

On Sensor Fusion for Head Tracking in Augmented Reality Applications

Ali O. Ercan and A. Tanju Erdem

Abstract—The paper presents a simple setup consisting of a camera and an accelerometer located on a head mounted display, and investigates the performance of head tracking for augmented reality applications using this setup. The information from the visual and inertial sensors is fused in an extended Kalman filter (EKF) tracker. The performance of treating accelerometer measurements as control inputs is compared to treating both camera and accelerometer measurements as measurements, i.e., fusing them in the measurement update stage of the EKF simultaneously. It is concluded via simulations that treating accelerometer measurements as control inputs performs practically as good as treating both measurements as measurements, while providing a lower complexity tracker.

I. INTRODUCTION

Augmented reality (AR) is applicable to many arenas such as medical education, remote robot control, entertainment, and cultural heritage [1]. In AR, the aim is to enrich the real world with virtual objects, rather than replacing it altogether (as in virtual reality). Therefore, the user observes virtual objects rendered on top of the video obtained from the real world.

Head mounted displays (HMDs) are commonly used in AR applications to display the virtual content on top of the real video. In order to achieve a realistic feeling of immersion, the rendering of the virtual content has to be in alignment with the real objects in the video. This requires a very accurate tracking of the 3D pose of the user's head [1], [2]. In the literature, many methods have been proposed for this purpose. Methods depending on technologies such as GPS [3] are not suitable for indoor applications. Methods involving placing artificial cues on the scene, such as IR light emitters, RFID tags, markers, etc., and corresponding suitable sensors on the HMD [2] may be not possible or desired, e.g. for applications such as cultural heritage, since placing external objects to historical scenes would not be acceptable. Thus, methods involving sensors placed only on the HMD have been used in the literature [1].

HMDs usually come with mono or stereo cameras. Therefore, using cameras for head tracking is a natural choice. Techniques that use only camera measurements and based on the popular structure-from-motion method (SfM) [4] generally perform satisfactorily [5] at slow head motion, however they become less accurate at high velocities and accelerations due to motion blur. Inertial sensors on the other hand measure the derivatives of the head pose and

hence can be employed to boost the tracking performance at high velocities and accelerations [6]. Therefore, hybrid methods where inertial sensors and cameras are used together have been proposed [7]. In the so-called “loosely coupled approach”, inertial sensor data is used to improve the SfM performance [8]. Further improvement is achieved when inertial and visual data are fused simultaneously in a single Kalman filter [9]. This approach is also referred to as “tightly coupled approach”.

In this paper, we investigate the use of inertial sensors and cameras for head tracking in a tightly coupled approach on a simple setup. Specifically, we consider a setup where a camera and accelerometer pair providing noisy measurements. We use these measurements in an extended Kalman filter (EKF) to track their motion. In the EKF, we compare the performance of treating accelerometer measurements as control inputs to the (more natural) method of treating both camera and accelerometer measurements as measurements, i.e., fusing them in the measurement update stage of the EKF simultaneously. A similar approach has been reported in robotics, where odometer data is treated as control inputs while range data is treated as measurements [10] to reduce the tracker computational complexity. In a recent paper, Bleser and Stricker [11] compared using accelerometer measurements as control inputs for AR applications. In this paper, we investigate *extensively* the performance of treating accelerometer measurements as control inputs *under varying object motion speeds and camera sampling rates*. To do so, a plausible camera measurement noise variance as a function of sampling rate is also proposed. We show via simulations that treating accelerometer measurements as control inputs performs practically as good as treating all measurements as measurements. We confirm that both methods perform better than using only camera measurements, and the accelerometer measurements and increased camera frame rates are most useful at high motion speeds. The conclusion of these observations is that one can reduce the computational complexity of the tracker by treating accelerometer measurements as control inputs without sacrificing from tracking accuracy.

The rest of the paper is organized as follows. In Section II, the problem setup, and the camera and accelerometer measurement models are presented. In Section III, the specifics of the EKF and the method of treating the accelerometer as control inputs is explained. Section IV presents simulation results.

II. MODEL

In this section, the problem setup is explained. For the sake of illustration, we assume a 2D world. However our results

This work has been supported by Turkish Scientific and Technical Research Council (TUBITAK) under project EEAG-110E053.

Ali O. Ercan and A. Tanju Erdem are with the Electrical and Electronics Department, Ozyegin University, Istanbul, Turkey, 34662. ali.ercan@ozyegin.edu.tr, tanju.erdem@ozyegin.edu.tr

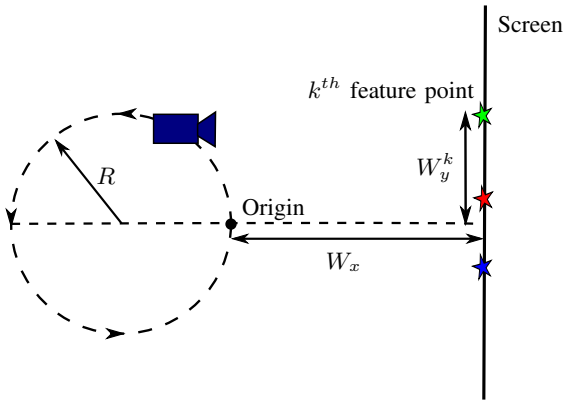


Fig. 1. Illustration of the problem setup.

can be readily extended to 3D¹. We consider an object (e.g., an HMD), equipped with a camera and an accelerometer, moving around on a circle of radius R with a period of P (see Figure 1). The goal is to track the position of a reference point on the object, using the measurements from the camera and the accelerometer. For simplicity, we assume the object does not undergo rotational motion (i.e., it only performs translational motion); the contribution of the paper does not necessitate the tracking of rotation or the presence of other measurement modalities such as gyroscopes. Extension of this work to 3-D including rotations is a topic of current research. Without loss of generality, we assume that the center of projection of the camera is the reference point to track, since any known offset can easily be accounted for in the measurement model. We assume that the reference point is at the origin at $t = 0$, however this knowledge is not available to the tracker. Under these considerations, the actual motion of the reference point is described by the following equations:

$$\begin{aligned} x(t) &= R \cos\left(\frac{2\pi t}{P}\right) - R \\ y(t) &= R \sin\left(\frac{2\pi t}{P}\right). \end{aligned}$$

Measurements

We assume that the camera frame rate is f_s^c and the accelerometer sampling rate is f_s^a . In practice, $f_s^a > f_s^c$ usually, due to high data rate of video. We assume that the first measurements of the camera and the accelerometer are synchronized. It is relatively easy to synchronize the measurements with external triggers that are present in most commercially available cameras today. Even when synchronization is not possible, one could fuse camera and accelerometer measurements in the EKF separately as they become available.

¹According to our setup, motion in the y dimension and the out-of-plane z dimension have the same dynamics and measurement characteristics, hence they are equivalent up to a rotation of the coordinate system. However, the motion in the x dimension is not equivalent to these two, because of the nonlinear dependence of the camera measurements to the x dimension. Therefore, only x and y dimensions are considered in this paper.

Accelerometer Measurements: We assume a capacitive accelerometer that is designed to operate in a $\Sigma\Delta$ A/D conversion feedback loop [12]. The advantages of this approach are two-fold. First, it improves the linearity of the accelerometer and reduces the memory effect, since the feedback stabilizes the proof mass in its rest position. Secondly, the amplifier noise and the quantization noise of the accelerometer are mostly shaped to out-of-band [13], and therefore they can be neglected. The noise due to Brownian motion of the proof mass and the mass residual motion noise due to digital feedback are not shaped, however. We assume that the mass residual motion noise is negligible compared to the Brownian noise [13] (otherwise it might be lumped in the Brownian noise as well). Under these considerations, assuming the Brownian noise power spectral density is given by S_b [$\text{m}^2/\text{sec}^4/\text{Hz}$], the accelerometer measurements can be modeled as follows:

$$\begin{aligned} z_x^a(n) &= \ddot{x}(n/f_s^a) + \nu_x^a(n) \\ z_y^a(n) &= \ddot{y}(n/f_s^a) + \nu_y^a(n), \end{aligned} \quad (1)$$

where $n \in [1, 2, \dots, \infty)$ is the sample index, the double dot denotes the second time derivative, and $\nu_x^a(n)$ and $\nu_y^a(n)$ are mutually independent white Gaussian noise processes with sample variance equal to $\sigma_a^2 = S_b f_s^a$. The subscripts x and y signify the two dimensions.

Here we assume that the accelerometer bias (including the gravity) is learned beforehand in a calibration stage and subtracted off from the measurements. The problem with this approach is, any error in the learning of the bias accumulates over time in the extended Kalman Filter. In practice, one can include the accelerometer bias and gravity into the parameter set that is tracked, and continuously correct for them using the help of camera measurements during tracking. This will be explored in the future extensions of this work.

Camera Measurements: We assume that there are N feature points on a screen (or wall) W_x units away from the origin. The vertical coordinates of these points are W_y^k , $k = 1, 2, \dots, N$. We assume that the locations of the feature points are learned beforehand at a calibration stage. We also assume that the local image processing can successfully detect these points on each video frame. For simplicity we ignored the probability of mis-detection or false-detection of these feature points, the cases of their occlusion by other objects or them being out of the field of view of the camera. These effects, if present, are expected to not alter the main message of the paper. However incorporating these issues to our work are topics of our current research. Assuming a perspective projection model [14], we get N feature point measurements per image:

$$z_k^c(n) = F \frac{W_y^k - y(n/f_s^c)}{W_x - x(n/f_s^c)} + \nu^c(n), \quad (2)$$

where, $k = 1, 2, \dots, N$ stands for the k^{th} feature point and F [pixels] is the focal length of the camera. To account for camera measurement noise, we assume mutually independent additive Gaussian random variables $\nu_k^c(n)$ with sample variance σ_c^2 .

To find the sample variance of camera measurement noise, we propose the following model. We assume that the errors in locating feature points in images are due to two independent sources. One is due to motion blur, and the other is due to camera readout noise. To compute the error due to motion blur, we consider the following. The object moves in the vertical direction $2R$ units in $P/2$ seconds ($f_s^c P/2$ frames). This corresponds to an average motion of $\frac{4R}{P f_s^c}$ units per frame. Each feature point then moves $\frac{4RF}{P f_s^c W_x}$ pixels per frame on the average. We assume that the local processing determines the location of a feature point to an accuracy of half of its average per frame motion. The error due readout noise is modeled as an independent additive random variable whose variance increases linearly with frame rate, as the image signal-to-noise ratio increases linearly with exposure time in the photon shot-noise dominated regime [15]. Thus, the sample variance of $\nu^c(n)$ becomes

$$\sigma_c^2 = \left(\frac{2RF}{P f_s^c W_x} \right)^2 + \alpha f_s^c, \quad (3)$$

where α depends on the camera properties as well as external factors such as lighting conditions.

It is interesting to note that, according to our proposed model, the accelerometer noise variance is linear in accelerometer sampling rate whereas the camera noise first decreases with frame rate (less motion blur) but increases after a point (photon shot noise dominates). Not only this model realistically captures the dependency of noise variances with respect to the sampling rates, but also it yields an interesting investigation on the best camera frame rate to accelerometer sampling rate ratio. In this paper, we explore using different camera frame rates and its effect on the tracking performance. Finding the best ratio is a topic of future research.

III. TRACKER

We use the accelerometer and camera measurements in an extended Kalman filter (EKF) to track the position of the reference point. EKF is extensively used in a wide variety of recursive non-linear estimation problems and the details of the EKF algorithm can be found in any standard textbook, e.g., [10]. Here, we will present the EKF algorithm very briefly with focus on the specifics to our problem.

First, we define the global sampling rate of the system f_s to be the least common multiple of the two sampling rates, namely f_s^c and f_s^a . We assume that the accelerometer measurements are available once every f_s/f_s^a time steps and the camera measurements are available every f_s/f_s^c time steps. If the least common multiple yields a global sampling rate that is prohibitively large for a real time operation, one could assume a smaller one and assume measurements happen at the closest sampling time. We also define the time period between two samples as $\Delta = 1/f_s$.

State Vector: Since the camera measurements give information about the position of the object and the accelerometer gives information about the acceleration, the system state that

we track is a 6-D vector consisting of the position, velocity and acceleration in the x and y dimensions:

$$u^T(n) := [x(n\Delta) \ y(n\Delta) \ \dot{x}(n\Delta) \ \dot{y}(n\Delta) \ \ddot{x}(n\Delta) \ \ddot{y}(n\Delta)]. \quad (4)$$

State Evolution: Given the state above, we assume in the EKF tracker that the object moves at a constant acceleration, and any deviations in the acceleration is modeled as additive noise:

$$u(n+1) = Au(n) + B\omega(n), \quad (5)$$

where

$$A = \begin{bmatrix} 1 & 0 & \Delta & 0 & \frac{\Delta^2}{2} & 0 \\ 0 & 1 & 0 & \Delta & 0 & \frac{\Delta^2}{2} \\ 0 & 0 & 1 & 0 & \Delta & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} \frac{\Delta^2}{2} & 0 \\ 0 & \frac{\Delta^2}{2} \\ \Delta & 0 \\ 0 & \Delta \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (6)$$

and $\omega(n) \in \mathbb{R}^2$ models the error in the constant acceleration assumption. We assume elements of $\omega(n)$ are independent and identically distributed Gaussian random variables with variance σ_ω^2 .

Measurement updates: In each iteration, the EKF first performs a ‘‘time update’’ stage:

$$\hat{u}(n|n-1) = A\hat{u}(n-1|n-1) \quad (7)$$

$$\hat{\Sigma}_u(n|n-1) = A\hat{\Sigma}_u(n-1|n-1)A^T + \sigma_\omega^2 BB^T, \quad (8)$$

where \hat{u} and $\hat{\Sigma}_u$ denote the estimates of the state and the state covariance matrix, and $(n|n-1)$ denotes ‘‘at time step n given the measurements up to time step $n-1$ ’’.

If there are measurements available, they are fused in the estimate of the state vector in a ‘‘measurement update’’ stage:

$$G := \hat{\Sigma}_u(n|n-1)J^T(J\hat{\Sigma}_u(n|n-1)J^T + V)^{-1} \quad (9)$$

$$\hat{u}(n|n) = \hat{u}(n|n-1) + G(z(n) - h(\hat{u}(n|n-1))) \quad (10)$$

$$\hat{\Sigma}_u(n|n) = \hat{\Sigma}_u(n|n-1) - GJ\hat{\Sigma}_u(n|n-1), \quad (11)$$

where $z(n)$ is the vector of all measurements at time step n

$$z(n) = [z_x^a(n) \ z_y^a(n) \ z_1^c(n) \ z_2^c(n) \ \dots \ z_N^c(n)]^T; \quad (12)$$

$h(u)$ is the non-linear measurement model according to Equations (1, 2)

$$h(u) = \left[u_5 u_6 F \frac{W_y^1 - u_2}{W_y^2 - u_1} \dots F \frac{W_y^N - u_2}{W_y^N - u_1} \right]^T, \quad (13)$$

where u_i denotes the i^{th} element of vector u ; J is the Jacobian of h in Equation (13) evaluated at $\hat{u}(n|n-1)$

$$J = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{F(W_y^1 - \hat{y}(n|n-1))}{(W_x - \hat{x}(n|n-1))^2} & \frac{-F}{W_x - \hat{x}(n|n-1)} & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{F(W_y^N - \hat{y}(n|n-1))}{(W_x - \hat{x}(n|n-1))^2} & \frac{-F}{W_x - \hat{x}(n|n-1)} & 0 & 0 & 0 & 0 \end{bmatrix}; \quad (14)$$

and V is a diagonal measurement noise covariance matrix with the diagonal entries being $\sigma_a^2, \sigma_a^2, \sigma_c^2, \dots, \sigma_c^2$.

If some of the measurements are not available at a specific time step, corresponding elements or rows and columns are deleted from $z(n), h(u), J$ and V . If no measurement is available, measurement update is not performed and the algorithm passes on to the next time update stage.

Note that in the case of accelerometer measurements only, the position estimate will diverge eventually since the acceleration measurement noise will accumulate through the inherent integrator of the EKF [1]. However, the system with only camera measurements does not suffer from the same problem, thus the camera measurements are crucial for tracking. On the other hand, when used with camera measurements, the accelerometer measurements help achieve more accurate results compared to camera measurements only. But how useful these measurements are, and when they are useful are important questions that are tackled in this paper. Also, we are mainly interested in the performance of treating accelerometer measurements as control inputs in the tracker. Next, we explain the changes in the EKF under this assumption.

Using Accelerometer as Control Input: Since the accelerometers provide measurements at a relatively high rate (more than 100 Hz) and the tracker should operate in real time for augmented reality applications, reducing its computational complexity as much as possible might become a practical necessity. In addition to the standard EKF described above, we implemented a modified tracker that assumes the accelerometer measurements as noisy control inputs. This allows the use of a constant velocity state evolution model, which needs only the position and the velocity of the reference point in the state vector. In return, the reduced size of the state vector provides lower complexity in the EKF operations. Moreover, since the sampling rate of the accelerometer is usually faster than that of the camera, this model allows to skip the measurement update step when only accelerometer measurements are available. With this assumption, the Equations (4 – 14) are modified as follows:

$$\tilde{u}^T(n) := [x(n\Delta) y(n\Delta) \dot{x}(n\Delta) \dot{y}(n\Delta)],$$

$$\tilde{u}(n+1) = \tilde{A}\tilde{u}(n) + \tilde{C}[z_x^a(n+1) z_y^a(n+1)]^T + \tilde{B}\omega(n),$$

$$\tilde{A} = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tilde{B} = \tilde{C} = \begin{bmatrix} \frac{\Delta^2}{2} & 0 \\ 0 & \frac{\Delta^2}{2} \\ \Delta & 0 \\ 0 & \Delta \end{bmatrix},$$

$$\tilde{z}(n) = [z_1^c(n) z_2^c(n) \dots z_N^c(n)]^T,$$

$$\tilde{h}(u) = \left[F \frac{W_y^1 - u_2}{W_y^2 - u_1} \dots F \frac{W_y^N - u_2}{W_y^N - u_1} \right]^T,$$

$$\tilde{J} = \begin{bmatrix} \frac{F(W_y^1 - \hat{y}(n|n-1))}{(W_x - \hat{x}(n|n-1))^2} & \frac{-F}{W_x - \hat{x}(n|n-1)} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{F(W_y^N - \hat{y}(n|n-1))}{(W_x - \hat{x}(n|n-1))^2} & \frac{-F}{W_x - \hat{x}(n|n-1)} & 0 & 0 \end{bmatrix}.$$

TABLE I
PARAMETERS USED IN SIMULATIONS.

Param	Explanation	Value
P	Object motion period	10 seconds
R	Object motion radius	1 unit
f_s^a	Accelerometer sampling rate	120 Hz.
f_s^c	Camera frame rate	30 Hz.
S_b	Accelerometer noise spectral density	$(218\mu\text{g}/\sqrt{\text{Hz}})^2$
N	Number of feature points	2
W_x	Depth of feature points	5 units
W_y	Height of feature points	$\{0, 1\}$ units
σ_ω	Noise std. dev. of motion model (5)	$\frac{\max(\dot{x}(t))}{100}$
α	Prop. constant for camera shot noise (3)	1/160 pixels ² /Hz.

While using the accelerometer measurements as control inputs allows a lower complexity tracker, it also means that the information from these measurements are incorporated only in the time update stage of the EKF. Hence in the measurement update stage, the innovation from the acceleration measurements is not fused with the past estimate of the probability density function. Therefore, this method is expected to perform inferior to the standard EKF where all measurements are treated as measurements. In the next section, we compare the performance of the two methods.

IV. SIMULATION RESULTS

In this section, we report simulation results that compares the tracking performance of the tracker under three scenarios: (i) when only camera measurements are available, (ii) when both camera and accelerometer measurements are available and treated as measurements, and (iii) when both modalities are available but the accelerometer measurements are treated as noisy control inputs. We do not report the result of using accelerometer measurements only, as the tracker diverges in that case. We compare the performance of these scenarios under different object motion speeds (i.e., object rotation periods P) and camera frame rates f_s^c . For a given (P, f_s^c) pair, we first simulate the actual object motion and generate the noisy camera and accelerometer measurements for 900 seconds. Then the obtained measurements are used in EKFs that implement the scenarios mentioned above, to track the position of the object. We repeat each such simulation for five times, and plot the RMSE tracking error means and standard deviations of these simulations.

First, we present a segment of an example simulation run in Figure 2. The parameters used in this simulation are given in Table I. In the table, the choice of S_b is based on the reported Brownian noise level of ST Microelectronics LIS331DLH accelerometer [16]. The other parameters are chosen arbitrarily. These parameters except for P and f_s^c are used in other simulations of this section throughout.

In Figure 2, the red line with diamonds shows the tracking result using only camera measurements. The other lines show the true object path and other two scenarios that we tested, namely using both camera and accelerometer measurements as measurements and using both modalities but accelerometer measurements as control inputs. We observe that using the accelerometer measurements together

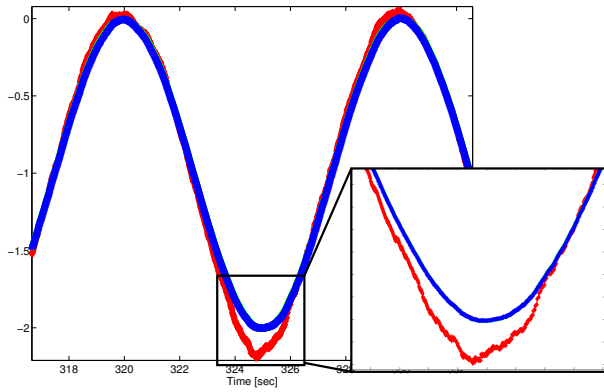


Fig. 2. An example simulation. The red line with diamonds is the tracking result using only camera measurements. The other two scenarios, namely using both camera and accelerometer measurements as measurements and using both modalities but accelerometer measurements as control inputs overlap without noticeable difference with the true object path.

with camera measurements improves the tracking accuracy. More interestingly, treating accelerometer measurements as control input does not seem to perform worse than treating them as regular measurements (their plots overlap without noticeable difference, see the zoom-in on one of the peaks of the position plot at the bottom right insert), despite the disadvantages explained in Section III. If true, this means one could save from computational complexity, while not compromising from tracking performance. Next, we present more simulations investigating this.

Figure 3 plots the means and standard deviations (as error bars) of the RMSE tracking performance in the x dimension for the three scenarios, with respect to varying object rotation period P . Here, the camera frame rate is $f_s^c = 30\text{Hz}$. The figure confirms that the accelerometer measurements increase the tracking accuracy, when the object motion is fast, which is expected. It is also observed that the performance of treating accelerometer measurements as control input is as good as treating both measurements as measurements (to within the variations in the simulations). These extended simulations confirm the observation that was made above; it seems like one can indeed save from computational complexity without losing performance. We are currently investigating the theoretical reason as to why this happens.

We also present the RMSE tracking performance in the y dimension, with respect to varying object rotation period P , in Figure 4. The conclusions of this figure is very similar to Figure 3. The main difference is that the RMSE values are lower in this dimension compared to x dimension. This makes sense, since according to our setup, the camera measurements are linear in the vertical position and non-linear in the horizontal position of the object (see Equation 2). Therefore, the EKF performs better in the y dimension.

We also investigated the effect of changing camera frame rate on the tracking performance. In Figures 5 and 6, similar plots are provided with camera frame rates $f_s^c = 20\text{ Hz}$ and

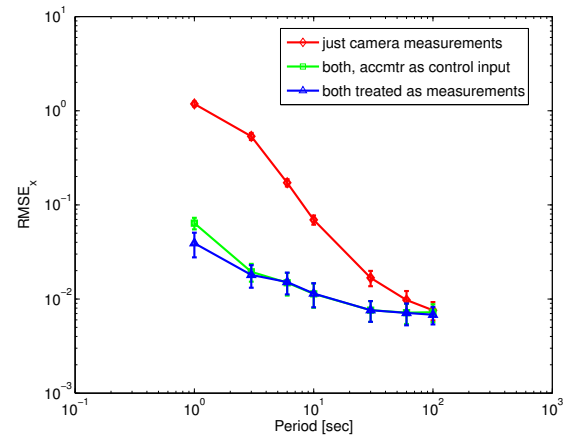


Fig. 3. RMSE tracking performance means and standard deviations in the x dimension for the three scenarios. Here, $f_s^c = 30\text{ Hz}$.

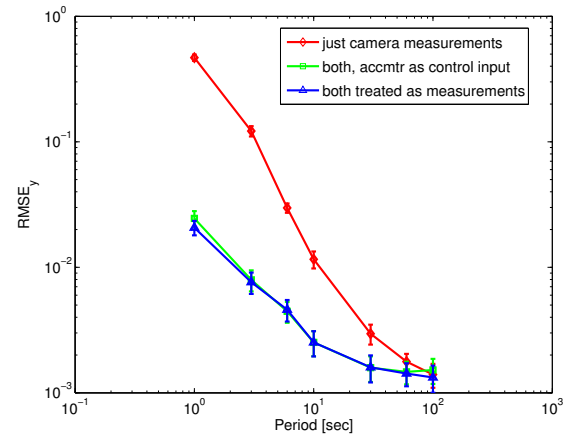


Fig. 4. RMSE tracking performance means and standard deviations in the y dimension for the three scenarios. Here, $f_s^c = 30\text{ Hz}$.

40 Hz, respectively. It is observed that, for the cases when both camera and accelerometer measurements are used, the motion blur term in the camera noise variance (Equation 3) dominates when the object motion is fast (P close to 1 sec), and the tracking performance increases as camera frame rate is increased. At $P = 1$ sec, the camera measurement noise standard deviation drops approximately by half as f_s^c increases from 20Hz to 40Hz, and correspondingly, the tracking RMSE decreases approximately by half. However, when the object motion is slow (P close to 100 sec), the motion blur is insignificant and the shot noise term dominates the camera noise variance, which increases as frame rate increases. The tracking performance of different camera frame rates are similar at around $P = 100$ sec therefore accelerometer noise is dominant. As a conclusion of these experiments, one can say that increasing camera frame rate only helps when the object motion is fast and the motion blur dominates. Otherwise, it might even hurt you (e.g., under poor lighting conditions). We also observed that varying the camera frame rate did not effect much the case when the motion is fast and only camera measurements are available.

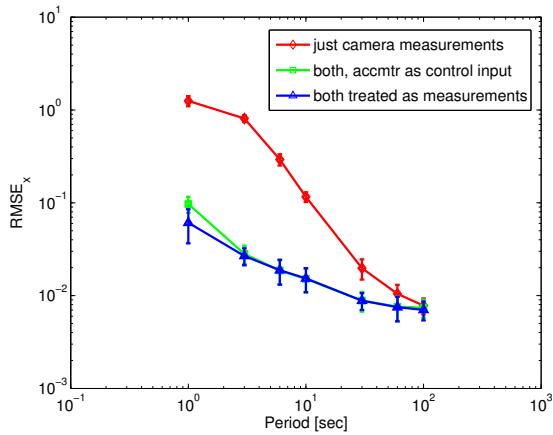


Fig. 5. RMSE tracking performance means and standard deviations in the x dimension for the three scenarios. Here, $f_s^c = 20$ Hz.

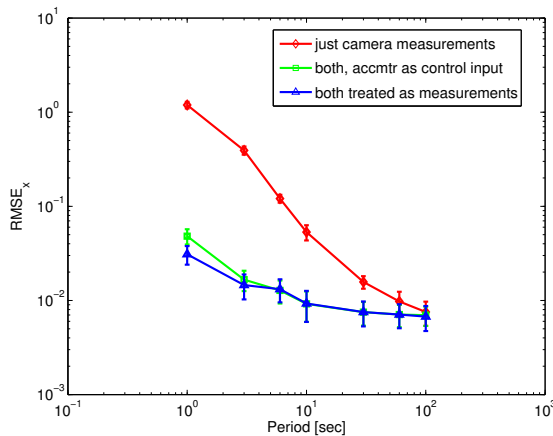


Fig. 6. RMSE tracking performance means and standard deviations in the x dimension for the three scenarios. Here, $f_s^c = 40$ Hz.

We are currently investigating the reason behind this.

In Table II, we present the CPU time of using both camera and accelerometer measurements in a standard EKF (T_{both}) relative to the CPU time of using accelerometer measurements as control input in the EKF (T_{aac}). As seen in the table, using accelerometer measurements as control input speeds up the computations about twice.

V. CONCLUSION

A hybrid system using inertial sensors and cameras to track the position of a head mounted display for augmented reality applications is investigated under a simple setup. The information from the sensors is fused in an extended Kalman filter (EKF). Specifically, the performance of treating accelerometer measurements as control inputs in the EKF is compared in simulations to treating both measurements as measurements, under varying motion speeds and camera frame rates. It is observed that the accelerometer measurements and increased camera frame rates are most useful to camera measurements under high motion speeds, however,

TABLE II

RELATIVE CPU TIMES FOR DIFFERENT CAMERA FRAME RATES.

	$f_s^c = 20$ Hz	$f_s^c = 30$ Hz	$f_s^c = 40$ Hz
$T_{\text{both}}/T_{\text{aac}}$	2.63	2.18	2.11

treating accelerometer measurements as control inputs does not suffer from performance loss compared to treating both measurements as measurements. This allows the use of a lower complexity EKF without sacrificing from tracking accuracy.

We are aware that our setup and model is very simple: it assumes a 2D world and does not consider many real life non-idealities such as mis- or false-detections and occlusion of feature points, rotational motion, accelerometer bias, etc. We are working on extending our results to incorporate these real life non-idealities. We will also verify our findings on an experimental setup. Exploration of the theoretical reasoning behind our conclusion is another work in progress.

REFERENCES

- [1] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, August 1997.
- [2] G. Papagiannakis, G. Singh, and N. Magnenat-Thalmann, "A survey of mobile and wireless technologies for augmented reality systems," *Journal of Computer Animation and Virtual Worlds*, vol. 19, no. 1, pp. 3–22, February 2008.
- [3] A. Schmeil and W. Broll, "Mara: an augmented personal assistant and companion," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*. New York, NY, USA: ACM, 2006, p. 141.
- [4] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.
- [5] G. Papagiannakis and N. Magnenat-Thalmann, "Mobile augmented heritage: Enabling human life in ancient pompeii," *International Journal of Architectural Computing*, vol. 5, no. 2, pp. 396–415, June 2007.
- [6] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *Int. J. Robotics Research*, vol. 26, no. 6, pp. 519–535, 2007.
- [7] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *Computer Graphics and Applications, IEEE*, vol. 21, no. 6, pp. 34–47, nov/dec 2001.
- [8] Y. Yokokohji, Y. Sugawara, and T. Yoshikawa, "Accurate image overlay on video see-through hmds using vision and accelerometers," *Proc. IEEE Virtual Reality*, pp. 247–254, 2000.
- [9] D. Strelow, "Motion estimation from image and inertial measurements," Ph.D. dissertation, Carnegie-Mellon University, 2004.
- [10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA & London, England: The MIT Press, 2006.
- [11] G. Bleser and D. Stricker, "Advanced tracking through efficient image processing and visual-inertial sensor fusion," *Computers & Graphics*, vol. 33, no. 1, pp. 59–72, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/B6TYG-4TYT1J-1/2/20348b115f35469af34fff0f2e05c1be>
- [12] B. Boser and R. Howe, "Surface micromachined accelerometers," *Solid-State Circuits, IEEE Journal of*, vol. 31, no. 3, pp. 366–375, Mar 1996.
- [13] H. Kulah, J. Chae, N. Yazdi, and K. Najafi, "Noise analysis and characterization of a sigma-delta capacitive microaccelerometer," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 2, pp. 352–361, Feb. 2006.
- [14] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ: Prentice Hall, 1998.
- [15] A. El Gamal and H. Eltoukhy, "CMOS image sensors," *Circuits and Devices Magazine, IEEE*, vol. 21, no. 3, pp. 6–20, May-June 2005.
- [16] *LIS331DLH 3-axis accelerometer*, ST Microelectronics. [Online]. Available: <http://www.st.com/stonline/products/literature/ds/15094/lis331dlh.pdf>