Journal of Engineering Science and Technology Review 7 (1) (2014) 148– 153

JOURNAL OF
**Engineering Science
and Technology Review**

www.jestr.org

Moving Target Tracking Based on Kalman Algorithm

Fei Cai* and Jian Cui

School of Civil Engineering, Shandong Jianzhu University, Jinan, 250101，China

Received 14 July 2013; Accepted 3 January 2014

Abstract

Problems concerning moving target tracking are being proposed and applied widely currently along with the maturity of computer technology and the fast advancement of computer vision. As a very important area in computer vision, moving target tracking is widely used in real life, for example, interactivity between human and computer, and so on. In this paper, Kalman filter, as a highly efficient recursive filter, is the common video method employed to track moving object in video. The moving state of moving object can be estimated in accordance with the location of the moving target. The tracking video is decomposed into frames in the initialization period and every frame is made up of a piece of picture; then, the targets in each frame are identified by means of color recognition; next, position the moving target, identify the center coordinates; at last, the coordinate of the previous and current frames is inputted and the location of the moving target in the present frame is estimated by Kalman filter.

Keywords: Computer Vision, Moving Target Tracking, Kalman filter

1. Introduction

Due to the wide applications of moving target in many fields, particularly in computer vision with the typical cases of video monitoring, image encoding, robot technology, etc., so its detection and tracking have been one of the focuses of applied vision research. [1-3] Moving target tracking based on Kalman algorithm means to predict and track moving targets through Kalman algorithm, where to recognize the moving target is the priority.

Ground surveillance aims to determining the situation of a large military ground with numerous targets under all weather and daylight conditions. And the practicable state of the art sensor technology is airborne ground moving target indicator (GMTI) radar that has the space-time adaptive processing (STAP).

In this paper, by providing high-quality tracking of military equipment, single vehicles as well as aggregations such as convoys, target tracking is used to support surveillance. Regarding the tracking of ground targets by means of airborne sensors, it often suffers from low visibility, high clutter, and high target density. Therefore, to improve the quality and continuity of tracking, previous information should be utilized as much as possible, which is highly desirable. The research on the identification of moving targets just provides us with the information and the important implications for us to understand the treatment of video images deeply.

2. Related Principles

2.1 Image identification

Identifying the moving target is one of the principal steps for moving target tracking [4-9]. Firstly, the obtained image information shall be made pre-processing (preliminary treatment) to remove noise and interference and make a correction of geometry, color, etc., which definitely enhance the noise signal. However, since weak image information is unable to be recognized, the noise signal has to be strengthened sometimes. Moreover, to understand the required things for image identification, images need to be segmented, or positioned and separated more exactly. And the images need further improvement and the restoration treatment so as to give the observer a clear image [10-11].

2.2 Kalman filtering algorithm

As a linear filtering theory and a commonly used prediction algorithm to track video motion, Kalman filtering algorithm seeks for a recursive estimation algorithm as the optimal principle of estimation based on minimum mean square error, whose basic idea lies in that: the state space model of signal and noise is used and the previous estimated value and present observed value are adopted to update the estimation for state variables, thus getting the present estimated value [12-13].

There are two special characteristics of Kalman filtering: first, concepts and terms of state space are used for the description of its mathematical model; second, its solution is calculating recursion. Generally, Kalman filtering is expressed by the system state model and a measurement model. Hence, the state space model is described as
System state model:

$$s(t) = \text{Ö}(t-1)s(t-1) + w(t) \tag{1}$$

and a measurement model

* E-mail address: feicaiswcd@126.com

ISSN: 1791-2377

$$z(t) = H(t)s(t) + v(t) \qquad (2)$$

Where, H(t) refer to state transition matrix and measurement matrix, respectively, and w(t) and v(t) are zero-related white Gaussian noise.

The prediction of the state vector s(t) of present time t origins from the average expected value of previous market and new measurement z(t).

The tasks of Kalman filtering are performed in two stages: prediction and calibration. The future situation is predicted in the prediction step and a priori estimation of states (t) is obtained. Feedback is intended to be received in calibration. The actual measurement and priori estimation are combined to get a modified posteriori estimation s^+ (t)(t) is, expressed as

$$s^+(t) = s^-(t) + K(t)(z(t) - H(t)s^-(t)) \qquad (3)$$

Where, K(t) is the weight and expressed as

$$K(t) = P(t)^- H(t)^T (H(t)P^-(t)H(t)^T + R(t))^{-1}$$
$$= \frac{P(t)^- H(t)^T}{H(t)P^-(t)H(t)^T + R(t)} \qquad (4)$$

where, P$^-$ 1(t) is covariance of priori estimate error, defined as

$$P^-(t) = E\left\lfloor e^-(t)e^-(t)^T \right\rfloor \qquad (5)$$

Here, $e^-(t) = s(t) - s^-(t)$ is the priori estimate error.

In addition, covariance of posteriori estimate error P$^+$ (t) is

$$P^+(t) = E\left\lfloor e^+(t)e^+(t)^T \right\rfloor \qquad (6)$$

Here, $e^+(t) = s(t) - s^+(t)$ is the posteriori estimate error. The recursion performance of prediction and calibration can be expressed as:

◆ Prediction

$$s^-(t) = \text{Ö}(t-1)s^+(t-1) \qquad (7)$$

$$P^-(t) = \text{Ö}(t-1)P^+(t-1)\text{Ö}(t-1)^T + Q(t-l) \qquad (8)$$

◆ Calibration

$$K(t) = P^-(t)H(t)^T (H(t)P^-(t)H(t)^T + R(t))^{-1} \qquad (9)$$

$$s^+(t) = s^-(t)K(t)(z(t) - H(t)s^-(t)) \qquad (10)$$

$$P^+(t) = (I - K(t)H(t))P^-(t) \qquad (11)$$

Cycles of prediction and calibration are repetitive. As can be seen in formula (9), measurement error R (t) is inversely proportional to K(t) obtained by Kalman. Less R(t) is a surplus measured heavily by the obtained K (t). In this case, the measurement becomes increasingly reliable, while the predicted result is opposite. On the other hand, when priori estimate error P$^-$ (t) is close to zero, the obtained K (t) will measure the surplus lightly and the actual measurement and the predicted result will change to the opposite.

So, if a more reliable one can be determined, then a close optimal result will be gained by the system. Known from formula (7) to (11), Kalman filtering has repetitive prediction and calibration cycles. To improve the recommended adaptive Kalman filter, firstly, moving model should construct the system state model used as Kalman filtering. The moving model will be appropriate to predict Kalman filtering. Then, regarding the calibration of Kalman filtering, a surveillance method of moving target is recommended.

In Kalman filtering algorithm, system state model is ideal for prediction. The system state model shall be determined as a moving model before using Kalman filtering algorithm. Due to the internal short time of frame, the interval for each frame of this moving object is supposed to be uniform. So, the speed of moving object can be substituted for position like a system parameter. In addition, the moving distance of each interval is uniform and can be set as:

$$d(t) = d(t-1) + (d(t-1) - d(t-2)) \qquad (12)$$

where, d(t-1) and d(t-2) are distances of moving objects between frame t-1 and t-2, respectively. In formula (1) and (13), the system state model of Kalman filtering can be expressed as:

$$s(t) = \text{Ö}(t-1)s(t-1) + w(t) = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} d(t-1) \\ d(t-2) \end{bmatrix} + \begin{bmatrix} w(t) \\ 0 \end{bmatrix} \qquad (13)$$

Here,

$$s(t) = \begin{bmatrix} d(t) \\ d(t-1) \end{bmatrix} s(t-1) = \begin{bmatrix} d(t-1) \\ d(t-2) \end{bmatrix}$$

and

$$\text{Ö}(t) = \begin{bmatrix} 2 & \text{-1} \\ 1 & 0 \end{bmatrix}.$$

At this stage, the position of moving object in the consecutive frame will be perceived. Once the dominant color in RGB color space which is used to increase the changing tolerance of light in the video is obtained, the color feature will be converted to HSI color space, even if the three constituent parts are relatively independent; then a color matching function is used to measure the space interval of moving objects. In HSI color space, the change of light is a main component affecting the strength.

Moreover, in order to indicate the position of the object, the center of matching object area shall be found. A smallest rectangle that can cover the whole moving target may be used to locate the moving target. Therefore, the center is $(cx,cy)=(\frac{left+right}{2},\frac{top+bottom}{2})$, where, $(left,top)$ and $(right,bottom)$ are the positions on the upper left corner and lower right corner of the rectangle, respectively. This rectangular area will expand some pixels as the search scope of the next frame. Therefore, the positions of the upper left corner and lower right corner in the search scope $(search\_left,search\_top)$ and $(search\_right,search\_bottom)$ are defined as:

$$search\_left=(pre\_left-(d(t-1)-d(t-2))),$$

$$search\_top=(pre\_top-(d(t-1)-d(t-2))),$$

$$search\_right=(pre\_right-(d(t-1)-d(t-2))),$$

$$search\_bttom=(pre\_bottom-(d(t-1)-d(t-2))) \quad (14)$$

Wherein, the first letters of pre_ indicate the previous frame. d(t-1) and d(t-2) are distances of moving objects between frame t-1 and t-2, respectively. In other words, the moving speed of an object shall be taken into consideration in the scope of the search. The center location of the search scope is based on the position predicted by the adaptive Kalman filtering algorithm.

3. Algorithm Introduction

3.1 System structure

The system is divided into the following modules, that is, video opening (including image opening), video frame capturing, frame processing, target recognition and location, forecasting and tracking moving targets and video reconstruction.

3.2 Video frame capturing

As the first step of video processing, video frame capturing disintegrates the video into frames, each with a picture. At first, we obtain the handle of video. Since c++ the video file name cannot be processed directly by c++, it should be transformed to handle what can be processed by c++. After opening the video file, the method provided in Open CV is used successively to take out one frame from the handle. Each frame taken out is named with the serial number of the frame as part of the name, and saved in the appointed place. In this way, every frame in the video is divided into the file in picture format.

3.3 Frame image processing

After video disintegration, operation is conducted for each frame to recognize moving objects. Starting from the first frame, the image of each frame is expressed as a binarybinary image. The method is to create an image expressed by gray level first through utilizing the cvCreateImage in OpenCV directly, transforming the original color frame image expressed by RGB into an image expressed by gray level; then, the same method is sued to create a binary image, and all pixel values of this image should be expressed as 8-bit character. The image expressed by gray level is converted into a binarybinary image. The pixel value at pixel point with the gray level pixel value less than 150 is set as eight-bit 0 and black, and that above 150 is set as eight-bit 1 and white. In this way, the black is the moving target, and the white is the backgroundbackground, therefore, we can distinguish the objects to be tracked quantitatively.

3.4 Positioning of moving target

Positioning of moving target is to determineis to find out the center coordinates and select the moving object by a rectangular box which is usedrectangular to track the moving object. Here, each pixel in frame image is traversed and the maximum and minimum x, y coordinates of the pixel with pixel value less than 150 are recorded. The mean value of maximum and minimum x coordinate value is the x coordinate of moving target center, similarly, that of maximum and minimum y coordinate value is the y coordinate of moving target center. In this way, the center coordinates of moving target are obtained. Rectangle can be therefore drawn in a frameframe image by the ImageDraw method in OpenCV.

3.5 Kalman filtering prediction algorithm

In section 2.2, we have already introduced the basic principle of Kalman filtering algorithm. is how to use Kalman algorithm is emphasized here. In our system, the following simplified Kalman filtering formulas are used:

Assuming the initial data is μ_0, the estimated value in t moment is $\hat{\mu}_t=\alpha_t\mu_{t-1}$, the key is to solve α_t. α_t is the weighted value changing with time and is deduced according to the Kalman filtering optimal estimation.

$$P_0=1$$

The original state of the algorithm is $\alpha_0=1$. For t in each moment, the Kalman filtering real-time update α_t calculation method of this algorithm is,

$$\begin{cases} K(t)=\dfrac{P_{t-1}\mu_{t-1}}{1+P_{t-1}\mu^2{}_{t-1}} \\ e(t)=\mu_t-\hat{\alpha}_{t-1}\mu_{t-1} \\ \hat{\alpha}_t=\hat{\alpha}_{t-1}+K(t)e(t) \\ P_t=P_{t-1}-K(t)\mu_t P_{t-1} \end{cases}$$

The estimated center coordinates -x and y (the values used in the first frame are exact values at present) in the previous frame as well as the x and y coordinates of the present frame are taken as the initial values, and are substituted into the formula. The obtained value is the estimated coordinate of the next frame. Thus, the estimated value of moving target coordinate for each frame can be obtained through the recursion like this, and such value is the motion state of moving target predicted by Kalman filtering algorithm.

3.6 Tracking results

The target location of each frame image has been achieved previously. The prediction of the motion path of target through Kalman algorithm has been accomplished. In the following, all frame images obtained by prediction will be reconstructed and the tracking for the moving target will be finally is achieved through creating a new video file and inserting each frame into the newly created video. So, the video file finally gone through the reconstruction is the tracking result of moving target.

4. System Realization

4.1 Development environment

The development environment of Visual C++6.0 and class library of Open CV is used by the system. As, ++a windows application program essentially, Visual C++ provides the development environment integrated with source program editing, compiling and debugging, which is called the integrated development environment. And the familiarity efficiency of programming can be directly impacted by the familiarity for that program.

Open CV is Intel&reg, an open source computer vision library, consisting of a series of C functions and a few C + + classes that have realized many general algorithms in image processing and computer vision. The system has used some C + + classes of Open CV in the process of image processing.

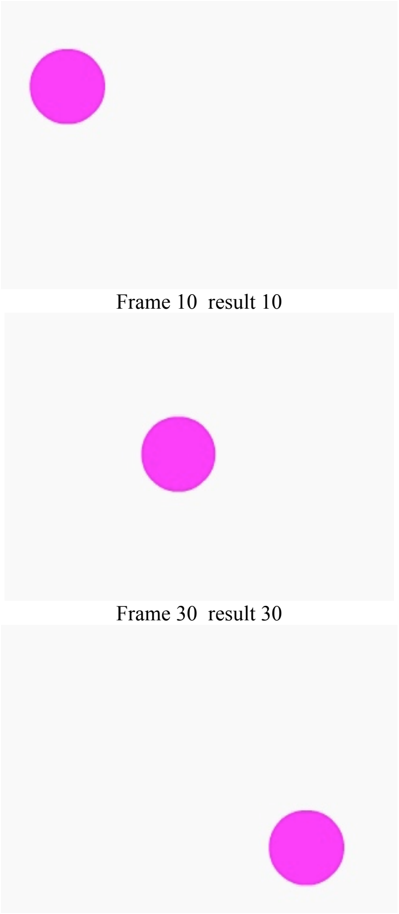
4.2 Brief introduction to the function

The realization of this system is totally divided into 10 classes of C About Dlg, C Main Frame, C video Analyze App, C video Analyze Doc, C video Analyze Module, C video Analyze View, Kalman, String Process, Target and Video Process respectively; C About Dlg, C Main Frame, C video Analyze App, C video Analyze Doc and C video Analyze Module among them have realized the framework of MFC, the main function of Struct Kalman is to record the prediction state during the Kalman prediction of the system, and Struct Target is used to record the position of moving objects.

4.3 Video frame capturing

Video frame capturing is the first step of motion tracking, and it is also the preprocessing step of the whole tracking. The moving objects and track it can be further identified through decomposing the video into frames of picture:

(1). Open the video file and obtain the handle of video file named video File Name, because the handle of the video file must be obtained from the processing for video in Open CV, so the handle of a video file will be returned after processing the received file name.
(2). Frames shall sequentially taken out from the video with the handle of p Capture, the read-only form shall be returned, and the frame taken out shall be obtained, which not only takes out the frames in the video files, but records the position of frame taking at this time, and saves into the incoming video handle p Capture, the video frame from the position of next frame when calling this function should be taken out next time.
(3). To facilitate unit testing, the taken out frame should be stored in the form of image file on hard disk, which is achieved by setting up Image Save function. The accepted parameters are Ipl Image * p Frame and char * image File Name, the frame to be saved is stored in the p Frame among them, and the storage position is stored in the image File Name.
(4). The results of video decomposition are obtained through decomposing each frame of the video, and saving each frame picture in jpg format, a few key frames are intercepted here with the results are shown in Fig. 1:



Frame 10 result 10

Frame 30 result 30

Frame 60 result 60

Fig. 1 Video decomposition

4.4 Frame processing

The main contents of frame processing are described in detail in the last chapter, which focuses on the transforming the frame into a binary image, so as to recognize the moving targets through the method of color recognition(as shown in Fig. 2), aiming at preparing for the location of the next step targets:

(1). Use the m Method provided in Open CV is used to create an image expressed by gray level.
Temp=cvCreateImage(cvSize(source->width,source->height),IPL_DEPTH_8U,1);
(2). The frame picture source expressed with RGB in video is transformed into picture of temp Cv Color (source, temp, CV_BGR2GR AY) expressed by RAY gray level;
(3). A binary image is created with each pixel expressing by an eight-bit byte
Destination=cvCreateImage(cvSize(source->width,source->height),IPL_DEPTH_8U,1);
(4). The picture temp expressed by gray level is transformed to a binary image destination. Since red and blue pellets are the moving objects used as the cases we tested, their pixel values are less than 150 after being transformed into the gray level pictures. Therefore, 150 is set as a threshold value, the pixel value of pixel point is represented with the gray level pixel value less than 150 to be eight-bit 0 and black, and the gray level pixel value of pixel point is represented with the gray level pixel value of 255 to be eight-bit 1 and white, so

the black is taken as a moving target while the white is the background so as to recognize the moving targets.

cvThreshold(temp,destination,150,255,CV_THRESH_BINARY);

(5). The result of frame processing is each frame image is represented to be an eight-bit binary image with the white background and the black moving targets, therefore, the moving targets in picture are recognized quantitatively.

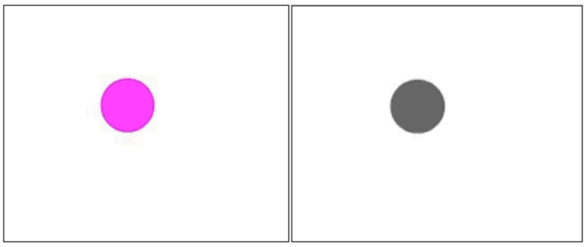


Fig. 2 Frame processing

4.5 Positioning of moving target

Positioning of moving target actually aims to get a coordinate as the indicator of moving Kalman algorithm, so as to conduct moving target tracking. The center coordinate of the pellet is adopted as the indicator of this tracking here:

(1). Firstly, we define a Target containing 4 data items, namely, x, y, semi_height and semi_width, respectively. The combination of x and y among them can indicate the center of object, along with the data of semi_height and semi_width, a rectangle to encircle the objects to be tracked can be calculated.

(2). The geometric center of object in binarization image shall be calculated as tracking indicator. When calculating the central position of moving target, the x coordinate of center coordinate of the pellet is the intermediate value of the maximum and minimum x coordinates, and the y coordinate of center coordinate is the intermediate value of the maximum and minimum y coordinate. Therefore, the location of moving object can be obtained.

(3). After the position of object is calculated, Image Draw function is introduced to test whether the calculation is correct. The two parameters received by the function are Ipl Image* image and Target* target, where image stores the image, and target represents the position to be indicated by painting.

(4). The purpose of target location is to find out the correct point to be tracked by the indicator, the center coordinate of the pellet needs to be found out here, then a rectangular box is used to circle the whole pellet and the rectangle whose position is the predicted moving position of the pellet is employed to track the moving targets.

4.6 Realization of motion tracking based on Kalman algorithm

Kalman filtering prediction algorithm serves as the core of the entire motion tracking system, playing a role in calculating the estimated value of the direction of the moving target, which is equivalent to the predicted value of motion tracking. With this predicted value above, the predicted value of the previous frame and observed value of the current value are used to calculate the predicted value of each frame, and then each frame after prediction is subject to video reconstruction for a full video again, which is the final result of motion tracking. Motion tracking is an integrated process, including acquisition of video frame, processing of video frame, drawing of frame, video frame reconstruction and many other technologies.

1. A simplified formula of Kalman used in this article in section 3.5 has been introduced, by which c++ language can be used to realize the code of Kalman algorithm.

2. A Video Writer for the purpose of writing frame in target video file can be created, and forms of member variables of Video Process class are saved by this Video Writer and initialized in constructed function

3. After the completion of a series of processes including analysis of video frame, prediction of object position, drawing of frame and so on, all of these processed frames shall be recombined as a new video file. During creating this video file, Video Writer is introduced by member variables and parameters are unnecessarily to be accepted. The required parameter is the frame to be written in.

4. A process needs to be established after the completion of all these preparations, and functions in Kalman can be realized through a combination of these methods. What this function requires is to accept video filename to be tracked. The result of the function after execution is to create a new video file where Kalman filtering prediction algorithm is applied for the realization of predicting the position of object and a rectangle is used to mark the predicted position.

5. Video moving target tracking is realized by the way of formula of Kalman filtering algorithm, and the achieved results are shown in Fig.3:

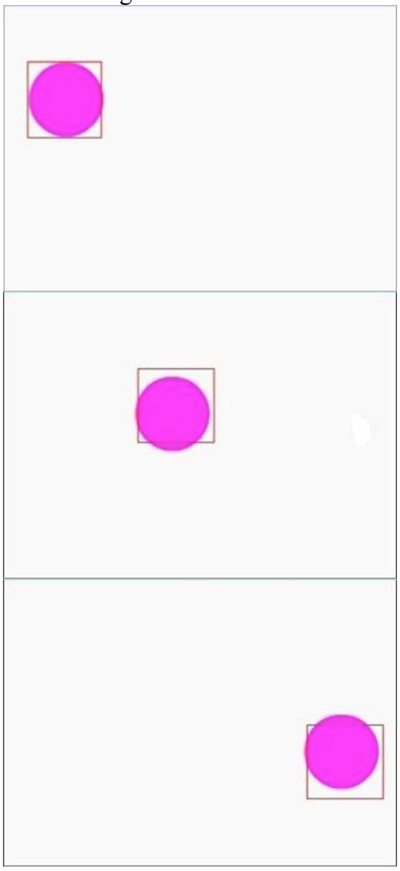


Fig. 3 Video moving target tracking

As seen from the resultsresults above, Kalman algorithm has better tracking and prediction functions in general. However, it is noteworthy that prediction is perfectly consistent with the motion of object when an movingobject is moving in linear motion route, indicating Kalman is quite accurate for uniform motion when the moving object is in linear motion route.

However, if a sudden change in the position of the object occurs, there will be a largerlarger discrepancy between the predictedpredicted result and actual motion, showing that it is difficult for the used Kalmanalgorithmuse to cope with this sudden change in a situationsituation.

5. Summary and Prospect

Thus so far, our system has realized video disintegration, target recognition and location, prediction of target motion, video reconstruction and other functions, basically achieving the tracking effect of moving target. In this article, a series of methods provided by Open CV is relied on to apply some principles of target recognition. Above all, Kalman filtering algorithm is used for the prediction and tracking of moving target Kalman as the core of the entire target tracking system, and target tracking is finally realized by using Kalman algorithm to predict the motion of moving object.

However, there are still many deficiencies still exist in our system in all aspects. Firstly, in terms of target recognition, colors are used to recognize for the reason that the background of the used example has easily distinguishable color from the color of moving object. But in common video files, it is sometimes difficult to distinguish the background color from color of moving target. What's more, color changes are resulted from the changes in lighting conditions, objects are thus difficult to be identified by way of color recognition; secondly, in the aspect of recognition and tracking of moving target, as our background is unchanged and we have known the information of moving object in advance, the color difference is used to recognize moving target. However, in actual situation, information of moving object is generally unknown to us, so moving object is difficult to be found out through information about colors and shapes; furthermore, the most simple and typical Kalman filtering algorithm is adopted here to track moving object, which is inadequate in accuracy and efficiency[14-16]. have a lot of work still needs to be done additionally in terms of expandability and adaptability of the system.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this article.

Acknowledgements

This work was supported by MOHURD program of China (No.2011-K9-21), Soft Science Research Project of Shandong Province（No.2013RKB 01284）and Innovation Team Development Plan of Education Ministry (No.IRT13075).

References

1. Z. Kalal, K. Mikolajczyk, J. Matas., *Tracking-Learning-Detection, PAMI*, 34, 2012, pp. 1409–1422.
2. Yilmaz, O. Javed, M. Shah. ''Object Tracking: A Survey'',. *ACM Comput. Surv*., 38,2006, pp.1–45.
3. Y. Rui, T. S. Huang, and S. F. Chang. ''Image Retrieval: Current Techniques'', *Promising Directions and Open Issues. J. Vis. Commun. Image Represent*., 10, 1999 , pp.39–62.
4. R. Datta, D. Joshi, J. Li, J. Z. Wang. ''Image Retrieval: Ideas, Influences and Trends of the New Age'', *ACM Computing Surveys,* 40 , 2008, pp.1–60.
5. J. Shi, C. Tomasi.: Good features to track. Conference on Computer Vision and Pattern Recognition, 1994.
6. M. Brand, V. Kettnaker. ''Discovery and Segmentation of Activities in Video'', *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22, Aug. 2000, pp.844-851.
7. R.T. Collins, A.J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, L. Wixson. A System for Video Surveillance and Monitoring, Technical Report CMU-RI-TR-00-12, Robotics Inst., Carnegie Mellon Univ., May 2000.
8. W.M. Hu, T.N. Tan, L. Wang, S.J. Maybank. ''A Survey on Visual Surveillance of Object Motion and Behaviors'', *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34, 2004, pp.334-352.
9. Y.A. Ivanov, A.F. Bobick. ''Recognition of Visual Activities and Interactions by Stochastic Parsing'', *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22, Aug. 2000, pp. 852-872.
10. J. Hu, H. Zhang, J. Feng, H. Huang, H. Ma. ''A scale Adaptive Kalman Filter Method Based on Quaternion Correlation in Object Tracking'', *2012 Third Int. Conf. Netw. Distrib. Comput*., Oct. 2012, pp.170–174.
11. R. Olfati-Saber, P. Jalalkamali. ''Collaborative Target Tracking Using Distributed Kalman Filtering on Mobile Sensor Networks'', *2011 American Control Conference on O'Farrell Street*, San Francisco, CA, USA, 2011.
12. M. S. Grewal, A. P. Andrews. Kalman Filtering: Theory and Practice Using MATLAB, John Wiley and Sons, Inc., 2001.
13. G. Welch and G. Bishop An Introduction to the Kalman Filter. Available from http://www.cs.unc.edu/welch/kalman/kalmanIntro.html
14. Z. Kalal, K. Mikolajczyk, J. Matas. ''Tracking-Learning-Detection'', *IEEE Trans. Pattern Anal. Mach. Intell*., 6, Dec. 2011, pp. 1–14.
15. E. Emami, M. Fathy. ''Object Tracking Using Improved CAMShift Algorithm Combined with Motion Segmentation'', *2011 7th Iran. Conf. Mach. Vis. Image Process*., Nov. 2011, pp.1–4.
16. H. Dong, S. Chen, J. Zhu, ''Research of Moving Targets Tracking Algorithm Based on Kalman Filtering'', *2010 Third Int. Conf. Intell. Networks Intell. Syst*., Nov. 2010, pp. 20–23.