Moving Camera Registration for Multiple Camera Setups in Dynamic Scenes

Evren Imre
h.imre@surrey.ac.uk
Jean-Yves Guillemaut
j.guillemaut@surrey.ac.uk
Adrian Hilton
a.hilton@surrey.ac.uk

Center for Vision, Speech and Signal Processing
University of Surrey
Guildford, UK

Abstract

Many practical applications require an accurate knowledge of the extrinsic calibration (*i.e.*, pose) of a moving camera. The existing SLAM and structure-from-motion solutions are not robust to scenes with large dynamic objects, and do not fully utilize the available information in the presence of static cameras, a common practical scenario. In this paper, we propose an algorithm that addresses both of these issues for a hybrid static-moving camera setup. The algorithm uses the static cameras to build a sparse 3D model of the scene, with respect to which the pose of the moving camera is estimated at each time instant. The performance of the algorithm is studied through extensive experiments that cover a wide range of applications, and is shown to be satisfactory.

1 Introduction

This manuscript presents a method to register a moving (principal) camera to a given reference frame, by the help of a set of fully calibrated static (witness) cameras, in dynamic scenes. The term *fully calibrated* signifies known intrinsic and extrinsic parameters, and *dynamic*, independently moving and nonrigid scene elements. This is a common scenario in live broadcasting and film production, therefore the proposed method addresses a practical problem. Our ultimate aim is to equip the existing free-viewpoint video algorithms, such as [6], with the ability to exploit any available moving cameras in generic dynamic scenes, and to facilitate 3D content production by augmented reality and stereoscopic rendering [15].

The core problem, pose recovery from three 3D-2D correspondences (the *P3P problem*) predates the computer vision field by more than a century [7]. However, it still receives some attention in the form of minimal polynomial equation solvers for full calibration [11], and non-minimal, globally optimal PnP solvers [9]. P3P solvers are used in multiview structure-from-motion (SfM) algorithms, which can simultaneously recover a sparse 3D model of the scene and the camera calibration parameters for a set of static cameras [26]. These techniques can be adapted to handle monocular moving cameras [28], as an alternative to simultaneous localisation and mapping (SLAM) [4]. Recently, in [10], the monocular SfM (MSfM) approach is extended to exclusively moving multiple camera setups, by first solving an individual MSfM problem for each camera, and then merging these solutions via the

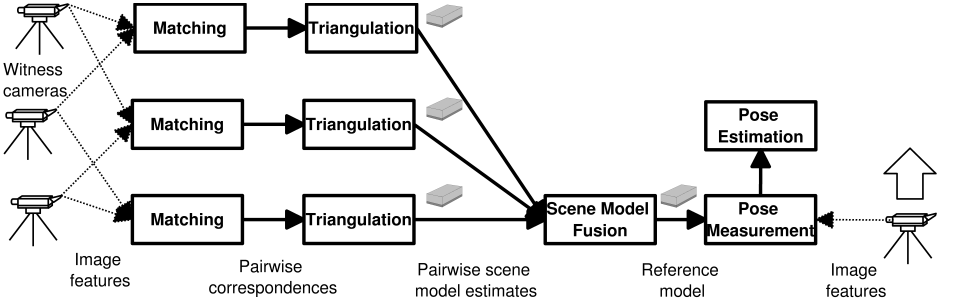

Figure 1: Overview of the proposed algorithm, for a 3 witness-1 principal camera setup.

3D-3D correspondences between the recovered sparse models. Although this is an effective strategy for combining multiple independent solutions, it is still vulnerable to the well-known SfM degeneracies, such as dominant planes and insufficient motion [8]. Moreover, MSfM techniques are not robust to large dynamic elements: In our experiments, Boujou [2] failed when the dynamic elements occupied more than 20-30% of the image. Such limitations are the reason why commercial matchmovers still retain extensive marker tracking and manual editing capabilities [1], despite the maturity of the available monocular solutions.

The proposed algorithm, illustrated in Figure 1, diverges from the literature by its use of a sparse scene model, *i.e.*, a *reference model*, computed from a set of fully calibrated static cameras. Then, it solves a P3P problem at every time instant, to obtain a measurement of the pose of the principal camera with respect to this model. An unscented Kaman filter (UKF) [12] smooths these measurements for jitter removal. The algorithm is robust to the MSfM issues mentioned above: The only degeneracy is the unlikely case of the entire reference structure lying on a line. And although a P3P solver requires known intrinsics for the principal camera, it provides resilience to occlusions by reducing the size of the minimal sample, and facilitates the use of the algorithm in dynamic scenes.

The rest of the paper is organized as follows: In the next section, the details of the proposed method are discussed. Then, its performance is experimentally evaluated in Section 3. The conclusions are presented in Section 4.

2 Proposed Method

2.1 Building a Reference Model

The first part of the algorithm locates a set of salient 3D point features in the scene. A scene feature is a triplet of the form $\{\mathbf{X}, \mathbf{G_X}, \mathbf{D}\}$, where $\mathbf{X}$ is the 3D scene coordinates of the feature, $\mathbf{G_X}$, its covariance, and $\mathbf{D}$, a 3D feature descriptor. Each observation of a scene feature by a witness camera has an associated 2D image feature, defined by a triplet $\{\mathbf{x}, \mathbf{g_x}, \mathbf{d}\}$, or, by its 2D image coordinates, covariance, and descriptor. All image features originating from a certain scene feature form a *correspondence cluster*, a connected graph with at most 1 vertex from each witness image (Figure 2.1). Each link in a cluster is an image feature correspondence. A cluster is sufficient to determine all components of a scene feature.

Since our target applications often require the witness cameras to be deployed in a way to maximize the scene coverage, wide baseline conditions prevail, for which Hessian-affine [21] features with SIFT descriptors [18] are recommended [24]. The covariance is assumed to be isotropic and identical for all image features, in accordance with [13].

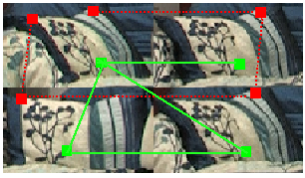

Figure 2: *Green:* A correspondence cluster, with two missing pairwise correspondences. *Red:* An inconsistent set, as it has two elements elements coming from the same image.

For each camera pair, the corresponding image features are identified as the pairs satisfying the following constraints:

- **Geometric consistency:** The pair is sufficiently conformant to the epipolar constraint, *i.e.*, has a Sampson error below a certain threshold [8].
- **Similarity:** The inverse Euclidean distance between the SIFT descriptors is above a certain threshold. In the wide baseline case, this constraint, by itself, has limited reliability.
- **Uniqueness:** The ratio of the similarity scores of the best and the second best candidate is above a threshold. This constraint eliminates ambiguous matches [18].
- **Reciprocity:** Both features are the best matches to each other [25].

The resulting high quality correspondence set is further processed to remove any inconsistencies (Figure 2.1, red/dashed), and to construct the correspondence clusters (Figure 2.1, green/solid). Ideally, each cluster belongs to a distinct scene feature. However, the wide baseline nature of the problem leads to missing links and, occasionally, fragmented clusters.

Finding the scene feature associated with each cluster can be posed as an N-view triangulation problem, for which optimal solutions exist [9]. However, the wide baseline assumption admits a simpler solution. As the first step, each cluster is transformed into a 3D point cloud, by converting the individual *links* to 3D features, as follows:

- **Coordinate (X) :** Computed from the 2D image coordinates via the optimal triangulation algorithm [8].
- **Covariance ($\mathbf{G_x}$) :** The covariance of the 2D image coordinates is propagated through the optimal triangulation operation by using the unscented transformation (UT) [12], which involves triangulating a judiciously selected set of samples representing the statistics of the input parameters (i.e, a 4D vector, the coordinates of the image feature pair), and computing the sample statistics of the resulting 3D points. The UT is superior to using Jacobians, as it approximates the transformed distribution, instead of the transformation, hence, can handle nonlinear functions more accurately [12].
- **Descriptor (D):** A 3D feature descriptor is simply the pair of descriptors belonging to the members of the link, *i.e.*, $\mathbf{D} = \{\mathbf{d_0}\,\mathbf{d_1}\}$.

Each member of the point cloud is a measurement of the scene feature. The second step involves recovering an estimate of the latter, from the former.

The coordinate and the covariance components of the scene feature are estimated by averaging over the point cloud via a Kalman filter (KF). A KF is the optimal estimator for Gaussian and independent measurements. The wide baseline correspondences ensure the validity of the Gaussianity assumption on the uncertainty of the coordinate measurements [23]. However, the independence assumption does not hold (as an image feature may appear in more than one link), rendering the KF solution a suboptimal fusion of pairwise optimal measurements. Nonetheless, KF is superior to ordinary mean, which does not take the uncer-

tainty information into account, and to simply picking the best measurement, which does not utilize the available uncertainty information. Another alternative, nonlinear minimization of the reprojection error across multiple views with respect to the scene feature coordinates, is known to be vulnerable to local minima [9], and would provide a poorer covariance estimate.

The descriptor component is a list of the descriptors of all members of the cluster (*i.e.* $\mathbf{D} = \{\mathbf{d_0}...\mathbf{d_{N-1}}\}$), which is a simple alternative to more sophisticated descriptors, such as [29]. It is an appropriate choice for the wide baseline case, as the descriptors belonging to different viewpoints are less likely to be redundant.

2.2 Estimating the Pose Trajectory

In this section, we seek to recover the *pose trajectory* of the principal camera. A pose is a triplet of the form $\{\mathbf{C}, \mathbf{q}, \mathbf{P}\}$, where $\mathbf{C}$ and $\mathbf{q}$ denote the camera position and the orientation quaternion, respectively, and $\mathbf{P}$ is their covariance. A pose trajectory is a sequence of poses belonging to a camera. It is estimated by first computing the instantaneous estimates of the pose, *i.e.* the pose measurements, at each time instant, and then filtering them via a UKF.

$\mathbf{C}$ and $\mathbf{q}$ are computed by using the *automatic geometry estimator* of [8], which needs an initial 3D-2D correspondence set. This set is constructed by identifying the correspondences between the image features extracted from the principal camera image and the scene features of the reference model. The matching algorithm enforces the constraints described in Section 2.1, except for the geometric constraint, as it is not available at the first pass. Also, it employs a different similarity metric, which is defined as

$$s = \max_{\mathbf{d} \in \mathbf{D}} \|\mathbf{d_p} - \mathbf{d}\|^{-1}, \tag{1}$$

where $\mathbf{d_p}$ is the descriptor of a principal camera image feature. In other words, a principal camera feature is matched to the most similar member of a correspondence cluster.

Three of the resulting 3D-2D correspondences are sufficient to compute the pose via a P3P solver. This triplet, and the corresponding pose measurement, are identified robustly by using SPRT-RANSAC [3], a variant of RANSAC that can quickly spot and discard unpromising hypotheses. In order to construct the hypotheses, Finsterwalder's method, a minimal polynomial solver with superior numerical stability [7], is employed. For the refinement of the pose measurement, Powell's dog leg algorithm is preferred over Levenberg-Marquardt, as it offers the same accuracy at lower computational complexity [17]. Finally, the recovered $\{\mathbf{C}, \mathbf{q}\}$ pair is supplied as a guide to the matching algorithm, enforced by a reprojection error constraint [8], and the estimation- guided matching steps are iterated until convergence.

As for the covariance, since P3P is a highly nonlinear operation, the UT is a more suitable choice than the first-order approximation. However, the implementation is not straightforward: The sample mean of a set of unit quaternions is not necessarily a unit quaternion, and does not recognize the fact that $\mathbf{q}$ and $\mathbf{-q}$ represent the same rotation. In order to avoid these issues, the sample mean is redefined as "the quaternion corresponding to the rotation matrix, which has the minimum total squared residue with the sample in the Frobenius norm sense" (as opposed to the ordinary vector mean minimizing the total squared Euclidean norm of the residues), as proposed in [20]. Another problematic property of the quaternion representation is its redundancy (*i.e.*, a 4D vector with 3 degrees of freedom), which implies a singular covariance matrix. This is remedied by employing the axis-angle form to represent the orientation uncertainty [19].

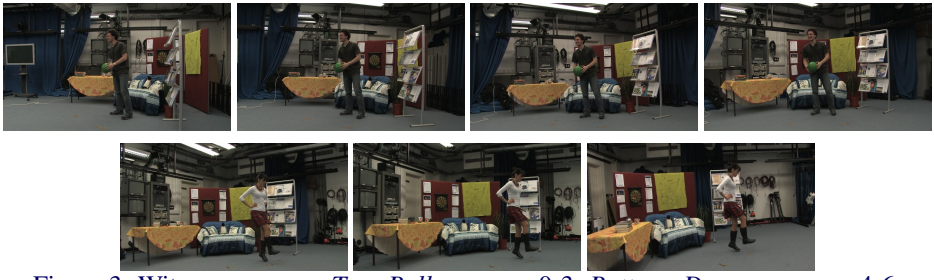

Figure 3: Witness cameras. *Top: Ball*, cameras 0-3. *Bottom: Dance*, cameras 4-6.

Since the pose measurements are computed independently at each time instant, a jitter in the pose trajectory is likely, which can be filtered out by a sequential state estimator. For this task, the UKF is chosen over EKF, as it replaces the linearisation involved in the time and measurement update steps by the UT [12], hence, can deal with large rotations more successfully. It also has the added benefit of providing a more accurate estimate of the covariance of the pose trajectory.

The UKF employs a constant translational and angular velocity model, *i.e.*,

$$\begin{aligned} \mathbf{C_{t+1}} &= \mathbf{C_t} + \delta_\mathbf{t} \\ \mathbf{q_{t+1}} &= \mathbf{q_t} \otimes Q(\phi_\mathbf{t}) \\ \delta_{\mathbf{t+1}} &= \delta_\mathbf{t} + v \\ \phi_{\mathbf{t+1}} &= \phi_\mathbf{t} + w \end{aligned} , \tag{2}$$

where δ and ϕ denote the translational and angular velocity, respectively, affected by independent Gaussian noise processes v and w. Q is an operator that maps an axis-angle vector to a quaternion, and $\otimes$ is the quaternion multiplication operator. The measurement function is identity, corrupted by a Gaussian noise process.

3 Experimental Results

The behavior and the performance of the proposed algorithm is studied qualitatively by analysing the variation of the pose trajectory with the size of the dynamic elements in the scene, and the number of cameras; and qualitatively through a number of applications, namely, free-viewpoint video, scene augmentation and stereoscopic rendering, all of which are highly sensitive to pose errors. The data used in the experiments is two indoor sequences, *Ball* and *Dance*, captured by a set of one hand-held principal camera, and 7 witness cameras, all having a resolution of 1920x1080. The sequences are 750 and 665 frames long, respectively, and both feature an actor in front of a static scene, but performing a different routine: In *Ball* the actor stays roughly at the same spot, whereas in *Dance*, the actress periodically moves across the scene, therefore occluding different parts of the reference model. The actors occupy about 5% of the images. The intrinsic parameters for the setup are estimated by using a calibration chart [30], whereas for the extrinsics of the witness cameras, [22] is used. Figure 3 is a sample from the witness cameras. The camera layout is depicted in Figure 4, together with the reference models and the estimated trajectories for *Ball* and *Dance*.

In the following discussion, the term *foreground* is used synonymously with the dynamic elements, whereas the static elements are referred to as *background*.

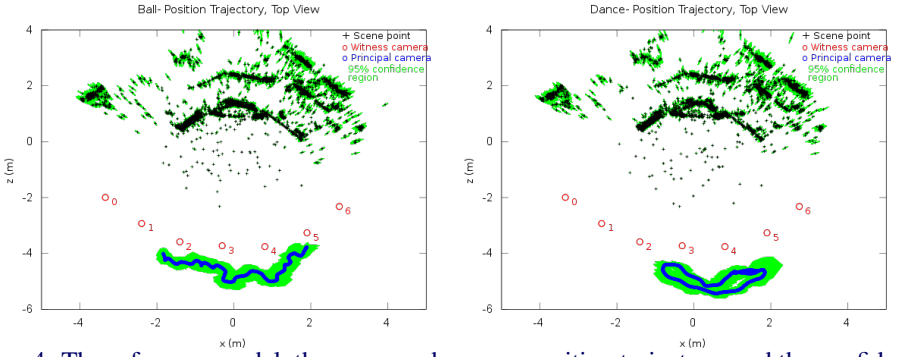

Figure 4: The reference model, the recovered camera position trajectory, and the confidence regions, as estimated by KF and UKF, respectively. Top view. *Left:* Ball. *Right:* Dance.

3.1 Pose Trajectory Estimation

In the absence of a ground truth pose, a direct quantitative evaluation of the registration error in the results presented in Figure 4 is not possible. However, we identified two indirect measures of performance:

- **Reprojection error:** An accurate pose estimate should be able to explain the 3D-2D correspondences not used in its computation as well, an observation whose validity is established by the success RANSAC. In accordance with this observation, for each sequence, we randomly removed half of the features in the reference model to form a test set, and estimated the pose trajectory by using the remaining half. Then, we computed the reprojection error over all visible features across the sequence (362 and 373 features/image on the average for *Ball* and *Dance*, respectively). If we had the ground truth pose and the reference structure, assuming a unit-variance image coordinate noise, the reprojection error would be distributed as χ^2 with 2 degrees of freedom. A comparison of this "ideal" and the measured error (Table 1) reveals that most of the observed error can be attributed to the inaccuracies in feature localization, and the pose estimates are highly reliable (obviously, the validity of this conclusion relies on that of the assumption on the magnitude of the noise on the image coordinates).
- **Comparison with Boujou:** In order to provide an MSfM alternative, the pose trajectories for the sequences are estimated via Boujou, in addition to the proposed algorithm. Then, the similarity transformation between the reference frame of the witness cameras, and that of Boujou is recovered by using the 3D-3D correspondences between the trajectories. Table 2 shows that, when transferred to the witness reference frame, the trajectories are closely aligned. This result should be interpreted with care: A close alignment does not imply anything on the accuracy of the registration with respect to the reference model, as, if a global similarity bias, such as a shift, existed, it would be incorporated into the mapping between the two reference frames, and hence leave no trace on the pose difference. On the other hand, for *relative* pose estimates, any global bias cancels out. Therefore, Table 2 indicates that Boujou and our algorithm have consistent relative pose estimates. Their accuracy in the witness reference frame follows from the reprojection error experiment.

MSfM algorithms are susceptible to large foreground objects, due to occlusions and mismatches they introduce. The deterioration of the performance can be quite rapid: In our

	5%	25%	Median	75%	95%
Ball (pixel^2)	0.083	0.519	1.402	2.935	5.221
Dance (pixel^2)	0.085	0.561	1.568	3.319	5.680
Ideal (pixel^2)	0.010	0.575	1.386	2.773	5.989

Table 1: Order statistics of the reprojection error. "Ideal" is the case of unit-variance image coordinate noise and error-free pose estimate.

	5%	25%	50%	75%	95%
Ball- Position difference (m)	0.003	0.005	0.008	0.012	0.028
Ball- Principal axis difference (degrees)	0.347	0.405	0.445	0.490	0.672
Dance- Position difference (m)	0.003	0.006	0.009	0.012	0.018
Dance-Principal axis difference (degrees)	0.135	0.229	0.371	0.420	0.481

Table 2: Order statistics of the pose difference. 50th percentile is the median.

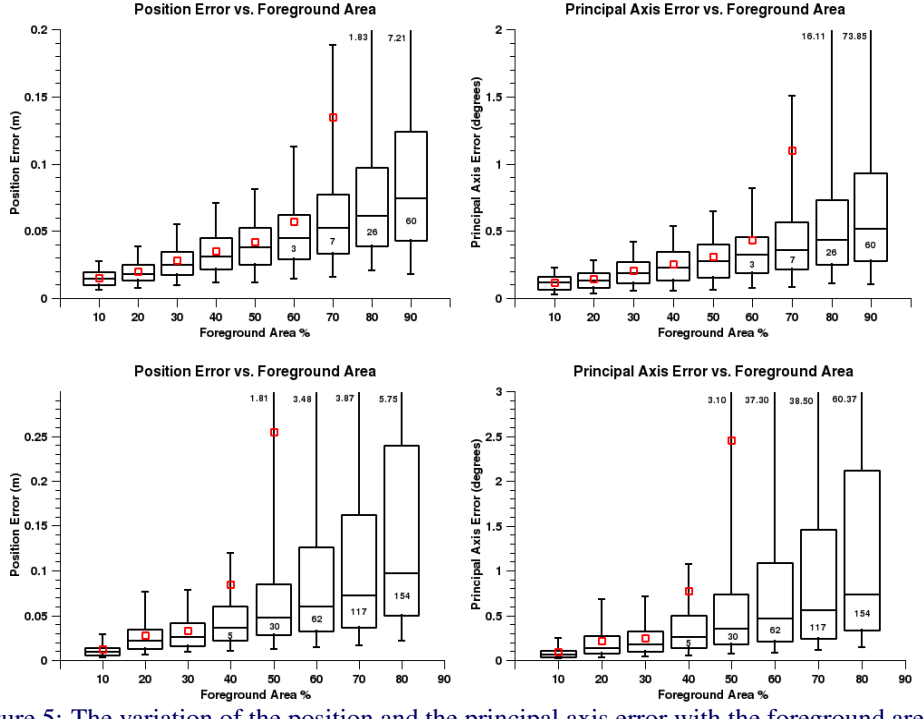

Figure 5: The variation of the position and the principal axis error with the foreground area. *Top: Ball*. *Bottom: Dance*. The numbers within the boxes indicate the count of the instances in which the algorithm failed to find a solution. The numbers at the upper boundary are the 95th percentile points.

experiments, Boujou failed to return an acceptable estimate at foreground area-to-image ratios beyond 20% and 30% for *Dance* and *Ball*, respectively. In order to understand the sensitivity of our algorithm to this issue, we simulated a tighter framing, by discarding all features outside of a bounding box around the foreground object in the principal camera images. The size of the bounding box is adjusted to match a specified foreground-to-image ratio. In each case, the performance is compared against that of the original sequence, *i.e.* the 5% foreground ratio case. The results in Figure 5 show that the performance remains acceptable until 40-60%, where the first instances of estimation failures appear. Beyond that, the

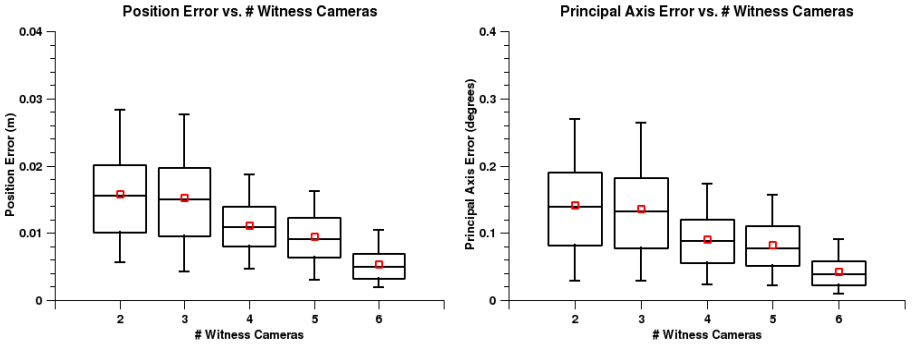

Figure 6: The variation of the position and the principal axis error with the size of the witness camera set, for *Ball*

quality of the estimates decrease dramatically, especially for *Dance*, which features a more challenging camera trajectory. Nevertheless, the median error still remains below 15 cm and 1 degree for the position and the principal axis error, respectively.

In order to assess the effect of the size of the witness set on the performance of the algorithm, the cameras are removed successively, in a way to maintain both high coverage, and a high overlap with the field-of-view (FoV) of the principal camera, *i.e.*, with a strategy that neither strongly favours, nor totally ignores the needs of the algorithm (the exact sequence is 3, 5, 1, 6 and 0). As in the foreground area experiments, the error is defined as the difference with the 7-witness case. The results, presented in Figure 6, show that the performance of the algorithm is effectively independent of the number of witness cameras, as long as a reasonable FoV overlap is maintained (as provided by Cameras 2 and 4). However, the rise in the error observed by the removal of Camera 6 (*i.e.*, the 3/4-witness transition) indicates the contribution of the peripheral cameras to the estimation process.

3.2 Applications

The qualitative assessment of the algorithm is performed through a number of computer vision/graphics applications, namely, free-viewpoint video, augmented reality and stereoscopic rendering, successful operation of which are closely related to the quality of the camera calibration. The other prerequisite for these applications is accurate multiple-view reconstruction, which is briefly discussed below, before presenting the experiment results.

The first stage of the reconstruction pipeline utilizes the background-cut algorithm [27] to compute an initial segmentation of the foreground objects for the witness cameras, by the help of the background images automatically extracted from the data. This segmentation is necessary to build a coarse scene model, via a visual hull reconstruction algorithm [16]; in order to mitigate potential artefacts due to segmentation errors, a conservative implementation is used. The next stage, the joint refinement of the segmentation and the scene model, is performed by using the dense reconstruction algorithm of [6], which computes a layered depth estimate for each camera through graph-cut minimization of a cost function involving colour, contrast, similarity and smoothness terms, over the entire camera set (including the principal camera). Finally, the individual depth maps are fused into a single 3D mesh representation for each time instant, through Poisson surface reconstruction [14].

The pipeline described above is used at each time instant for actor modelling, however, only once for set modelling, as the latter is static. As seen in Figure 7, the reconstructed scene

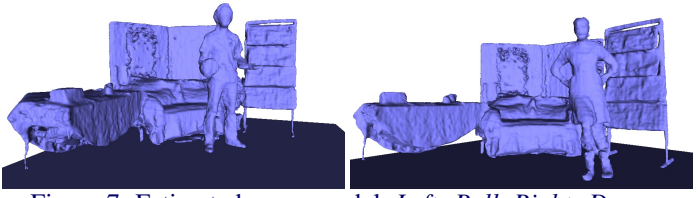

Figure 7: Estimated scene model. *Left: Ball. Right: Dance.*

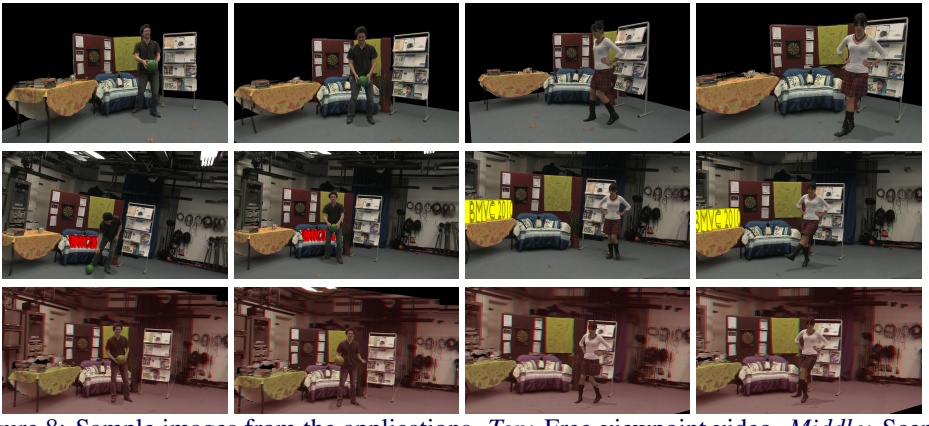

Figure 8: Sample images from the applications. *Top:* Free-viewpoint video. *Middle:* Scene augmentation. *Bottom:* Stereoscopic rendering, in red/cyan anaglyph format. *Left:* Ball. *Right:* Dance. Full video sequences are provided as supplementary material, and at `http://www.guillemaut.org/publications/10/ImreBMVC10/videos/`

models are free from any severe artefacts, such as missing body parts or large protrusions, typical symptoms of poor extrinsic calibration.

Application 1 – Free-viewpoint video: The estimated scene model is used to produce views from novel (virtual) viewpoints via the view-dependent texture mapping method proposed in [8]: Each pixel in the virtual image is textured by blending the colours observed in the two nearest cameras, weighted by the angles separating each camera from the virtual camera. Figure 8 presents some sample images synthesized for a number of virtual viewpoints. In this application, an incorrect pose estimate would manifest itself through blurring or distortions, caused by the novel view texture rendered from the image patches belonging to different parts of the foreground object. However, the absence of obvious artefacts, and the seamless transition from the real world seen by the principal camera to the virtual world imply a reliable camera pose estimate.

Application 2 – Scene augmentation: Scene augmentation involves incorporating virtual objects into a real-world scene. An erroneous pose estimate would introduce a drift or an instability in the apparent image position of the virtual object with respect to the real image content. Moreover, it would deteriorate the scene model estimate, and therefore lead to incorrect occlusions. An example of this application can be seen in Figure 8 where a virtual advertisement has been added to the principal camera's video sequence, as well as virtual shadows cast by the actor. No jitter or poor occlusion performance is observed in the position of the advertisement, due to the accuracy of the camera pose estimate.

Application 3 – Stereoscopic rendering: In this application, the estimated scene model (Figure 7) is used to convert the principal camera's monoscopic output into a stereoscopic sequence. This is achieved by synthesising two virtual camera viewpoints located on either

side of the principal camera. Example images can be seen in Figure 8. The synthesised video appears very realistic and does not show any significant artefacts such as the texture mapping onto an incorrect depth layer, which would occur with an inaccurate calibration.

4 Conclusion

This paper presents an algorithm to estimate the pose of a moving camera in a dynamic scene, by the help of a set of fully calibrated witness cameras. This is accomplished by first using the witness cameras to build a reference model, then solving the P3P problem with respect to this model, and finally eliminating the jitter via an UKF. The proposed algorithm addresses a case commonly encountered in practice, and is shown to be remarkably robust to large foreground objects. It also fills the gap between the monocular, and the more general, multiple moving camera techniques. The method has two limitations:

- The algorithm assumes constant and known intrinsics for the principal camera throughout the entire shot. However, this can be remedied by replacing the P3P solver with [7].
- The span of the principal camera motion is ultimately limited by the coverage of the witness cameras. The solution to this problem lies in the realm of SLAM.

However, within its application domain, *e.g.*, free-viewpoint video, scene augmentation and stereo rendering, it has a satisfactory performance, as demonstrated through experiments.

Acknowledgements: This work was supported by TSB/EPSRC project "i3Dlive: interactive 3D methods for live-action media" (TP/11/CII/6/I/AJ307D, TS/G002800/1). The authors wish to thank Martin Klaudiny and Cemre Zor for their performances in *Ball* and *Dance*, respectively.

References

[1] 3d equalizer. URL http://www.sci-d-vis.com/.

[2] Boujou. URL www.2d3.com.

[3] O. Chum and J. Matas. Optimal randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1472–1482, August 2008.

[4] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1052–1067, 2007.

[5] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *SIGGRAPH*, pages 11–20, 1996.

[6] 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.

[7] 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*, pages 592–598, 1991.

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

[9] R. I. Hartley and F. Kahl. Optimal algorithms in multiview geometry. In *Lecture Notes in Computer Science*, volume 4843, pages 13–34, 2007.

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

[11] K. Josephson and M. Byröd. Pose estimation with radial distortion and unknown focal length. In *Proc. CVPR*, pages 2419–2426, 2009.

[12] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proc. IEEE*, 92(3):401–422, March 2004.

[13] Y. Kanazawa and K. Kanatani. Do we really have to consider covariance matrices for image feature points? *Electronics and Communication in Japan*, 86(1), 2003.

[14] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Symp on Geometry Processing*, pages 61–70, 2006.

[15] S. Knorr, M. Kunter, and T. Sikora. Super-resolution stereo- and multi-view synthesis from monocular video sequences. In *Proc. 3DIM*, 2007.

[16] A. Laurentini. The visual hull concept for silhouette-based image understanding. *PAMI*, 16(2):150–162, 1994.

[17] M. I. A. Lourakis and A. A. Argyros. Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment?. In *ICCV*, pages 1526–1531, 2005.

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[19] F. L. Markley. Attitude error representations for kalman filtering. *Journal of Guidance, Control, and Dynamics*, 2(2):311–317, March-April 2003.

[20] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman. Quaternion averaging. Technical Report 20070017872, NASA Goddard Space Flight Center, Greenbelt, MD, 2007. URL ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.../20070017872_2007014421.pdf.

[21] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. v. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

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

[23] J. Montiel, J. Civera, and A. Davison. Unified inverse depth parametrization for monocular slam. In *Proc. Robotics: Science and Systems*, 2006.

[24] P. Moreels and P. Perona. Evaluation of feature detectors and descriptors based on 3d objects. In *Proceedings of 10th IEEE international Conference on Computer Vision*, volume 3, pages 800–807, October 2005.

[25] D. Nistèr, O. Narodistsky, and J. Bergen. Visual odometry. In *Proc. CVPR*, 2004.

[26] M. Pollefeys. Automatic 3d modeling with a hand-held camera images. In *2nd International Symposium on 3D Data Processing, Visualization and Transmission, Tutorial Notes*, 2004.

[27] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum. Background cut. In *ECCV*, volume 3954, pages 628–641, 2006.

[28] P. H. S. Torr, A. W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*, 32(1):27–44, 1999.

[29] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3d model matching with viewpoint-invariant patches (vip). In *CVPR*, pages 1–8, 2008.

[30] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000.