CONSIDERATIONS ABOUT KALMAN FILTRATION APPLIED TO SURFACE RECONSTRUCTION METHODS

D.E. MIHAI[1], E. STRĂJESCU[2]

Lucrarea descrie posibilitatea evitării unor situații neplăcute care apar in procesarea datelor experimentale, in particular vizând aplicația de reconstrucție de suprafață si anume situațiile particulare in care funcțiile sau derivate ale lor trec prin valori extreme. Oricare ar fi cauza erorilor, fie ca sunt generate de sistemul măsurat in cazuri particulare, fie ca sunt generate de cazuri neacoperite de teorie sau chiar de teorie in sine rezultatul este blocarea algoritmului, obținerea de date eronate sau propagare incontrolabilă de erori.

This paper describes the possibility to avoid some critical situations which can arise in experimental data processing, more precisely in applications of surface reconstruction, spotting the particular situations where functions or there derivatives pass by extreme values. No matter which is the error source, either they are generated by the measured system in some particular cases, either they are generated by special cases uncovered by the theory or even generated by the theory itself, the consequences are the failure of the algorithm, the obtaining of strange data values or the propagation of incontrollable errors .

Keywords: Kalman, filtration, surface reconstruction, recursion, numerical processing.

1. Introduction

In the 60's R.E. Kalman has published an article, a very famous one, where he described a recursive solution for solving the problems of discrete data filtration. Since this article, mainly because of the great evolution that had place in the domain of numerical data processing, the Kalman filtration has been the subject of an intense research and development of various applications.

Main applications in industry has been developed for systems with continuous data acquisition based on a sequence of observations of a system state to provide accurate continuously updated information about evolution of different parameters.

[1] Eng., ADG Design SRL, Bucharest, ROMANIA
[2] Prof., Dept. of Machine tools and production systems, University "Politehnica" of Bucharest, ROMANIA

2. Theoretical considerations

Presuming that there exists a system that can be described by a **m** dimensional vector that is measured at time intervals k = 0, 1, 2, ..., each observation being affected by noise, for each measurement we have:

$$Z_k = H \cdot x_k + v_k \quad (1)$$

where:
Z_k are the measurable unit;
v_k are the random values of the measurement;
x_k is the state of the system at the step k;
H is the observation matrix (a link between the observed and the true process).

Formulated this way the problem cannot be solved. What is missing is a link between the current state x_k and the x_{k-1} state.

To solve this problem, we can presume that the link between two successive states is a linear function and that there exists an additional vector, “noise”, that is equivalent to the dynamical system uncertainities.

The Kalman filtration is trying to solve the general problem to estimate a state „x” of a time dependent discrete process that is ruled by a differential linear equation like:

$$x_k = A \cdot x_{k-1} + B \cdot u_{k-1} + w_{k-1} \quad (2)$$

The variables w_k and v_k are presumed to be independent of each other and to have a normal probability distribution.

The **A** matrix ($n \times n$ dimension) in the differential equation (2) makes the link between the state of the system k - 1 and the state of the system k, besides of any link function or process noise. In a normal way, in a real process, the **A** matrix can be modified at each step of iteration. Anyway, it can be considered that the **A** matrix is constant without diminishing the generalization of the problem.

The **B** matrix ($n \times l$ dimension) is correlated to the optional control value of the process $u \in R^l$ at the state x of the system.

The **H** matrix ($m \times n$ dimension) in the measure equation (equation 1) is correlated to the Z_k measure. Practically, the **H** matrix can also be modified at each step.

The Kalman filter is estimating a process using a feedback form: the filter is estimating the process state at a given time and than is obtaining a feedback like a noisy measurement. More, the equations of the Kalman filtering are divided in two groups: time dependent equations and measurement dependent equations.

The time dependent equations can be used to make predictions (time predictions) starting with the current state and errors estimations to obtain estimations for the next step of the process.

The equations that depends of the measuring process result can be considered resposible of the system feedback, for example, by incorporating a new measure in a measure estimation to obtain a better estimation for the next step.

The time dependent equations can be considered prediction equations.

The measure dependent equations can be considered correction equations.

3. Solving the differential equation of the surface height

From the illumination equation:

$$E(x,y) - I_s = 0 \tag{3}$$

where:
E is the measured reflectance map
I_s is the the iluminaton

can be obtained the approximate equation:

$$f\left(Z^{n-1}(x,y)\right) + \left(Z(x,y) - Z^{n-1}(x,y)\right) \cdot \frac{df}{dZ(x,y)}\left(Z^{n-1}(x,y)\right) = 0 \tag{4}$$

The above equation can be rewritten as:

$$Z^n(x,y) = Z^{n-1}(x,y) + K^n\left(-f\left(Z^{n-1}(x,y)\right)\right) \tag{5}$$

where K^n has to satisfy three conditions:

- K^n is approximate equal with the inverse of $\frac{df}{dZ(x,y)}\left(Z^{n-1}(x,y)\right)$; (I)
- K^n approaches zero when $\frac{df}{dZ(x,y)}\left(Z^{n-1}(x,y)\right)$ approaches zero; (II)
- K^n approaches zero when $Z^n(x,y)$ approaches zero. (III)

The operators involved in K^n definition has been identified like:

- a constant, having unique value and non zero, which is used to avoid divi -

ding by zero, $W_{x,y}$;

- the anticipation operator

$$\delta^n_{x,y} = E\left[\left(Z^n(x,y) - Z(x,y)\right)^2\right] \tag{6}$$

- from equation (5) and condition (I):

$$M_{x,y} = \frac{df}{dZ(x,y)}\left(Z^{n-1}(x,y)\right) \tag{7}$$

With these three elements can be found many K^n functions which can comply to all the three conditions from above.

The best known function is:

$$K^n = \frac{S^n_{x,y} \cdot M_{x,y}}{W_{x,y} + S^n_{x,y} \cdot M^2_{x,y}} \tag{8}$$

The above function gives very good results and has been implemented in many algorithms but is quite inefficient from the point of view of system resources. Even it represents a general possibility to solve the problems described above (chapter 2), the whole solution is difficult to implement in an industrial system.

Because at moment of screening or of various manipulations the surface suffers in few times normalizations, we preferred to rewrite the equation (5) and to work on a “surface” already normalized. We used the quotes referring to the surface because at the moment of calculations in fact the surface is not yet calculated, however the first and the successive approximations are also normalized. From the view point of screening and graphical manipulations this trick has not any importance, but it reduces quantity of operations and avoid manipulations of big numbers and powers of big numbers. During the tests we observed an important diminution of execution times.

Equation (5) rewritten looks like (we kept the same notations like above):

$$Z^n(x,y) = Z^{n-1}(x,y) - \frac{f(z^{n-1}(x,y))}{W_n + K^{*n}} \tag{9}$$

Obviously, K^* does not comply anymore with the condition (I), but by rewriting the equation in essence the things are the same and the generality of the solution is not affected in any way.

In this conditions it is possible to write:

$$K^{*n} = \frac{\delta_{x,y}^{n}}{\alpha} \cdot M_{x,y} \tag{10}$$

where α is an integer above 1. Good results has been obtained for α = 2.

With this algorithm (and the C++) code can be easily viewed bellow in the table 1:

Table 1.

for(i=0;i<width;i++){	- see note 1
for(j=0;j<height;j++){	
Zn[i][j]= 0.0;	
Zn1[i][j]= 0.0;	
Si[i][j] = 0.0;	
Si1[i][j]= 0.01;	
}	
}	
for(t=1;t<=Ival;t++){	
for(i=0;i<width;i++){	
for(j=0;j<height;j++){	
if(j-1 < 0 || i-1 < 0)	
p = q = 0.0;	- see note 2
else {	
p = Zn1[i][j] - Zn1[i][(j-1)];	- see note 3
q = Zn1[i][j] - Zn1[i-1][j];	
}	
pq = 1.0 + p*p + q*q;	
PQs = 1.0 + Ps*Ps + Qs*Qs;	
Eij = Ll[i][j][0];	
fZ = -1.0*(Eij - MAX(0.0,(1+p*Ps+q*Qs) / (sqrt(pq)*sqrt(PQs))));	- see note 4
dfZ = -1.0*((Ps+Qs)/(sqrt(pq)*sqrt(PQs))-(p+q)* (1.0+p*Ps+q*Qs)/(sqrt(pq*pq*pq)*sqrt(PQs)));	
Y = fZ + dfZ*Zn1[i][j];	
K = Si1[i][j]*dfZ/2;	
Si[i][j] = (1.0 - K*dfZ)*Si1[i][j];	- see note 5
Zn[i][j] = Zn1[i][j] – fZ/(Wn + K);	
}	
}	
for(i=0; i<width; i++) {	
for(j=0; j<height; j++) {	
Zn1[i][j] = Zn[i][j];	- see note 6
Si1[i][j] = Si[i][j]; }}	

Notes:
1. This sequence of code initialize all variables, surface heights to zero, the estimated values to zero, the intermediate values to non zero.
2. This sequence of code show that the image of the borders *p* and *q* are considered to be zero, for all the other points are approximated from neighbors.
3. This sequence of code initialize the measure anticipation from the values table.
4. This sequence of code calculates *fz* and *dfz* ;
5. This sequence of code estimates heights ;
6. This sequence of code transfers heights back to the value table for processing.

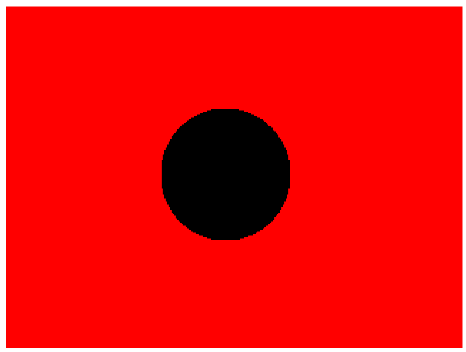

Fig. 1. Synthetic image

A numerical application of the above algorithm for surface reconstruction of the synthetic image from the fig. 1 (320 × 240 pixels) is shown bellow.

Stage 1: RGB filtering for input data preparation in Kalman filtration. The mean RGB value before the filtration is 76. The mean value after the filtration is 68. The mean values has been converted to the integers because of the RGB format.

Stage 2: The process variables have been initialized to the shown values: the initial surface heights to zero, the estimated values to zero, the anticipation operator to 0.01.

Stage 3: The algorithm has been run to the prepared image (after the RGB filtration) and bellow will be presented only the statistical parameters of the first 5 steps (for space and relevance reasons), all the other steps are following the same trends and only getting the surface approximation better and better. The values shown bellow were collected after a full cycle completion.

Step 1:

The mean value of the anticipation operator: 0.009999;

The mean value of the surface heights: - 0.8124;

The maximum height: 0.0809;
The minimum height: - 0.8256;
Skewness: - 0.0233.

Step 2:

The mean value of the anticipation operator: 0.009999;
The mean value of the surface heights: - 1.6218;
The maximum height: 0.1775;
The minimum height: - 1.6483;
Skewness: - 0.02315.

Step 3:

The mean value of the anticipation operator: 0.009998;
The mean value of the surface heights: - 2.4301;
The maximum height: - 2.4699;
The minimum height: 0.2754;
Skewness: - 0.02301.

Step 4:

The mean value of the anticipation operator: 0.009998;
The mean value of the surface heights: - 3.24;
The maximum height: 0.3712;
The minimum height: - 3.2937;
Skewness: - 0.02311.

Step 5:

The mean value of the anticipation operator: 0.009998;
The mean value of the surface heights: - 4.0461;
The maximum height: 0.4723;
The minimum height: - 4.1123.
Skewness: - 0.02296.

Stage 4: The surface heights are transfered back to the heights array. The surface is now ready for further calculations.

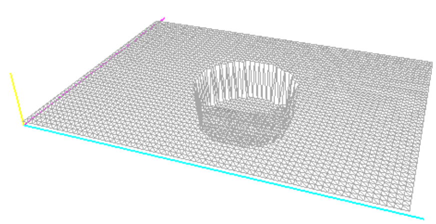

Fig. 2. Surface reconstruction

The final result of the reconstruction can be observed in fig. 2.

4. Conclusions

The Kalman filtration implementation for the data that result in the surface measurement and illumination processing of a surface, can help to avoid the particular situations where functions or their derivatives are experimenting extreme. Either generated by particular positions of light source and CCD sensor, or generated by surface particularities, the extremes of functions and their derivatives lead to errors generated by the computer, to the algorithm freezing, or, even worse, to a unreal surface, very rough and very far from what had to be reconstituted.

The proposed function has been tested with an algorithm used for reconstruction of surfaces from images acquired from a single camera. The algorithm has been tested against well known algorithms and proved itself to be competitive from point of view of the quality of reconstructed surface and more important from point of view of execution times. The functions and the Kalman filtration implementation is mainly useful because by a calculation trick can solve difficult problems, in other ways impossible to be solved.

R E F E R E N C E S

[1]. *Greg Welch, Gary Bishop*, An Introduction to the Kalman Filter, TR 95-041, Department of Computer Science University of North Carolina at Chapel Hill Chapel Hill, NC 27599-3175, 2004.

[2]. *Horn and Brooks*, Shape from Shading, MIT Press, 1989.

[3]. *J. Oliensis*, Shape from shading as partially well constrained problem, CVGIP, 1991.