3D SSD Tracking with Estimated 3D Planes

Dana Cobzas* Peter Sturm
INRIA Rhone-Alpes
655 Av. de l'Europe, 38330 Montbonnot, France
{cobzas,sturm}@inrialpes.fr

Abstract

We present a tracking method where full camera position and orientation is tracked from intensity differences in a video sequence. The camera pose is calculated based on plane equations, and hence does not depend on point correspondences. The plane based formulation also allows additional constraints to be naturally added, e.g. perpendicularity between walls, floor and ceiling surfaces, co-planarity of wall surfaces etc. A particular feature of our method is that the full 3D pose change is directly computed from temporal image differences without making a commitment to a particular intermediate (e.g. 2D feature) representation. We experimentally compared our method with regular 2D SSD tracking and found it more robust and stable. This is due to 3D consistency being enforced even in the low level registration of image regions. This yields better results than first computing (and hence committing to) 2D image features and then from these compute 3D pose.
Keywords: *visual tracking, structure estimation*

1. Introduction

In visual tracking the pose of an object or the camera motion is estimated over time based on image motion information. Some applications such as video surveillance only require that the target object is tracked in image space. For other applications such as augmented reality and robotics full 3D camera motion is needed. In this paper we concentrate on tracking full 3D pose.

One way to classify tracking methods is into feature-based and registration based. In feature-based approaches features in a (usually apriori) 3D model are matched with features in the current image. Commonly a feature detector is used to detect either special markers or natural image features. Pose estimation techniques can then be used to compute the camera position from the 2D-3D correspondences. Many approaches use image contours (edges or curves) that are matched with an apriori CAD model of the object [11, 14, 6]. Most systems compute pose parameters by linearizing with respect to object motion. A characteristic of these algorithms is that the feature detection is relatively decoupled from the pose computation, but sometimes past pose is used to limit search ranges, and the global model can be used to exclude feature mismatches [11, 2].

In registration based tracking the pose computation is based on directly aligning a reference intensity patch with the current image to match each pixel intensity as closely as possible. These methods assume that the change in location and appearance of the target in consecutive frames is small. Image constancy can be exploited to derive efficient gradient based schemes using normalized correlation, or a sum-of-squared differences (e.g. L_2 norm) criterion, giving the technique its popular name SSD tracking. Unlike the two previous approaches which build the definition of what is to be tracked into the low level routine (e.g. a line feature tracker tracks just lines), in registration based tracking any distinct pattern of intensity variation can be tracked. The first such methods required spatial image derivatives to be recomputed for each frame when "forward" warping the reference patch to fit the current image [12], while more recently, efficient "inverse" algorithms have been developed, which allow the real time tracking for the 6D affine [7] and 8D projective warp [3]. A more complicated appearance model can be used to compensate changes in intensity [7] or can be learned as a mixture of stable image structure and motion information [10].

In this paper we extend the registration-based techniques by constraining the tracked regions to 3D planes. This will allow tracking full 3D camera position like in the model-based approaches but eliminates the need for explicit feature matching. The update is based on the same SSD criterion as the classical registration-based methods with the difference that the update is done directly on the 3D parameters and not on the 2D warp parameters. The approach is thus different from previous approaches that first estimate the homography warp from salient points and then the 3D motion parameters from the homography [15]. The 3D plane parameters are estimated and optimized in a training phase (typically ≈ 100 frames) using structure-from-motion techniques. The algorithm does not require complete scene de-

* Acknowledgments to NSERC Canada for supporting this work.

composition in planar facets, but works with few planar patches identified in the scene. Man-made environments usually contain planar structures (e.g. walls, doors). Some advantages of using a global 3D model and local surface patches are that only surfaces with salient intensity variations need to be processed, while the 3D model connects these together in a physically correct way. We show experimentally that this approach yields more stable and robust tracking than previous approaches, where each surface patch motion is computed individually.

Related work of incorporating a 3D model into registration based tracking involve a full 3D model (3D patches defined by estimated 3D points) of the regions that are tracked [5]. Another similar approach is presented by Baker et al. [16] where the 3D model is calculated from a 2D active appearance model (AMM) and used to improve the tracking. In the proposed technique we loosen this constraint and require only the plane parameters to be estimated. Any regions on these planes can then be tracked.

The rest of the paper is organized as follows: The next section describes the tracking algorithm, then Section 3 presents the method for estimating plane equations from images. The complete tracking system is presented in Section 4 and its qualitative and quantitative evaluation in Section 5 followed by conclusions and a discussion in Section 6.

2. Tracking 3D planes

We consider the problem of tracking the motion of a camera looking at a rigid structure using image registration. The structure is represented by a set of 3D planes that are estimated a-priori as described later in Section 3. Full 3D camera motion is tracked by registering image regions on corresponding planes through the induced homography.

2.1. Homography induced by a plane

It is well known that images of points on a plane in two views are related by a homography [8]. For planes in general position this homography is uniquely determined by the plane equation. A 3D plane is represented as $\pi = [\mathbf{n}^T, d]$, where $\mathbf{n}$ is the unit normal and d is the signed distance from the origin to the plane. For points $\mathbf{X}$ on the plane $\mathbf{n}^T\mathbf{X} + d = 0$. If the world coordinate system is aligned with the first camera coordinate system, the calibrated projection matrices have the form:

$$P_0 = K[I|\mathbf{0}] \quad P_t = K[R|\mathbf{t}] \tag{1}$$

where K is the camera matrix (internal parameters) and $R, \mathbf{t}$ represents the 3D motion of the second camera with respect to the first one. Now, the homography induced by the plane π has the form:

$$H = K(R - \mathbf{t}\mathbf{n}^T/d)K^{-1} \tag{2}$$

Image points in the two views I_1, I_2 are then related by $\mathbf{u}_2 = H\mathbf{u}_1$. If the image points are normalized with respect to camera internal parameters $\mathbf{x} = K^{-1}\mathbf{u} = [R|\mathbf{t}]X$ the homography becomes:

$$H = R - \mathbf{t}\mathbf{n}^T/d \tag{3}$$

In the tracking problem formulation the goal is to directly estimate camera motion $R, \mathbf{t}$ that corresponds to the homography that best aligns the image points in two views assuming that the plane parameters are known.

2.2. Region-based tracking for planes

Assume we have estimated parameters in the plane equations for several planar regions in the scene. Let $\mathbf{x}_k = \{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_{K_k}\}$ denote all the (interior) normalized image pixels that define the projection of the planar region $\pi_k = [\mathbf{n}_k^T, d_k]$ in image I. We refer to $I_0 = T$ as the *reference image* and to the union of the projections of the planar regions in T, $\cup_k T(\mathbf{x}_k)$ as the *reference template*. The goal of the tracking algorithm is to find the (camera) motion $P_t = [R_t, \mathbf{t}_t]$ that best aligns the reference template with the current image I_t. The problem is formulated as finding an incremental motion update $\Delta\mathbf{p}$ from frame I_{t-1} to I_t that is added to the current motion. The model is defined so it is aligned with the first frame (template). A more precise formulation follows next (refer to Figure 1).

As described in the previous section the image motion in image t for each individual planar region k can be perfectly modeled by a homography warp $H(\mathbf{x}_k; P_t, \pi_k) = R_t - \mathbf{t}_t\mathbf{n}_k^T/d_k$. In the following we denote the homography warp by $H(\mathbf{x}_k; \mathbf{p}_t)$ where $\mathbf{p} = [\alpha_x, \alpha_y, \alpha_z, t_x, t_y, t_z]^T$ are column vectors of the 3D motion parameters that define the camera motion (Euler angles and translation). The main difference from the previous approaches in registration based tracking [3] is that we directly compute 3D motion parameters unified over the whole scene as opposed to 2D warp parameters for each individual patch.

Under the common image constancy assumption (e.g. no illumination variation, no occlusion) used in motion detection and tracking [9] the tracking problem can be formulated as finding $\mathbf{p}_t$ such as:

$$\cup_k T(\mathbf{x}_k) = \cup_k I_t(H(\mathbf{x}_k; \mathbf{p}_t)) \tag{4}$$

$\mathbf{p}_t = \mathbf{p}_{t-1} \circ \Delta\mathbf{p}$ (where '$\circ$' denotes the composition operation) can be obtained by minimizing the following objective function with respect to $\Delta\mathbf{p}$:

$$\sum_k \sum_{\mathbf{x}} [T(\mathbf{x}_k) - I_t(H(\mathbf{x}_k; \mathbf{p}_{t-1} \circ \Delta\mathbf{p}))]^2 \tag{5}$$

The update in position $\Delta\mathbf{p}$ is based on the image difference between the template image and the current image warped in the space of the template, the update in position being place on the side of the current image. As a consequence,

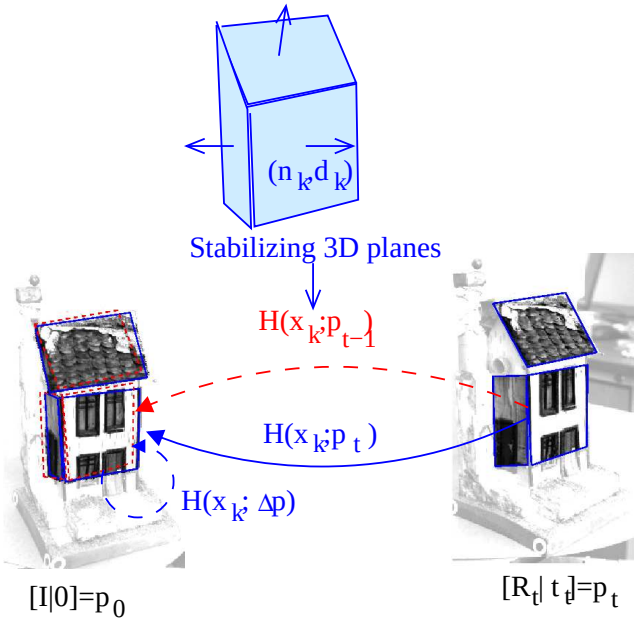

Figure 1. Overview of the 3D plane based tracking system. In standard SSD tracking 2D surface patches are related through a homography H between frames. In our system a 3D planes are estimated (from video alone), and global 3D pose change $\Delta\mathbf{p}$ is computed, and used to enforce a consistent update of all the surface warps.

the computations are performed in the space of the current image.

For efficiency, we solve the problem by an inverse compositional algorithm [3] that minimizes the error between the template image and the current image warped in the space of the template image, with the update on the template image (see Equation 7). As shown below, working in the space of the template image allow more computations to be done only once at the initialization speeding up the tracking. H becomes the homography from the image t to the template image (inverse of Equation 3). The goal is to find $\Delta\mathbf{p}$ that minimizes

$$\sum_k \sum_{\mathbf{x}} [T(H(\mathbf{x}_k;\Delta\mathbf{p})) - I_t(H(\mathbf{x}_k;\mathbf{p}_{t-1}))]^2 \qquad (6)$$

where in this case the 3D motion parameters are updated as:

$$P_t = P_{t-1} \circ \mathrm{inv}(\Delta P) \qquad (7)$$

where $\mathrm{inv}(P) = [R^T | -R^T\mathbf{t}]$ for $P = [R|\mathbf{t}]$. As a consequence the homography warp update is:

$$H(\mathbf{x}_k;\mathbf{p}_t) = H(\mathbf{x}_k;\Delta\mathbf{p})^{-1} \circ H(\mathbf{x}_k;\mathbf{p}_{t-1}) \qquad (8)$$

Performing a Taylor expansion of Equation 6 gives:

$$\sum_k \sum_x [T(H(\mathbf{x}_k;\mathbf{0})) + \nabla T \frac{\partial H}{\partial \mathbf{p}}(\mathbf{x}_k;\mathbf{0})\Delta\mathbf{p} - I_t(H(\mathbf{x}_k;\mathbf{p}_t))] \qquad (9)$$

As the motion of the template image is zero (the model is aligned with the template frame) $T = T(H(\mathbf{x}_k;\mathbf{0}))$. Denoting the image derivatives by M

$$M = \sum_k \sum_x \nabla T \frac{\partial H}{\partial \mathbf{p}} \qquad (10)$$

equation 9 can be rewritten as:

$$M\Delta\mathbf{p} \simeq \mathbf{e}_t \qquad (11)$$

where $\mathbf{e}_t$ represents the image difference between the template regions and warped image regions, and the motion $\Delta\mathbf{p}$ is computed as the least squares solution to Equation 11.

The image derivatives M are evaluated at the reference pose $\mathbf{p} = \mathbf{0}$ and they are constant across iterations and can be precomputed, resulting in an efficient tracking algorithm that can run in real time (see Section 4).

3. Estimating planes equations from images

The tracking algorithm described in Section 2.2 requires knowledge of the plane parameters for each planar region that is tracked. The plane equations are estimated from images in a bootstrapping phase. Salient feature points on each plane are tracked using standard (2D image-plane) SSD trackers as in [3, 7]. The grouping of the points depending on the plane can be easily solved by having the user mark planar regions in the first frame.

We first present the algorithm that computes a plane equation from images of points on the plane in two images. It is a special case of the structure from motion problem where the camera is internally calibrated and the feature points belong to a physical plane. The homography induced by the plane H is robustly computed using RANSAC from 4 or more corresponding points. Knowing that it is of the form $H = R - \mathbf{t}\mathbf{n}^T/d$, the motion and structure parameters $\{R, \frac{1}{d}\mathbf{t}, \mathbf{n}\}$ can be computed [13]. There are in general four solutions but only at most two are physically valid by imposing the positive depth constraint (model points are in front of the camera).

In a more general case, when multiple planes are viewed in several images, a reference view is chosen and the corresponding plane homographies that relate the reference view with additional views are computed. The motion for each frame is averaged over the motions estimated from each plane homography and the plane parameters are averaged over the ones computed from several views. Assuming a smooth motion between adjacent views only the solution that corresponds to the motion closest to the motion of the previous frame is chosen. For the first pair one of the

two physically valid solutions is chosen. The scale of the scene is also disambiguated by fixing the distance to one plane. At the end a nonlinear optimization using Levenberg-Marquardt algorithm over all the frames is performed. The error that we optimize is the symmetric transfer error for points related through a homography:

$$\{R_2, \mathbf{t}_2, \dots R_m, \mathbf{t}_m; \mathbf{n}_1, d_1, \dots, \mathbf{n}_k, d_k\} = \\ \operatorname{argmin} \sum_t \sum_k \sum_{\mathbf{x}_{tk}} d^2(\mathbf{x}_{tk}, H_{tk}\mathbf{x}_{1k}) + d^2(\mathbf{x}_{1k}, H_{tk}^{-1}\mathbf{x}_{tk}) \quad (12)$$

This is not exactly the maximum likelihood estimator under Gaussian noise but is more practical in our case as it will give the best motion and plane structure without explicitly computing the 3D points coordinates.

3.1. Incorporating constraints between planes

Known constraints between planes such as perpendicularity or parallelism of walls can potentially stabilize the tracking. We impose constraints by a minimum parametrization of the plane parameters as in [4].

Consider two planes $\pi_1 = [\mathbf{n}_1^T, d_1]$,$\pi_2 = [\mathbf{n}_2^T, d_2]$. A perpendicularity constraint can be algebraically expressed by a vanishing dot product between the plane normals:

$$n_{11}n_{21} + n_{12}n_{22} + n_{13}n_{23} = 0 \quad (13)$$

This bilinear constraint can be enforced by eliminating one plane parameter. We chose to eliminate the parameter n_{ik} such that the absolute value of the corresponding parameter on the second plane n_{jk} is maximal over all the parameters.

For the other type of constraint when the planes are parallel we impose that the normals of the two planes are the same. This eliminates all parameters that represent the unit normal of one plane.

$$n_{1k} = n_{2k}, k = 1, 2, 3 \quad (14)$$

The resulting plane parameters and the originally recovered motions are then optimized using the same Equation 12. A full parametrization of the planes is recovered for every plane from Equations 13,14. A potentially somewhat more accurate approach would involve obtaining a minimal parameterization of 3D points on constrained planes and estimating the structure of those points and the camera motion from feature correspondences. This would allow defining a maximum likelihood estimator under Gaussian image noise. The plane parameters are then computed from the estimated 3D points.

4. Tracking system overview

We incorporated the proposed plane tracking algorithm into a system that first initializes plane equations from 2D image tracking over a limited motion and then switches to track points on the estimated 3D planes.

Bootstrapping phase

1. The user marks planar regions in the first frame and specifies plane constraints (parallelism, perpendicularity) as applicable. Feature points inside these regions are tracked using standard SSD 2D trackers.
2. Plane parameters are first initialized by averaging close form solutions from homographies and then a minimal parametrization is optimized together with the estimated motion over all the training frames as described in Section 3.
3. The 3D planes are related to the current frame using the 2D tracked points. This will align the origin of the world coordinate system with the current frame. Then the plane based tracking is initialized by computing the derivative images M (Equation 10).

Tracking phase

The tracking now continues with 2D surface patches integrated in the 3D model of the planes that enforces a globally consistent motion for all surface patches as described in Section 2.2.

1. An incremental position update $\Delta\mathbf{p}$ is computed based on image differences between the regions in the reference template and the warped regions from the current image (Equation 11).
2. The global camera position is updated based on Equation 7.

5. Experiments

Two important properties of tracking methods are convergence and accuracy. Tracking algorithms based on optimization and spatio-temporal derivatives (Equation 9) can fail to converge because the image difference between consecutive frames I_{t-1}, I_t is too large (more than just few pixels), and the first order Taylor expansion (Equation 9) around $\mathbf{p}_{t-1}$is no longer valid, or some disturbance causes the image constancy assumption to be invalid.

In the numerical optimization the pose update $\Delta\mathbf{p}$ is computed by solving an overdetermined equation system, Equation 11. Each pixel in a tracking patch provides one equation and each model freedom (DOF) one variable. The condition number of the linearized motion model M affects how measurement errors propagate into $\Delta\mathbf{p}$, and ultimately if the computation converges or not. In general, it is more difficult to track many DOF. In particular, the homography warp (that incorporates scaling and out-of-plane rotations) causes less apparent image change compared to a 2D translational warp. By tracking a connected 3D model, the tracking convergence is no longer solely dependent on one surface patch alone, and the combination of differently located and oriented patches can give an accurate 3D pose estimate even when each patch would be difficult to track individually.

In the first experiment we compared the tracking stability for the plane based tracker and the traditional homography based tracker. The results are shown in Figure 2 (above). When three regions are individually tracked using an 8DOF homography by the algorithm from [3] (top images) the first region is lost already after 70 frames. The condition numbers for M vary between $4 * 10^6$ and $1 * 10^7$, indicating a numerically ill conditioned situation. When instead the regions are related by the global model, pose is successfully tracked through the whole sequence of 512 frames (middle, bottom of Figure 2). The condition number of the 6DOF (3 rot, 3 trans) model is 1000, which is significantly better than for the 8DOF homography. Imposing constraints on the estimated planes (e.g. roof planes perpendicular to front plane) further stabilizes the trackers (last row of Figure 2) . One of the trackers (the window on the tall house) starts drifting at about frame 250 when using the unconstrained model (middle row of Figure 2). The experiment is illustrated also in `video1` [1] where the red trackers use 8DOF homography the green trackers use general 3D planes and the blue ones constrained 3D planes. The planes that become occluded are eliminated using a Z-buffer algorithm.

One of the main advantages of the proposed approach over traditional SSD tracking is that actual 3D camera pose can be tracked. This is useful for example in robotics or augmented reality applications. In the next experiment we evaluate the accuracy of tracking in an indoor lab scene tracked by a moving camera. Ground truth was obtained by measuring the camera path and performing a Euclidean calibration of the model. Figure 3 shows two tracked frames, and the sequence can be seen in `video2` [1].

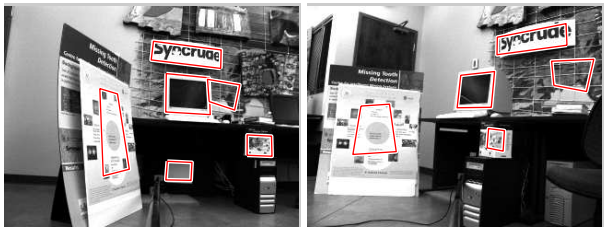

Figure 3. Tracking 3D planes. Pose accuracy experiment. `video2` **[1]**

The first test trajectory is a straight line in the horizontal plane of 1m. Figure 4 (left) illustrates the recovered trajectory. To measure the accuracy of the tracking algorithm we calibrated the 3D model for the planes assuming given real dimensions (distance from camera to one plane) so we could get the translation in meters. Here the parallelism constraints imposed between planes (e.g. back wall and Syncrude sign) had a very small influence on the pose accuracy. We found that the trajectory had 0.41 cm mean deviation from a straight line and 3.15 cm mean deviation from the horizontal plane. The recovered line length was 1.10 m, that result in an error of 10% with respect to the measured ground truth. The camera was not rotated along the first trajectory, that corresponds to the measured rotation (error was less than 1.4 degree on average).

We tracked the second trajectory along two perpendicular lines in the horizontal plane. In this experiment, the physical motion was not particularly smooth and the

recorded data therefore also somewhat jumpy. We measured the angle between the two lines fitted to the recovered positions (see Figure 4) as 76°. Hence it had a considerable angular error with respect to the ground truth. The MATLAB implementation of the plane tracking runs at about 3Hz.

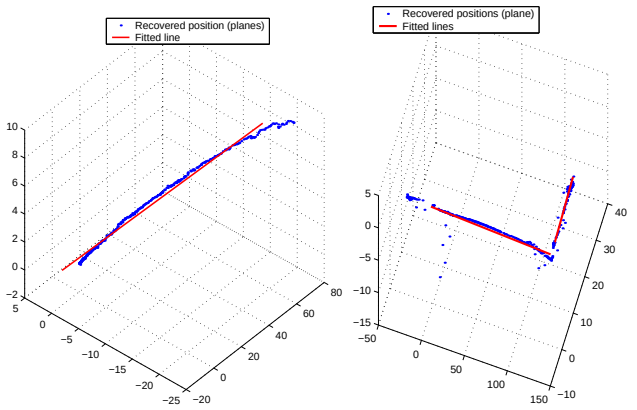

Figure 4. Recovered positions (in 3D space) for the straight line trajectory (left) and the 2 perpendicular lines trajectory (left). The red line are the fitted 3D lines to each line segment.

6. Discussion

We have presented a tracking algorithm that extends the existing SSD homography tracking by computing a global 3D position based on precomputed plane equations. The parameters of the 3D planes are estimated from an initial sequence (about 100 frames) where feature points on the planes are tracked using regular SSD translational trackers. Constraints between planes are also incorporated using a minimal parametrization of the planes. We showed that the proposed tracking algorithm is more stable due to the reduced DOF compared to tracking individual homographies and can handle a large range of motion.

A main advantage of the method is that it tracks full 3D camera position that might be required in applications like robotics or augmented reality. The pose is computed directly from image derivatives with respect to pose parameters that guarantes the best 3D pose update from the linearized model. This is unlike the other model based approaches where 3D pose is estimated from tracked 2D image correspondences.

The present version of the algorithm does not handle partial occlusions and illumination variation. This problem can be solved by using a robust norm like in [7].

References

[1] On-line mpeg movies of the experiments are available. See `videoX` at http://www.cs.ualberta.ca/∼dana/CRV05.

[2] M. Armstrong and A. Zisserman. Robust object tracking. In *Second Asian Conference on Computer Vision*, pages 58–62, 1995.

[3] S. Baker and I. Matthews. *Lucas-Kanade 20 Years On: A Unifying Framework*. Technical Report CMU-RITR02-16, 2002.

[4] A. Bartoli and P. Sturm. Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. *IJCV - International Journal of Computer Vision*, 52(1):45–64, 2003.

[5] D. Cobzas and M. Jagersand. 3d ssd tracking from uncalibrated video. In *ECCV 2004 Workshop on Spatial Coherence for Visual Motion Analysis (SCVMA)*, 2004.

[6] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *PAMI*, 24(7):932–946, July 2002.

[7] G.D. Hager and P.N. Belhumeur. Effi cient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, October 1998.

[8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[9] B.K.P. Horn. *Computer Vision*. MIT Press, Cambridge, Mass., 1986.

[10] Allan D. Jepson, David J. Fleet, and Thomas F. El-Maraghi. Robust online appearance models for visual tracking. *PAMI*, 25(10):1296–1311, 2003.

[11] D.G. Lowe. Fitting parameterized three-dimensional models to images. *PAMI*, 13(5):441–450, May 1991.

[12] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Int. Joint Conf. on Artificial Intelligence*, 1981.

[13] Y. Ma, S. Soatto, J. Kosecka, and S Sastry. *An Invitation to 3D Vision*. Springer, 2004.

[14] E. Marchand, P. Bouthemy, and F. Chaumette. A 2d-3d model-based approach to real-time visual tracking. *IVC*, 19(13):941–955, November 2001.

[15] Gilles Simon, Andrew W. Fitzgibbon, and Andrew Zisserman. Markerless tracking using planar structures in the scene. In *IEEE and ACM International Symposium on Augmented Reality (ISAR)*, 2000.

[16] Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade. Real-time combined 2d+3d active appearance models. In *Proc. of International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.