Real-Time Tracking for Augmented reality

M.Badeche
Department of computer science, Hadj Lekhdar University
Batna, Algeria
Badeche_mohamed@yahoo.fr

M.Benmohamed
Lire Laboratory,University of Mentouri
Constantine, Algeria
ibnm@yahoo.fr

ABSTRACT

In real time applications of augmented reality, it is always matter of tracking; one of the most promising techniques in this sense is tracking based pattern.

In this paper we describe a method based on the discrete Kalman filter for a real-time tracking of a 2D pattern for augmented reality. Objective being to make Real-time tracking of a 2D pattern through a video stream on the simple personal computer; the process can be resumed as follow: in each frame we do an estimation by Kalman filter of predict corner location, after we proceed to correct this estimation by a measure around predict location, with Harris corner detection.

the results obtains are very satisfactory, because all the process precede on real time and succeed to accurately track the pattern through video stream.

KEYWORDS: *Augmented reality, corner tracking, Real-time tracking, Kalman filter, corner detection.*

1. Introduction

The specificity of augmented reality which differs from virtual reality is that the first one aims to enhance or to augment the user's view with virtual objects generated by computer while the world of the second is completely artificial.

In pattern-based augmented reality it is matter of tracking a pattern (like in Figure 1) through a video stream and of adding the virtual object in every frame to his hypothetical location according to pattern's one.

Figure 1. Augmentation based on 2D pattern (left figure contains a 2D pattern, and that of the right shows the scene after augmentation) ***[3]***

In certain applications of augmented reality, time's factor is very critical, what impose that the process of tracking and augmentation has necessarily to take place during the transition of two frames in a live-video stream. From there appears the need to develop accurate but especially fast methods to allow to stay in these very tight temporal constraints.

1.1. Domains of application

The potential applications of augmented Reality are multiple: we find in particular applications in field of medicine, maintenance, manufacturing, industry and even entertainment.

1.2. Real-time Application in Medicine

Consider a system of augmented reality which tries to enhance the surgeon's view in full surgical operation by the insertion, in real time, of the other views stemming from sensors such as X-rays images (figure 2). It is evident that a delay in the augmentation of images can cause a disaster!

Figure 2. Augmentation with the internal image of patient's brain stemming from X-Ray data [1]

1.3. Tracked Pattern

To facilitate the identification of the Pattern considering the enormous raw information brought by an image, we opted for the 2D pattern (used in [4]) illustrated in figure 3 and which presents many advantages:

- It is strongly contrasted, what facilitates its detection.
- It is recognizable in a unique way, what allows associating him a unique orientation.
- It is suited for corner's detection through Harris's detector (2). The corners detection is desirable seen that it is not sensitive to radiometric changes.

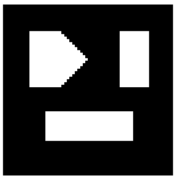


Figure 3. Kind of pattern to be tracked

2. The Discret Kalaman Filter

In 1960, R.E.KALMAN published his first-rate article describing a recursive solution for the problem of the linear filtering of discreet data. At this time and due to big progress realized in the field of numeral calculation and appearance of the first computers Kalman's filtering was subject to numerous extensive searches and application.

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown [7].

The Kalman filter addresses the general problem of trying to estimate the state $x \in 3^{n}$ of a discrete-time controlled process that is governed by the linear stochastic difference equation.

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}$$

With the measurement that is $z \in 3^{m}$:

$$z_k = Hx_k + v_k$$

The random variables w_k and v_k represent the process and measurement noise (respectively). They are assumed to be independent (of each other), and with Normal Probability Distributions.

$$P(w) \sim N(0,Q)$$

$$P(v) \sim N(0,R)$$

- In practice, the process noise covariance Q and measurement noise covariance R matrices might change with each time step or measurement, however here we assume they are constant.
- A (N×N) is a matrix which makes link between the state's variable of the system for two successive stages.
- B (N×N) makes link between the optional control values and the state's variable of the system.
- H (M×N) is the matrix which makes link between the state's variable of the system and the measures.

2.1. Estimation Process

Kalman's filter uses two sets of equations to estimate the value of the state's variable. The estimation of the state's variable in every stage passes by two phases: a phase of prediction and a phase of correction of the prediction:

2.1.1 Prediction Phase

This phase allows predicting the current value of the state's variable by taking into account the previous state. Variable predicted is called a priori state estimate. It allows also predicting the error covariance matrix by taking into account the covariance matrix of the previous state. This matrix is called the posteriori error covariance. So this phase uses two equations to predict the value of the state's variable which are:

$$x^-_k = Ax_{k-1} + Bu_k$$

$$P_k^- = AP_{k-1}A^T + Q$$

With:
x^-_k : priori state estimate.
P_k^- : Estimation of posteriori error covariance matrix.
x_{k-1} : Estimation of the state of the previous stage.
P_{k-1} : Estimation of the error covariance matrix of the previous stage..
u_k : control vector.
Q : the process noise covariance.
A, B : Associate the state at step k with the state at the step k-1.

2.1.2 Correction Phase

This phase allows to correct errors made in the phase of prediction by taking into account measure zk. Variable estimated is called a priori state estimate. It allows also estimating a posteriori error covariance matrix. This phase uses three equations to estimate the value of the state's variable and the error covariance matrix which are the input parameters for the following stage. These equations are:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

$$x_k = x_k^- + K_k (z_k - H x_k^-)$$

$$P_k = (I - K_k H) P_k^-$$

With:
x_k : a priori state estimate.
P_k : Estimation of posteriori error covariance matrix.
K_k : the Kalman gain.
R : measurement error covariance.
I : Identity matrix.

To use Kalman's filter, the measurement noise covariance (R) and the process noise covariance (Q) have to be estimated in advance. The determination of the process noise covariance is generally more difficult because the direct observation of the process state is not always possible.

There are two main stages for the evaluation of this state's variable; a priori evaluation (prediction) and a posteriori evaluation (correction). The state's variable of the system is estimated in the first place from values of the previous stage and then it is corrected by measures in the stage of correction.

3. Filter Tracking

In our application, Kalman filter is used to track corners localized on a determined pattern through a video sequence. So it is enough to specify what represents the state's variable and how to make measure to estimate the location of the corners in the view, and from this specification we will deduct the Kalman filter equations.

Our system is modeled by the following two equations:

$$x_k=Ax_{k-1}+w_{k-1}$$

$$z_k=Hx_k+v_k$$

Where x_k represents the state's vector at the step k :

$$x_k=[i_k\ j_k\ v_{xk}\ v_{yk}]^T$$

i_k and j_k represent the coordinates of the predicted point x_k,
v_{xk} and v_{yk} represent respectively the speed according to the X axis and the speed according to the Y axis.
z_k is the observation vector or the measure made at step k:

$$z_k=[z_x\ z_y]^T$$

z_x, z_y represent the coordinates of the measure z_k of the point x_k. In our application, measure is made by HARRIS corner detector.

A is the transition matrix or the matrix which associates current state to previous one.

$$A=\begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

H is the matrix which associates measure to the state's variable.

$$H=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

W_{k-1} is the process random vector of error with normal probability distribution $N(0,Q)$. vk is the measurement random vector of error with normal probability distribution $N(0,R)$.

Once the equations of our system described we can deduct those of Kalman filter to estimate in every frame the location of corners to be tracked on the model through video stream. As explained there are two sets of equations: equations of the prediction and equations of the correction.
The prediction equations are:

$$x^-_k=Ax_{k-1}$$

$$P_k^-=AP_{k-1}A^T+Q$$

The correction equations are:

$$K_k=P_k^-\ H^T\ (H\ P_k^-\ H^T+\ R)^{-1}$$

$$x_k=x_k^-+K_k\ (z_k-H\ x_k^-)$$

$$P_k=(I-K_k\ H)\ P_k^-$$

3.1. Tracking Algorithm

The adaptation of Kalman filter to our application consists of two main stages: tuning the filter parameters and tracking corners of pattern in a live video stream. Tuning of the filter is the process of initialization of the parameters of discrete Kalman filter (Q, R, X0, P0), it makes generally in an empirical way. Corner Tracking consists of the prediction of corners location and the correction of this prediction by taking into account measure made by Harris corner finder. It can be described by the following algorithm:

1- Initialize Q, R, x0 z0 and P0 (tuning the filter).
2- For every step k … do:

- Predict P_k^-, x_k^- by using the equations of prediction.
- Calculate the gain K_k and update P_k, x_k by using the equations of correction by taking into account measure made in this step.
- The output x_k and P_k are the input for the next step.

3.2. Filter Tunig

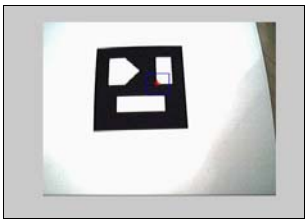
Tuning the filter has a direct impact on the accuracy of the filter during the Tracking of corners. Consequently, an appropriate choice must be made. In the case of our application, Q has to be relatively small with regard to R; it is because we don't tolerate errors at the level of the theoretical estimation as at the level of the measure.
Besides, to ameliorate performance and to reduce computational cost we can reduce the search for corners on a part of the image strikingly smaller size of which corresponds to the covariance matrix P which represents the reliable margin of the prediction filter. The choice of P has to satisfy the following two conditions:

- Size of search window has to be big enough to correctly track the corner along the video stream.
- Size of search window must be small enough to don't make a mistake and track another corner.

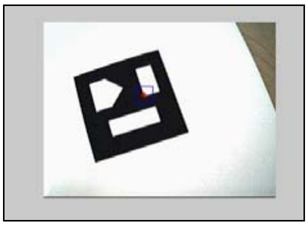
These two conditions must be satisfied to avoid the problem of loss of point during the process of the tracking.

4. Experimental Results

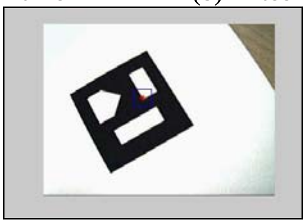
We notices that the tracking by the discret Kalman filter is very satisfactory (figure 4); indeed the efficiency of the filter is due to the fact that it is more that a simple algorithm, it is a mathematical tool demonstrated in theory.



(a) First frame



(b) Fifteenth frame



(c) Frame N°21

Figure 4. Tracking of a corner through a video stream

A tool which expresses better the process of tracking is the graph showing the actual position of the corner to be tracked with regard to that predicted with Kalman filter. Such a tool can represent a help for the tuning of the filter (Figure 5).

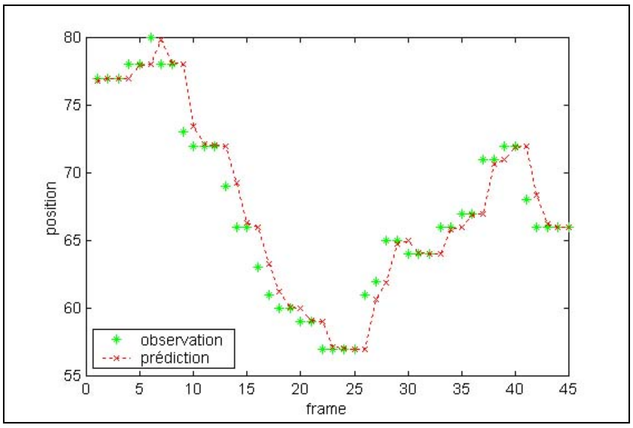
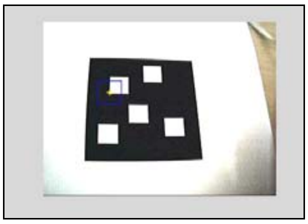
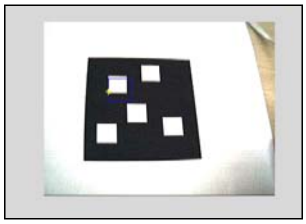


Figure 5. Abscissas of the predicted location with regard to the observed location)

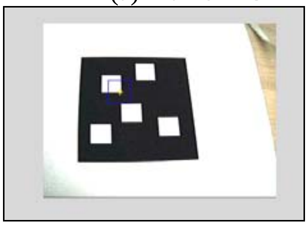
4.1. Filter Tuning



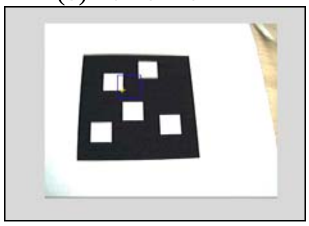
(a) Frame N°3



(b) frameN°6

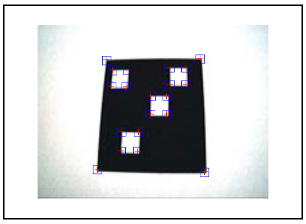


(c) Frame N°7

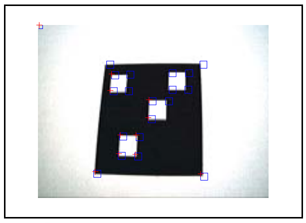


(d) frame N°10

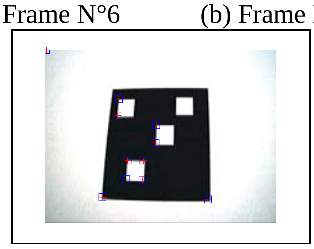
Figure 6. With a too big size of search window, System loss the corner and track another



(a) Frame N°6



(b) Frame N°7



(c) Frame N°8

Figure 7. Loss of corner because of a size of search window witch is too small

In the practice, the local search window must be enough small for do not make a mistake about corner during the track (Figure 6), but relatively big to allow a more free dynamics of the corner movement (Figure 7). In our method, the tracking is made robust by the introduction of the Ransac algorithm.

4.2. Real-Time Performance

In what follows, we are going to discuss method in term of computational cost on standard PCs; we begin in the first place with the tracking of a video stream on several machines with different physical characteristics to see the impact of these on the speed of computation of our application and we ends to give detailed measures of every phase of the tracking, what is going to allow us to end on the efficiency and the realism of our method.

The following table (Table 1) as well as its graph (figure 8) shows the Tracking Performance of ten video frames with various resolutions according to the machine to which method was applied.

	0,48/32	1/128	2,1/96	2,1/128	2,8/256	3,2/512
320/240	4238	2430	1427	1335	961	739
160/120	1162	775	330	325	239	187
80/60	525	328	125	122	88	75

Table 1. Tracking performance of ten video frames according to the resolution and the characteristics of the machine of execution

- Lines represent frame resolution in pixel.
- Columns represent the characteristics; speed of the processor in Giga hertz (GHz) and capacity of memory in Mega byte (Mb).
- The intersections of lines and columns represent time of tracking of a video stream to a given resolution and with a PC of given characteristics. Time is measured in ms.

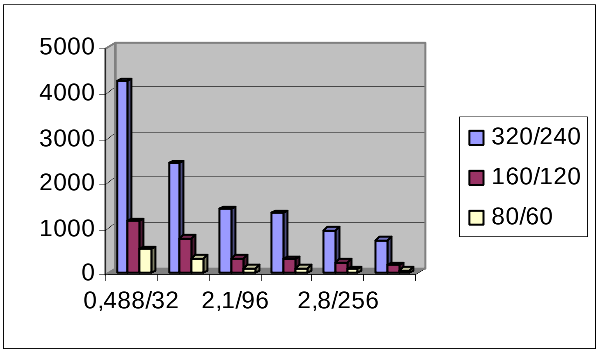


Figure 8. Table 1 Graphic

According to the graph this high (Figure 8) we make the following conclusions:

1. Tracking operate on real-time; the limit of ten images per second necessary to guarantee the fluidity of the video stream [Vallino. J, 1998] is easily reached with standard personal computers; a computer having 2.8 GHZ as speed of the microprocessor and 256 Mb of RAM is enough to reach the speed required to track images with a resolution of 320/240.
2. The reduction of the resolution, leads the increase of the speed of the tracking but at the same time, a decrease of the precision. However with 320/240 resolution images (and even less) the quality of images is preserved and the accuracy is widely satisfactory.
3. By an interpolation we can deduct approximately the time of execution for a given resolution.
4. The two characteristics which are speed of the processor and capacity of the memory influence at the time on the computational cost of the tracking.

The following table (Table 2) as well as its graph (Figure 9) shows processing time of each phase in the tracking of ten video frames on a PC (3.2 GHz / 512 Mb).

	One Corner Tracking (ms)	24 Corners Tracking (ms)	Corner Finder (ms)
320/240	692	739	470
160/120	156	187	160
80/60	60	75	50

Table 2. Tracking Performance

- Lines represent frame resolution in pixel
- Columns represent the details of the process of tracking.
- Intersections represent the computational cost in 'ms'.

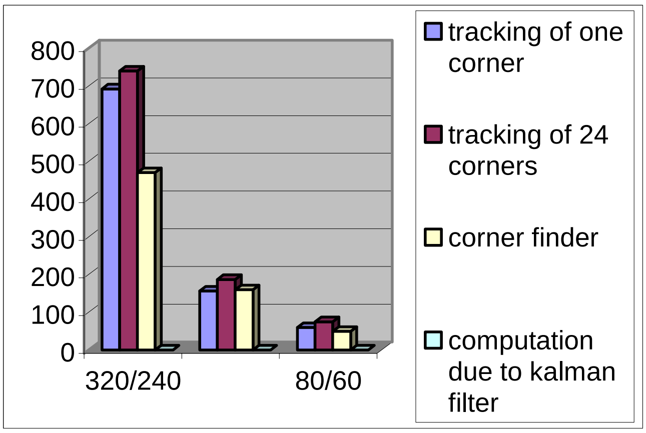


Figure 9. Table 2 Graphic

According to obtained results we extract the following conclusions:

1. The computational cost of Kalman filter is not significant with regard to the total cost of the tracking.
2. In the practice, the number of pattern's corners has no impact at the computational cost. Consequently, to follow a model with 24, 20 or 16 corners does not affect the cost of tracking.
3. Except for time lost in the pure details of the programming such as allocation / liberation of images and matrices, it is the phase of corner's detection which is the greediest in term of computational cost.

5. Conclusion

The Tracking method presented here shows an interesting accuracy and especially a satisfactory processing time on simple PC.

We notice essentially that:

- Tracking operate on real-time.

- In spite of his probabilistic origin relatively complicated, the computational cost of Kalman filter is negligible. So we have accuracy due to the use of such tool demonstrated mathematically and on the other hand, engendered computational cost is not penalizing.
- There is a relative freedom in the choice of the pattern to be followed because the number of corners does not influence time processing.

6. REFERENCES

[1] Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B. "Recent Advances in Augmented Reality". IEEE Computer Graphics and Applications. November/December 2001.

[2] Harris. C and Stephens. M. A combined corner and edge detector, Proceedings of the Fourth Alvey Vision Conference, pp. 147-151, 1988.

[3] Hirokazu Kato, Mark Billinghurst, Ivan Poupyrev. *ARToolKit User Manual*, *Version 2.33*. Human Interface Technology Lab, University of Washington. November 2000.

[4] Malik, S., Roth, G., McDonald, C. In Vision Interface 2002. Calgary, Alberta, Canada. May 2002. pp. 399-406.

[5] Ronald T. Azuma. "A Survey of Augmented Reality". Presence: Teleoperators and Virtual Environments 6, 4. August 1997. pp. 355-385.

[6] Vallino. James R. *Interactive Augmented Reality*. PhD Thesis, University of Rochester, Rochester, NY. November 1998.

[7] Welch. G, Bishop. G. An introduction to Kalman filter. SIGGRAPH'2001 course 8., pp.19-33, 2001, http://www.cs.unc.edu/~{welch, gb}.