wong, P.-A. Heng, and H. Bao. Robust metric reconstruction from challenging video sequences. In *CVPR*, 2007. 6

[28] Z. Zhou, H. Jin, and Y. Ma. Robust plane-based structure from motion. In *CVPR*, pages 1482–1489, 2012. 8

[29] M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multi-ransac algorithm and its application to detect planar homographies. In *ICIP (3)*, pages 153–156, 2005. 4