Improved NetArgus:
A Suite of Wi-Fi Positionng & SNMP Monitor

Tryfon Theodorou[1], George Violettas[1], Antigoni Polychroniadou[2], Christos Georgiadis[1]
[1]University of Macedonia, Greece, [2]Royal Holloway University of London
{georgevio|theodorou|it0855|geor}@uom.gr
156, Egnatia Street, 54006, Thessaloniki, Greece

Keywords: Wi-Fi positioning; Kalman Filter; SNMP; FSPL; RTLS.

Abstract: In this paper, field research was conducted in order to enrich and strengthen the "NetArgus" application that is able to monitor either a wired or a wireless network with all its possible aspects. Original algorithms were implemented in an effort to find the position of a moving station utilizing the signal level of wireless local area network (WLAN) devices. Exhausting sampling and research were done, in order to resolve various practical problems associated with the signal level of the wireless stations as well as the involved hardware. Various flaws and omissions of 802.11x protocols were discovered, such as the lack of definition of the signal level and its width range. Moreover, a plethora of positioning practical problems has been adequately managed. Hence, an application was constructed to draw paths and manipulate proper information from various network moving clients, using the Simple Network Management Protocol (SNMP) as well as low level operating system libraries. Last but not least, "Kalman Filter" was implemented and fully tested in order to correct the positioning unavoidable mistakes.

1 INTRODUCTION

This paper is an extended-updated and improved version of (G.E. Violettas et al., 2009). The aim of the essay was to use the Wi-Fi available information for finding the position of a station. SNMP protocol was used for a central management and possible central guidance of such devices. In addition, *Kalman filter* (KF) is implemented in order to derive the correct positioning estimates.

Practical problems of positioning were solved, by creating the *NetArgus* application, which draws information from network devices using various low level operating system libraries like Microsoft WMI and NDIS revealing several flaws and lacks. It was proven that only NDIS library is functional.

The *NetArgus* application is open source, it permits any kind of parameterization and it exploits all the new technologies. This modular application consists of 3 basic segments, providing the ability to install every part of it to the most convenient part of the network. The GUI user part is location independent. This application manipulates all the necessary information. Thus, an SQL database stores in real time the SNMP available information together with the signal level information sent by the moving stations through the application created for that purpose. A server application derives data from the database and it calculates the position of the moving stations in real time. It also stores and manipulates all the provided SNMP information. The client application pictures in a 2-axis graphical interface the position of the network elements, in real time too.

A theoretical barrier of the 802.11 signal level strength measurement was discovered, due to limitations of the implementation of this specific variable as an integer (the gap between two consecutive integer numbers in a logarithmic function is resulting in a minimum error). For the purposes of this updated version, extensive practical research was done in order to add Kalman Filter (KF) to the application. Hence, we describe the theory of KF, its implementation and the extensive testing of it.

2 RELATED WORK

The systems introduced to find the location of a moving target are called Real Time Location System (RTLS). Cisco (CISC Systems, 2006) has implemented such a system addressed mainly to closed spaces. It is a commercial (not open sourced) system, cooperating with various hardware vendors, which combines software and hardware (Wi-Fi tags). Aeroscout is the first company introduced in 2003 an RFID tag compatible with Wi-Fi. As a result, it is possible to find the position of the target carrying this tag. The closest to this implementation is the "Ekahau" system, a not open source commercial system. It operates in open and close spaces and it seems to dominate the market, basically because of the accuracy, but also because it is not using any hardware, being able to cooperate with various systems and vendors.

Most of the papers investigated were vague (probably on purpose, due to the future commercial applications). A demanding job is described in (Interlink Networks, 2002). It describes the possibility of physically locating the position of a wireless intruder into a corporate network. An interesting job is also described in (P. Bahl et al., 2000). They faced the same problems, such as the extraction of the signal level info (without revealing much), the statistical error, and the distance from the broadcasting station. In (A. Seppänen et al., 2003) the extraction of the signal level through SNMP is described. It will be shown, that this is not always possible. The essay in (A. Kotanen et al., 2003) has some very interesting ideas, and it was the main inspiration for the KF work. So far the authors are not aware of such a work combining SNMP with Wi-Fi positioning.

3 DISCRETE TIME OPTIMAL FILTERS & PREDICTORS

The objective of such systems is to calculate "best" estimates of a state vector sequence x_k in the discrete interval $0 \le k \le k_{\max}$. Those estimates are based on all available (previous) information about the system and measurement(s) taken during the time interval. The time increment $t_k - t_{k-1}$ is usually constant, although not necessarily so.

The estimation problems fall into 3 categories:

1) If all the measurements are known prior to the appliance of the filter, then a *smoother* can be applied to the data.

2) If the time of calculation is synchronous with the time of the computation, then the filter has to take into consideration only previous data and is called a filter, as it "filters out" the noise of the system.

3) If an estimate of the system's position in future time is needed, then the filter is called a predictor.

Usually the *smother* is producing better results due to the usage of more data.

3.1 Kalman Filter

The purpose of a filter is to compute the state estimate, while an optimal filter minimizes the spread of the estimate- probability density in the process. A recursive optimal filter will propagate the conditional probability density function from one sampling to the next, taking into account system dynamics and inputs (R.F. Stengel,. 1994). Computing the weighting factors (also called *filter gains*), which optimally combine samples and future states, is the crucial step of the computation and the only part that needs extensive practical work.

"*Kalman Filter opens the way to the maximum likelihood estimation of the unknown parameters in a model…*"(A.C. Harvey, 1990). A KF may be applied for prediction and smoothing on models put in state space form. It is usually described by the following linear difference equation (R.E. Kalman, 1960):

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{1}$$

with a measurement $z \in \Re^m$ that is

$$z_k = Hx_k + v_k \tag{2}$$

The random variables w_k and v_k are process & measurement error factors. They are independent, white and with normal probability distributions, i.e. we assume that $p(w) \sim N(0,Q)$ and $p(v) \sim N(0,R)$. The $n \times n$ matrix A in (1), relates the previous step at *k-1* to the current step *k*. The $n \times 1$ matrix *B* in (1) relates the optional control input $u \in \Re^l$ to the state *x*. The $m \times n$ matrix H in (2) relates the state to the measurement z_k (2).

According to the above, applied KF needs only the previous position (state) of the system to calculate its current state. Combined with the measured position (state), the system outputs the current estimated position. If the position of a moving station on a given moment is x_k and this position is sampled every time fraction *t*, then the position of the moving station is described as follows:

$$x_k = T_t x_{k-1} + R_t a_k \tag{3}$$

Where α is the acceleration of the moving station (if the speed of the moving station is constant, then obviously α=1) and $T_t = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$ and $R_t = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}$.

3.2 Kalman Filter in Practice

For the needs of the current paper, KF in its simple form was used. The best practical explanation found for KF, was the one in (G. Welch and G. Bishop, 1995). Eq. (1) and (2) are directly depicted from it. In addition KF in practice needs to define the following states: $\hat{x}_k^- \in \Re^n$ to be the "*a priori*" state estimate at state k given knowledge, and $\hat{x}_k \in \Re^n$ the "*a posteriori*" state estimate at step k , given the measurement z_k .

Then the estimate errors can be defined as follows: $e_k^- \equiv x_k - \hat{x}_k^-$, as the *a priory* error, and its error covariance

$$P_k^- = E\left(e_k^- e_k^{-T}\right) \tag{4}$$

$e_k \equiv x_k - \hat{x}_k$, as the *a posteriori* error, and its error covariance

$$P_k = E\left(e_k e_k^T\right) \tag{5}$$

Then, the final KF equation can be described as

$$\hat{x}_k = \hat{x}_k^- + K\left(z_k - H\hat{x}_k^-\right) \tag{6}$$

where $z_k - H\hat{x}_k^-$ is the residual reflecting the discrepancy between the predicted measurement $H\hat{x}_k^-$ and the actual measurement z_k . If those two are equal, then obviously the residual equals to zero, or in other words there is no difference between the predicted and the measured state of the system. According to (G. Welch and G. Bishop, 1995). one form of K that minimizes

(5) is (7): $K_k = P_k^- H^T \left(HP_k^- H^T + R\right)^{-1}$ (7)

A practical representation of the two discrete ("predict" & "update") phases of the KF, are described in Fig. 1. Notice that when initial estimates $x_{k-1} = x_0$ and $P_{k-1} = P_0$ are known, they can be inserted once, at the start of "predict" state. Otherwise if the filter begins at the middle of the process, x_{k-1} and P_{k-1} are set to an arbitrary high (or respectively low) value, so the system will immediately ignore them, taking into consideration the first actual measurement(s).

As the filter was implemented, w_{k-1} in (1) was ignored. Matrixes Q & R were first arbitrary set, and there was a lot of discussion whether those values can be set in general. Since more than one (different) systems were not available, it remains an open question. The wisest strategy would be to set those matrixes after long observation, of each particular system.

3.3 Kalman Filter Appliance Results & Comparison Charts

For the implementation of the KF, there was a big set of data available. This data was collected in an open area system, roughly described as follows:

In an area of approximately 1,2Km^2, there were 13 APs implemented, covering the area at a non optimal way (there were "holes" – dead zones of signal coverage in the area).

A moving station recorded its own movement repeatedly in the area. Most of the research was based on this set of data. Time sampling was arbitrary set to 1 sec. This can be easily altered depending on each particular system needs (speed of movement is usually determining the time sampling). The area was transferred to a digital map of 600X600 pixels. Axes X,Y are mapped to each dimension of the map, by a scale of 1 meter to 1 pixel (1:1).

As Fig.2-3 is demonstrating, the recorded movement was far from being described as optimal. There was a lot of distortion and noise. Most important, there were several samples taken indicating positions completely outside the area drawn on the map. These are the result of the weaknesses of the system described at (G.E. Violettas et al., 2009) plus the high noise of the surrounding area.

After the appliance of the KF as depicted in Fig. 2-3, an almost optimal movement is demonstrated, in perfect accordance with the theory of KF.

Because KF is taking into consideration all the data available to predict the position, if a really distant point is inserted into KF it obviously affects the prediction (although in a very small fraction due to the nature of the filter). In order to avoid data that is obviously useless, before the data is passed to the KF there should be a simpler filter asking the following question:

If a station is moving with a speed of 5m/s, is it possible to alter its position by 10 meters from one second to the next and then return to a normal speed again? Obviously not, so this measurement should be disqualified.

Because of this problem (i.e. when a much distanced point suddenly "appears" inside the smooth-like movement), an algorithm for disqualifying this kind of data was implemented. More specifically, the algorithm is examining each set of data checking if it is different from the previous and the next set of data, more than one threshold. If it is, then this set of data is disqualified, and it is not taken into consideration at all. The threshold of the algorithm is still an open debate, as it highly relates to the nature of the movement (e.g. the threshold should be much smaller for a pedestrian walk than the one for a moving car).

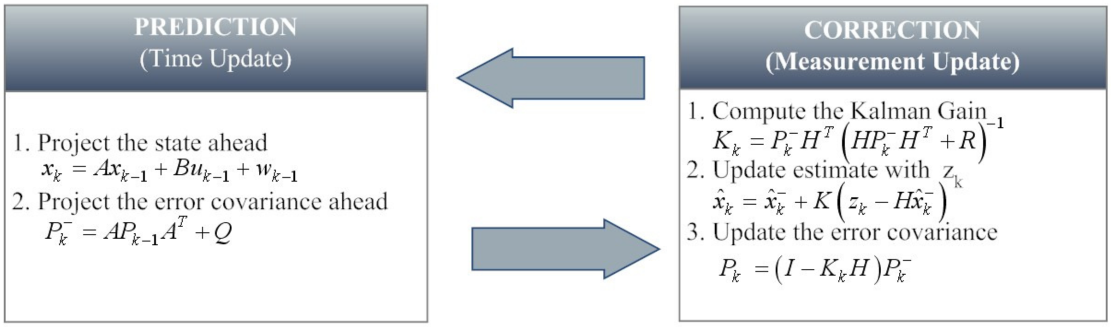
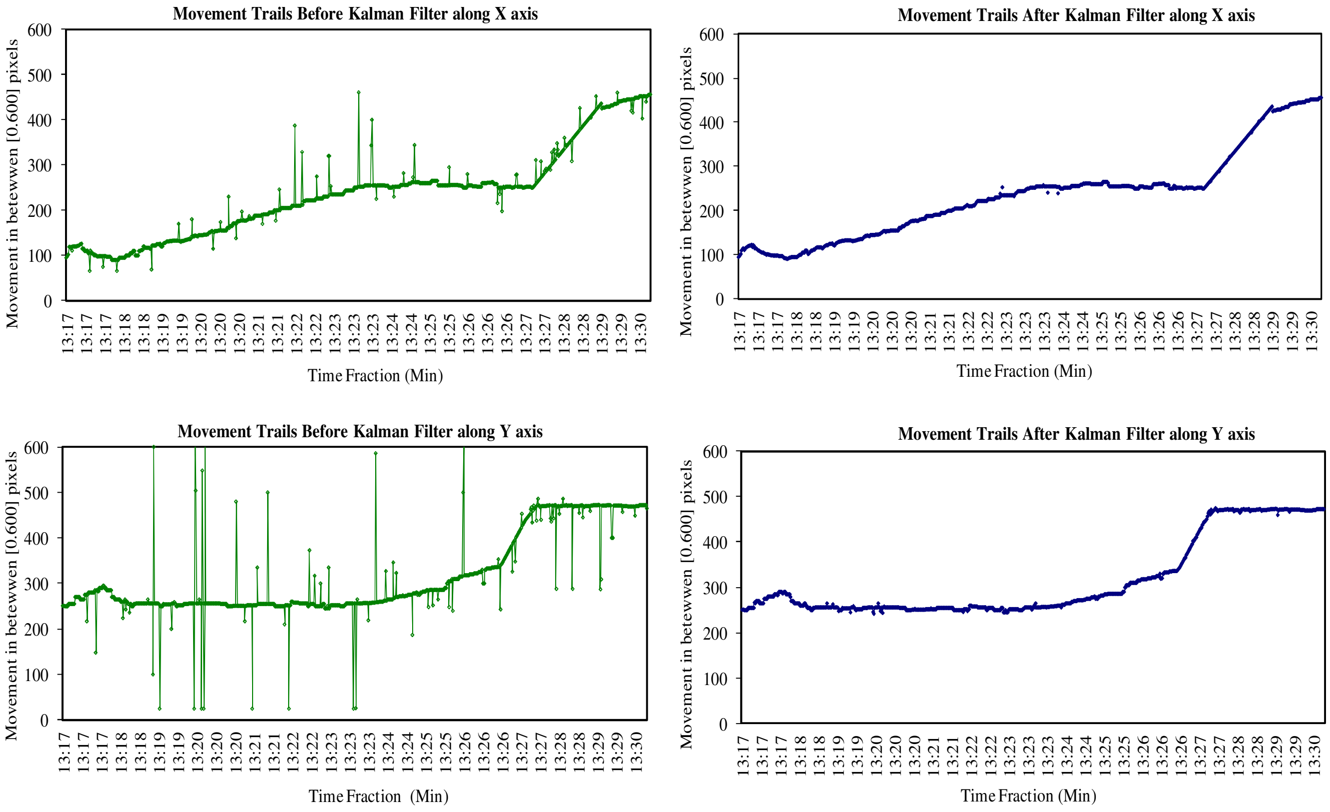


Figure 1. Kalman Filter in Action


Movement Trails Before Kalman Filter along X axis

Movement Trails After Kalman Filter along X axis

Movement Trails Before Kalman Filter along Y axis

Movement Trails After Kalman Filter along Y axis

Figure 2 . Movement Trails along axes X,Y, before & after Kalman Filter .

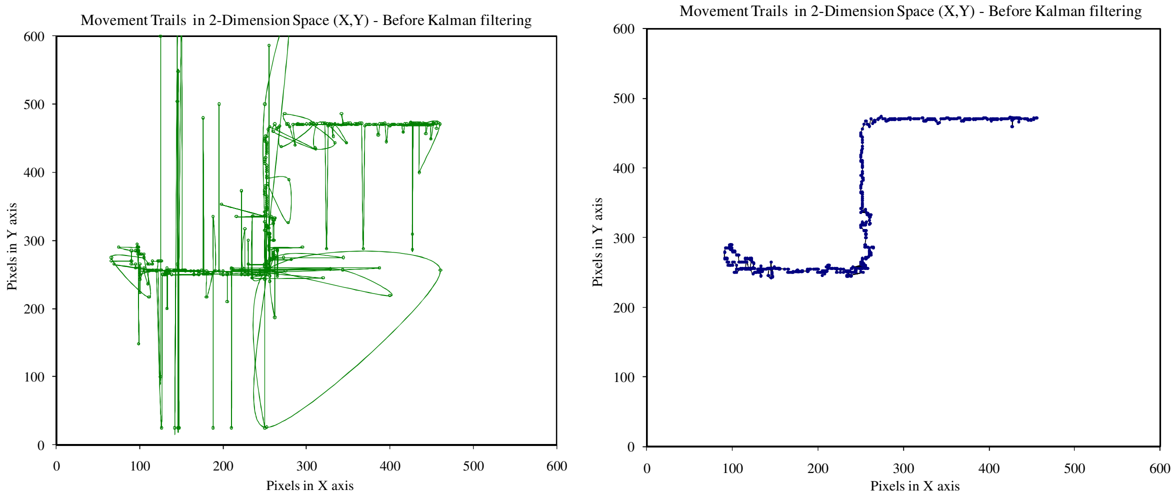


Figure 3 . Movement Trails along 2-D space maped, before & after Kalman Filter.

4 ARCHITECTHURE AND IMPLEMENTATION OF THE “NETARGUS” SYSTEM

The “*NetArgus*” application collects the available SNMP information from the network devices in order to construct the network structure and to investigate its status. *NetArgus* is a 3-tier network application that operates over TCP/IP networks. The system uses Microsoft SQL Server 2005 for data storage. The *NetArgus System* application is divided into three separate applications, which communicate using the common database and TCP/IP network messages. In detail:

4.1 NetArgus Client, Server, Viewer

This is the part of the software that is installed in the moving client’s PC. It sends in real time the signal level of all the APs in its region thought the wireless network card. The collected information is stored in the database. *NetArgus Client* also collects information about the AP’s Mac Address, SSID, Security protocol, Network type, speed, etc. in order to fill the database with as much information as it can.

It should be noted, that the application can easily carry SNMP information or any extra information provided by another I/O device. E.g. in Fig. 13, the NetArgus Client (at the upper right corner) is transmitting to the NetArgus Server the details (id) of a specific cargo named “Container CODE 1”, inserted by an RFID reader, any other source or even manually.

The application runs as a process in the background and it can be started or stopped using a tray icon. The signal level sampling time and the database location can be configured by the user through a windows form.

The second part of the application is the Server-side application. It reads information from the moving client (with the NetArgus application installed) and by using the triangulation algorithm described in (G.E. Violettas et al., 2009)., computes the absolute position of the above client, storing that information into the server database.

The server also does the entire SNMP related job. It stores the SNMP collected information, in addition to the collected information from the Wi-Fi signaling.

This is the graphical representation module that is used for presenting the network devices and the network structure on a map. Moreover it shows in real time the movement of the mobile clients in the covered area (Fig. 4).

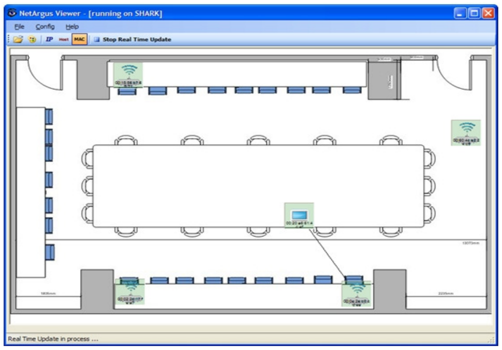


Figure 4. *NetArgus Viewer*, with Classroom map

5 CONCLUSION

The aim of this paper was to create an application for finding the position of a station. So, a three-tier distributed application was created, following all the modern design patterns. The positioning finding algorithms are running in real time, finding the trail of the moving station(s). However, due to the fact that noise exists, the addition of the KF addressed this problem with very accurate results. Thus, KF corrects (optimizes) those trails. The *NetArgus Client* is installed in every moving station we care of finding its position. It continuously sends to the database the received Signal level from all the APs within its reach. The application was tested in various harsh environments in real time. It did find the position of the station with a relative accuracy. Valuable results were gathered according to the quality of the received signal, the distance between station and the hardware used, specially the differences between various vendors.

6 FUTURE WORK

Various additions can be easily made in SNMP, such as exporting results of routing loads of a network and the possible optimization through dynamic routing protocols (RIP, BGP, IGMP), or possible implementation of network watch applications e.g., clever water leakage sensors. As for 802.11 protocols, there is a wide area open. This is the roaming protocol between the transmitting APs. The 802.11 protocol gives some general directions, but every manufacturer implements its own methods (most of the times without documentation) making the possibility of roaming between devices of different vendors, almost impossible. Moreover, the positioning finding algorithms can be improved and expanded specially on the direction of including smart devices such as telephones, active tags etc.

Augmented Reality (AR) is another very promising field of interest. For years, AR's usage has been restricted to high-end settings. Now, though, AR is beginning to move into more consumer settings, and the technology may be ready to become more commonplace and commercially successful. AR provides a digital enrichment of the physical world. More specifically, it projects images or information that augments what users see. For example, when users view their surroundings via the camera, an application displays information about nearby shops, restaurants, and other landmarks. According to *"Computer"* magazine (Steven J. Vaughan-Nichols, 2009) "*location-based services (LBS)—used with smart phones and other types of mobile technology—are a major driving force behind AR's entering the mainstream*".

The *NetArgus* can be firstly used for Finding the Physical Position of a Transmitting 802.11 Station. More specifically, with the use of any kind of vector maps (e.g. Google Earth), or any kind of DEM maps with Latitude & Longitude information, measuring up the signal strength at three random spots (Fig. 5), we can determine the co-ordinates of the broadcasting station in question.

Figure **5**. Triangulation of a known Target AP.

Secondly, *NetArgus* can be used for Mapping and Tagging the Signal Level of all the Transmitting Surrounding Stations in a given space (Fig.6), where several stations are broadcasting. This can lead to an optimization of the signal coverage through the physical re-location of the APs towards an optimum hexagon cell layout.

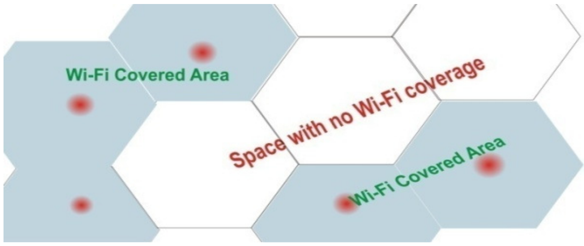


Figure 6. Empty space 802.11 signal coverage

REFERENCES

G.E. Violettas, T.L. Theodorou, and C.K. Georgiadis. (2009)"NetArgus: An SNMP Monitor & Wi-Fi Positioning, 3-tier Application Suite". La Boca/Cannes, France: icwmc, , pp. 346–351.

CISCO Systems. (2006) "Wi-Fi Based Real-Time Location Tracking: Solutions and Technology". *C11-346575-00*, vol. White Paper.

AeroScout, "Technology Overview" Available: http://www.aeroscout.com/content/technology.

Interlink Networks. (2002) Inc.. "A Practical Approach to Identifying and Tracking Unauthorized 802.11 Cardsand Access Points".

P. Bahl, V.N. Padmanabhan, and A. Balachandran. . (2000) "A software system for locating mobile users: Design, evaluation, and lessons". *online document, Microsoft Research, February*.

A. Seppänen, J. Ikonen, and J. Porras. (2003) "EXTRACTING AND USING POSITION INFORMATION IN WLAN NETWORKS". *WAWC'03*.

A. Kotanen, M. Hannikainen, H. Leppakoski, and T.D. Hamalainen. (2003) "Positioning with IEEE 802.11 b wireless LAN". *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003*, vol. 3..

R.F. Stengel. (1994) *Optimal Control and Estimation*. New York, NY: Dover Publications.

A.C. Harvey. (1990) *Forecasting, Structural Time Series Models and the Kalman Filter.* Cambridge: Cambridge University Press.

R.E. Kalman. (1960) "A new approach to linear filtering and prediction problems". *Journal of basic Engineering*, vol. 82, pp. 35–45.

G. Welch and G. Bishop. (1995) "An introduction to the Kalman filter". *University of North Carolina at Chapel Hill, Chapel Hill, NC.*

P.A. Ruymgaart. (1988) *Mathematics of Kalman-Bucy Filtering*, Berlin: Springer-Verlag.

Steven J. Vaughan-Nichols. (2009) "Augmented Reality: No Longer a Novelty?". *Computer*, vol. 42, pp. 19-22.