Accurate Pose Estimation using Single Marker Single Camera Calibration System

S. Pati[§], L. Wang, O. Erat, P. Fallavollita, N. Navab

Chair for Computer Aided Medical Procedures, Computer Science Department, Technische Universität München

Abstract

Visual marker based tracking is one of the most widely used tracking techniques in Augmented Reality (AR) applications. Generally, multiple square markers are needed to perform robust and accurate tracking. Various marker based methods for calibrating relative marker poses have been already been proposed. However, the calibration accuracy of these methods relies on the order of the image sequence and pre-evaluation of pose-estimation errors, making the method offline. Several studies have shown that the accuracy of pose estimation for an individual square marker depends on camera distance and viewing angle. We propose a method to accurately model the error in the estimated pose and translation of a camera using a single marker using an online method based on the Scaled Unscented Transform (SUT). Thus, the pose estimation for each marker can be estimated with highly accurate calibration results independent of the order of image sequences compared to cases when this knowledge is not used. This removes the need for having multiple markers and an offline estimation system to calculate camera pose in an AR application.

Keywords: *single marker calibration, scaled unscented transform, visual marker based tracking, kalman filter.*

1. Description of Purpose

Performing fully automatic real-time vision-based camera tracking is an extremely challenging problem. To simplify this problem, the environment is populated with one or more fiducial markers of known appearance [2]. This essentially reduces the problem to distinguishing (and labeling) the markers and estimating their pose(s). There are many mature tracking systems in wide use in the academic and commercial communities, like *ARToolKit* [3], *ARToolKitPlus* [4] and *ARTag* [5]. These tools are great assets for researchers from the augmented and mixed reality communities. But in all of these systems, the pose is always recovered with a certain degree of error as the corners detected by the camera have error because of sampling or because of intrinsic camera parameters (refer Figure 1). Quantification of these tracking errors is vital for at least two reasons. Firstly, the statistical error can be utilized by applications to adapt their interfaces depending upon the level of error [4, 6]. Secondly, if the estimated poses are to be used in another estimation algorithm [6] such as a particle filter [7] or a Kalman filter [8], performance can be highly optimized if a statistical characterization of the errors in the estimates is known. To overcome this problem, several studies have been carried out to characterize the errors of such marker tracking systems [1, 2, 3, 4]. In all these studies, the characterizations are carried out empirically, i.e, the tracking system is used to measure the pose of a platform, while the real pose is being estimated by an external and more accurate tracking or measurement system. By comparing these results, the error characteristics can be derived. These studies have either divided up the region around the marker(s) into known error zones or sought to find generalizations about the tracker's overall performance. But, there are two important disadvantages with this approach. First, since the errors are highly dependent on the configuration of the markers and camera properties [12], it is unclear whether the errors calculated for one configuration can be used for a different configuration in a different situation. Second, the analysis has only been carried out for a few select tracking systems and would therefore need to be repeated for any further systems that might be developed or used. This is obviously not practically feasible. In this paper, we develop a generic technique for predicting the statistics of tracking error in real-time. In theory, this is much more practical than empirical methods as the area of application for this is far wider. The method is based on the *Scaled Unscented Transform (SUT)* [13, 14] and it is used in collaboration with OpenCV to develop an algorithm that calculates the possible error in estimated marker pose. Since this method is very generic, we believe that it is applicable to various other classes of tracking systems where such an online error estimation model is necessary. Although this study is based on single-marker-single-camera systems, the theory and logic can be extended to multi-marker-single-camera systems as well. Thus, many more applications for tracking are concievable.

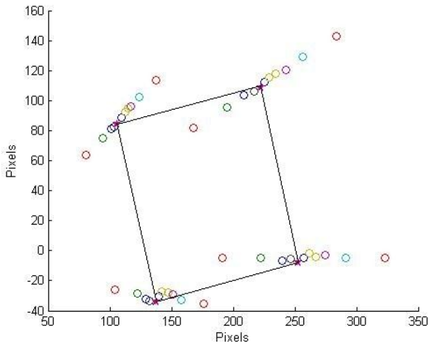

Figure 1: Corner Locations for hand held marker – the solid line represents the original marker and corners and hollow circles are the corner locations predicted by SUT.

2. Methodology and Theory

In the fields of probabilistic signal processing, one of the most important and fundamental tasks is to estimate the statistical properties of a random variable which has been transformed. For example, the Kalman Filter (KF) [16, 17] uses the transformations in two random variables to predict the possible future state(s) of a system. But KF can only be used for linear transformations. In the case of non-linear transformations, there are no general closed-form solutions [18] and to counter this, many approximations have been proposed [19, 20, 21]. The most widely used approach of all of this is the Extended Kalman Filter (EKF) [22]. It simply linearizes all non-linear models and then applies the KF algorithm so that all the KF equations are satisfied. Nevertheless, this cannot be done without putting considerable thought into KF's two well-known drawbacks - firstly, if assumptions of local linearity are violated, linearization can produce highly unstable filters [23] and secondly, the derivation of the Jacobian matrices are nontrivial in most applications and can lead to serious implementation difficulties. *Julier et. al.* have introduced the "unscented transform" and it's various derivatives in [13, 14, 23, 24]. In this transform, a set of weighted "sigma points" are chosen such that certain defining characteristics of the random variable (like their first two moments) are preserved. Each of these sigma points undergoes the nonlinear transformation and various statistical properties of the transformed set are calculated. Although this transform resembles the Monte Carlo method on a superficial level, no random sampling is done, thus reducing the required number of points ([2n+1] points for an n-dimensional space) for prediction.

Let **A** be the random variable that stores measurements from a system. Its mean is denoted as $\boldsymbol{\mu}_{\mathbf{A}}$ and variance as $\mathbf{V}_{\mathbf{A}}$. Let **B** be the random variable that gives the future states of **A** and it is related to **A** by a nonlinear transformation,

$$\mathrm{B} = \mathrm{f}[\mathrm{A}] \qquad (1)$$

We need to find the mean and variance of **B**, which are $\boldsymbol{\mu}_{\mathbf{B}}$ and $\mathbf{V}_{\mathbf{B}}$, respectively.

Let $\mathrm{A} = \delta\mathrm{A} + \mu_{\mathbf{A}}$, where $\delta\mathrm{A}$ is a zero mean random variable with variance $\mathbf{V}_{\mathbf{A}}$. Expanding $\mathrm{f}[.]$ about $\boldsymbol{\mu}_{\mathbf{A}}$ using Taylor Series expansion we get,

$$\mathrm{f}[\mathrm{A}] = \mathrm{f}[\delta\mathrm{A} + \mu_{\mathbf{A}}] = \mathrm{f}[\mu_{\mathbf{A}}] + \nabla\mathrm{f}\delta\mathrm{A} + \left(\frac{1}{2}\right)\nabla^2\mathrm{f}.\,\delta\mathrm{A}^2 + \left(\frac{1}{3!}\right)\nabla^3\mathrm{f}.\,\delta\mathrm{A}^3 + \cdots \qquad (2)$$

Using expectations and simplifying further,

$$\mu_{\mathbf{B}} = \mathrm{E}[\mathrm{B}] = \mathrm{f}[\mu_{\mathbf{A}}] + \nabla\mathrm{f}\delta\mathrm{A} + \left(\frac{1}{2}\right)\nabla^2\mathrm{f}.\,\delta\mathrm{A}^2 + \left(\frac{1}{3!}\right)\nabla^3\mathrm{f}.\,\delta\mathrm{A}^3 + \cdots \qquad (3)$$

And,

$$\mathrm{V_B} = \mathrm{E}[(\mathrm{B} - \mu_{\mathbf{B}})(\mathrm{B} - \mu_{\mathbf{B}})^{\mathrm{T}}]$$

$$= \nabla\mathrm{f}\,\mathrm{V_A}\,(\nabla\mathrm{f})^{\mathrm{T}} + \left(\frac{1}{4}\right)\nabla^2\mathrm{f}\,\mathrm{E}[\delta\mathrm{A}^3](\nabla\mathrm{f})^{\mathrm{T}} + \left(\frac{1}{2}\right)\nabla\mathrm{f}\,\mathrm{E}[\delta\mathrm{A}^3](\nabla^2\mathrm{f})^{\mathrm{T}} + \left(\frac{1}{2}\right)\nabla^2\mathrm{f}(\,\mathrm{E}[\delta\mathrm{A}^4] - \mathrm{E}[\delta\mathrm{A}^2\mathrm{V_A}] - \mathrm{E}[\mathrm{V_A}\delta\mathrm{A}^2] + {\mathrm{V_A}}^2)(\nabla^2\mathrm{f})^{\mathrm{T}}$$

$$+ \left(\frac{1}{3!}\right)\nabla^3\mathrm{f}\mathrm{E}[\delta\mathrm{A}^4](\nabla\mathrm{f})^{\mathrm{T}} + \cdots \qquad (4)$$

A set of (p + 1) weighted points (called sigma points [13, 14]) ***S = {W, A'}*** (such that $\sum_{i=1}^{p} W_i = 1$) are chosen to reflect certain properties of **A** [23, 24]. Once the set has been derived, the prediction method is pretty straight-forward. First, each point is instantiated through the nonlinear function, $\boldsymbol{B_i}' = \boldsymbol{f}[\boldsymbol{A_i}']$**.** The estimated mean and variances of **B** are,

$$\mu_{\mathbf{B}} = \sum_{i=1}^{p} W_i B_i{}' \qquad (5)$$

$$\mathrm{V_B} = \sum_{i=1}^{p} W_i\,.\{B_i' - \mu_{\mathbf{B}}\}\{B_i' - \mu_{\mathbf{B}}\}^{\boldsymbol{T}} \qquad (6)$$

We implemented the above Scaled Unscented Transform (SUT) and pose estimation algorithms in C++ using Visual Studio 2008. Input image is thresholded and contours are detected in binary image. Polygon estimation is done on detected contours and polygons having four sides (rectangles) are selected and filtered by size of the area they cover. Initial corners that are obtained by polygon estimation are used to find the approximate sides of rectangle. Using sub pixel accuracy method exact points that are located on sides of the rectangle are calculated. Using these exact points on edges real corners are calculated. Since we know that the markers are square in the real world, the corners in image plane can be mapped to their real world counterparts and the perspective is warped. When the perspective correction is done, marker identification number gets read and by using the orientation information obtained by the marker id, corner orders are corrected. After correcting the orders of the corners, pose estimation can be done using algorithms used in standard AR tools [3, 4, 5]. The detected corners are given to SUT which gives the coordinates of the new corners (the total number is based on the number of iterations SUT performs). Each of these new corner locations gives a new pose matrix. After all the poses are estimated, an average can be taken, thereby giving a much more accurate value of the camera pose, thus making the camera calibration algorithm a completely online system.

3. Ground Truth

We tested our algorithm by analyzing images of a marker with the camera positioned **-10º, 0º** and **+10º** with respect to the z-axis using a C-arm fluoroscope that is augmented with a video camera (i.e. CamC system). Since there is no projective transformation of the marker at **0º**, the spread that the corners exhibited was taken as ground truth and it was used for all subsequent error modeling. This spread can be manually adjusted (using parameters **p** & **q**) in accordance with how the camera is oriented (very low when camera rotation with respect to z-axis is 0º and progressive increase with increase in this angle). As the orientation of the marker with respect to the camera keeps changing, the error will keep spreading further till a point where the error becomes ∞ (no corners are visible, i.e., camera can see side view of marker).

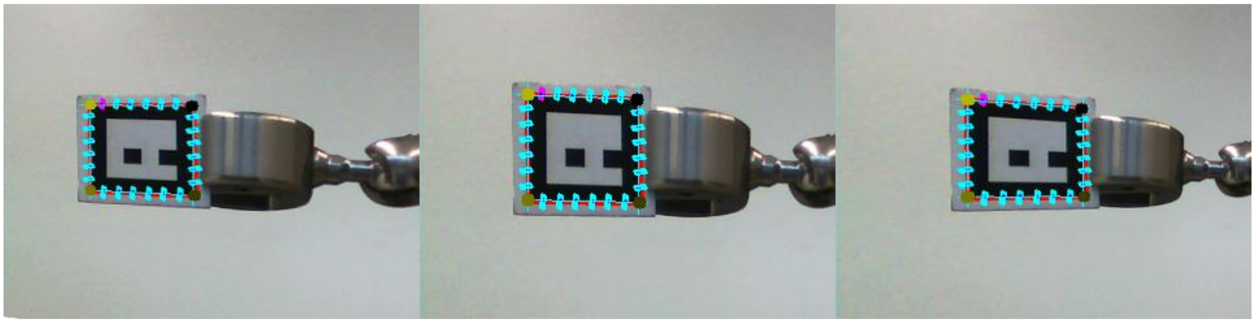

Figure 2: Images obtained by camera when camera is placed -10º, 0º and +10º with respect to z-axis respectively

4. Results

The proposed algorithm was tested on markers whose pose was known and on hand-held markers (random pose). Both were done using uncalibrated cameras and the SUT was performed for **10** iterations, which yielded 20 possible estimated poses by the SUT. For 10 iterations, the run time efficiency was about 1.31ms per iteration. For every iteration, SUT outputs 2 error values – one in the positive direction and another in negative. No *a priori* knowledge was given to SUT regarding the pose or location of the camera. For all cases, the variance of the rotation and translation with respect to each of the axes from the originally estimated values was calculated. From the values, it is clear that the variance for the marker placed perfectly vertical is the least and as the orientation increases, the variance also keeps increasing. Unsurprisingly, as the spread value is increased, the variance increases proportionately (see Table 2).

	Error Values (Variance from Originally Estimated Values) calculated for (p,q) = (1,1)					
Deviation	X-axis		Y-axis		Z-axis	
	Rotation	Translation	Rotation	Translation	Rotation	Translation
-10	2.0771e-04	4.1322e-12	6.4632e-05	5.5087e-11	3.9340e-04	6.1922e-04
0	6.6422e-06	2.0048e-11	3.1623e-10	2.9735e-09	1.1189e-12	3.3114e-04
+10	9.6048e-04	1.0124e-11	5.2735e-05	1.4708e-11	4.1381e-04	2.4338e-04

Table 1: Error Values for fixed markers (known pose) for 10 iterations

	Error Values (Variance from Originally Estimated Values)					
(p, q)	X-axis		Y-axis		Z-axis	
	Rotation	Translation	Rotation	Translation	Rotation	Translation
(1,1)	7.5368e-04	3.9860e-11	4.2096e-05	9.3654e-11	3.6985e-04	1.0326e-04
(2,2)	1.9654e-02	2.3651e-09	8.3612e-04	1.7428e-09	5.5634e-02	3.1475e-03
(3,3)	9.9625e-02	9.7412e-06	5.2537e-02	4.9214e-06	4.8296e-01	7.9615e-02
(4,4)	5.6412e-01	5.6981e-04	6.1746e-01	1.3624e-03	9.9851e-01	6.7821e-01
(5,5)	4.9501	1.4210e-03	2.3641	9.9631e-03	6.7531	2.4338

Table 2: Error Values for hand-held markers (random poses) for 10 iterations. These are variances around the ground-truth.

5. New or breakthrough work to be presented

There have many improvements made to pose estimation algorithms but this is the first study that has successfully implemented an **online method** of estimating the error model in the pose retrieved using popular AR algorithms. By using such an improved error model, the calibration properties of a camera can be calculated online irrespective of the scene geometry and its intrinsic properties, thereby virtually eliminating the need for pre-calibration. The application for this study can extend to any AR or tracking problem.

6. Conclusions

In this work, we have proposed a novel technique to evaluate errors inherent in the estimated pose form AR markers. Although we have tested this only with fiducial markers for the CamC system [24], the concept can be extended to any optical marker (reflective, LED-based). Not only is tracking accuracy improved, it gives more working room for developers and interface designers to make their product more flexible for the end user.

This work has not been published nor will be published elsewhere.

References

[1] Freeman RM, Julier SJ, Steed AJ. A Method for Predicting Marker Tracking Error. University College London

[2] Newman J, Wagner M, Bauer M, MacWilliams A, Pintaric T, Beyer D, Pustka D, Strasser F, Schmalstieg F, Klinker G. Ubiquitous Tracking for Augmented Reality. Technical University of Munich. In Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality, 2004. (ISMAR 2004), pages 192-201, Arlington, VA, USA, 2-5 November, 2004.

[3] ARToolKit: http://www.hitl.washington.edu/artoolkit, [cited 22nd May, 2007].

[4] ARToolkitPlus: http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php, [cited 22nd May, 2007].

[5] Fiala. M. ARTag - A fiducial marker system using digital techniques. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.

[6] MacIntyre B, Coelho EM and Julier SJ. OSGAR: A Scenegraph with Uncertain Transformations. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, 2004.

[7] Pupilli M and Calway A. Real-time Camera Tracking Using Known 3D Models and a Particle Filter. International Conference on Pattern Recognition, 2006.

[8] Pustka D. Handling Error in Ubiquitous Tracking Setups. PhD thesis, Technische Universitat Munchen, 15 August 2004.

[9] Malbezin P, Piekarski W, Thomas BH. Measuring ARToolKit Accuracy in Long Distance Tracking Experiments. In the First IEEE International Augmented Reality Toolkit Workshop, 2002.

[10] Abawi DF, Bienwald J, Dörner R. Accuracy in Optical Tracking with Fiducial Markers: An Accuracy Function for ARToolKit. In Proceedings of the Third IEEE and ACM ISMAR, 2004.

[11] Bauer M, Schlegel M, Pustka D, Navab N, Klinker G. Predicting and Estimating the Accuracy of Optical Tracking Systems. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, 2006.

[12] Hoff W, Vincent T. Analysis of head pose accuracy in Augmented Reality. In Proceedings of the IEEE Transactions on Visualization and Computer Graphics, 2000

[13] Julier SJ. Unscented Filtering and Nonlinear Estimation. University College London; IEEE Review, 92(3), 2004.

[14] Julier SJ. The Scaled Unscented Transform. University College London

[15] OpenCV – Willow Garage, USA

[16] Kalman RE. A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82 (1): 35–45 (1960)

[17] Kalman RE, Bucy RS. New Results in Linear Filtering and Prediction Theory. (1961)

[18] Kushner HJ. Dynamical Equations For Optimum Non-linear Filtering. Journal of Differential Equations, vol. 3, pp. 179–190, 1967

[19] Gordon NJ, Salmond DJ and Smith AFM. Novel Approach to Nonlinear/non-Gaussian Bayesian State Estimation. IEE Proceedings-F, vol. 140, no. 2, pp. 107–113, April 1993.

[20] Sorenson HW and Stubberud AR. Non-linear Filtering by Approximation of the a posteriori Density. International Journal of Control, vol.8, no. 1, pp. 33–51, 1968.

[21] Daum FW. New Exact Nonlinear Filters. In Bayesian Analysis of Time Series and Dynamic Models, J. C. Spall, Ed., chapter 8, pp. 199–226. Marcel Drekker, Inc., 1988.

[22] Uhlmann JK. Algorithms for multiple target tracking. American Scientist; vol. 80, no. 2, pp. 128–141, 1992.

[23] Julier SJ and Uhlmann JK. A New Extension of the Kalman Filter to Nonlinear Systems. In The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando FL, USA, 1997, SPIE, Multi Sensor Fusion, Tracking and Resource Management II.

[24] Wang L, Traub J, Weidert S, Heining SM, Euler E., and Navab N. Parallax-free long bone x-ray image stitching. In G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, and C. Taylor, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009, volume 5761 of Lecture Notes in Computer Science, pages 173–180. Springer, 2009.