Zbigniew GOMÓŁKA, Bogusław TWARÓG, Paweł KRUTYS

University of Rzeszow, Institute of Technology, Department of Electrical Engineering and Computer Science

Prediction mechanism for the face tracking algorithm

***Abstract**. The paper presents a modified version of the algorithm dedicated for face tracking in video stream. The detection procedure involves gradient technique and the prediction of the face coordinates by the modified adaptive Kalman filter. The proposed combination of face contour detection and coordinates extraction can be used in modern interfaces to facilitate user interaction with the computer.*

***Streszczenie.** W pracy przedstawiono zmodyfikowaną wersję algorytmu śledzenia twarzy w strumieniu video z wykorzystaniem techniki gradientowej macierzy oraz predykcji współrzędnych twarzy poprzez zmodyfikowany adaptacyjny filtr Kalmana. Zaproponowana metoda wyodrębniania konturu i współrzędnych twarzy może znaleźć zastosowanie w nowoczesnych interfejsach ułatwiających interakcję użytkownika z komputerem. (**Algorytm śledzenia twarzy z wykorzystaniem mechanizmu predykcji**)*

Keywords: face tracking, Kalman filter, prediction.
Słowa kluczowe: śledzenie twarzy, filtr Kalmana, predykcja.

Introduction

With the accelerating growth of information technology, growing interest in new techniques of human interaction with computers and machines (HCI). One such example is the idea of a traffic control in a virtual 3D scene by tracking the position of the observer's face. Known in the literature, face-tracking algorithms utilize two basic modules: gradient detection module and matrix of hits also referred to as the hit-matrix which is the certain form of the Hough's accumulator space. In this paper w will try to improve the speed up hit-matrix computations by using Kalman filter.

Assumed edge detection method

The first stage of a classical algorithm for face detection is to determine the matrix of gradients, which will include local changes of illumination in the image obtained in early processing step. Besides determination of the gradients matrix in the vertical and horizontal directions, the suitable gradient matrices for angles of 45o and 135o is calculated, as shown on the Fig.1.

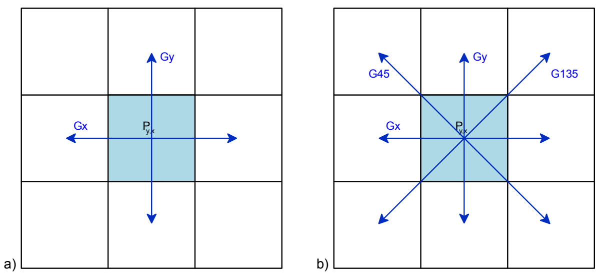

Fig. 1. Calculated gradients a) only for the vertical axis b) with respect to the axis of skew.

Gradient matrices are determined as follows:

(1) $$Gx_{y,x} = P_{y,x-1} - P_{y,x+1}, Gy_{y,x} = P_{y-1,x} - P_{y+1,x}$$

(2) $$G45_{y,x} = P_{y-1,x-1} - P_{y+1,x+1},$$
$$G135_{y,x} = P_{y-1,x+1} - P_{y+1,x-1}$$

where: $P_{y,x}$ - the brightness value at the position [y,x], $Gx_{y,x}$ - horizontal gradient value at the considered point $P_{y,x}$, $Gy_{y,x}$ - value of the vertical gradient at the point $P_{y,x}$, $G45_{y,x}$ - value of the gradient at an angle 45° in the point $P_{y,x}$, $G135_{y,x}$ - value of the gradient at an angle 135° in the point $P_{y,x}$. For each point of the image we can calculate the length of the vector gradient, using the following formula:

(3) $$G_{y,x} = \sqrt{(Gx_{y,x})^2 + (Gy_{y,x})^2},$$

and similarly for pairs of inclined gradients:

(4) $$G2_{y,x} = \sqrt{(G45_{y,x})^2 + (G135_{y,x})^2}.$$

The points, where the length of the vector Gx or Gy is less than some empirically accepted values can be excluded from further calculations. We can also assume that these points are caused by noise and do not belong to the edges in the image. This is similar circumvention known as the adaptive manner of edge detection used by Canny-Deriche edge detector. The idea behind the procedure of searching for the face plant of the selected point presents Fig.2.

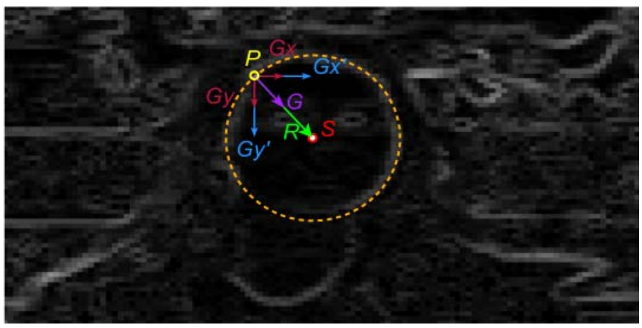

Fig. 2. Determination of the circle center containing the face for the matrix of perpendicular gradients G $_x$ and G $_y$ respectively.

Description of the standard terminology that is used to describe consecutive symbols: *P* - currently analyzed image point, *Sy,x* - the probable measure of oval face center we are searching for, R_o - expected radius of the contours of observed face (in the figure marked as *R*), *Gy,x* - gradient vector at point *P* (which is determined from the values of gradients in the matrices *Gxy,x* and *Gyy,x*), *Gx'y,x Gy'y,x* - horizontal and vertical component of the *R* vector. Subsequently the formulas for the coordinates of the circle *S* can be derived:

(5) $$R_o^2 = (Gx')^2 + (Gy')^2, G^2 = (Gx)^2 + (Gy)^2,$$

(6) $$\frac{Gx}{G} = \frac{Gx'}{R_o}, \frac{Gy}{G} = \frac{Gy'}{R_o}.$$

After the successive transformations we obtain[2]:

$$Gx' = \frac{R_o Gx}{\sqrt{Gx^2 + Gy^2}}, Gy' = \frac{R_o Gy}{\sqrt{Gx^2 + Gy^2}}. \quad (7)$$

Eventually, searched coordinates of the circle S are formulated as:

$$S = (x + Gx', y + Gy'), \quad (8)$$

where: *x,y* - coordinates of the analyzed point.

A similar analysis for the *G45y,x* and gradients *G135y,x* Fig. 3 shows.

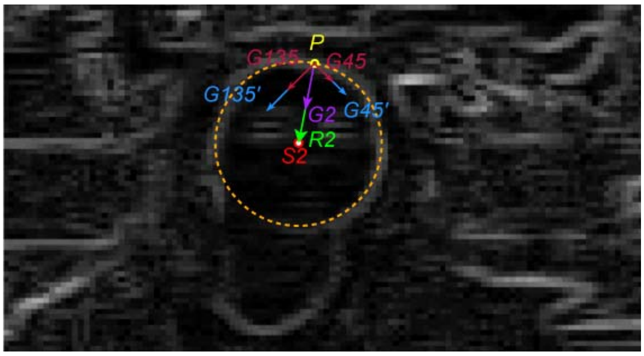

Fig. 3. Determination of the circle containing the face for the gradient matrix pair *G45* and *G135*

Assumed description: *P* - currently analyzed image point, *S2y,x* - the predicted measure for an oval face we are searching for, $R_o 2$ - expected radius of the oval face, *G2y,x* - gradient vector at the point *P* (which can be determined from the values of gradients in the matrices *G45y,x* and *G135y,x*) respectively, *G45'y,x* - component of the inclined axis 45^o for *R* vector, *G135'y,x* - component of the inclined axis 135^o for *R* vector too. On the basis of of Fig.3 we can derive formulas for the coordinates of the circle center *S2*:

$$R_o 2^2 = (G45')^2 + (G135')^2, \quad (9)$$

$$G2^2 = (G45)^2 + (G135)^2, \quad (10)$$

$$\frac{G45}{G2} = \frac{G45'}{R_o 2}, \frac{G135}{G2} = \frac{G135'}{R_o 2}. \quad (11)$$

Eventually, after transformations we obtain:

$$G45' = \frac{R_o \cdot G45}{\sqrt{G45^2 + G135^2}}, \quad (12)$$

$$G135' = \frac{R_o \cdot G135}{\sqrt{G45^2 + G135^2}}. \quad (13)$$

Sought coordinates of the circle center *S2* are:

$$S2 = (x + \frac{G45'}{\sqrt{2}} + \frac{G135'}{\sqrt{2}}, y + \frac{G45'}{\sqrt{2}} - \frac{G135'}{\sqrt{2}}), \quad (14)$$

In the language of image processing, these operations can be interpreted as a convolution of image with the modified Prewitt kernels:

$$f_{Gx} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}, f_{Gy} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}, \quad (15)$$

$$f_{G45} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, f_{G135} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad (16)$$

In the initial experiments were used as input images free images from the database of images VIS_DB [3]. Thus obtained gradient map stands the input information to the determination of hit-matrix stage in which the work parameter algorithm will be adopted value of the radius *R*.

Stabilized hits matrix

The classic version of the parametric Hough transform uses an array of so-called accumulators of best matches. In this paper, we use a similar mechanism called the hit-matrix. For the localized coordinates of circle center *S* and *S2* the increment of matrix elements values at the coordinates of the designated zones, whose size is equal to the size of the analyzed image is made. The initial values of this matrix are zero. The way in which elements values of the hit-matrix are increased and its scope presents Fig.4. The matrix is modified by the values indicated in this figure, at positions of the found point *S* or *S2* and its eight neighbors. Since the calculated coordinates in a video stream may be subject to some small error due to optical or *Salt & Pepper* noise and momentary changes in scene lighting conditions, we assumed weighted mechanism for updating the values of the hit-matrix. Incrementing in a fuzzy manner values of the matrix elements provides a stabilizing effect of a sequence of computational results. Example array of hits, which was constructed for a certain portion of the video stream is illustrated in Fig.5

$S_{y-1,x-1}$ +1	$S_{y-1,x}$ +2	$S_{y-1,x+1}$ +1
$S_{y,x-1}$ +2	$S_{y,x}$ +3	$S_{y,x+1}$ +2
$S_{y+1,x-1}$ +1	$S_{y+1,x}$ +2	$S_{y+1,x+1}$ +1

Fig. 4. Neighborhood function for hit-matrix updating.

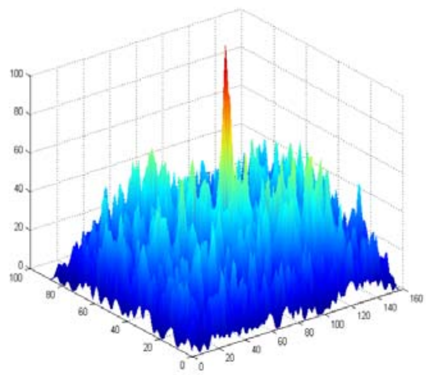

Fig. 5. Sample hit matrix in the 3D view.

To increase the speed of the algorithm it was assumed that the hit-matrix will be constructed for the selected frames only, while between them, the face tracking prediction will be conducted.

Kalman filter in the task of motion estimation

Kalman filter is a recursive data processing algorithm, which makes state estimation of dynamical system distorted by noise [4]. We define system state vector x consisting of n variables that describe essential properties of the system. Kalman filter addresses the general problem of state estimation process $x \in \Re n$, which is controlled by a stochastic differential equation:

(17) $$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1},$$

with the use of measurement $z \in \Re m$

(18) $$z_k = Hx_k + v_k.$$

Random variables w_k and v_k respectively denotes the noise of process and measurement. It is assumed that they are independent from each other and correspond to the white noise with normal distribution:

(19) $$p(w) : N(0, Q), p(v) : N(0, R).$$

Conditional probability for the above model can be written as follows [4]:

(20)
$$p(x_k \mid x_{k-1}) : N(Ax_{k-1} + Bu_k, Q), p(z_k \mid x_k) : N(Hx_k, R).$$

In practice, the covariance matrices of process noise *Q* and measurement noise *R* can be changed with each step or time measurement cycle, but here we assume that they are permanent. The A matrix of dimensions *n* × *n* in the differential equation involves a state in the previous time step $k-1$ with the state in the current time step *k* in case of signal absence of excitation function and noise process. The matrix *A* may vary at each time step, but here we assume that it is constant. The matrix *B* of size *n* × *1* involves an optional excitation signal of state (control) $u \in \Re^l$ with the state *x*. Subsequently the *H* matrix, with dimensions *n* × *n* in the measurement equation involves the state with the measurement of z_k. More over, the *H* matrix can vary at each time step or measurement, however for the conducted experiments it was assumed that it is constant too. In addition, the state estimate $\hat{x}_k^- \in \Re^n$ is defined as *a priori* in step *k*, taking into account knowledge about the process $\hat{x}_k \in \Re^n$ before step *k*, as the estimation of a posteriori in step *k*, taking into account the measurement z_k. Taking this into account we can now define the estimation errors a priori and *a posteriori* respectively as:

(21) $$e_k^- \equiv x_k - \hat{x}_k^-, e_k \equiv x_k - \hat{x}_k.$$

and covariance estimation error a priori and a posteriori as:

(22) $$P_k^- = E\left[e_k^- e_k^{-T}\right] P_k = E\left[e_k e_k^T\right].$$

Finally, we can rewrite a certainty a priori and a posteriori of the estimated state as:

(23) $$Bel(x_k)^- : N(\hat{x}^-, P_k^-), Bel(x_k) : N(\hat{x}, P_k).$$

And the conditional probability for the state x_k based on the measurement z_k:

(24) $$P(x_k \mid z_k) = N(\hat{x}, P_k).$$

In this paper, to take advantage of the Kalman filter, we began by finding the equation that calculates the estimation of *a posteriori* state $\hat{x}_k$ as a linear combination of estimation *a priori* state $\hat{x}_k^-$ and the weighted difference between the measurement z_k and prediction of measurement $H\hat{x}_k^-$ shown below:

(25) $$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-).$$

The $z_k - H\hat{x}_k^-$ difference is also referred to as innovation or residual. This residual reflects the discrepancy between prediction of measurement $H\hat{x}_k^-$ and the actual measurement z_k. More over, residual with a value of zero means that they are completely compatible. The matrix *K* of dimensions *n* × *m* is the coefficient of amplification, which minimizes the covariance error a posteriori. One of the possible solution for *K* we defined as:

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

The equations of the time update phase can be written as:

(27) $$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}, P_k^- = AP_{k-1}A^T + Q,$$

where $\hat{x}_k^-$ and P_k^- means the estimated value of the state and covariance *a priori*, $\hat{x}_{k-1}$ and P_{k-1} the optimum estimated *a posteriori* values obtained in the previous step.

Practical implementation of the Kalman filter applied in presented application

Designing of the model system for the filtration of facial movement can be summarized as follows: model of the system and matrix of state becomes:

(28) $$S_k = S_{k-1} + w_{k-1}, x = \begin{bmatrix} S_x \\ S_y \end{bmatrix}$$

Hence the system, the measurement, uncertainty of the system and measurement matrices will be of the form:

(29) $$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} q, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} r.$$

To ensure the good efficiency of signal tracking procedure we extended the mathematical model of the system to comply the motion velocity:

(30) $$S_k = S_{k-1} + w_{k-1} + V_{k-1}dt, V_k = V_{k-1} + w_{k-1}.$$

The state matrix we can rewrite in the form:

(31) $$x = \begin{bmatrix} S_x & S_y & F & V_x & V_y \end{bmatrix}^T$$

Regarding above we assumed the final form of the state matrix *A*

(32) $$A = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where: *dt* - denotes time slice update between consecutive cycles. Measurement matrix, because of the measurable values *Sx* and *Sy* is written as:

(33) $$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Matrices of systems uncertainty and measurement will be of the following form:

(34) $$Q = \begin{bmatrix} q_S & 0 & 0 & 0 \\ 0 & q_S & 0 & 0 \\ 0 & 0 & q_V & 0 \\ 0 & 0 & 0 & q_V \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} r$$

The module for the radius value control we can simple rewrite as:

(35) $$F_k = F_{k-1} + w_{k-1}, x = [F], A = [1], H = [1] Q = [q], R = [r]$$

In the final stage, Kalman filter used in the application take the form as follow and the state matrix becomes:

(36) $$x = \begin{bmatrix} S_x & S_y & F & V_x & V_y \end{bmatrix}^T$$

Thus eventually we can rewrite following formulas:

(37) $$A = \begin{bmatrix} 1 & 0 & 0 & dt & 0 \\ 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

$$Q = \begin{bmatrix} q_S & 0 & 0 & 0 & 0 \\ 0 & q_S & 0 & 0 & 0 \\ 0 & 0 & q_F & 0 & 0 \\ 0 & 0 & 0 & q_V & 0 \\ 0 & 0 & 0 & 0 & q_V \end{bmatrix}, R = \begin{bmatrix} r_S & 0 & 0 \\ 0 & r_S & 0 \\ 0 & 0 & r_F \end{bmatrix}$$

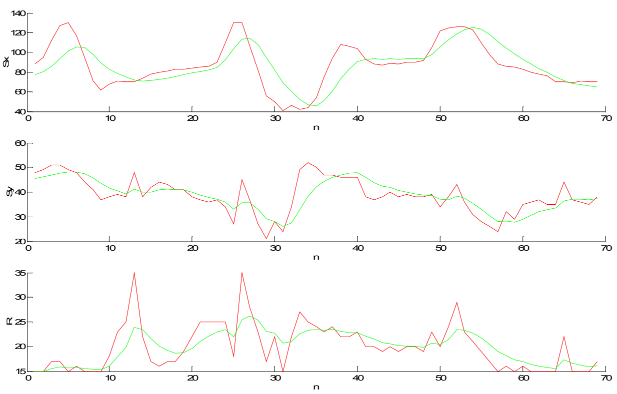

Fig. 6. Kalman filtration of facial motion in the scene for the parameters *qS*=0.1, *qF*=0.5, *qV*=1, *rS*=5, *rF*=5

We chosen empirically the following values of coefficients, also assumed as the default EyeTracker application settings q_S =0.1, q_F =0.5, q_V =1, r_S =5, r_F =5. The effect of the Kalman filter simulation for these parameters can be seen in Fig. 6. However, the EyeTracker offers the ability to edit the matrix *Q* and *R* in the course of its operation, so these parameters can be changed. Fig.7 shows examples of recorded video sequences and extracted frames with marked both real and estimated face coordinates.

Conclusions

Based on the performed tests we can formulate following conclusions. It is possible to create system enabling location of the facial image in real time and use this information to control the virtual world. The proposed model used in this study fulfills its role as a fast algorithm and also the exact location of the face tool in the video stream. Important advantage of the filter is its low computational complexity compared to alternative solutions (such as particle filters). Natural development of the conducted experiments could be elaboration of a mechanism for adaptation this method in Field Programmable Gate Array environment.

Fig. 7. Sample simulation results

REFERENCES

[1] Gomolka Z., Lewandowski T.,The Biometric Signals Processing, *Advances in soft computing, Springer Verlag*, Volume 45/2008, 637-644

[2] Kukharev G., Masicz P, Masicz Pi, Modified gradient method for face localization*, Enhanced methods in computer security, biometric and artificial intelligence systems*, Kluwer Academic Press, Springer, New York, USA, (2002), 165-176

[3] Baza obrazow testowych twarzy VIS Faculty of Computer Science and Information Technology, University of Technology Szczecin. http://face-recognition.eu/index.php?id=20

[4] Welch G., Bishop G., An introduction to the Kalman filter, *University of North Carolina at Chapel Hill*, Chapel Hill, 1995

[5] Swit G., Virtual reality control with the use of the observer in streaming video, *Engineering thesis in Polish*. PWSTE Jaroslaw 2010.

[6] Dudek-Dyduch E., Gomolka Z., Neural network design without learning, *in proceedings of the sixth international conference on Neural networks and soft computing,* New York: Physica-Verlag, (Advances in Soft Computing), (2003), 179-187

[7] Dudek-Dyduch E., Gomolka Z., Hybrid system - wavelet decomposition and hierarchical neural networks, *in Neural networks and soft computing: proceedings of the fifth conference, Polish Neural Network Society in cooperation with IEEE Neural Networks Council and Department of Computer Engineering Technical University of Czestochowa*, PNNS, (2000), 186-195

[8] Twarog B., Pekala R., Bartman J., Gomolka Z., The changes of air gap in inductive engines as vibration indicator aided by mathematical model and artificial neural network, *Dynamical Systems and Differential Equations*, Volume (2007), 1005-1012

***Authors**: dr Zbigniew Gomółka, dr inż. Bogusław Twaróg, mgr Paweł Krutys, Uniwersytet Rzeszowski, Instytut Techniki, Zakład Elektrotechniki i Informatyki, ul. Rejtana 16c, 35-959 Rzeszów, E-mail: zgomolka, btwarog, pkrut@univ.rzeszow.pl*