Hidden View Synthesis using Real-Time Visual SLAM for Simplifying Video Surveillance Analysis

C. Mei
LAAS-CNRS

E. Sommerlade, G. Sibley, P. Newman, I. Reid
University of Oxford

Abstract— **Understanding and analysing video data from static or mobile surveillance cameras often requires knowledge of the scene and the camera placement. In this article, we provide a way to simplify the user's task of understanding the scene by rendering the camera view as if observed from the user's perspective by estimating his position using a real-time visual SLAM system. Augmenting the view is referred to as hidden view synthesis. Compared to previous work, the current approach improves by simplifying the setup and requiring minimal user input. This is achieved by building a map of the environment using a visual SLAM system and then registering the surveillance camera in this map. By exploiting the map, a different *moving* camera can render hidden views in real-time at 30Hz. We discuss some of the challenges remaining for full automation. Results are shown in an indoor environment for surveillance applications and outdoors with application to improved safety in transport.**

I. INTRODUCTION

When robots are used in remote locations or when an area is under surveillance, feedback is often provided to the user using video data. Analysing this data can be a challenging task and requires an understanding of the underlying scene and expert knowledge. Overhead views can help obtain an overview of the scene but necessitate a method of describing or summarising the objects in the environment - which is itself challenging - and details are often lost. An alternative that we advocate in this work is hidden view synthesis or see-through approaches. These provide an attractive and simple way of presenting the user with simplified visual feedback from different cameras.

The technical difficulty of hidden view synthesis lies in estimating accurately the changing position of the user in real-time, finding his position with respect to the surveillance camera and rendering a realistic (and understandable) view. Although rendering novel views by pose estimation is not new, combining visual SLAM (Simultaneous Localisation and Mapping) with place recognition to overcome the technical challenges and provide a scalable and simple approach has not been done before. Previous methods have either required building a CAD model of the environment or used an extra camera for the registration.

The proposed setup consists of a camera held by a user and moving freely in the environment. This camera will be referred to as the user's camera or the SLAM camera. We wish to enrich this view with images from different cameras in the environment to enable synthesising hidden views. Theses cameras will be called 'surveillance' cameras as this is one of the main target applications.

After describing related work in Section II, we will give an overview of the system in Section III. The technical aspects of registering a given map with a surveillance camera view will be detailed in Section IV, with the problem of relocalisation in Section V. Finally we will show some experimental results indoors and outdoors (Section VI) and discuss some of the challenges and future work (Section VII).

II. RELATED WORK

Previous work on hidden view synthesis has often required a complex setup stage or extensive user input. In [12], the authors use markers in the environment to experiment with see-through capabilities for vehicle navigation. A year later, in [7], the authors describe a mixed reality system that combines surveillance cameras with a GPS/Inertial/monocular system to provide the user with a video feedback of occluded scenes. They require a CAD model of the environment and calibrated surveillance cameras. The camera localisation is done at the user's demand and the system then uses the output of the inertial sensor to provide the position for the video feedback until the system drifts and a new localisation process is instantiated.

Synthesising hidden views that look realistic or are effective from the user's perspective is also an important aspect of this research. In [1], the authors evaluate the quality of see-throughs.

Multiple 2-D planes can also be exploited to provide a 3-D experience as in [13] and [9]. This requires multiple cameras to be effective.

The closest related work is the recent approach proposed by Barnum *et. al* [2]. The authors show how by exploiting projective invariants (and in particular homologies), it is possible to render realistic looking views in presence of dynamic moving objects that do not belong to a specific plane. Our approach can be seen as complementary. We show how a SLAM system can provide a method for transferring information between the surveillance camera and the user's view. This simplifies the tracking and avoids the use of a transfer camera as in to provide for the positioning of the reference view. Our method can benefit directly from the approach by [2] for modelling out-of-plane objects.

III. OVERVIEW

To synthesise a realistic view from a surveillance camera in the user's view, it is necessary to transform the image according to the observed scene. In the general case, this is a complicated task as it requires the full 3-D modelling of

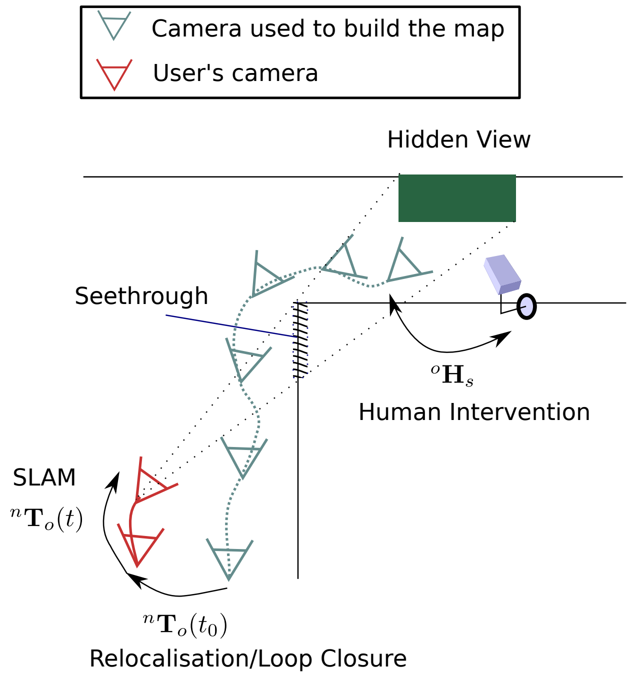


Fig. 1. The user's camera cannot see the hidden view so we require an indirect way of computing its position with respect to the surveillance camera. The idea proposed in this article is to build a map beforehand and register one of its views with the surveillance camera (this gives us $^s\mathbf{H}_o$). At run-time, the system relocalises with the map (this provides $^n\mathbf{H}_o$) and can then transfer the hidden view live to the camera.

the scene. A simpler approach which is often exploited for hidden view synthesis is to assume the observed scene is planar or that the observed objects are distant. We can then exploit the projective property of planar homographies that links all image points belonging to a plane between views through a 3×3 matrix (with 8 degrees of freedom) [6]. The choice of plane is important as it will affect the quality of the rendered view. A typical choice is a background wall, the rendering will then look realistic for objects close to the plane (this constraint can be partly lifted by using homologies [2]). Let $\mathbf{p}_s$ and $\mathbf{p}_n$ be the projections of points on a plane in the surveillance camera's view and the user's view respectively. Let $^s\mathbf{H}_n$ be the associate homography: $\mathbf{p}_s = {}^s\mathbf{H}_n\mathbf{p}_n$.

Our task is now to evaluate $^s\mathbf{H}_n$ at any given time and then use the homography to render the synthesised view. What makes this challenging is the need for a way to evaluate the position of the camera at every time-step *without* observing the same part of the scene. This could be achieved using a *transfer* camera as in [2] but this is not always possible in practice. An alternative is to evaluate the current position of the camera with respect to an old view of the scene taken in a similar place by a different camera whose position was known with respect to the surveillance camera view. This is possible for example if a map of the environment has been built beforehand and this is the approach we propose. As depicted in Fig. 1, our method involves four steps, the first two "prepare" the environment for hidden view synthesis and only need to be done once (or when the environment changes significantly). The last two steps happen at run-time to provide the user with the visual feedback from the surveillance camera:

1) Build a SLAM map. Several SLAM systems have been proposed in the literature (e.g. [5], [8], [11]). Figure 7 shows an example of a map built for hidden view synthesis in an indoor scene.
2) Associate one of the keyframes of the map with the surveillance camera's view. Identifying candidate views can be done automatically but we found that the final geometric association was very challenging in practise and manual intervention was required (Section IV). This step provides a homography $^s\mathbf{H}_o$ between a planar region in the surveillance camera view and the map.
3) Relocalise/Localise. At run-time, we start by relocalising the camera with respect to the old map and then track the map to find the $\mathbb{SE}(3)$ transform $^n\mathbf{T}_o$ between the current position and the pose registered with the surveillance camera. We can then compute the desired homography $^s\mathbf{H}_n$ that links the surveillance camera to the see-through view.
4) Synthesise the hidden view. Given the homography, we can now warp the surveillance camera's image to enhance the user's experience. The images are blended by weighted averaging to provide a see-through effect.

This setup is motivated by the ease of use of building a map using a SLAM system without requiring any artificial fiducials. It can be seen as an alternative to building a CAD model as in [7]. It should be noted that pre-calibration of the surveillance cameras is not required. (In fact, an added benefit of a SLAM system is to provide a mean of calibrating the surveillance cameras.)

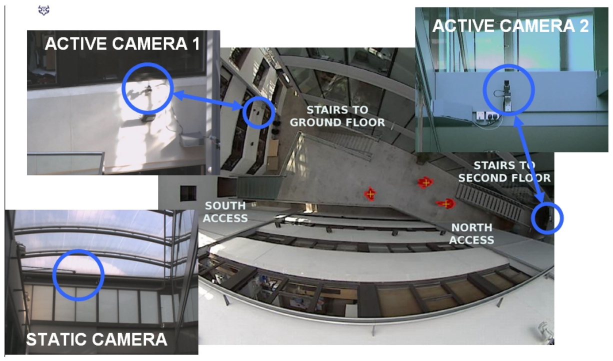
a) SLAM system: In this work we used the RSLAM stereo system described in [11]. Using a stereo system proved more robust in indoor scenes compared to available monocular solutions and did not require an initial stage to provide a baseline as in [8]. The map building and localisation task runs at frame rate (30Hz) and scales to large environments enabling the proposed viewing method to be applied to surveillance tasks in large buildings or outdoors environments.

b) Surveillance system: Figure 2 shows the position of the different cameras used in the indoor experiments and the architecture of the network surveillance system. The cameras provide live video images through the network that are also saved in a SQL database for further processing or monitoring tasks. More details of the system can be found in [3].

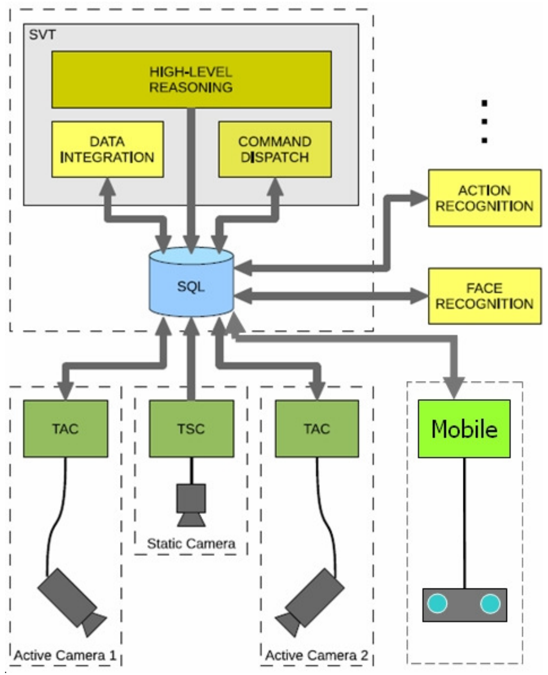
We will now discuss in more detail step 2. (Section IV) and step 3. (Section V) that are similar in objective but required developing different methods in practice.

IV. ASSOCIATING THE SLAM AND SURVEILLANCE CAMERAS

Finding a correct association between the surveillance camera and one of the keyframes proved to be very challenging in practice. A naive registration using SIFT features



(a) Position of the different cameras in the surveillance area



(b) Surveillance system architecture

Fig. 2. The indoor experiments were undertaken in an environment equipped with three surveillance cameras (Fig. 2(a)) that provide live video data through the network and save the images in a SQL database (Fig. 2(b)). This setup makes it possible to also analyse the video data for other tasks such as action and face recognition.

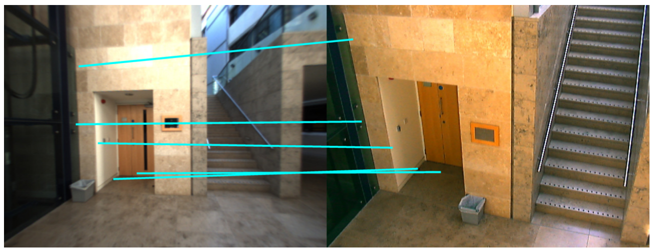


Fig. 3. Finding feature correspondences between a camera on the ground and images taken by a surveillance camera proved to be challenging. Out of 850 extracted SIFT features, only 5 were matched using the default settings. (The code used is D. Lowe's online implementation [10].)

[10] did not generally provide enough reliable matches (Fig. 3). This can be explained by the large changes in viewpoint between views from the ground and images provided by the surveillance cameras. The strong changes in illumination, different noise levels and focal length are other sources of failure. As this step only needs to be done once and is important to provide good quality novel view rendering, we developed a user interface for the registration. We consider this registration step to be part of the initial setup phase together with the building of the SLAM map.

Figure 3 shows an example of features being associated between two views. Each point selected by the user is refined to subpixel precision. In the case of highly structured scenes with little texture, we provide an interface to compute the homography from lines.

The planar homography ${}^{o}\mathbf{H}_s$ can be found by solving the least-square problem resulting from the constraints provided by the associated points $(\mathbf{p}_s, \mathbf{p}_o)$ or lines $(\mathbf{l}_s, \mathbf{l}_o)$ selected by the user [6] (with $\mathbf{H}^{-\top}$ the inverse of the transpose of $\mathbf{H}$):

$$\mathbf{p}_o = {}^{o}\mathbf{H}_s \mathbf{p}_s, \quad \mathbf{l}_o = {}^{o}\mathbf{H}_s^{-\top} \mathbf{l}_s$$

The user is also asked to select a region of interest to mask out the parts that do not belong to the registration plane (Fig. 5(a), 5(b)). After these two steps, we have the homography ${}^{o}\mathbf{H}_s$ between a keyframe and the surveillance camera but we have not linked the transforms to the SLAM map. The final step consists in projecting the SLAM landmarks in the vicinity of the keyframe and removing those that do not belong to the region of interest. Figure 5(c) shows the map points projected in the view in red crosses. Those belonging to the region of interest are circled in blue. We still need to ensure the points belong to the plane as the region of interest can encompass non-planar parts (as is the case in Fig. 5(c)). We thus apply a RANSAC step to find robustly the best planar fit. The green diamonds in Fig. 5(c) show the remaining landmarks. Let L_o denote this set of landmarks.

The landmarks L_o belonging to the plane can be used to find the final homography used to synthesise the hidden view. If we assume we have managed to register with respect to the old map (Section V), at each time step t we know the position of the user's camera with respect to the SLAM map and thus the $\mathbb{SE}(3)$ transform ${}^{n}\mathbf{T}_o(t)$. We can thus transfer the landmarks from the registered keyframe to the current frame: $L_n(t) = {}^{n}\mathbf{T}_o(t) L_o$. By projecting in the image we can compute the interframe homography ${}^{n}\mathbf{H}_o(t)$ from:

$$\mathrm{Proj}(L_n(t)) = {}^{n}\mathbf{H}_o(t)\mathrm{Proj}(L_o)$$

and compose to find the final desired homography between the user's view and the surveillance camera:

$${}^{n}\mathbf{H}_s(t) = {}^{n}\mathbf{H}_o(t){}^{o}\mathbf{H}_s$$

with Proj the world-to-image projection function.

V. RELOCALISATION

At runtime, we need to relocalise the camera with respect to the old map to find an initial value ${}^{n}\mathbf{T}_o(t_0)$ and compute ${}^{n}\mathbf{H}_s(t_0)$ for rendering. The subsequent estimates ${}^{n}\mathbf{T}_o(t)$ are

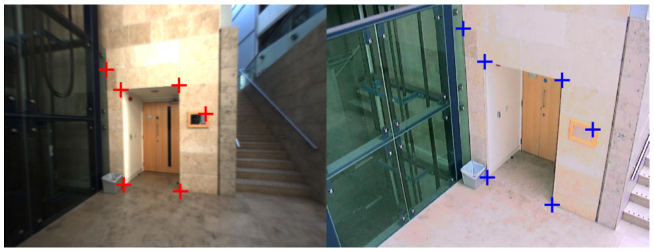
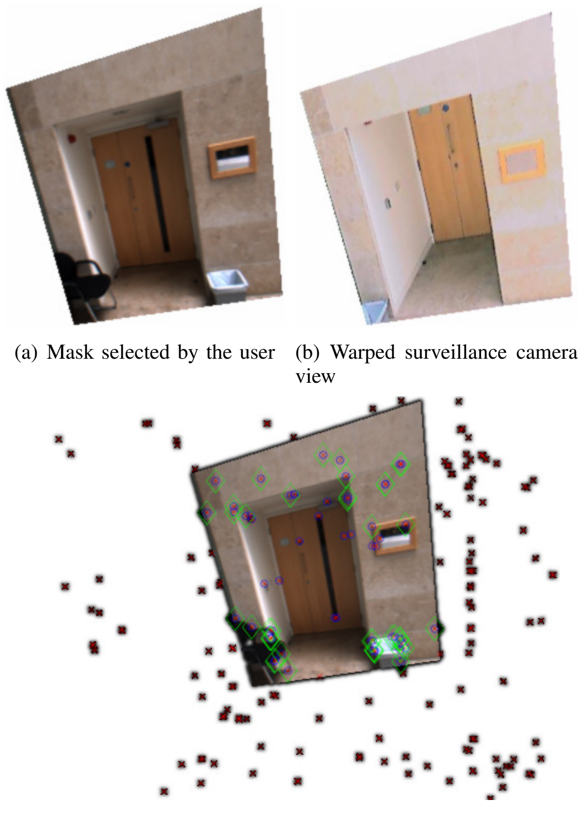


Fig. 4. Left: keyframe obtained when building a map of the scene. Right: view from a surveillance camera. Large changes of viewpoint and illumination make it difficult to automatically find the dominant plane viewed by the SLAM camera and the surveillance camera. It is however a simple task for a user and only needs to be done once. In this image, points have been selected between the two views and provide the input to compute the planar homography ${}^{o}\mathbf{H}_s$ linking a plane in the surveillance camera view to the same observed plane in a keyframe belonging to the SLAM map.



(a) Mask selected by the user (b) Warped surveillance camera view

(c) Selected features points

Fig. 5. Figure 5(a) shows the mask selected by the user to outline the dominant plane. Figure 5(b) shows the camera view warped using the homography ${}^{o}\mathbf{H}_s$ computed from the points selected and matched between the views by the user. Figure 5(c) shows the SLAM landmarks projected in the view with red crosses, the landmarks left over after region of interest selection (blue circles) and the final points after RANSAC (green diamonds). The RANSAC step is important as some landmarks projected in the region of interest do not belong to the dominant plane (e.g. points on the door).

then provided by the SLAM system. Relocalisation is sometimes referred to as loop closure or place recognition and a large body of literature has been dedicated to this topic. [15] offers a recent survey in the context of visual SLAM. Current approaches scale well with the size of the environment and in this work we use the FABMAP algorithm [4] available online. It should be noted that such an approach would not work for registering the surveillance camera with the SLAM camera directly as it relies on feature extraction and matching which, as seen previously, fails due to the strong viewpoint change.

VI. EXPERIMENTAL RESULTS

The system was tested on indoor and outdoor sequences. The quality of the system can be assessed based on (i) the correct reprojection of the surveillance camera view in the current camera view, (ii) the motion of people or objects between the occlusion boundaries and (iii) the stability of the tracking.

Figure 7(a) shows an overhead view of the scene used for testing the system. It is a large indoor space with a surveillance camera at one of the entrances. Figures 7(b) and 7(c) show the map built in the environment for this experiment. Although no ground truth is available, the structure of the scene gives a qualitative idea of the precision of the reconstruction and motion estimate. Figure 6 shows the synthesised hidden view provided by the surveillance camera. This sequence does not contain any motion between occluding boundaries but highlights the precision and stability of the tracking that can be assessed more clearly in the accompanying video.

The outdoor setup consisted of two computers communicating through an ad-hoc network. This proved challenging in practice because of the low bandwidth and unreliable connection. The transmitted image was reduced to a low resolution of 128×96 to ensure a frequency of 5Hz. This resolution had the effect of rendering the registration between the surveillance camera and a SLAM keyframe more difficult. Figure 8 shows four different sequences with hidden view synthesis in different types of environments. The quality of the estimates can be seen on the occluding boundaries where little or no ghosting effect is present as would be the case with imprecise registration. This experiment highlights the simplicity and naturalness of see-through views for improving safety in transport and analysing surveillance data.

VII. DISCUSSION AND FUTURE WORK

By building a visual map of the environment, we showed that hidden view synthesis can be greatly simplified and provide a natural way of exploiting video data from another camera. In future work, we wish to investigate further how to switch seamlessly between cameras and in particular how to choose which hidden view to render (and how to present the information to the user) in large multi-camera settings. The choice of view could be decided by the events of interest detected by the system using a probabilistic event handling mechanism [14]. To further facilitate the deployment of the

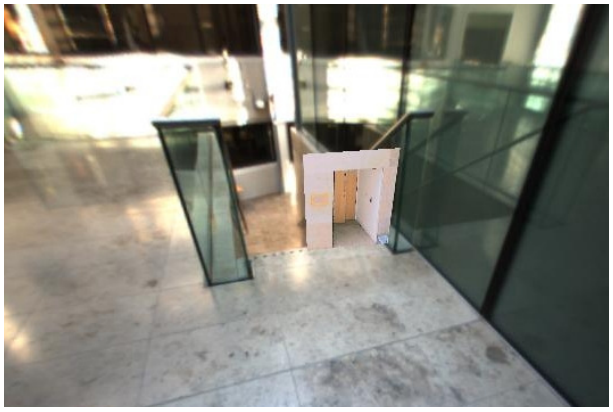
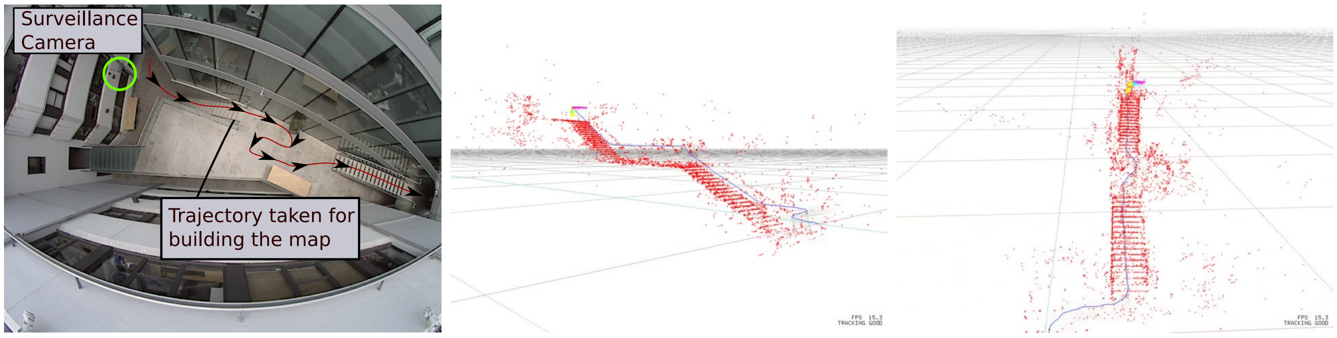


Fig. 6. Synthesised hidden view for the indoor sequence.



(a) Overview (b) Map side view (c) Map top view

Fig. 7. An indoor sequence was used to test the system. (a) shows a top view of the scene with the surveillance camera and the trajectory taken to build the map. (b) and (c) show the estimated trajectory and map. The structure of the steps are apparent. The first floor appears parallel to the ground and the staircase are aligned indicating that the pose and map are correctly estimated.

system, we will investigate how to automatically register the surveillance cameras and SLAM views. One possible direction of research is to exploit the Manhattan world assumption to reduce the search space.

VIII. ACKNOWLEDGEMENTS

The work reported in this paper was funded by EPSRC platform grant EP/D037077. We would also like to thank Shalima Deeljore for her help setting up the experiments.

REFERENCES

[1] B. Avery, B. H. Thomas, and W. Piekarski. User evaluation of see-through vision for mobile outdoor augmented reality. In *ISMAR*, 2008.

[2] P. Barnum, Y. Sheikh, A. Datta, and T. Kanade. Dynamic seethroughs: Synthesizing hidden views of moving objects. In *ISMAR*, 2009.

[3] N. Bellotto, E. Sommerlade, B. Benfold, C. Bibby, I. Reid, D. Roth, C. Fernández, L. V. Gool, and J. Gonzàlez. A distributed camera system for multi-resolution surveillance. In *Proc. of the 3rd ACM/IEEE Int. Conf. on Distributed Smart Cameras (ICDSC)*, Como, Italy, 2009.

[4] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *IJRR*, 27(6):647–665, 2008.

[5] A. J. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *PAMI*, 29(6):1052–1067, June 2007.

[6] R. Hartley and A. Zisserman. *Multiple View geometry in Computer vision*. Cambridge university press, 2000.

[7] Y. Kameda, T. Takemasa, and Y. Ohta. Outdoor see-through vision utilizing surveillance cameras. In *ISMAR*, 2004.

[8] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *ISMAR*, Nara, Japan, 2007.

[9] T. Koyama, I. Kitahara, and Y. Ohta. Live mixed-reality 3d video in soccer stadium. In *ISMAR*, 2003.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2(60):91–110, 2004.

[11] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. RSLAM: A system for large-scale mapping in constant-time using stereo. *IJCV*, 2010.

[12] T. Nakatsuru, Y. Yokokohji, D. Eto, and T. Yoshikawa. Image overlay on optical see-through displays for vehicle navigation. In *ISMAR*, 2003.

[13] I. O. Sebe, J. Hu, S. You, and U. Neumann. 3d video surveillance with augmented virtual environments. In *IWVS*, 2003.

[14] E. Sommerlade and I. Reid. Probabilistic surveillance with multiple active cameras. In *ICRA*, May 2010.

[15] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós. A comparison of loop closing techniques in monocular slam. *RAS*, 2009.

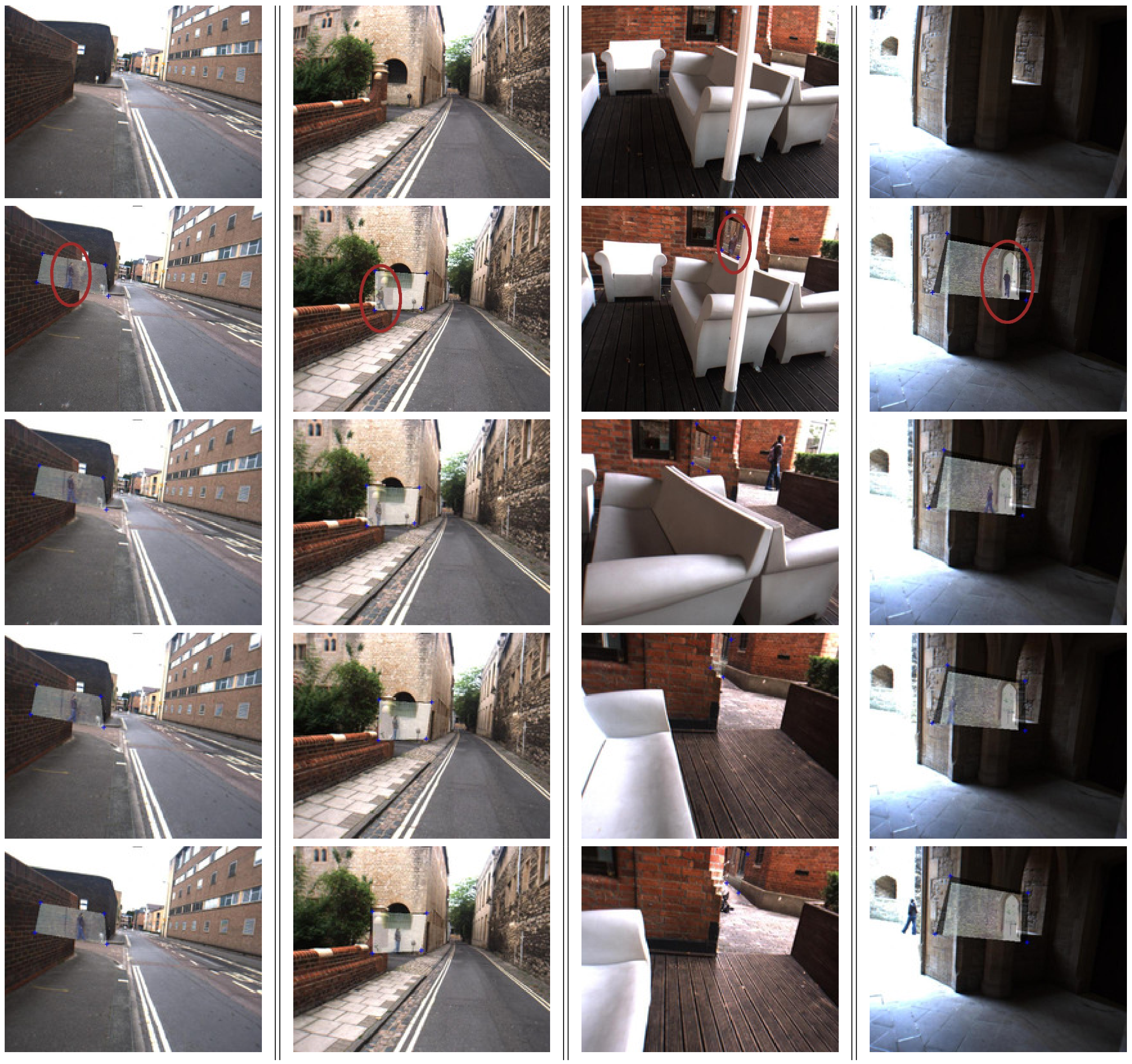


Fig. 8. Four different sequences showing hidden view synthesis in a moving camera running a SLAM system. Each column corresponds to a different sequence. The first row shows the images of the scene without the rendered view and subsequent rows show increasing time steps. In the second row, a occluded person can be seen thanks to the rendering and is highlighted by a red ellipse. The third and fourth rows for the first two sequences show the person on the occluding boundary of the hidden view. The overlap highlights the precision of the system. The last column is a mixture of indoors and outdoors. These different experiments can be seen in more detail in the accompanying video.