Augmenting TV Show via Uncalibrated Camera Small Motion Tracking in Dynamic Scene

Yizhen Lao*
Hunan University
Changsha, China
lyz91822@gmail.com

Jie Yang*
MGTV
Changsha, China
yangjie@mgtv.com

Xinying Wang
MGTV
Changsha, China
xinying@mgtv.com

Jianxin Lin
Hunan University
Changsha, China
linjianxin@hnu.edu.cn

Yu Cao
Hunan University
Changsha, China
lcaoyuyu@hnu.edu.cn

Shien Song†
MGTV
Changsha, China
shien@mgtv.com

ABSTRACT

To augment the TV show in post-production, we propose a novel solution to uncalibrated camera small motion tracking in a dynamic scene that simultaneously reconstructs the sparse 3D scene and computes camera poses and focal lengths of each frame. The critical elements of our approach are a robust image feature tracking strategy in dynamic scenes followed by automatic local-window frames slicing, local and global bundle adjustment optimization initialized by a homography-based uncalibrated relative rotation solver. The proposed method allows us to add the virtual objects (elements) into the reconstructed 3D scene, then composite them back into the original shot while perfectly matched perspective and appear seamless.

The evaluation of a large variety of real TV show sequences demonstrates the merits of our method against state-of-the-art works and commercial software products.

CCS CONCEPTS

• **Computing methodologies** → **Virtual reality**; **Reconstruction**.

KEYWORDS

TV show, Camera tracking, Structure-from-Motion, Augmented reality

ACM Reference Format:
Yizhen Lao, Jie Yang, Xinying Wang, Jianxin Lin, Yu Cao, and Shien Song. 2021. Augmenting TV Show via Uncalibrated Camera Small Motion Tracking in Dynamic Scene. In *Proceedings of the 29th ACM Int'l Conference on Multimedia (MM '21), Oct. 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475543

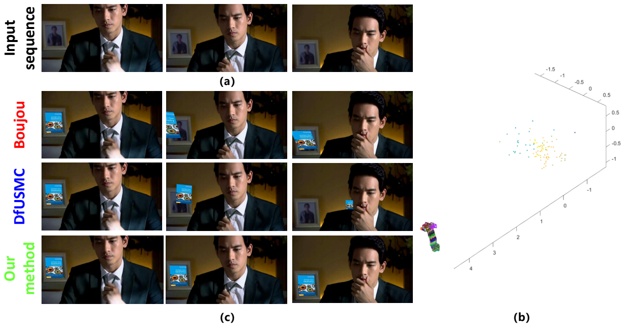


Figure 1: (a) An uncalibrated sequence of TV show shot with small camera motion in dynamic scene (large area of moving objects). (b) The proposed method in this paper can simultaneously reconstruct the sparse 3D scene and computes camera poses, focal lengths of each frame. (c) Augmentation results (reproject a virtual book cover back to the picture frame on the cabinet) produced by commercial camera tracking software Boujou [1] (top), state-of-the-art uncalibrated small motion tracking approach [8] (middle) and our method (bottom).

1 INTRODUCTION

Augmenting the sequence of "shots" by producing visual effects (VFX) is a well-known and vital task in filmmaking. The main idea is to enhance the camera captured the real world by accurate 3D registration of virtual and real objects, which is close to augmented reality (AR). An accurate camera tracking is essential to convincingly composite Computer Generated (CG) images onto live-action footage by ensuring that the virtual camera in a render matches the movement of the actual camera. Thus, camera tracking (or matchmove) is a crucial task and one of the first to perform in the VFX pipeline.

Recently, augmenting TV shows such as variety shows, series, sport event broadcasting, and animation becomes a new but hot topic for TV station [9, 16, 37]. In the field of movie VFX production or mobile phone AR, camera pre-calibration and multiple sensors information are available without mention that the camera baseline (translation) is usually significant. In contrast, the shot of a TV show

*Both authors contributed equally to this research.
†Corresponding author.

MM '21, October 20–24, 2021, Virtual Event, China.
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475543

is common with (i) uncalibrated camera with dramatical lens zoom, (ii) highly dynamic scene, and (iii) extreme narrow-baseline motion. These differences make the classical camera tracking approaches fail when using a TV show as input. Therefore, the task of TV show augmentation becomes an open but challenging problem.

This paper focuses on augmenting TV show shots by using a novel uncalibrated camera small motion tracking in the dynamic scene. Note that despite multiple ways to augment input shots, throughout this work, we focus on one of the most common augmentation tasks, namely, inserting virtual objects into the original shot while perfectly matched perspective and appear seamless.

1.1 Related Work and Motivations

Inserting virtual objects into a real shot requires the object's location must appear seamlessly as the camera motion. The most common solution is camera tracking with estimations of all internal and external parameters (focal lengths and camera poses etc.) with an adequate degree of stability. The existing works can be divided into three major categories:

(1) Additional Information Assistance. BBC proposed an on-set pre-visualization system called Free-D [28] by sticking barcoded markers on the studio ceiling. Schweighofer et al. use multiple cameras rig to recover the camera motion with pre-calibration [23]. Similarly, Yu and Kim [34] proposed using an auxiliary camera that faces the ground to estimate the motion of the main shooting camera and 3D points in a dynamic scene. Besides, an internal measurement unit (IMU) that can assist camera tracking has been reported in [11]. However, all these approaches fail in handling TV show shots since the post-produce is based on post-edited sequences without calibration information such as focal length of each frame or measurements from auxiliary cameras and IMU.

(2) Structure-from-Motion The most popular way of recover the motion of a moving camera is using structure-from-motion (SfM) algorithms [36]. SfM is an active field of computer vision and robotic communities, with a large number of works on accurate 2D feature tracking [6], visual 3D scene reconstruction [36] and visual simultaneous localization and mapping (SLAM) [27]. The majority of SfM assume the multiple views are well pre-calibrated with pre-knowledge about the focal length and lens distortion [18, 30]. This dependency on pre-calibration defeats their feasibilities to the TV show. Recently, some uncalibrated SfM methods are presented such as VisualSfM [31], COLMAP [22] and OpenMVG [17]. Nonetheless, note that all the classical SfM methods are based on epipolar geometry, which first estimates the fundamental matrix and follows its decomposition into the relative pose. This pipeline requires a wide baseline to ensure the numerical stability of epipolar solving [36]. Unfortunately, TV show shots are commonly under small camera motion, leading to an extraordinary narrow baseline and defeating conventional SfM solutions.

3) Uncalibrated AR. Note that uncalibrated AR solution solve similar task in this paper such as [14, 24, 26]. However, all these approaches assume wide translation and static scenes while the TV show shots are with highly dynamic scenes and small motion.

4) Commercial Matchmove Software. In the movie post-production stage, camera tracking can be achieved by using commercial matchmove software [7] such as Boujou [1], ACTS [35] and Camera-Tracker tool [5] in Adobe After Effect. However, even such commercial software fails to track the camera robustly in a dynamic scene where a moving foreground object such as a real actor occupies a large part of the background that frequently occurs in the TV show.

5) Structure-from-Small-Motion (SfSM). Recently, 3D reconstruction from small motion calibrated sequence was first studied in [33] and then extends to rolling shutter case in [12]. One of the most closed existing works to this paper is the depth from uncalibrated small motion clip (DfUSMC) [8], which is feasible to handle the uncalibrated small camera motion in the TV show. However, note that all [8, 12, 33] assume a rigid scene without considering the captured moving objects, which does not hold in a TV show shot.

Motivations. From the discussion above, we found out that to augment TV show robustly, an uncalibrated camera small motion tracking method in highly dynamic scenes is still absent in current literature. Such a solution could benefit the post-production and post-edit of TV show or even the social sharing short-form videos.

1.2 Contributions

To augment the TV show in post-production, we propose a novel solution to tackle uncalibrated camera small motion tracking in dynamic scenes, enabling reconstructing the sparse 3D scene simultaneously and computing camera poses, focal lengths of each frame. The key elements of our approach are a robust image feature tracking strategy in dynamic scenes followed by automatic local-window frames slicing, local and global bundle adjustment optimization initialized by a homography-based uncalibrated relative rotation solver. The proposed method allows users to add the virtual objects (elements) into the reconstructed 3D scene, then composite them back into the original shot while perfectly matched perspective and appear seamless. We evaluate the proposed solution on an extensive data set of 1000 real shots consisted of variety shows, series, and animations collected from the industry. Our main contributions can be summarized as follows:

- We present a novel uncalibrated camera small motion tracking approach for dynamic scenes called **Dynamic-UCSMT**. It can compute the 3D scene, camera pose, and focal length of each frame precisely by giving an uncalibrated TV show shot with even small motion.
- We propose a completed pipeline to augment the TV show shot in the post-production step by using Dynamic-UCSMT. We collected a large dataset of real TV show shots, and extensive evaluations demonstrate the superiority of our method to existing works and commercial software. The dataset will be publicly available in the future as we hope interest to the research community.

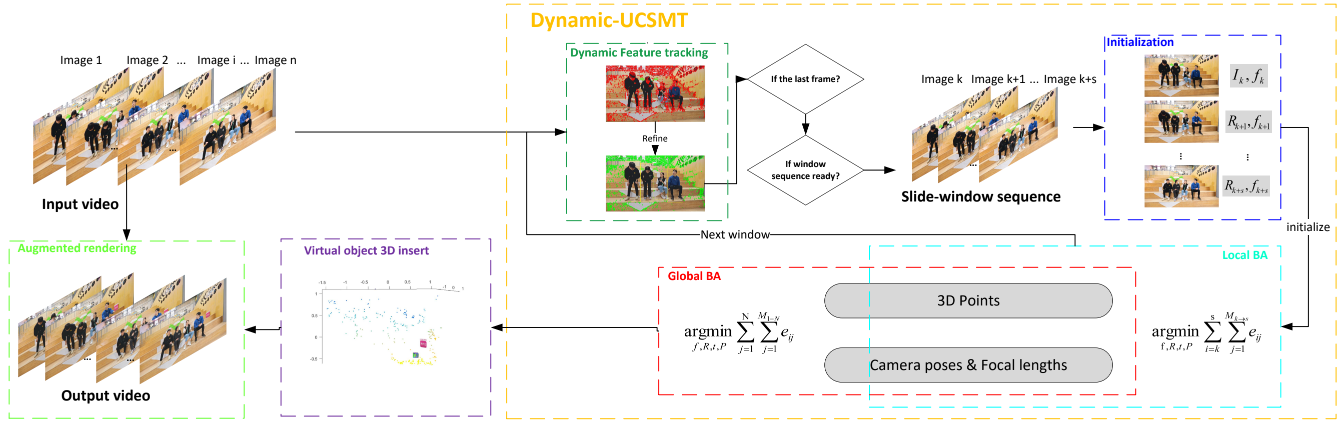


Image 1
Image 2 ... Image i ... Image n
Input video
Dynamic-UCSMT
Dynamic Feature tracking
Refine
If the last frame?
If window sequence ready?
Image k Image k+1 ... Image k+s
Slide-window sequence
Initialization
I_k, f_k
R_{k+1}, f_{k+1}
R_{k+s}, f_{k+s}
initialize
Next window
Local BA
Global BA
3D Points
Camera poses & Focal lengths
$\underset{f,R,t,P}{\operatorname{argmin}} \sum_{j=1}^{N} \sum_{j=1}^{M_{1-N}} e_{ij}$
$\underset{f,R,t,P}{\operatorname{argmin}} \sum_{i=k}^{s} \sum_{j=1}^{M_{k\to s}} e_{ij}$
Virtual object 3D insert
Augmented rendering
Output video

Figure 2: TV show video augmentation pipeline: (1) We first perform a static feature tracking followed by automatic local-window frames slicing (section. 2.2.2). Then a local BA to recover the 3D points and camera motion within local-window frames (section. 2.2.3), which is initialized by an analytical homography-based uncalibrated relative rotation solver (section. 2.2.4). Finally we refine the sparse 3D reconstruction, camera poses and focal lengths of the whole frames with a global BA (section. 2.2.3). (2) In the end, we augment the original video by placing virtual objects into the reconstructed 3D scene (section. 2.3.1) and reprojecting in each frame based on the recovered camera motion (section. 2.3.2).

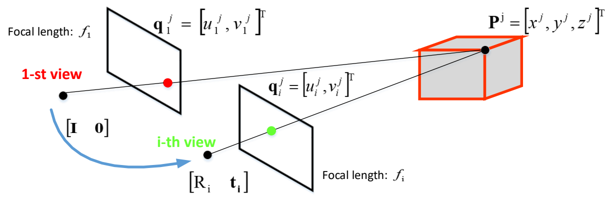


Figure 3: Camera model for small motion (section. 2.2.1). A 3D point $\mathbf{P}^j$ is projected into the 1^{st} and i^{th} cameras with distinct focal lengths f_1 and f_i as 2D points $\mathbf{q}_1^j$ and $\mathbf{q}_i^j$ respectively.

2 OUR SOLUTION

2.1 Overview

Fig. 2 illustrates a conceptual overview of the proposed method, which consists of two main steps: By given the original uncalibrated video, (1) the 3D scene, camera 6D motion, and focal length of each frame is estimated by performing dynamic-UCSMT. (2) Then, we augment the original video by inserting virtual objects into the reconstructed 3D environment and projecting them based on the recovered camera poses and focal lengths.

2.2 Dynamic-UCSMT

2.2.1 ***Camera Model for Small Motion.*** Existing small motion tracking approaches [12, 33] assumes a well-calibrated camera with accurate and constant focal length. However, in TV show shots, the focal length can significantly change within few frames without value recorded, violating the pre-calibrated assumption. Therefore, the camera tracking in augmenting the TV show should self-calibrate the focal length in each frame. Note that different from the camera tracking solutions such as Boujou [1] and [8] recover the focal lengths and lens distortion parameters simultaneously. In this paper, we propose to use the perspective projection model by ignoring lens distortion since the modern camera lenses used in TV shows are developed to have slight distortion according to [2]. This is also verified by the observations in the experiments that the estimated radial distortion parameters from [1, 8] are extremely small, which leads to distortion less than 1 pixel. Thus, a 3D point $\mathbf{P}^j = \left[x^j, y^j, z^j\right]$ is projected into i^{th} camera as a 2D point $\mathbf{q}_i^j = \left[u_i^j, v_i^j\right]$ as follows:

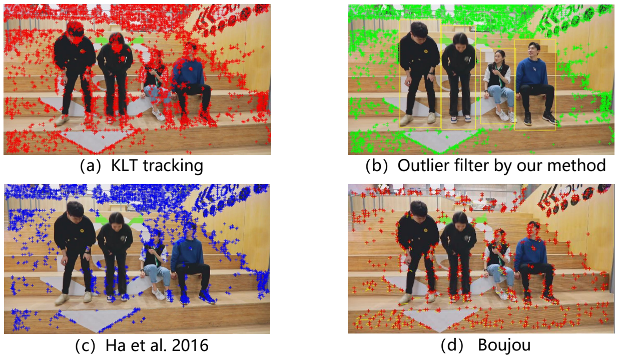
$$s_i^j \left[\mathbf{q}_i^j, 1\right]^\top = \mathbf{K}_i \left[\mathbf{R}_i \quad \mathbf{t}_i\right] \left[\mathbf{P}^j, 1\right]^\top$$
$$\text{with} \quad \mathbf{K}_i = \begin{bmatrix} f_i & 0 & c_x \\ 0 & f_i & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where s_i^j is a scale factor, $\mathbf{K}_i$ is the calibration matrix of i^{th} camera that contains focal length f_i and principal points c_x and c_y, $\mathbf{R}_i$ and $\mathbf{t}_i$ describe the camera pose w.r.t. the world coordinate system.

Since small motion leads to a small baseline and makes the conventional SfM approaches fail, Yu et al. [33] proposes to use small angle approximation for the camera rotation parameterization. This strategy enables reducing the complexity in the bundle adjustment step, which has been validated by [8, 12]. Thus, based on this insight as mentioned above, we parameterize the camera pose as:

$$\mathbf{R}_i = \begin{bmatrix} 1 & -r_i^z & r_i^y \\ r_i^z & 1 & -r_i^x \\ -r_i^y & r_i^x & 1 \end{bmatrix}, \quad \mathbf{t}_i = \begin{bmatrix} t_i^x \\ t_i^y \\ t_i^z \end{bmatrix} \tag{2}$$

where $\mathbf{r}_i = \left[r_i^x, r_i^y, r_i^z\right]^\top$ and $\mathbf{t}_i = \left[t_i^x, t_i^y, t_i^z\right]^\top$ are rotation vector and translation vector of i^{th} camera.


(a) KLT tracking

(b) Outlier filter by our method

(c) Ha et al. 2016

(d) Boujou

Figure 4: (a) Conventional KLT tracking result on an input TV show. (b) Outliers in moving objects filtered by object detection (yellow bounding boxes). (c) Feature tracking results produced by [8]. (d) Feature tracking results produced by [1].

2.2.2 *Feature Tracking and Local-window Slicing in Dynamic Scene*.

The narrow baseline in the TV show scene makes the feature tracking easier, but on the other hand, a large number of dynamic humans captured in the input video causes failure of classical camera tracking methods.

Feature Tracking. In the first frame, we use Harris corner [10] and Kanade-lucas-Tomasi (KLT) tracker [29] to match the features over the sequence. Note that [1, 8] using epipolar geometry with RANSAC loop to filter the dynamic points in moving objects [19]. However, since the human close-up shot is common in the TV show, the majority of the feature matches could be detected on the dynamic objects, which defeats the RANSAC performance [20]. To address this problem, we utilize dynamic object detection to assist the dynamic feature filtering process. Specifically, as shown in Fig. 4(b):

(1) After the KLT tracking, we perform YOLOv3 [21] to detect humans in the first frame and remove the feature matches within the detected bounding boxes.
(2) In the next step, since small translation within the short local-window sequence, we compute the homography between consecutive frames to filter out the remained outliers. Note that with small camera motion, we found out that the homography-based computation achieves significant numerical stability to epipolar-based estimation, which is used in [1, 12].

The extensive experiments (e.g., Fig. 4) in real TV show sequences demonstrate the superiority of our feature tracking method in the dynamic scene.

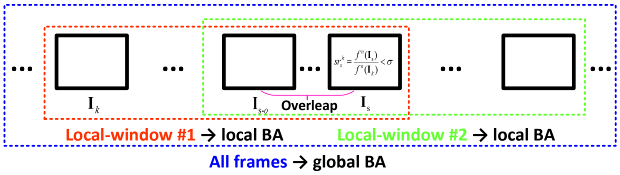

...
I_k
...
I_s-o
...
$sr_s^k = \frac{f^n(\mathbf{I}_s)}{f^n(\mathbf{I}_k)} < \sigma$
Overleap
I_s
...
...
Local-window #1 → local BA
Local-window #2 → local BA
All frames → global BA

Figure 5: Illustration of automatic local-window slicing algorithm.

Local-window Slicing. Based on the observation that the survival rate of detected features is low over the whole sequence raised by TV show shot styles such as bokeh, faster zoom, and reaction shot, and thus affects the final camera tracking quality. We conduct a sliding window to track the camera motion locally tanks to an automatic local-window slicing algorithm. As shown in Fig. 5, We assume the frame $\mathbf{I}_k$ as the first frame a local window sequence. Then the feature survival rate of s^{th} frame is defined as follow:

$$sr_s^k = \frac{f^n(\mathbf{I}_s)}{f^n(\mathbf{I}_k)} \tag{3}$$

where $f^n(\mathbf{I}_k)$ is the number of detected features in frame $\mathbf{I}_k$ and $f^n(\mathbf{I}_s)$ is the number of successfully tracked features in frame $\mathbf{I}_s$. The slicing algorithm follows two steps:

(1) We track the feature points from the k^{th} and compute the survival rate over each frame sequentially. Once sr_s^k is smaller than a threshold σ, we slice the k^{th} to s^{th} as a local-window sequence as input of a local BA (section. 2.2.3).
(2) To ensure the continuity over the whole sequence, we set the $(s-o)^{\text{th}}$ frame as the first frame of the next local-window, which leads to overlap with o frames length between two consecutive local-windows.

2.2.3 *Bundle Adjustment*.

BA is a non-linear optimization technique in 3D vision which iteratively refines the camera poses and 3D points simultaneously [30]. With the proposed automatic local-window slicing algorithm, we design a sliding-window BA paradigm with local and global optimization.

Local BA. We first formulate a local BA to compute the camera poses, focal lengths, and 3D points observed by the frames in local-window sequence by minimizing the reprojection error as follows:

$$\begin{aligned} &\underset{\mathbf{f},\mathbf{r},\mathbf{t},\mathbf{P}}{\operatorname{argmin}} \sum_{i=k}^{s} \sum_{j=1}^{M_{k\to s}} \left\| \mathbf{q}_i^j - \pi(\mathbf{K}_i \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \end{bmatrix} \left[\mathbf{P}^j, 1\right]^\top \right\|^2 \\ &\text{with,} \quad \pi([x,y,z]^\top) = [x/z, y/z] \end{aligned} \tag{4}$$

where k and s is the first and last frame of the local-window sequence, $M_{k\to s}$ is the number of survival features. $\mathbf{f}$ is the set of local lengths over all frames, while $\mathbf{r}$ and $\mathbf{t}$ are the sets of rotation and translation vectors from $(k+1)^{\text{th}}$ to s^{th} frame (the pose of the k^{th} frame is fixed as $[\mathbf{I}, \mathbf{0}]$). $\mathbf{P}$ is the set of 3D point coordinates. The initialization of $\mathbf{f}$, $\mathbf{r}$, $\mathbf{t}$ and $\mathbf{P}$ is introduced in section. 2.2.4.

Global BA. To obtain an accurate estimation globally, based on the results of local BAs, we use a global BA to refine the camera motion, focal lengths, and 3D scene. To this end, we first unify all the local-window sequences into the world coordinate system by using the overlapping (o frames). Thus we have the rough estimation of $\mathbf{f}$, $\mathbf{r}$, $\mathbf{t}$ and $\mathbf{P}$ from all the local-windows as initialization values to the cost function:

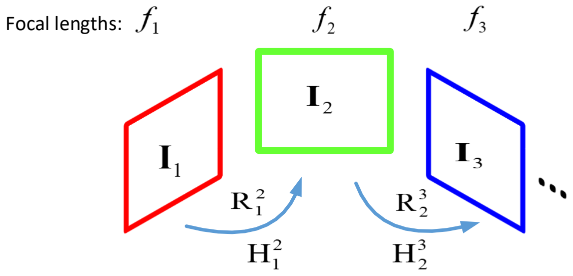


Figure 6: Geometric model for uncalibrated homography-based relative rotation and focal lengths solver.

$$\underset{\mathbf{f},\mathbf{r},\mathbf{t},\mathbf{P}}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{j=1}^{M_{1\to N}} v_{ij} \left\| \mathbf{q}_i^j - \pi(\mathbf{K}_i \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \end{bmatrix} \mathbf{P}^j) \right\|^2$$
$$\text{with,} \quad v_i^j = \begin{cases} 1 & \text{if} \quad \mathbf{P}^j \quad \text{is observed by i-th frame} \\ 0 & \text{if} \quad \mathbf{P}^j \quad \text{is not observed by i-th frame} \end{cases} \tag{5}$$

where N is the number of all frames, $M_{1\to N}$ is the number of features once observed by at least one local-window sequence. v_i^j is the visibility indicator that shows if a 3D point $\mathbf{P}^j$ is captured by i^{th} frame or not.

BA Optimization and discussions. We use the Levenberg-Marquardt (LM) algorithm to refine the camera poses, focal lengths, and 3D points by minimizing the cost functions in Eq. (4) and (5). We derive the closed-form Jacobian matrix in both local and global BAs via chain rule to speed up the iterative optimization. The proposed BAs have two main advantages over the BA approaches used in existing camera small motion tracking solutions:

(1) Our formulations in Eq. (4) and (5) have a significantly lower number of parameters compared to the numbers in [8, 12] thanks to ignoring the radial distortion and rolling shutter effect, which are not apparent in TV show. This simplification and a more reasonable initialization strategy enable our BAs to converge during our experiments rapidly.

(2) We found out that in the proposed local and global BA scheme, which can track the reliable features along the whole sequence, thus can robustly recover the accurate 3D reconstruction, camera poses and focal lengths in the highly dynamic scenes that fairly common in TV show compared to the all-frames optimization used in [8, 12].

2.2.4 ***Initialization.*** The initial parameters of the BAs are vital to the performance of non-linear optimization. [8] use all-zeros setting to the camera poses and a rough guess to the focal lengths based on the image size, which we found out that could easily fail in TV show shot with extremely short translation but large lens zoom and rotation. In contrast, to ensure the robustness of camera tracking, we present a novel linear solver to compute the relative rotation and focal lengths of two uncalibrated cameras to initial the parameters in our local BA. The initialization of global is based on the results from all the local BAs.

Homography-based Uncalibrated Relative Rotation Solver. Based on the fact that two views epipolar-based approach is unstable for small baseline [8], we can ignore the translation between two consecutive views and compute their relative rotation and focal lengths simultaneously with a analytical solution. As shown in Fig. 6, let the two frames $\mathbf{I}_1$ and $\mathbf{I}_2$ hold for pure rotation assumption which leads to a homography transformation $\mathbf{H}_1^2$ consisted by the calibration matrices $\mathbf{K}_1$, $\mathbf{K}_2$ (where only focal lengths f_1 and f_2 are unknown parameters) and the relative rotation $\mathbf{R}_1^2$. Thus, based on Eq. (1) and (2), we obtain the definition of $\mathbf{H}_1^2$ as follows:

$$\mathbf{H}_1^2 = \mathbf{K}_1 \mathbf{R}_1^2 \mathbf{K}_2^\top = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$$
$$\text{with,} \quad \begin{cases} H_{11} = \frac{f_1 - c_x * r_y}{f_2} \\ H_{12} = \frac{c_x * r_x - f_1 * r_z}{f_2} \\ H_{13} = c_x + f_1 r_y - \frac{c_y (c_x r_x - f_1 r_z)}{f_2} - \frac{c_x * (f_1 - c_x * r_y)}{f_2} \\ H_{21} = \frac{-c_y r_y + f_1 r_z}{f_2} \\ H_{22} = \frac{f_1 + c_y r_x}{f_2} \\ H_{23} = c_y - f_1 r_x - \frac{c_x (c_y r_y - f_1 r_z)}{f_2} - \frac{c_y (f_1 + c_y r_x)}{f_2} \\ H_{31} = \frac{-r_y}{f_2} \\ H_{32} = \frac{r_x}{f_2} \\ H_{33} = \frac{c_x r_y}{f_2} - \frac{c_y r_x}{f_2} + 1 \end{cases} \tag{6}$$

where H_1^2 can be further normalized by forcing H_{33} to 1. Therefore, with 4 feature-matches obtained from the feature tracking step (section. 2.2.2), we can compute the values of $H_{11}, H_{12}, ..., H_{32}$ in $\mathbf{H}_1^2$ [3] and obtain a polynomial system with 8 equations and 5 unknown parameters (r_x, r_y, r_z, f_1 and f_2). Thanks to the automatic Grobner basis solver generators such as [13, 15], the linear solver to the homography-based uncalibrated relative rotation can be obtained.

Initialization Setting. Finally, we can initialize parameters for local BA by setting the rotation vectors and focal lengths as the results of linear uncalibrated relative rotation solver described above, translation vectors to zero which mentioned to be reasonable for the small motion [33], and 3D point coordinates as $\hat{\mathbf{P}}^j = \hat{z}^j \mathbf{K}_k^{-1} [\mathbf{q}_k^j, 1]^\top$ where $\hat{z}^j$ is a random depth value.

2.3 Augment TV Show Shot

Based on the recovered camera motion, focal lengths, and 3D scene, we can augment the video by inserting 3D virtual objects into the reconstructed scene followed by projection in each frame. It is essential to realize that, in this paper, we focus on the camera small tracking problem, which is the cornerstone for the post-process, such as augmenting the TV show. Therefore, we provide a brief description of the video augmentation.

2.3.1 ***Virtual Objects Insert.*** As shown in Fig. 7(a), the user can easily place virtual objects in the user-defined location within the

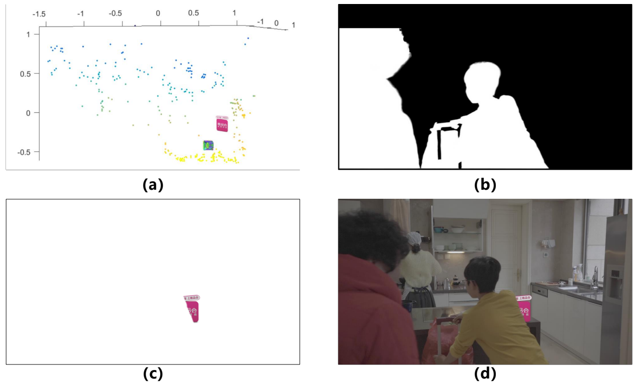


Figure 7: (a) An example of a virtual object insert in the reconstructed 3D scene. An example of occlusion handling pipeline: (b) we use [4] to segment the human. (c) Virtual object reprojection by considering the depth and the intersection with human segmentation masks. (d) The reprojected virtual objects are rendered into each frame.

reconstructed 3D scene. Note that inserted virtual objects can be either 2D or 3D.

2.3.2 ***Virtual Objects Reprojection and Occlusion Handling***. To composite the inserted virtual objects back into the original shot while perfectly matched perspective and appear seamless, we conduct a virtual objects projection and follow by an occlusion handling procedure.

Virtual Objects reprojection. After inserting the virtual objects, we can then augment the whole video sequence by reprojecting inserted virtual objects back to each frame based on the recovered camera poses by using OpenGL [25].

Occlusion handling. Based on the observations in our experiments, the human movement causes most of the virtual objects' occlusion. Thus, we propose the following pipeline to hide virtual objects behind real things adaptively:

1. In each frame, we check if the detected human bounding boxes provided by [21] have an intersection with the reprojected virtual objects. If so, as shown in Fig. 7(b), we use DeeplabV3+ [4] to extract the accurate masks of human.
2. The depth of detected humans is roughly estimated by using [32]. If virtual objects' depths are larger than the human depth, as shown in Fig. 7(b), we cut the reprojected virtual objects based on the intersection between them and the human masks.

3 EXPERIMENTS

3.1 Implementation

We run our method on an Intel(R) Xeon(R) Gold 5220 CPU with 8G RAM. We set the threshold of survival rate of successful tracked feature σ as 0.3 and the number of overlapping frames between two local-windows o as 5. For a TV show shot of 1280 × 720 resolution, our solution takes to process. Specifically, we spend to feature tracking, BAs, final reprojection, and render.

3.2 Dataset and Comparison Methods

Dataset. We assemble a comprehensive dataset of 1000 real production shots (3s 10s, 25fps) collected from real variety shows, series, and animations of **Mango TV**[1]. To know the strength and weaknesses of a method in different situations, we carefully selected TV shots with 5 categories based on camera motion and scene type, namely, (I) interview, (II) quick rotation, (III) faster zooming, (IV) small translation, and (V) crowd. We will release our dataset to the research community after acceptance.

Comparison Methods. We denote the proposed camera tracking method as **D-UCSMT**. We compared our method with two other methods:

- **DfUSMC**[2]: Depth from uncalibrated small motion clip [8] which address the uncalibrated camera small motion problem that is strongly related to this paper.
- **Homo**[3]: Since the narrow baseline could easily defeat the epipolar geometry, thus, it is interesting to investigate homography based camera rotation tracking under the pure-rotate assumption. To this end, we extend the classical image stitching pipeline [3] with linear homography solver and BA to camera rotation tracker followed by re-projection and render steps as our method.
- **Boujou** [1]: Commercial camera tracking software Vicon Boujou (version 5.0.2), which is famous for VFX in the past decades.

3.3 Quantitative Comparison

We use two metrics for quantitative analysis and then show our results:

Reprojection Error. We select 400 shots from our TV show dataset and manually chose four features points which will be tracked over all frames by KLT [29] as ground truth projections. Each feature is be localized with less than 0.1 pixels bidirectional error in simultaneously tracking forwards and backward between two consecutive frames to ensure the accuracy of the measurement.

In this experiment, we use DDfUSMC [8], Homo [1], Boujou [1] and the proposed D-UCSMT to recover the camera motion and reconstruct the 3D scene. The four selected ground truth features are forced as input of them. To evaluate the camera tracking quality, we reproject these four reconstructed 3D points back to each frame and compute the root-mean-square error (RSME) between the four reprojection and image measurements (ground truth). Note that since Homo [1] does not reconstruct 3D scenes, thus we use recovered homography matrices to transform the four features to the rest frames instead.

Fig. 8 shows the RSMEs in each frame of 10 example shots consists of 5 categories while the average RSMEs of each example shot is reported in table. 1. We can observe that DDfUSMC [8], Homo [1]

[1] https://w.mgtv.com/

[2] https://github.com/hyowonha/DfUSMC

[3] https://github.com/ppwwyyxx/OpenPano

Table 1: **Quantitative comparisons on 200 shots: the numbers show that RSME of reprojection error of 4 reconstructed 3D points.**

	Interview		Quick rotation		Faster zooming		Small translation		Crowd		Average **(400 shots)**
	#14	#330	#23	#67	#79	#237	#2	#9	#89	#318	
DfUSMC [8]	122.6	121527.2	8.9	18.4	1.7	60.1	**0.7**	72.1	0.9	0.4	1263.0
Homo [3]	1.0	27.5	2395.0	2.4	2.5	1.3	3.2	2.8	1.1	67.2	136.2
Boujou [1]	22.6	1808.0	377.1	5.2	0.4	1.8	3.6	1.9	**0.4**	87.8	96.8
D-UCSMT	**0.9**	**2.2**	**2.0**	**2.3**	**0.3**	**1.2**	**0.7**	**0.7**	0.7	**0.3**	**1.5**

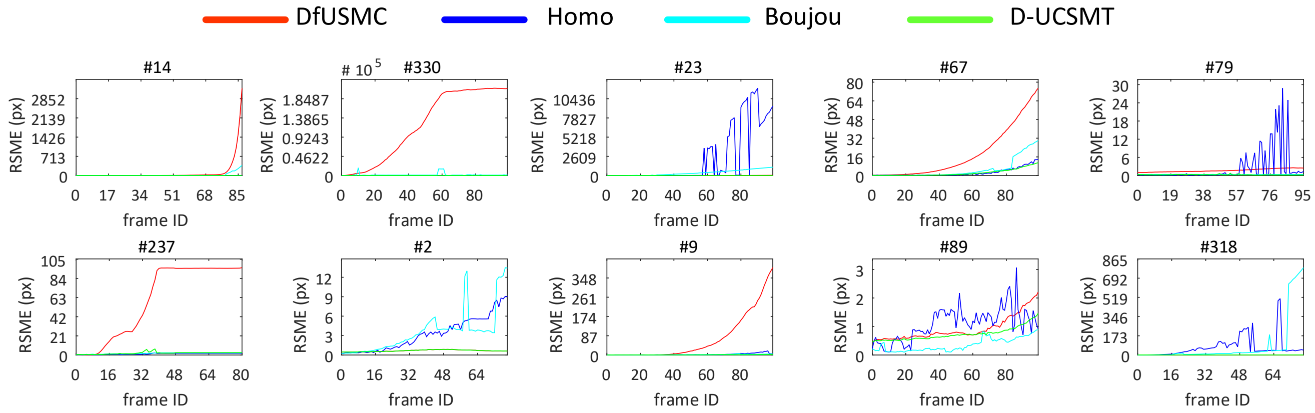


Figure 8: Reprojection RSME (in pixels) on each frame of ten shots by DDfUSMC [8], Homo [1], Boujou [1] and the proposed D-UCSMT.

and Boujou [1] achieve accurate tracking results in certain categories. For example, DfUSMC fails in interviews and faster zooming scenes, while Homo and Boujou are unsuitable for handling quick rotation and crowded scenes. In contrast, D-UCSMT provides stable results in every category.

Finally, the average RSMEs over the **400 shots** are computed and reported in table. 1 show the D-UCSMT achieves significant robust performance in TV show scenes against to DDfUSMC [8], Homo [1] and Boujou [1].

Intersection Over Union (IOU). We select 50 shots from our TV show dataset and manually label the minimum bounding boxes in each frame of a certain real object. Besides, we design a 3D edge skeleton for these real objects and augment the shots by projecting them. If the camera motion is well tracked, the projected 3D edge skeleton can perfectly fit the corresponding real object.

Thus, as shown in Fig. 9, we evaluate the performances based on IOU between minimum bounding boxes of a real object and its corresponding projected virtual edge skeleton. Table. 2 summarizes the average IOUs of 5 examples shots (check the comparison videos in supplemental material) and the average IOUs over **50 shots** which shows that D-UCSMT provides the most stable project virtual object over DfUSMC [8], Homo [1], Boujou [1] under different scenes.

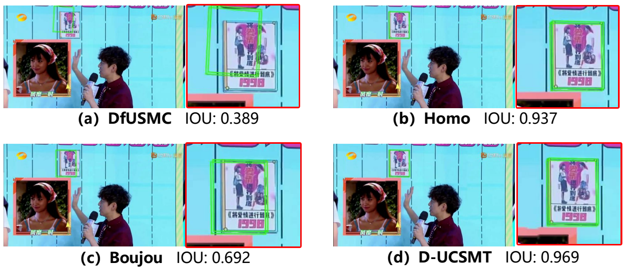

(a) **DfUSMC** IOU: 0.389
(b) **Homo** IOU: 0.937
(c) **Boujou** IOU: 0.692
(d) **D-UCSMT** IOU: 0.969

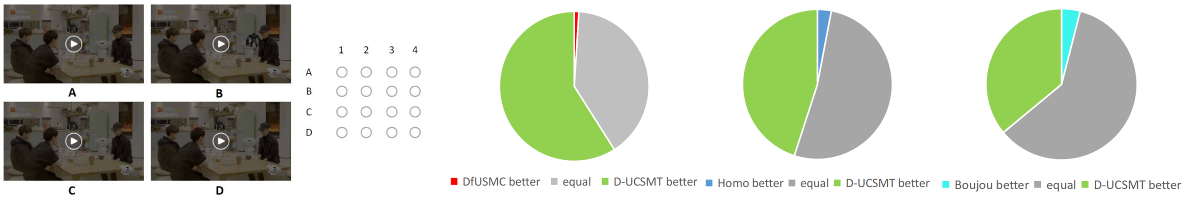
Figure 9: An example frame from sequence 'poster' augmented by DfUSMC [8], Homo [1], Boujou [1] and the proposed D-UCSMT. We use IOUs between the bonding box of ground-truth object and projected virtual edge skeleton are used to evaluate the performances.

3.4 Visual Comparisons

We surveyed 20 users to provide preferences for 550 augmented TV show shots by DfUSMC [8], Homo [1], Boujou [1] and D-UCSMT based on their visual perception. These augmented shots are displayed to the users in random order. The users are unaware of which technique is used to produce the augmentation results. Fig. 10(a) shows such an interface for the user study. Every result could be assigned one of the four values (1-4), with 1 denoting the better than the others.

Table 2: Quantitative comparisons on 50 shots: the numbers show that IOU between minimum bounding boxes of a real object and its corresponding projected virtual edge skeleton.

	#poster	#box	#makeup mirror	#photo frame	#sofa	Average IOU (50 shots)
DfUSMC [8]	0.575	0.951	0.840	0.974	0.789	0.836
Homo [1]	0.948	0.952	0.947	0.972	0.944	0.923
Boujou [1]	0.763	**0.955**	0.943	0.971	0.941	0.906
D-UCSMT	**0.950**	0.954	**0.962**	**0.981**	**0.964**	**0.969**



(a)

(b)

Figure 10: (a) Comparison interface for user surveyed. (b) User surveyed results by comparing our method on 550 shots with three methods: DfUSMC [8], Homo [1] and Boujou [1].

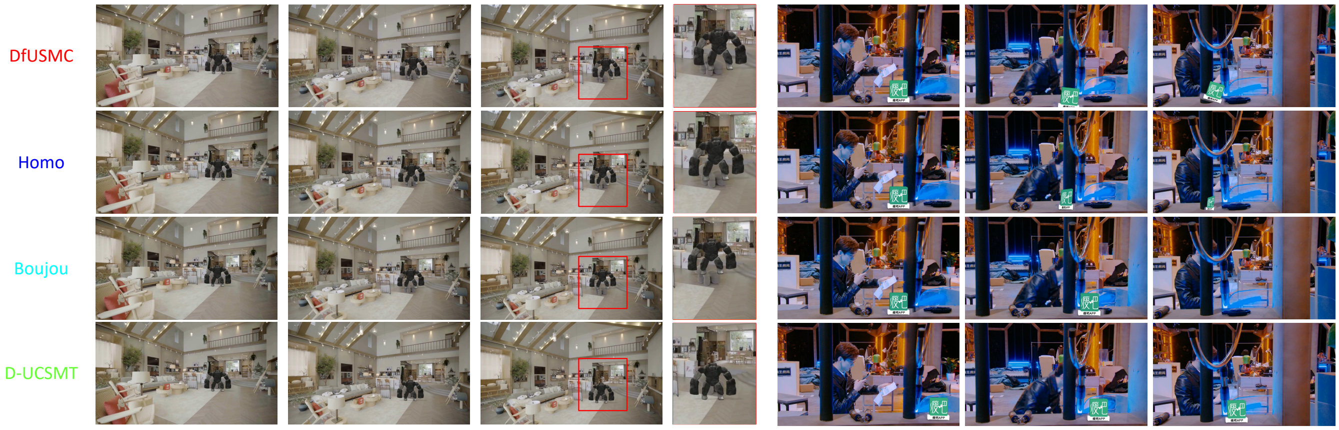


Figure 11: Two examples of visual comparison. (left) Virtual rock man standing on the floor, but be rendered to fly up by DfUSMC [8], Homo [1], Boujou [1] while be perfectly matched by D-UCSMT. (right) A virtual billboard is inserted on the desk but be rendered with significant drift in the later frames by DfUSMC [8], Homo [1], Boujou [1] while D-UCSMT provides stable projections.

Two examples of augmented shots in Fig. 11 show that the proposed method enables virtual objects to insert and projection stably. The statistic results in Fig. 10(b) on 550 shots demonstrates the effectiveness and superiority to the state-of-the-art approaches DfUSMC [8], Homo [1] and Boujou [1].

4 CONCLUSION

In this paper, we have introduced a practical solution that can track uncalibrated cameras with small motion and reconstruct the 3D scene accurately called D-UCSMT. It matches the TV show slotting style, namely, faster zooming, small motion, and quick rotation without intrinsic information (focal length). Thus, we can use the proposed D-UCSMT to simultaneously reconstruct the sparse 3D scene and compute camera poses and focal lengths of each frame by giving an uncalibrated TV show shot as input. As a result, we can further add the virtual objects (elements) into the reconstructed 3D scene, then composite them back into the original shot while perfectly matched perspective and appear seamless. The evaluation of a large variety of real TV show shots demonstrates the effectiveness of our method in augmenting TV shows against state-of-the-art works and commercial software products.

ACKNOWLEDGMENTS

This work has been sponsored by the Fundamental Research Funds for the Central Universities (HNU: 531118010543).

REFERENCES

[1] [n.d.]. Vicon Boujou, howpublished = http://www.vicon/,.

[2] Alastair Barber, Darren Cosker, Oliver James, Ted Waine, and Radhika Patel. 2016. Camera tracking in visual effects an industry perspective of structure from motion. In *Symposium on Digital Production*.

[3] Matthew Brown, David G Lowe, et al. 2003. Recognising panoramas.. In *ICCV*.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*.

[5] Mark Christiansen. 2013. *Adobe After Effects CC Visual Effects and Compositing Studio Techniques*. Adobe Press.

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2018. Superpoint: Self-supervised interest point detection and description. In *CVPR workshops*.

[7] Tim Dobbert. 2006. *Matchmoving: the invisible art of camera tracking*. John Wiley & Sons.

[8] Hyowon Ha, Sunghoon Im, Jaesik Park, Hae-Gon Jeon, and In So Kweon. 2016. High-quality depth from uncalibrated small motion clip. In *CVPR*.

[9] Jungong Han, Dirk Farin, and Peter HN de With. 2007. A real-time augmented-reality system for sports broadcast video enhancement. In *ACM MM*.

[10] Christopher G Harris, Mike Stephens, et al. 1988. A combined corner and edge detector.. In *Alvey vision conference*.

[11] Juan David Hincapié-Ramos, Kasim Ozacar, Pourang P Irani, and Yoshifumi Kitamura. [n.d.]. GyroWand: IMU-based raycasting for augmented reality head-mounted displays. In *ACM Symposium on Spatial User Interaction*.

[12] Sunghoon Im, Hyowon Ha, Gyeongmin Choe, Hae-Gon Jeon, Kyungdon Joo, and In So Kweon. 2015. High quality structure from small motion for rolling shutter cameras. In *ICCV*.

[13] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. 2008. Automatic generator of minimal problem solvers. In *ECCV*.

[14] Kiriakos N Kutulakos and James R Vallino. 1998. Calibration-free augmented reality. *TVCG* (1998).

[15] Viktor Larsson, Kalle Astrom, and Magnus Oskarsson. 2017. Efficient solvers for minimal problems by syzygy-based reduction. In *CVPR*.

[16] Zahid Mahmood, Tauseef Ali, Nazeer Muhammad, Nargis Bibi, Imran Shahzad, and Shoaib Azmat. 2017. EAR: Enhanced Augmented Reality System for Sports Entertainment Applications. *KSII Transactions on Internet & Information Systems* (2017).

[17] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. 2016. Openmvg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*. Springer, 60–74.

[18] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics* (2015).

[19] David Nistér. 2004. An efficient solution to the five-point relative pose problem. *TPAMI* (2004).

[20] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. 2012. USAC: a universal framework for random sample consensus. *TPAMI* (2012).

[21] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[22] Johannes L Schonberger and Jan-Michael Frahm. 2016. Structure-from-motion revisited. In *CVPR*.

[23] Gerald Schweighofer, Sinisa Segvic, and Axel Pinz. 2008. Online/realtime structure and motion for general camera models. In *2008 IEEE Workshop on Applications of Computer Vision*.

[24] Yongduek Seo and Ki Sang Hong. 2000. Calibration-free augmented reality in perspective. *TVCG* (2000).

[25] Dave Shreiner, Graham Sellers, John Kessenich, and Bill Licea-Kane. 2013. *OpenGL programming guide: The Official guide to learning OpenGL, version 4.3*.

[26] RA Smith, Andrew W Fitzgibbon, and Andrew Zisserman. 1999. Improving Augmented Reality using Image and Scene Constraints.. In *BMVC*.

[27] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. 2017. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications* (2017).

[28] Graham Thomas. 2006. Mixed reality techniques for TV and their application for on-set/pre-visualisation in film production. In *International Workshop on Mixed Reality Technology for Filmmaking*.

[29] Carlo Tomasi and T Kanade Detection. 1991. Tracking of point features. *IJCV* (1991).

[30] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. 1999. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*.

[31] Changchang Wu et al. 2011. VisualSFM: A visual structure from motion system. (2011).

[32] Ruichao Xiao, Wenxiu Sun, Jiahao Pang, Qiong Yan, and Jimmy Ren. 2018. Dsr: Direct self-rectification for uncalibrated dual-lens cameras. In *3DV*.

[33] Fisher Yu and David Gallup. 2014. 3d reconstruction from accidental motion. In *CVPR*.

[34] Jung-Jae Yu and Jae-Hean Kim. 2010. Camera motion tracking in a dynamic scene. In *2010 IEEE International Symposium on Mixed and Augmented Reality*.

[35] Guofeng Zhang, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. 2009. Consistent depth maps recovery from a video sequence. *TPAMI* (2009).

[36] Richard Hartley Andrew Zisserman. 2004. Multiple view geometry in computer vision. (2004).

[37] Stefanie Zollmann, Tobias Langlotz, Moritz Loos, Wei Hong Lo, and Lewis Baker. 2019. Arspectator: Exploring augmented reality for sport events. In *SIGGRAPH Asia*.