Calibration of Nodal and Free-Moving Cameras in Dynamic Scenes for Post-Production

Evren Imre, Jean-Yves Guillemaut, Adrian Hilton
Center for Vision, Speech and Signal Processing
University of Surrey
Guildford, UK
Email: h.imre@surrey.ac.uk

***Abstract*—In film production, many post-production tasks require the availability of accurate camera calibration information. This paper presents an algorithm for through-the-lens calibration of a moving camera for a common scenario in film production and broadcasting: The camera views a dynamic scene, which is also viewed by a set of static cameras with known calibration. The proposed method involves the construction of a sparse scene model from the static cameras, with respect to which the moving camera is registered, by applying the appropriate perspective-n-point (PnP) solver. In addition to the general motion case, the algorithm can handle the nodal cameras with unknown focal length via a novel P2P algorithm. The approach can identify a subset of static cameras that are more likely to generate a high number of scene-image correspondences, and can robustly deal with dynamic scenes. Our target applications include dense 3D reconstruction, stereoscopic 3D rendering and 3D scene augmentation, through which the success of the algorithm is demonstrated experimentally.**

***Keywords*-camera calibration; structure-from-motion**

I. Introduction

A hybrid multi-camera setup, involving a moving (principal), and a set of static (witness) cameras is frequently employed in film production and broadcasting, as such a setup saves editing time and facilitates post-production. In the latter case, the plausibility of many special effects depends on whether accurate camera calibration information is available. Moreover, the prominence of this setup, and the importance of calibration, is on the rise due to the popularity of 3D films, and new applications, such as 3D-TV. For static cameras, the necessary level of calibration accuracy can be attained via manual calibration techniques [1] [2]. However, the fact that a principal camera may have variable calibration parameters (*i.e.*, that it can move and zoom) leaves through-the-lens calibration as the only viable approach, despite the challenges posed by dynamic scene content. The algorithm we propose aims to address this problem, *i.e.*, recovers the intrinsic and the extrinsic parameters of a camera in general or nodal motion, and viewing a scene with dynamic elements, given a set of cameras with known calibration.

The through-the-lens calibration literature of the last decade is dominated by a strategy originally developed for unordered image collections [3]: Solve a 3D-2D registration (*i.e.*, P*n*P, "perspective n-point") problem with respect to a sparse, point-based scene model for each image. The scene model is built incrementally, by triangulating the wide-baseline correspondences gleaned from the consecutive images. The registration involves finding scene-image (3D-2D) correspondences, and a minimal P6P [4], or a P3P [5] solver, depending on the availability of the intrinsics. For monocular video sequences, monocular structure-from-motion (MSfM) replaces wide-baseline matching by tracking, and constructs the scene model from a subset of images with sufficiently large baseline [6] [7]. Alternatively, simultaneous localization and mapping (SLAM) approaches cast the problem into a state estimation framework that jointly computes the scene model and the calibration parameters [8]. In the case of multiple monocular sequences, [9] proposes registering the individual MSfM solutions to a common reference frame by establishing correspondences between their scene models. Finally, [10] handles a hybrid static-moving camera setup by treating the witness set as an unordered image collection for building a scene model, to which the principal camera is registered via a P3P solver.

Although the modern calibration algorithms offer mature tools, they are not without their limitations: MSfM is vulnerable to dominant planes and insufficient camera motion [4]. SLAM algorithms aim for real-time operation, and are superior to MSfM in terms of accuracy only in the case of small processing budget [11]. Moreover, all of the methods above operate under the static scene assumption, and their performance is susceptible to large dynamic objects.

Our approach constructs a scene model from a witness set, computes the calibration measurements at each frame via guided matching and RANSAC, and removes the measurement jitter through an unscented Kalman filter (UKF). It has the following capabilities and novel features:

- It is robust to dynamic scenes, as it can refresh the scene model at each time instant, and therefore make use of both the static and the dynamic elements in calibration. The accompanying increase in the computational complexity is mitigated by the use of guided matching, and dynamic witness subset selection.

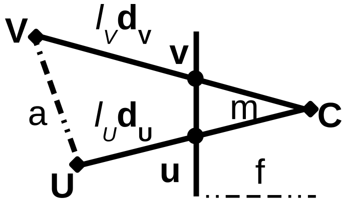

Figure 1. Geometry of the orientation and focal length estimation problem.

- It can estimate the orientation and the focal length of a nodal camera, given the camera center, through a novel P2P solver that requires 2 scene-image correspondences.
- It can handle a free-moving camera, both with and without known intrinsics, via P3P [5] and P4P [12] solvers, respectively. The unknown intrinsics case is stabilized by accepting the P4P solution, only when the P3P solution with the current estimate of the intrinsics is significantly inferior.
- It can dynamically identify a subset of witness cameras (an *active* witness set), whose combined field-of-view (FoV) is likely to contain the largest number of 3D-2D correspondences with the principal camera, in order to limit the processing to the most promising portions of the scene model.

The rest of the paper is organized as follows: In the next section, the new P2P solver is introduced. The full calibration algorithm is discussed in Section 3. In Section 4, the performance of the algorithm is evaluated quantitatively and qualitatively through our target applications, namely, 3D reconstruction, stereoscopic rendering and augmented reality. Section 5 concludes the paper.

II. Orientation and Focal Length Estimation from Two 3D-2D Correspondences

The relative orientation and focal length estimation problem is often encountered in the context of image stitching and calibration of pan-tilt-zoom cameras. In either case, a feature-based solution essentially involves the estimation of a homography from image correspondences [13] [4]. The 3D counterpart, orientation and focal length with respect to a scene model, given the camera center, requires two 3D-2D correspondences. The method, presented below, follows a similar strategy to the 3-point pose estimation algorithm [5]: Locate the scene points in the camera reference frame (defined by the camera center, the image plane, and the principal axis vector), and then recover the 3D homography that maps them to the world reference frame. In addition to the camera center, we make the common assumption that all intrinsics except for the focal length are known, and the image points are normalized accordingly. Figure 1 illustrates the geometry of the problem.

An image point with the coordinate vector $\mathbf{u}$ can be expressed as a 3D point on the image plane as

$$\mathbf{u_c} = \begin{bmatrix}\mathbf{u^T} & f\end{bmatrix}^T, \tag{1}$$

where f is the focal length. The corresponding scene point, $\mathbf{U}$, and the direction vector of the projection ray, $\mathbf{d_U}$, in the camera reference frame, are

$$\begin{aligned}\mathbf{U} &= l_U \mathbf{d_U}\\ \mathbf{d_U} &= \frac{1}{\|\mathbf{u_c}\|}\mathbf{u_c}\end{aligned}, \tag{2}$$

where l_U denotes the distance of $\mathbf{U}$ to the camera center, which can be computed from the known world coordinates of the scene point and the camera center. $\mathbf{v_c}$, $\mathbf{V}$, $\mathbf{d_V}$, and l_V are defined similarly for the image point $\mathbf{v}$.

In order to recover the only unknown, f, we observe that the cosine of the angle m (Figure 1) can be computed either by applying the law of cosines on the triangle $\mathbf{UVC}$, or as the dot product of $\mathbf{d_U}$ and $\mathbf{d_V}$. This defines a quadratic equation in f^2, *i.e.*,

$$\begin{aligned}a^2 &= l_U^2 + l_V^2 - 2l_U l_V \cos m\\ &= l_U^2 + l_V^2 - 2l_U l_V \mathbf{d_U^T d_V}\\ &= l_U^2 + l_V^2 - 2l_U l_V \frac{\mathbf{u^T v} + f^2}{\sqrt{(\mathbf{u^T u} + f^2)(\mathbf{v^T v} + f^2)}}\end{aligned}, \tag{3}$$

where a is the distance between the scene points. Upon solving Equation 3, $\mathbf{U}$ and $\mathbf{V}$ can be recovered via Equation 2. The coordinates of 2 scene points in the camera and the world reference frame are sufficient to recover the rotation relating the reference frames [14].

III. Camera Calibration Estimation

A. Calibration Pipeline

The calibration algorithm, summarized in Figure 2, requires a set of witness cameras as input. The set could be manually calibrated, or, could be the output of an automatic calibration algorithm, such as Bundler [15]. In the case of multiple principal cameras, *e.g.*, a nodal and a free camera tracking the same object, the calibration algorithm can be run iteratively, employing the estimated calibration sequences as "dynamic witness cameras" for the remaining principal cameras. The initialization involves the construction of a sparse scene model from all camera pairs [10], and the estimation of the initial calibration measurement, by solving the appropriate PnP problem. The resulting scene model is a collection of 3D features, each of which is composed of a 3D coordinate, the covariance of the coordinate, and a SIFT descriptor [16] for each image in which the feature is observed. In the main loop, the algorithm goes through the following steps:

Active witness set selection: This step aims to identify a subset of the witness cameras with a FoV that not only overlaps that of the principal camera, but is also promising in terms of potential correspondences with the scene and the

Input: Witness set, available calibration parameters of the principal camera, scene type (static or dynamic), # active witnesses
Output: Principal camera calibration estimate
Initialization: Build the scene model and estimate the initial calibration.
Main loop:
1. Identify the active witness set (Section III-B).
2. If the scene is dynamic, rebuild the scene model from the active witness set.
3. Estimate the missing calibration parameters of the principal camera.
4. Update the calibration state estimate

Figure 2. Calibration algorithm

principal camera image. Removal of the “irrelevant” witness cameras is of interest for two reasons:

- Contamination of the scene model with unpromising features degrades the matching performance through repetitive texture and false positives. A scene feature that is not visible in a member of the active witness set is temporarily removed from the model.
- For a set of n witness cameras, the number of image pairs to be processed for a sparse scene model is $\frac{n(n+1)}{2}$. In the dynamic scene case, it is not feasible to use the entire witness set at each time instant.

Assuming a k-camera subset is requested, the camera selection procedure uses the current estimate of the calibration, and a sparse 3D lattice covering the scene, to rank all k-element subsets of the witness set. The details of the algorithm are discussed in Section III-B

Handling of dynamic scenes: Camera calibration algorithms rely on the observations of a static scene in different images, and make no assumptions with regard to synchronization. However, in film production and broadcast environments, cameras are often synchronized. Images taken synchronously contain all the necessary information to build a scene model including both the static and the dynamic elements. Of course, the reliability of this model in other time instants relies on the size of the area occupied by the dynamic elements, and reusability can be improved by tracking the dynamic elements across the sequence. However in the current algorithm, when the shot contains a large dynamic object, we build a new, single-use scene model at each time instant, by using all camera pairs in the *active* witness set (which, in practice, contains just two cameras, *i.e.*, $k = 2$). This approach solves the occlusion problem, however, the features on the dynamic objects might be of lower quality due to motion blur.

Computation of calibration measurements: This step solves one of the following calibration problems, as determined by the missing calibration elements:

- **Orientation:** P2P algorithm described in Section II
- **Orientation and focal length:** P2P algorithm described in Section II
- **Pose:** P3P algorithm of [5]
- **Pose, focal length and lens distortion:** P4P algorithm of [12]

Depending on the actual problem, the necessary number of 3D-2D correspondences is obtained via iterative application of guided matching and calibration estimation [4]. The matcher seeks feature correspondences between the scene and the principal camera image (Hessian-affine features [17] with SIFT descriptors), within the search regions defined by the *guide*, the current calibration estimate. The similarity score of two features is the minimum Euclidean distance between the descriptor of the 2D point, and those in the descriptor list of the 3D point [10]. The matching procedure keeps only the correspondences that are strongly similar, unambiguous and satisfying the left-right consistency [10]. A RANSAC engine then selects a minimal correspondence set, and the associated PnP solution. The engine employs the SPRT variant [18], which culls the unpromising hypotheses without evaluating the entire correspondence set, and a PROSAC-inspired [19] approach that processes the more likely correspondences first (as determined by the reprojection error with respect to the guide), but with full stochastic sampling. After the nonlinear refinement stage, the resulting calibration becomes the new guide, and the matcher-solver loop is repeated, until convergence.

The above process performs quite successfully for all cases, except for P4P. In our experiments, we observed that, even when the intrinsics are constant, the successive focal length measurements computed by the P4P solver may exhibit some variation, introducing a significant jitter to the position measurements. We identified the likely cause as the focal length-depth ambiguity, which is not trivial to resolve just by the reprojection error. In order to alleviate this problem, we employed a multiple-hypothesis approach, and at each frame, in addition to the P4P solution, we computed a competing P3P solution, by using the current estimate of the intrinsics. The P4P solution is preferred only if it is considerably superior to the P3P in terms of the number of inliers. The increase in the computational cost is affordable, as the P3P solver is about 6 times faster than P4P.

The covariance of a calibration measurement is estimated via the unscented transformation (UT) [20], as justified by the highly nonlinear nature of the PnP solvers. The operation involves applying deterministic offsets to the minimal correspondence set (as determined by its covariance), passing the new sets through the appropriate PnP solver, and computing the sample statistics of the resulting calibration set (with due attention to quaternions [21].

Computation of calibration estimates: If the calibration measurements are computed independently from each other, they inevitably exhibit some jitter. A sequential state estimator can alleviate this problem in two ways: Filtering out the noise, and providing an initial estimate to the guided matcher. In order to avoid linearization errors, we choose UKF [20]. In the most general case, a constant velocity model for position, orientation, focal length and lens distortion is employed. Concretely,

$$\begin{aligned}
\mathbf{C_{t+1}} &= \mathbf{C_t} + \delta_\mathbf{t} \\
\mathbf{q_{t+1}} &= \mathbf{q_t} \otimes Q(\phi_\mathbf{t}) \\
f_{t+1} &= f_t + \gamma_t \\
\mu_{t+1} &= \mu_t + \lambda_t \\
\delta_\mathbf{t+1} &= \delta_\mathbf{t} + \Delta \\
\phi_\mathbf{t+1} &= \phi_\mathbf{t} + \Phi \\
\gamma_{t+1} &= \gamma_t + \Gamma \\
\lambda_{t+1} &= \lambda_t + \Lambda
\end{aligned} \quad , \tag{4}$$

where $\mathbf{C}$, $\mathbf{q}$, f and μ stand for the camera center, the orientation quaternion, the focal length, and the lens distortion, respectively. δ, ϕ, γ and λ are the associated change rates, which are corrupted by the independent Gaussian noise processes Δ, Φ, Γ and Λ. The quaternion uncertainty, and the rotation vector (ϕ) are represented in axis-angle form [22], and Q is an operator that converts an axis-angle vector to a quaternion. The measurement function is identity, corrupted by a Gaussian noise process.

In more restricted cases, the known calibration parameters are dropped from the state.

B. Active Witness Set Selection

Intuitively, the active witness set should share a large FoV with the principal camera. However, covisibility of a feature does not necessarily imply that it can be matched. A better criterion is the *matchable volume* (MV), the volume that contains scene features that can be potentially related to the image features observed by the principal camera. A feature in Ω_{ij}, the MV of the cameras C_i and C_j, satisfies the following conditions:

- **Visibility:** The feature is visible in both cameras.
- **Viewpoint difference:** The angle between the projection rays to C_i and C_j is less than a certain value.
- **Scale difference:** The projection of a surface patch around the feature should cover similar areas in the images.

It should be noted that evaluating the matchability of a feature set for an image pair amounts to an inexpensive simulation of the matching process. The actual values of the thresholds should reflect the limitations of the underlying feature descriptor [23].

Table I

DETAILS OF THE DATA. *Calibration* INDICATES THE KNOWN CALIBRATION PARAMETERS- THE INTRINSICS (K) AND/OR THE CAMERA CENTER (C)-, WHICH IMPLIES THE USE OF A CERTAIN *Solver*.

Sequence	Motion Type	Calibration	Solver
Unicycle1	Nodal	C and K	P2P
Odzemok	Free	K	P3P
Unicycle2	Nodal	C	P2P
Juggler	Free	None	P4P

The MV between the principal camera and an active witness set depends on the scene type. In the static case, if a point is in the MV of a principal camera and any member of the set, it is in the MV of the entire set. Therefore, given an index set I, representing a subset of witness cameras, and the principal camera C_p,

$$MV_{static} = \bigcup_{i\in I} \Omega_{ip}. \tag{5}$$

In the dynamic case, there is an additional constraint: If a feature cannot be matched in at least one pair of witness cameras, no corresponding scene feature can be instantiated. Therefore, there must be a pair $(i;j) \in I$, for which the feature is in Ω_{ij}. Moreover, in order to be matchable to C_p, it must be in either Ω_{ip} or Ω_{jp}. This can be expressed as,

$$MV_{dynamic} = \bigcup_{i\in I} \bigcup_{i\in I, j\neq i} (\Omega_{ip} \cup \Omega_{jp}) \cap \Omega_{ij}. \tag{6}$$

In order to determine a k-element active witness set, the Equations 5 or 6 are evaluated for all k-element subsets of the witness set. The evaluation is not carried out exactly, over the entire volume, but only at the vertices of a regular lattice covering the scene, usually about several hundred points (depending on the size of the scene). Alternatively, an existing scene model can be used, if it provides reasonably uniform coverage. Ω_{ij} should contain only the vertices which can be triangulated accurately (*e.g.*, with a low covariance), so that narrow-baseline witness camera pairs are excluded.

IV. EXPERIMENTAL RESULTS

We studied the performance of the algorithms through quantitative and qualitative experiments on four 125-frame sequences, all of which are captured with 7 HD cameras, in a 6 witness-1 principal configuration. All sequences include a performing actor, who can occupy as much as 30% of the image. The full calibration of the static cameras, as well as the partial calibration of the principal cameras, is obtained via [1] and [2]. Further details of the sequences, and sample images, are presented in Table I and Figure 3.

In the following discussion, the terms *foreground* and *background* refer to the dynamic, and the static scene elements, respectively. The checkerboard pattern in the background *does not* help the algorithm: Due to the repetitive nature of regular patterns, the matcher considers such features as ambiguous, and does not include them in the final correspondence list.

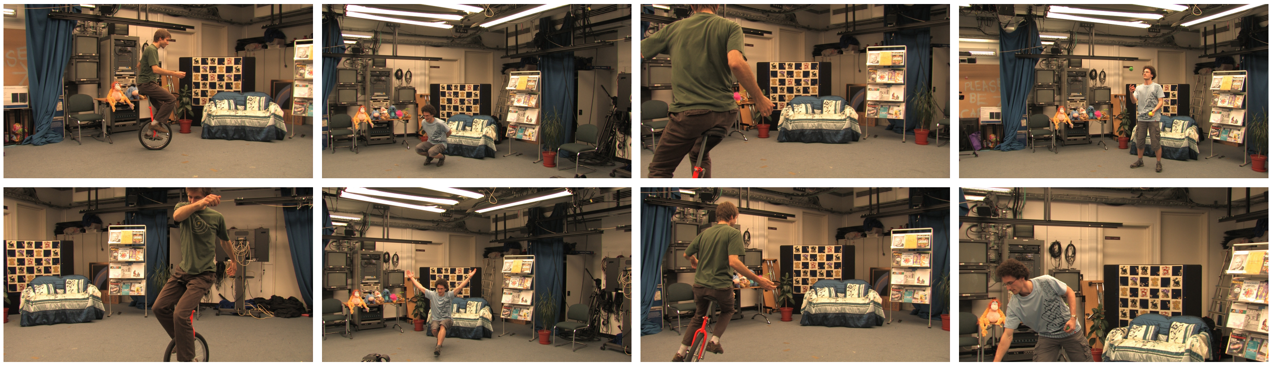

Figure 3. Initial and final images of the data: *Left-to-right: Unicycle1, Odzemok, Unicycle2, Juggler*

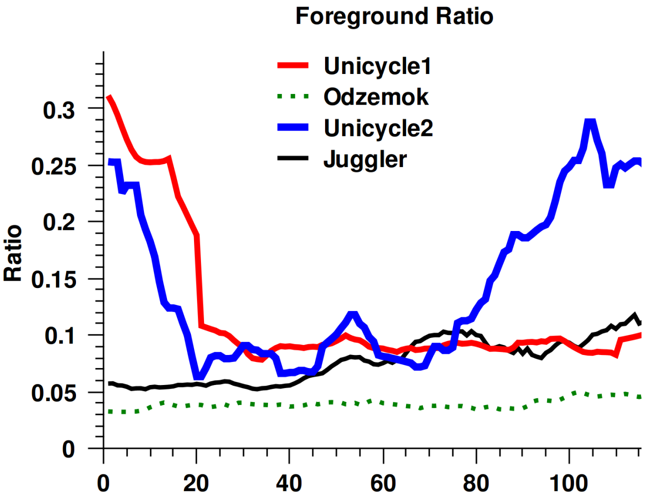

Figure 4. Ratio of the area of the foreground object to the image.

Table II
ORDER STATISTICS OF THE SQUARE-ROOT OF THE REPROJECTION ERROR (PIXEL). "IDEAL" IS THE CASE OF UNIT-VARIANCE IMAGE COORDINATE NOISE AND ERROR-FREE POSE ESTIMATE.

	5%	25%	Median	75%	95%
Unicycle1	0.284	0.734	1.215	1.741	2.296
Odzemok	0.314	0.753	1.198	1.730	2.445
Unicycle2	0.268	0.715	1.260	1.882	2.972
Juggler	0.196	0.570	1.060	1.823	2.866
Ideal	0.318	0.759	1.177	1.668	2.449

Table III
ORDER STATISTICS OF THE SQUARE-ROOT OF THE REPROJECTION ERROR (PIXEL) WITH A 6-CAMERA WITNESS SET.

	5%	25%	Median	75%	95%
Unicycle1	0.359	0.872	1.364	1.824	2.246
Odzemok	0.253	0.653	1.095	1.635	2.356
Unicycle2	0.355	0.844	1.311	1.821	2.618
Juggler	0.314	0.810	1.346	1.940	2.995

A. *Calibration Estimation*

The test data is processed by the proposed algorithm to recover the missing calibration parameters (Table I). In all experiments, a 2-camera active set, and the dynamic-scene assumption are utilized. The results are presented in Figures 5 and 6. Figure 4 illustrates the variation of the foreground area, which is computed by projecting the foreground object to the principal camera (Section IV-B). As seen in the graphs, the estimated quantities have only an insignificant amount of jitter, and their initial and final values do not appear to be in conflict with Figure 3. The initial high-uncertainty of the pose estimate in Figure 6 reflects the initial uncertainty of the UKF, which rapidly diminishes as more measurements arrive.

In our experiments with Boujou, with the default parameters, we observed that as long as the checkerboard pattern is visible, the performance remains satisfactory. However, when the camera is zooming, a certain amount of manual intervention becomes necessary to mitigate the jitter, and abrupt jumps. Moreover, when the foreground ratio is above 20%, the calibration estimates detoriorate considerably.

The lack of ground truth calibration information prevents a direct assessment of the accuracy of the estimates. Instead, we use an indirect measure: distribution of the reprojection error. Assuming that each pixel coordinate in the image measurement set is corrupted by an independent noise process with a standard deviation of 1 pixel, in the absence of any calibration errors, the reprojection error would be distributed as χ^2 with 2 degrees of freedom (*Ideal* in Table II). In order to assess the performance of the algorithm, we employ cross-validation over the scene model, *i.e.*, partition the scene model into a "training" and a "test" set by randomly discarding half of the scene points, run the algorithm on the former, and calculate the distribution of the reprojection error on the latter. The results, presented in Table II, show that the error can be mostly attributed to the image measurement noise, and the calibration is reasonably accurate. However, it should be mentioned that 1-pixel standard deviation assumption is somewhat pessimistic- in

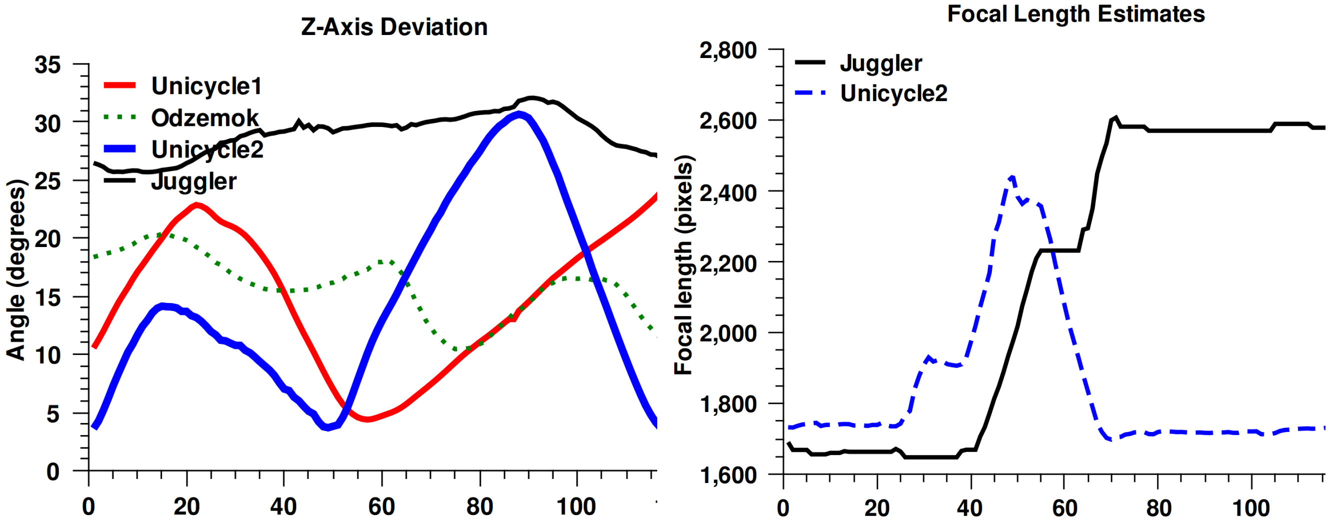

Figure 5. *Left:* Angle between the principal vector and the z-axis. *Right:* Focal length estimates.

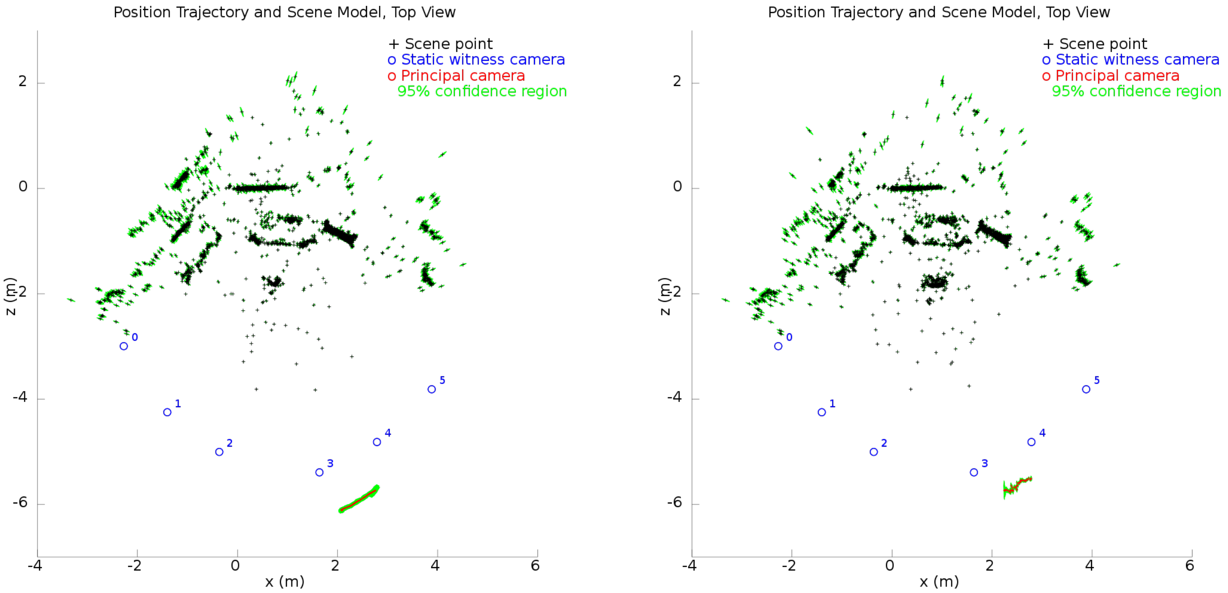

Figure 6. Top-view of the pose trajectories: *Left: Odzemok. Right: Juggler*. The trajectories are the lines closest to the x-axis.

Table II, *Juggler* seems to be slightly superior to *Ideal*, the perfect calibration case.

In order to evaluate the effectiveness of the active witness set selection procedure, we repeated the previous experiment with a 6-camera active witness set. Since the order statistics are similar, we conclude that the pair of cameras included in the active set are the ones that provide sufficient information for an accurate calibration, and hence the selection procedure achieves its objective.

B. *Applications*

For a qualitative study of the performance of the algorithm, 3 tasks are chosen, on the basis of their sensitivity to calibration errors: Dense 3D reconstruction, scene augmentation and stereoscopic rendering. The scene models (Figure 7) obtained via the dense 3D reconstruction algorithm are utilized by the other applications. In order to render the floor and the walls, the room is modeled as a cube.

Dense 3D reconstruction: Dense 3D reconstruction involves building a conservative visual hull [24] from the foreground masks extracted via background cut [25], which

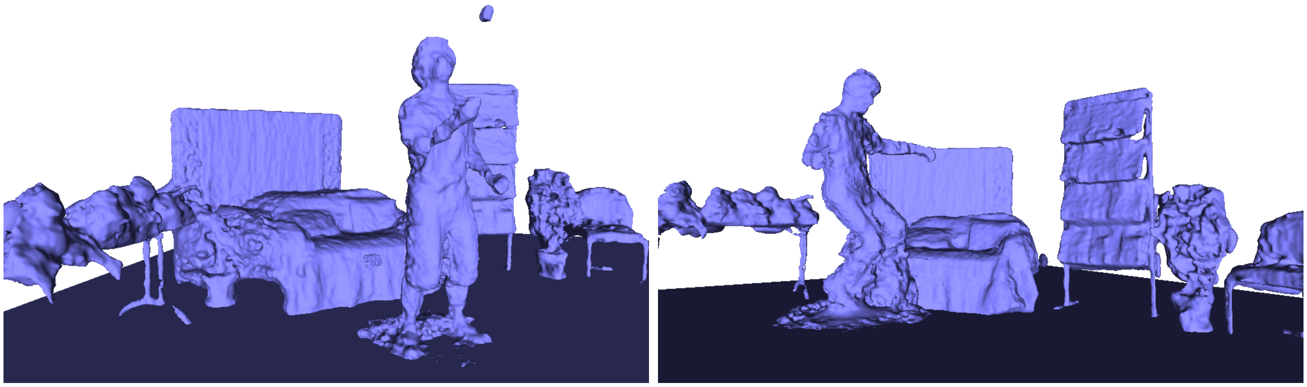

Figure 7. Estimated actor and set model. *Left: Juggler*. *Right: Unicycle2*.

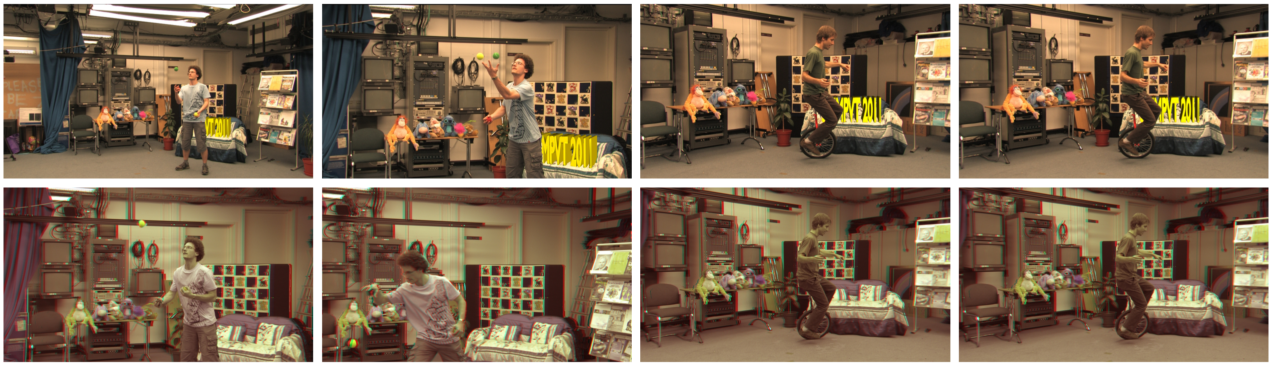

Figure 8. Sample images from the applications. *Top:* Scene augmentation. *Bottom:* Stereoscopic rendering, in red/cyan anaglyph format. *Left: Juggler*. *Right: Unicycle2*. Full video sequences are available at http://www.guillemaut.org/publications/11/Imre3DIMPVT11/videos

is then refined by the joint segmentation-reconstruction algorithm of [26]. The resulting view-dependent depth-maps are merged into a global mesh representation through Poisson surface reconstruction [27]. An incorrect principal camera calibration can chop off parts of the visual hull, and corrupt the depth maps, leading to disturbing visual artifacts. However, the actor models in Figure 7 are successfully reconstructed without any obvious artifacts. The quality of the reconstruction can also be inferred from the performance of the scene augmentation and the stereoscopic rendering applications.

3D Scene augmentation: Scene augmentation involves planting virtual objects in a real scene, and is of high importance to post-production. Any calibration errors are manifested either as a drift in the location of the virtual object, or as an incorrect occlusion. Figure 8 depicts several examples, in which the scene, as seen by the principal camera, is augmented by a virtual advertisement. The images do not exhibit any symptoms of poor calibration.

Stereoscopic 3D rendering: This application utilizes the scene geometry, and the principal camera calibration, to generate a stereoscopic sequence from a monocular input. This is achieved by synthesizing novel views of the scene for two virtual viewpoints located on either side of the principal camera. A poor calibration leads to visible artefacts such as texture mapping onto an incorrect depth layer. The images in Figure 8 appear very realistic, and are free of such problems.

V. Conclusion

This paper proposes a method for calibrating a moving camera in the presence of a dynamic object, given a witness set. This is a setup commonly encountered in film production, where accurate calibration is valuable for the post-production process. The algorithm first builds a reference structure, with respect to which the calibration is measured. The calibration measurements are then smoothed by a UKF. The algorithm can handle general and, via a novel P2P solver, nodal motion both with known and unknown intrinsics. Moreover, it can dynamically determine an active witness set to focus the processing on more promising portions of the available data. This capability makes it possible to robustly deal with dynamic scenes, by refreshing the scene model at each frame. The performance of the algorithm is demonstrated both quantitatively, and through several applications (dense 3D reconstruction, scene augmentation and steroscopic rendering).

Acknowledgment

This work is supported by the TSB project "i3Dlive: interactive 3D methods for live-action media" (TP/11/CII/6/I/AJ307D). The authors would like to thank Dr. Teo de Campos and Martin Klaudiny for their performances in *Unicycle1, Unicycle2, Juggler* and *Odzemok*.

References

[1] Z. Zhang, "A flexible new technique for camera calibration," *PAMI*, vol. 22, no. 11, pp. 1330–1334, 2000.

[2] J. Mitchelson and A. Hilton, "Wand-based multiple camera studio calibration," University of Surrey, CVSSP, Tech. Rep. VSSP-TR-2/2003, 2003.

[3] M. Pollefeys, "Automatic 3d modeling with a hand-held camera images," in *3DPVT Tutorial Notes*, 2004.

[4] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., 2003.

[5] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nolle, "Analysis and the solutions of the three point perspective pose estimation problem," in *Proc. CVPR*, 1991, pp. 592–598.

[6] G. Klein and D. W. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proc. ISMAR*, 2007, pp. 225–234.

[7] M. Farenzena, A. Bartoli, and Y. Mezouar, "Efficient camera smoothing in sequential structure-from-motion using approximate cross-validation," in *Proc. ECCV*, 2008.

[8] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *PAMI*, vol. 29, no. 6, pp. 1052–1067, 2007.

[9] N. Hasler, B. Rosenhahn, T. Thormählen, M. Wand, J. Gall, and H.-P. Seidel, "Markerless motion capture with unsynchronized moving cameras," in *Proc. CVPR*, 2009, pp. 224–231.

[10] E. Imre, J.-Y. Guillemaut, and A. Hilton, "Moving camera registration for multiple camera setups in dynamic scenes," in *Proc. BMVC*, 2010, pp. 38.1–12.

[11] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-time monocular slam: Why filter?" in *Proc. ICRA*, 2010, pp. 2657–2664.

[12] K. Josephson and M. Byröd, "Pose estimation with radial distortion and unknown focal length." in *Proc. CVPR*, 2009, pp. 2419–2426.

[13] R. Szelisky, "Image alignment and stitching: A tutorial," *Handbook of Mathematical Models in Computer Vision*, pp. 273–292, 2005.

[14] K. Kanatani, "Analysis of 3-d rotation fitting," *PAMI*, vol. 16, no. 5, pp. 543–549, 1994.

[15] N. Snavely, S. M. Seitz, and R. Szelisky, "Modeling the world from internet photo collections," *IJCV*, no. 80, pp. 189–210, 2008.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[17] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. v. Gool, "A comparison of affine region detectors," *IJCV*, vol. 65, no. 1-2, pp. 43–72, 2005.

[18] O. Chum and J. Matas, "Optimal randomized RANSAC," *PAMI*, vol. 30, no. 8, pp. 1472–1482, 2008.

[19] O. Chum and J. Matas, "Matching with prosac- progressive sample consensus," in *Proc. CVPR*, 220-226 2005.

[20] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[21] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, "Quaternion averaging," NASA Goddard Space Flight Center, Greenbelt, MD, Tech. Rep. 20070017872, 2007. [Online]. Available: ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa. ../20070017872_2007014421.pdf

[22] F. L. Markley, "Attitude error representations for kalman filtering," *Journal of Guidance, Control, and Dynamics*, vol. 2, no. 2, pp. 311–317, 2003.

[23] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *PAMI*, vol. 27, no. 10, pp. 1615–1630, 2005.

[24] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *PAMI*, vol. 16, no. 2, pp. 150–162, 1994.

[25] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum, "Background cut," in *ECCV*, vol. 3954, 2006, pp. 628–641.

[26] J.-Y. Guillemaut, J. Kilner, and A. Hilton, "Robust graph-cut scene segmentation and reconstruction for free-viewpoint video of complex dynamic scenes," in *Proc. ICCV*, 2009.

[27] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Symp on Geometry Processing*, 2006, pp. 61–70.