Initialization and System Modeling in 3-D Pose Tracking

Danica Kragic
Computational Vision and Active Perception
Centre for Autonomous Systems
KTH, Sweden

Ville Kyrki
Laboratory of Information Processing
Lappeenranta University of Technology
Finland

Abstract

Initialization and choice of adequate motion models are two important but seldom discussed problems in 3D model-based pose (position and orientation) tracking. In this paper, we propose an automatic initialization approach suitable for textured objects. In addition, we define, study and experimentally evaluate three motion models commonly used in visual servoing and augmented reality.

1. Introduction

During the last few years, different pose tracking algorithms have been proposed for visual servoing, augmented reality (AR), activity interpretation. One reason for the multitude of approaches are camera-object configurations: moving camera/static object (visual servoing, visual navigation, AR), static camera/moving object (activity interpretation, surveillance), moving camera/moving object(visual servoing, AR). Some of the proposed approaches are more suitable for textured objects [11, 9] while others assume objects of uniform color [12, 5, 2]. However, most methods still fail in case of drift or jitter, or if the geometrical model of the object is simple and appearance of the object and background complex. Also, initialization of the tracking process and the modeling of motion have received little attention. One of the problems of wireframe based approaches [2, 4] is that they are not suitable for textured objects and realistic background, since it is difficult to distinguish between the background and object edges, as well as multiple edges on the object itself. Classical methods of camera registration in image sequences produce accurate tracking over short sequences, but suffer from drift and decreased performance for significant view changes. To avoid the problems, object modeling using several pre-registered views has been proposed [11]. Another idea is to combine the edge-based tracking with automatically initialized model-free point trackers [6]. The latter approach uses extended Kalman filter in integrating the measurements, as originally proposed in [1]. We use it as the base for our motion modeling study as the Kalman filtering approach is also well suited for integrating the predictions made by the motion model.

2. Tracking System

Our tracking system is based on Kalman filter with alternating system modeling (prediction) and measurement modeling (update) phases, Fig.1. The initialization stage estimates the initial system state (initial object pose). The measurement modeling part includes the actual visual tracking algorithms. In this work, we are using the integration approach proposed in [6] and the main focus is on the system modeling and initialization steps.

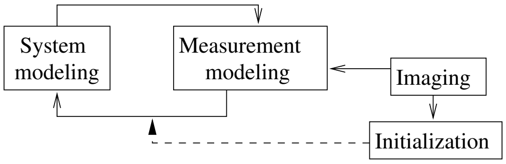

Figure 1. Tracking framework.

A number of effective model-based tracking methods have been proposed during the last few years. Many of them still assume manual initialization of the pose. The objects considered for manipulation in our framework are highly textured and therefore not suited for matching approaches based on, for example, line features [5, 12]. Techniques based on feature matching, [8, 10] are more suitable for our purposes. Similar to the work presented in [10], to initialize the pose in general settings, wide baseline matching has to be performed. Our approach considers two stages: an offline learning stage and an online matching stage.

The object recognition is based on Receptive Field Cooccurrence Histogram (RFCH) method, [3]. A RFCH is a statistical representation of the occurrence of several descriptor responses within an image. In our approach, image descriptors used are color intensity, gradient magnitude and

Laplace response. Instead of just counting the descriptor responses, RFCH is built from pairs of descriptor responses. The pixel pairs can be constrained based on, for example, their relative distance. This way, only pixel pairs separated by less than a maximum distance are considered. Thus, the histogram represents not only how common a certain descriptor response is in the image but also how common it is that certain combinations of descriptor responses occur close to each other. For a set of tracked objects, histograms are first generated offline and during recognition stage, the whole image is parsed with overlapping windows for which a RFCH is estimated. These are then compared with the stored histogram of the object using the χ^2 metric. An example of a recognition result can be seen in Fig. 2.

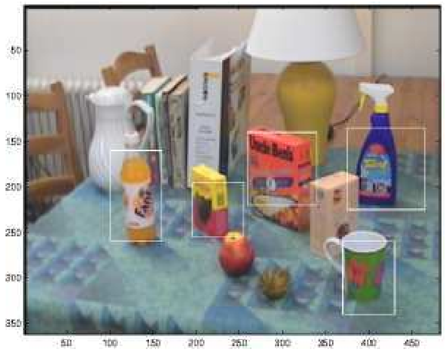

Figure 2. Object recognition based on RFCH

Once the object is localized, pose initialization step is performed. Here, SIFT features proposed in [7] are used for matching. Similar to the work presented in [10], in the training stage a viewset is defined for each object. However, instead of having each viewset represent all possible appearances of a feature under different viewing conditions, our viewset represents a set of points related to a particular object pose taken from one view. At run time, given a part of the image occupied by the object, SIFT feature detector is applied to extract scale invariant features. The detected features are matched to points at each viewset using a Nearest Neighbor search. The viewset for which the number of matches is maximal is then used to estimate the current pose of the object.

3. System Modeling

Pose measurements are integrated with the prediction given by the system model using an extended Kalman filter (EKF). EKF estimates the state $\mathbf{x}$ of a system by using a system model $f(\mathbf{x})$, which models the time dependencies of the system. A measurement model $h(\mathbf{x})$ is used to link the internal state to a set of measurable quantities $\mathbf{y}$. The uncertainties are modeled with Gaussian random variables. Let $\mathbf{P}$ denote the covariance of the internal state, $\mathbf{Q}$ be the covariance of the model error, and $\mathbf{S}$ be the covariance of the measurement error.

The EKF estimation consists of two steps. In the prediction step, the evolution of the system is predicted using the system model by

$$\mathbf{x}_{i+1|i} = f(\mathbf{x}_i) \qquad \mathbf{P}_{i+1|i} = \mathbf{F}_i \mathbf{P}_i \mathbf{F}_i^T + \mathbf{Q}_i, \tag{1}$$

where $\mathbf{F}_i$ is the gradient of $f(\cdot)$ evaluated at $\mathbf{x}_i$. In the update step, the Kalman gain is first computed as

$$\mathbf{K}_{i+1} = \mathbf{P}_{i+1|i} \mathbf{H}_{i+1}^{k^T} \left(\mathbf{H}_{i+1}^{k} \mathbf{P}_{i+1|i} \mathbf{H}_{i+1}^{k^T} + \mathbf{S}_{i+1} \right)^{-1} \tag{2}$$

where $\mathbf{H}_i^k$ is the gradient of the measurement function $h(\cdot)$ evaluated at $\mathbf{x}_{i+1}^k, \mathbf{x}_{i+1}^0 = \mathbf{x}_{i+1|i}$. Then, the state and state covariance are updated as

$$\mathbf{x}_{i+1}^{k+1} = \mathbf{x}_{i+1}^k + \mathbf{K}_{i+1} \left(\mathbf{z}_i - h(\mathbf{x}_{i+1}^k) \right) \tag{3}$$

$$\mathbf{P}_{i+1} = \mathbf{P}_{i+1|i} - \mathbf{K}_{i+1} \mathbf{H}_i^N \mathbf{P}_{i+1|i}. \tag{4}$$

We study three different motion models. The first two are zeroth order, with no predictive capability. The difference between the models is the center point of the rotation. In Model 1, the object rotates around its own origin, which corresponds to a moving object and stationary camera. In Model 2, the rotation is around the camera frame origin, corresponding to a moving camera and stationary object. It is important to note that these models indeed provide different predictions, because their covariances differ. Model 3 is a constant velocity model. We study here how much the choice of a model affects the tracking accuracy. All models are linear, but have differences in the gradient of the prediction function $\mathbf{F}$ and the prediction covariance $\mathbf{Q}$. It should be noted that the differences in $\mathbf{Q}$ have considerable effect on system characteristics.

Models 1 and 2 use the 6DOF pose vector as the system state, i.e., $\mathbf{x} = (X, Y, Z, \phi, \theta, \psi)^T$. Due to the well-known problems with the non-uniqueness of Euler angles [9], we adopt the approach proposed in [13], where the orientation is represented externally, outside the Kalman filter state, and the angles ϕ, θ, ψ only describe incremental changes. Unlike their quaternion based approach, we represent the external orientation using a rotation matrix. Thus, after each time step the rotation angles are integrated into matrix $\mathbf{R}_0$ and reset to zero. Models 1 and 2 predict the internal state according to $\mathbf{x}_{i+1|i} = \mathbf{x}_i$. Thus, the gradient of the state update is the identity matrix, $\mathbf{F} = \mathbf{I}_6$. The motion is modeled as a Wiener process, with independent uncorrelated noise sources for both translational and rotational motion. Thus, for Model 1, where the motion is with respect to the object origin, the state prediction covariance is

$$\mathbf{Q}_1(\Delta t) = \begin{pmatrix} \Delta t\, \sigma_p^2\, \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \Delta t\, \sigma_\phi^2\, \mathbf{I}_3 \end{pmatrix} \tag{5}$$

where Δt is the time step, σ_p^2 and σ_ϕ^2 are the variances of the translational and rotational motion, respectively. In Model 2, the translation is affected by the rotation around the camera center and the system can be written as

$$\begin{aligned}\mathbf{p}_{i+1} &= \mathbf{R}(\mathbf{w}_\phi)\mathbf{p}_i - \mathbf{w}_p \\ \boldsymbol{\phi}_{i+1} &= \boldsymbol{\phi}_i - \mathbf{w}_\phi\end{aligned} \tag{6}$$

where $\mathbf{w}_p$ and $\mathbf{w}_\phi$ are the noise sources for camera translation and rotation. Writing $\mathbf{R}(\mathbf{w}_\phi, \mathbf{p}_i) \equiv \mathbf{R}(\mathbf{w}_\phi)\mathbf{p}_i$, we can then approximate it with a first order Taylor series as

$$\mathbf{R}(\mathbf{w}_\phi, \mathbf{p}) = \mathbf{R}(\mathbf{0}, \mathbf{p}) + \frac{\partial \mathbf{R}(\mathbf{w}_\phi, \mathbf{p})}{\partial \mathbf{w}_\phi}\mathbf{w}_\phi = \mathbf{p} + \mathbf{A}\mathbf{w}_\phi \tag{7}$$

where

$$\mathbf{A} = \begin{pmatrix} 0 & p_z & -p_y \\ -p_z & 0 & p_x \\ p_y & -p_x & 0 \end{pmatrix}. \tag{8}$$

(6) and (7) allow us to write $\mathbf{Q}_2$ as

$$\mathbf{Q}_2 = \begin{pmatrix} \mathbf{Q}_{pp} & \mathbf{Q}_{p\phi} \\ \mathbf{Q}_{\phi p} & \mathbf{Q}_{\phi\phi} \end{pmatrix} \tag{9}$$

where

$$\begin{aligned} \mathbf{Q}_{pp} &= \Delta t\, \sigma_p^2\, \mathbf{I}_3 + \Delta t\, \sigma_\phi^2\, \mathbf{A}\mathbf{A}^T & \mathbf{Q}_{p\phi} &= -\Delta t\, \sigma_\phi^2\, \mathbf{A} \\ \mathbf{Q}_{\phi p} &= -\Delta t\, \sigma_\phi^2\, \mathbf{A}^T & \mathbf{Q}_{\phi\phi} &= \Delta t\, \sigma_\phi^2\, \mathbf{I}_3. \end{aligned}$$

Thus, the translation is additionally affected by the rotation around the camera center.

In Model 3 the system state includes also velocities, $\mathbf{x} = (X, Y, Z, \phi, \theta, \psi, \dot{X}, \dot{Y}, \dot{Z}, \dot{\phi}, \dot{\theta}, \dot{\psi})^T$ with prediction

$$\mathbf{x}_{i+1|i} = \left(\begin{smallmatrix} \mathbf{I}_6 & \Delta t \mathbf{I}_6 \\ \mathbf{0} & \mathbf{I}_6 \end{smallmatrix}\right) \mathbf{x}_i. \tag{10}$$

The noise is considered to only affect the velocities and the noise covariance is thus

$$\mathbf{Q}_3 = \begin{pmatrix} \frac{1}{3}\Delta t^3 \sigma_p^2 \mathbf{I}_3 & \frac{1}{2}\Delta t^2 \sigma_p^2 \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \frac{1}{2}\Delta t^2 \sigma_p^2 \mathbf{I}_3 & \Delta t \sigma_p^2 \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{1}{3}\Delta t^3 \sigma_\phi^2 \mathbf{I}_3 & \frac{1}{2}\Delta t^2 \sigma_\phi^2 \mathbf{I}_3 \\ \mathbf{0} & \mathbf{0} & \frac{1}{2}\Delta t^2 \sigma_\phi^2 \mathbf{I}_3 & \Delta t \sigma_\phi^2 \mathbf{I}_3 \end{pmatrix}. \tag{11}$$

4. Experimental evaluation

Experiments were performed on a recorded sequence to allow repeated tests and two example images are shown in Fig. 3. The length of the sequence is 173 seconds. It has been recorded by moving a camera mounted on a robot arm and therefore the ground truth has been generated by recording the robot trajectory. This is important since most pose tracking results are usually presented only qualitatively.

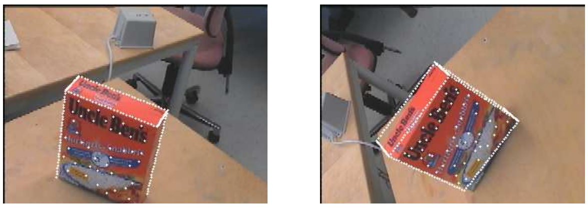

Figure 3. Two images from the test sequence.

4.1. Initialization

The initialization accuracy was examined using all the images in the sequence. An initialization attempt was considered failed if the translation error exceeded 15cm or if the rotation error was above 15 degrees. The errors for successful initializations are seen in Fig. 4. Only 1.5% of the attempts failed, and no two consecutive frames were failures. The mean translation error for the successful attempts was approximately 2 cm and rotation error 4 degrees, which demonstrates that the approach is valid for initializing the tracker.

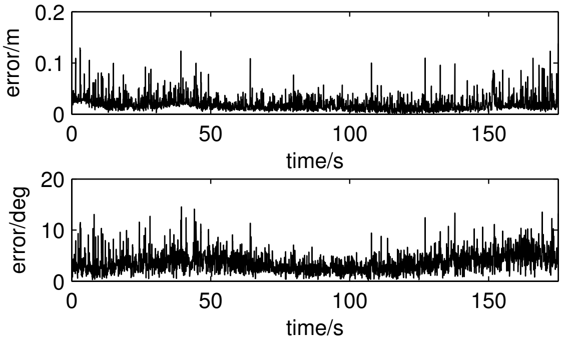

Figure 4. Initialization accuracy.

4.2. Motion models

Fig. 5–7 show the error magnitudes for system models described in Sec. 3.. At around 50 sec, an interesting phenomena occurs, resulting from a temporarily erroneous measurements of rotation. Models 1 and 3 have increasing but smooth error, while Model 2 has a different kind, oscillating error. This is the result of Model 2 restricting the rotation of the object to co-occur with suitable translations. All models had similar average errors, Model 1 having slightly lower error of 1.7 cm in translation and 3.8 degrees in rotation compared to 1.9 cm and 3.9 degrees for Model 2, and 1.7 cm and 4.0 degrees for Model 3.

The effect of the sampling rate was inspected by lowering the sample rate to 1/2, 1/3, etc. The errors for different sampling rates are shown in Table 1. The sudden increase in the errors at 1/3 sampling rate for Models 1 and 2 occurs because at that rate the object motion at one point of the sequence is fast enough so that one of the tracked edges

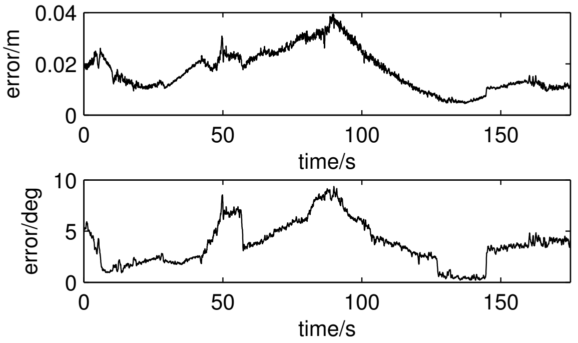

Figure 5. Model 1 (moving object).

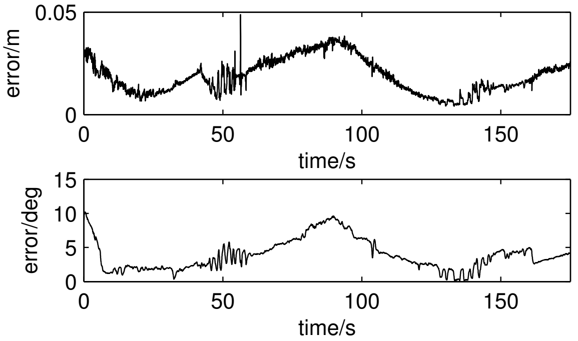

Figure 6. Model 2 (moving camera).

is confused with another. Constant velocity prediction of Model 3 allows it to run without this breakdown even for 1/4 sampling rate, but the breakdown happens at 1/5 sampling. This demonstrates the importance of having a good predictive system model in high speed tracking.

5. Summary and Conclusions

We have presented a method for initialization of pose tracking based on robust feature matching and object recognition. In addition, we have implemented and evaluated three different motion models. An important observation is that even when a model is not predictive, the model poses constraints on the motion through the structure of the prediction covariance. It should be noted that these results are not restricted to the particular model-based tracking methods used in this study, but they are applicable to all Kalman filter based tracking. Experimental results present an evaluation of the accuracy of the proposed models. We found that while the overall accuracy of the system did not heavily depend on the motion model, the model choice had a remarkable effect on other characteristics, such as the required sampling rate. Also, the frame of reference for motion must be selected carefully in order to guarantee desired behavior.

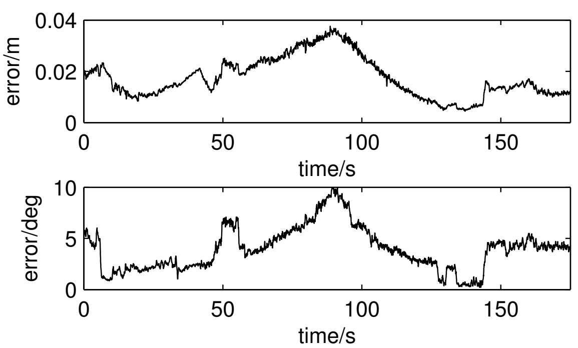

Figure 7. Model 3 (constant velocity).

Table 1. Errors with lower sampling rates.

	Model 1	Model 2	Model 3
1/1	0.017 / 3.8	0.019 / 3.9	0.018 / 4.0
1/2	0.019 / 4.4	0.020 / 4.8	0.019 / 4.5
1/3	0.038 / 13.8	0.037 / 13.4	0.019 / 4.8
1/4	0.038 / 13.8	0.038 / 13.6	0.020 / 5.1
1/5	0.039 / 13.9	0.041 / 17.6	0.038 / 13.8

References

[1] E. D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision and Appl.*, 1:223–240, 1988.

[2] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. Patt. Anal. and Machine Intell.*, 24(7):932–946, 2002.

[3] S. Ekvall and D. Kragic. Receptive field cooccurrence histograms for object detection. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'05*, 2005.

[4] C. Harris. Tracking with rigid models. In *Active Vision*, pages 59–73. 1992.

[5] D. Koller, K. Daniilidis, and H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 1993.

[6] V. Kyrki and D. Kragic. Integration of model-based and model-free cues for visual object tracking in 3d. In *Int Conf on Robotics and Automation*, pages 1566–1572, 2005.

[7] D. G. Lowe. Object recognition from local scale-invariant features. In *Int Conf Computer Vision*, 1999.

[8] C. Schmid and R. Mohr. Local greyvalue invariants for image retreival. *Patt. Anal. and Machine Intell.*, 1997.

[9] G. Taylor and L. Kleeman. Fusion of multimodal visual cues for model-based object tracking. In *Australiasian Conference on Robotics and Automation*, 2003.

[10] L. Vacchetti, V. Lepetit, and P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In *Computer Vision and Pattern Recognition*, 2004.

[11] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3d tracking using online and offline information. *IEEE Trans. Patt. Anal. and Machine Intell.*, 26(10):1385–1391, 2004.

[12] M. Vinzce, M. Ayromlou, M. Ponweiser, and M. Zillich. Edge projected integration of image and model cues for robust model-based object tracking. *International Journal of Robotics Research*, 2001.

[13] G. Welch and G. Bishop. SCAAT: Incremental tracking with incomplete information. In *25th annual conference on Computer graphics and interactive techniques*, 1997.