15

Object Visual Tracking using Window-Matching Techniques and Kalman Filtering

Flávio B. Vidal and Victor H. Casanova Alcalde
Electrical Engineering Department, University of Brasilia
Brazil

1. Introduction

Object visual tracking aims to determine the image configuration of a target region of an object as it moves through a camera's field of view. The visual tracking process consists on matching the target region in successive frames of a sequence of images taken at closely-spaced intervals. Visual tracking has become an important process on various applications as: vision-based control (Hutchinson et al., 1996; Papanikolopoulos et al., 1992), industrial robotics (Sumi et al., 2007), biomedicine (Shen et al., 2006), surveillance (Urtasun et al., 2006), aerial target tracking (Yau et al., 2001), aircraft and car traffic monitoring and control (Rostamianfar et al., 2006).

Algorithms that combine digital image processing and visual servo control techniques are being applied to the solution of complex problems such as object tracking from a sequence of images (Hager et al., 1998). Visual tracking can be considered an estimation process acting together with digital image processing techniques. For the estimation process a stochastic filtering approach using Kalman filter can be applied (Veeraraghavan et al., 2006) and the particle filter (Shen et al., 2006). A visual tracking algorithm in (Babu et al., 2007) combines mean-shift tracker with a modified window-matching algorithm in order to avoid drift during partial object occlusion. Other algorithm (Brassnet et al., 2007) uses particle filtering for object tracking based on multiple cues with adaptive parameters and its performance is investigated and evaluated with synthetic and natural sequences and compared with the mean-shift tracker. These estimation approaches can be applied to visual servo control in association with window-matching techniques yielding better results (Tan et al., 2005).

Here, an object tracking algorithm is proposed that combines the window-matching techniques and optimal estimation theory based on the linear stochastic Kalman filtering (Kalman, 1960). The window-matching algorithm is modified and a Kalman filtering stage is coupled to improve the tracking performance. The main objective of this work was to develop the structure of a tracking algorithm not yet its final and efficient implementation, so it was developed within the Matlab computational environment.

The chapter is organized as follows, in Section 2 the object visual tracking problem is stated together with the solving methods. Section 3 discusses the window-matching techniques and presents a window-matching algorithm (WM) for tracking purposes. Section 4 deals with the use of the Kalman filtering (K) to improve the object visual tracking, a new algorithm (WM+K) is then presented. Section 5 then shows the application of the WM+K

Source: Kalman Filter, Book edited by: Vedran Kordić,
ISBN 978-953-307-094-0, pp. 390, May 2010, INTECH, Croatia, downloaded from SCIYO.COM

algorithm to the following tracking situations: a) ball on a warming up table tennis game; b) vehicle in urban traffic scenery; c) somebody on a two-people meeting and walking scene; and d) a bottle floating on the sea.

2. Object visual tracking

Visual tracking is much related to the correspondence subproblem in vision-based motion analysis. The correspondence problem deals with determining which elements of a frame correspond to which elements of the next frame of the sequence, then, it can be applied for tracking purposes by determining the movement of an entire target region over a long sequence of images. Due to the small spatial and temporal differences between consecutive frames, the correspondence problem can also be stated as the problem of estimating the apparent motion of the image brightness pattern, the so called *optical flow*. The solution of the correspondence problem can roughly follow two strategies (Trucco & Verri, 1998): differential methods and window-matching methods. Differential techniques are based on the spatial and temporal variations of the whole image brightness, generating then the optical flow. Methodologies for motion detection based on differential techniques can be modified to perform object tracking in a sequence of images (Vidal & Casanova, 2005). However, these techniques demand numerical calculation of derivatives that could be impracticable in circumstances where there is a high level of noise, reduced number of frames or the effect of aliasing in the image acquisition process.

Window-matching techniques (Anandan, 1989) are based on the assessment of the degree of similarity among regions in sequential images, so that an object may be recognized and its position inferred in subsequent frames. Window-matching techniques can be applied to object tracking and to other issues in computing vision.

3. Visual tracking based on window-matching techniques

Window-matching methods are based upon an analysis of the grey level pattern around a point of interest and the search for the most similar pattern in the subsequent frame. They are also called *region similarity methods*. Having defined a window $W(x, y)$ around the point $\mathbf{p}(x, y)$, similar windows $W'(x+i, y+j)$ displaced an integer number of pixels are considered. The estimated image displacement corresponds to the minimal of a distance function between the intensity patterns of the two considered windows, which is then obtained by minimizing the function

$$f(W, W'(i, j)) \tag{1}$$

In case correlation functions between distances are used, the problem would be to maximize the cost function. The window-matching techniques assume that: a) the grey level intensity pattern is constant between two successive images; b) there is not a high degree of ambiguity between the texture of the region of interest and other regions of the image.

3.1 Window-Matching (WM) methods for motion detection

According to (Barron et al., 1994), there are several ways of evaluating similarities among grey level intensity patterns in sequential images. The nature and rigidity of the performed motion directly affect the success (or failure) of the method implementation. The choice of the region of interest (ROI) must be a careful task in order to faithfully reproduce the image actual characteristics.

Problems concerned to bidimensional approximations on image tracking usually happen when the ROI is subjected to complex form and illumination changes. One solution for this problem may be using methods for updating the interest region from the preceding image position in order to minimize geometry and lighting changes (Hager et al., 1998). However, this procedure brings up an undesirable effect, known in the literature as *feature drift*. That happens due to the fact that the ROI new position has a small aggregated error, which continuously builds up with the image motion, so compromising the tracking action.

The regions on the image are represented by squared windows of $N \times N$ dimension. The idea is to calculate motion between a region center around a certain point of interest $\mathbf{p}(x, y)$ on image I_1 that will be displaced by integer values i, j (along the horizontal and vertical directions, respectively) in the subsequent image I_2.

To measure similarity the well known SSD (Sum of Squared Differences) cost function will be used here, which is defined as

$$\sum_{i,j=-N/2}^{N/2} [I_1(x, y) - I_2(x+i, y+j)]^2 \tag{2}$$

Minimizing equation 2 represents minimizing the distance of similarity, and then it means finding, on the subsequent image, the most similar region to the current image. On tracking several objects with independent motion, occlusion can happen. Consequently, some objects may partial or totally disappear in some images; this will cause errors in the object trajectory. To deal with this problem local trajectory restrictions can be imposed and/or uncompleted trajectories can be allowed, since this last approach is properly considered.

Solving equation 2 usually encounters the problem of window-matching between regions with little texture information. Anandan (1989) proposed a methodology to evaluate the reliability of the similarity results obtained with the SSD function. In this work small windows of 5×5 pixels are used for window matching purposes with the candidate points for minimizing the SSD function. The method validation is based on the fact that, on establishing a window matching along a scanning direction, if there is a slight difference in the distance of similarity between windows, it will not be possible to determine the matched window. On the other hand, if there are acceptable similarity distance variations along the scanned direction, it can be concluded that there is a matched window.

3.2 A WM tracking algorithm based upon similarity distance measurement

An algorithm applying the window-matching method by using a similarity distance measurement was developed. The stages of such algorithm are shown in Figure 1. The algorithm is a modified version of the one introduced in (Barron et al., 1994). One of the modifications was the insertion of an additional stage to check the result reliability.

To measure the region similarity the algorithm builds an image distance matrix from which the minimum (or maximum, when the correlation functions were used) values can be obtained. A pre-processing stage was not implemented in the proposed algorithm, as suggested in Anandan (1989), because it is desirable to keep the visual information related to the object boundaries. The adopted methodology demands a substantial computational effort. In order to minimize this effort some procedures are reported by Giachetti (2000).

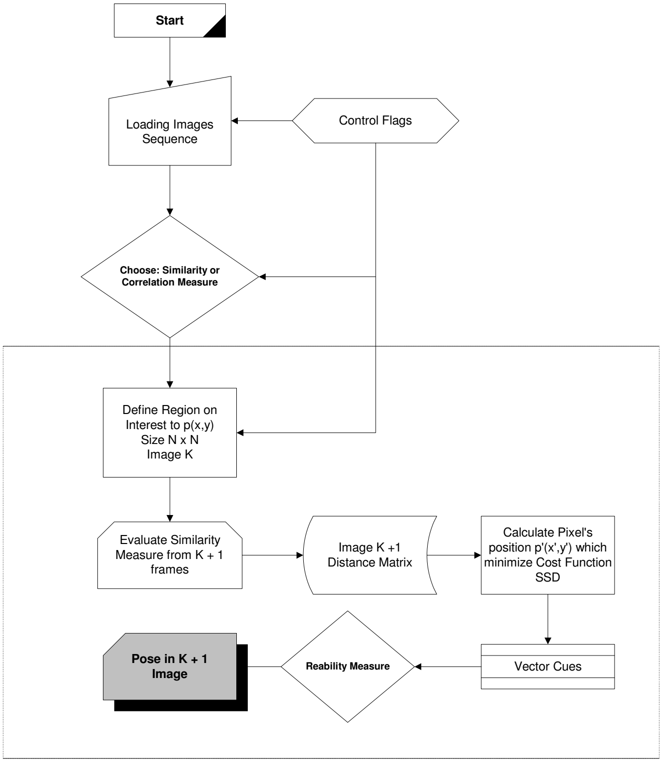


Fig. 1. Window-matching algorithm (WM) for tracking purposes

4. A Kalman filter stage into a WM tracking algorithm

Kalman filtering is a recursive procedure for optimal estimation of the state of a dynamic system, on the basis of noisy measurements and an uncertain model of the system dynamics. For object tracking purposes Kalman filtering can be used to estimate: a) the position of a moving feature point in the next frame, i.e. where to look for the feature; and b) the uncertainty of the estimation, i.e. the degree of confidence of finding the feature in the next frame in a region around the predicted point.

On tracking objects from frame to frame in long sequences of images, there is a fact; the motion of the observed scene is usually continuous, being then possible to make prediction on the motion of the image points, at any instant, based on their previous trajectories. Then object visual tracking can be approached as a problem of state estimation of a dynamic system motion, being the state vector $\mathbf{x} = \begin{bmatrix} x & y & u & v \end{bmatrix}^T$, consisting of the *2D* position $\mathbf{p} = \begin{bmatrix} x & y \end{bmatrix}^T$ and its respective velocity vector $\dot{\mathbf{p}} = \begin{bmatrix} u & v \end{bmatrix}^T$. As a new frame of the image sequence is acquired and processed at each instant $t_k = t_0 + k.\Delta t$, with $k = 0,1,2,\ldots$ and Δt a certain sampling time between frames. Assuming a short sampling time, the state vector does not change much, thus the system model describing the motion dynamics is a time-discrete dynamic equation as

$$\begin{aligned} x_k &= x_{k-1} + u_{k-1}.\Delta t \\ y_k &= y_{k-1} + v_{k-1}.\Delta t \\ u_k &= u_{k-1} \\ v_k &= v_{k-1} \end{aligned} \tag{3}$$

For simplicity (not considering real time) $\Delta t = 1$, thus a state equation that now will include a noise vector $\mathbf{w}_{k-1}$ to represent the system noise is

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}.\mathbf{x}_{k-1} + \mathbf{w}_{k-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{4}$$

The position vector as determined by the window-matching (WM) procedure will be the measurement vector $\mathbf{z}_k$; it will include a noise vector $\mathbf{v}_k$ to represent the measurement uncertainty. The measurement model will then be

$$\mathbf{z}_k = \mathbf{H}_k.\mathbf{x}_k + \mathbf{v}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} .\mathbf{x}_k + \mathbf{v}_k \tag{5}$$

The noise signals in the noise vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ are considered having Gaussian distribution and zero mean. The corresponding system covariance matrix $\mathbf{Q}_{k-1}$ and the measurement covariance matrix $\mathbf{R}_{k-1}$ are also inputs to the Kalman filter at time t_{k-1}. In the proposed algorithm the window-matching procedure will supply the Kalman filter with "noisy" position observations $\mathbf{z}_k$ from which optimal position and velocity estimates $\hat{\mathbf{x}}_k$ at time t_k will then be obtained.

To initialize the Kalman estimation, arbitrary high values for the process covariance matrix $\mathbf{P}_0$ must be assigned because the filter dynamics takes into account the confidence level of the estimates according $\mathbf{P}_0$. In many cases, undesirable estimates are obtained as a result of bad numerical conditioning of $\mathbf{P}_0$ causing therefore a filter biasing. The implemented filter dynamic equations are

$$\begin{aligned} \mathbf{P'}_k &= \mathbf{\Phi}_k.\mathbf{P}_{k-1}.\mathbf{\Phi}_k^T + \mathbf{Q}_{k-1} \\ \mathbf{K}_k &= \mathbf{P'}_k.\mathbf{H}_k^T.\left(\mathbf{H}_k.\mathbf{P'}_k.\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} \\ \hat{\mathbf{x}}_k &= \mathbf{\Phi}_{k-1}.\hat{\mathbf{x}}_{k-1} + \mathbf{K}_k.\left(\mathbf{z}_k - \mathbf{H}_k.\mathbf{\Phi}_{k-1}.\hat{\mathbf{x}}_{k-1}\right) \\ \mathbf{P}_k &= \left(\mathbf{I} - \mathbf{K}_k.\mathbf{H}_k\right)\mathbf{P'}_k\left(\mathbf{I} - \mathbf{K}_k.\mathbf{H}_k\right)^T + \mathbf{K}_k.\mathbf{R}_k.\mathbf{K}_k{}^T \\ k &= 1,2,3,\ldots,n \end{aligned} \tag{6}$$

With $\mathbf{I}$ as the 4×4 identity matrix, and $\mathbf{K}_k$ being the Kalman filter gain. The optimal estimation given by the filter output is vector $\hat{\mathbf{x}}_k$ at time t_k, representing the image position and velocity being its uncertainties described by the diagonal elements of the $\mathbf{P}_k$ matrix.

4.1 A tracking algorithm based on Window-Matching and Kalman filtering (WM+K)

The window-matching (WM) or similarity algorithm shown in Fig. 1 was then modified by inserting a Kalman filter stage. While the window-matching algorithm is running, the Kalman filter processes the resulting measurements generating then outputs to indicate the error tolerance during the WM algorithm execution. In case the similarity algorithm returns values that do not match the conditions previously established, these WM results are dropped and the Kalman filter position estimates are taken as solutions. In this way misleading results, specially those ones produced by feature drift are corrected. Figure 2 shows the developed algorithm (WM+K).

Complementary strategies could be introduced in order to improve this window matching with Kalman filtering. These strategies demand more complex techniques, most of which depend on the use of non-linear models to represent more accurately the attempted tracking. Here linearized models were used to implement the proposed algorithm.

5. Applications of the WM+K tracking algorithm

As mentioned before the objective of this work was to develop a window-matching algorithm with stochastic filtering and verify its performance. For this developing stage the Matlab computational environment was chosen. However, its more critical routines in terms of processing speed were written in *C* language and then converted to *mex* functions (*mexfiles*), a feature in the Matlab environment. *Mexfiles* can be called from within Matlab, so improving the processing speed. The final version of the algorithm will be fully written in C language, provisions for real-time operation will be as well pursued.

The developed tracking algorithms were then applied to these situations with different degree of complexity:

a. Tracking the ball on a warming up table tennis game;
b. Tracking a vehicle in urban traffic scenery;
c. Tracking people meeting and walking in public buildings (two cases); and
d. Tracking a bottle floating on the sea.

First, the WM tracking algorithm will be applied, then the WM+K algorithm, showing also, in this latter case, the measurement inputs from the WM stage.

5.1 A table tennis sequence[1]

Figure 3 shows the initial, two intermediates and final frames of an image sequence displaying a man practicing for a table tennis game. The problem is tracking the ball during this warming up period of the player. In this sequence the image size is *352×240* pixels with the images in PNG format. For this game sequence the sub-region and searching window sizes were *40×40* and *10×10* pixels respectively.

First, the WM algorithm was applied to the ball tracking problem, then for comparison, the combined WM+K algorithm was applied to the same table tennis sequence. Figure 4 shows, for the same four frames in Fig. 3, the ball tracking results obtained from the WM algorithm. The larger square (blue) is the subregion window and the smaller one (green), the searching window. It can be observed that the WM algorithm is capable of tracking the ball.

[1] Downloaded from http://www.cs.cmu.edu/afs/cs/project/cil/ftp/html/vision.html

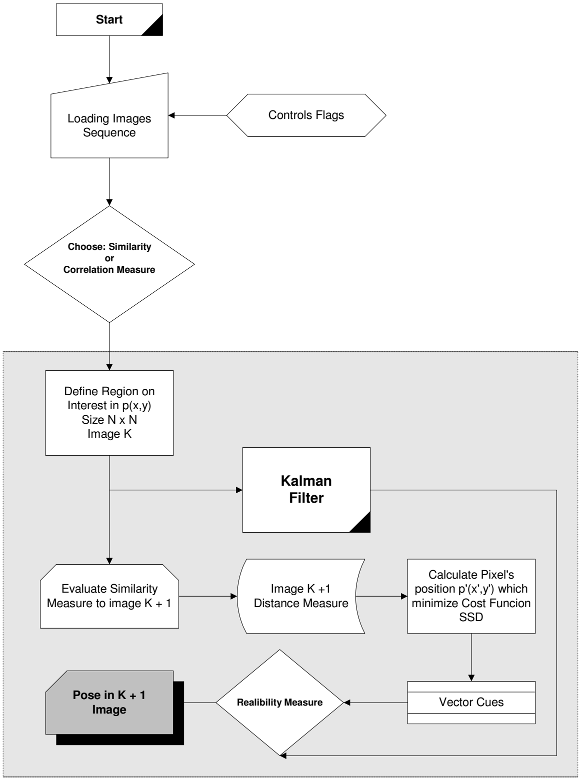


Fig. 2. A Window-matching with Kalman Filter Algorithm (WM+K) for tracking purposes

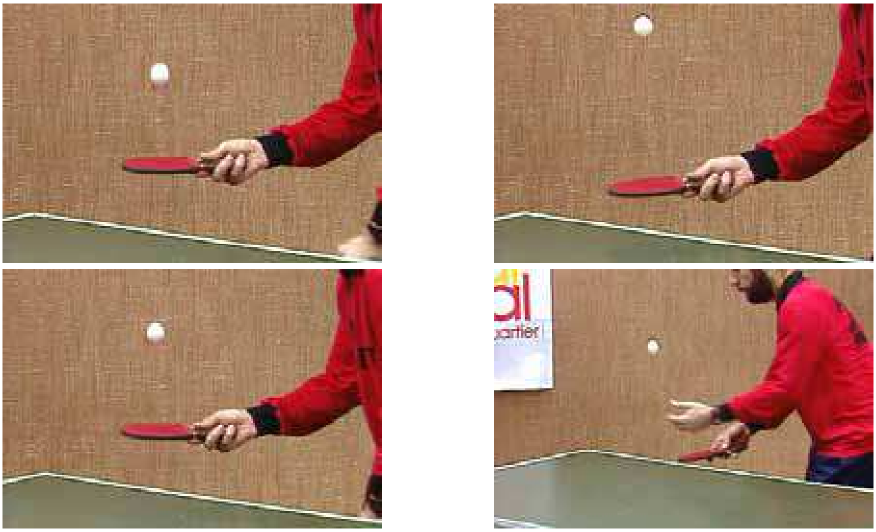


Fig. 3. A table tennis game sequence: initial, two intermediate and final frames

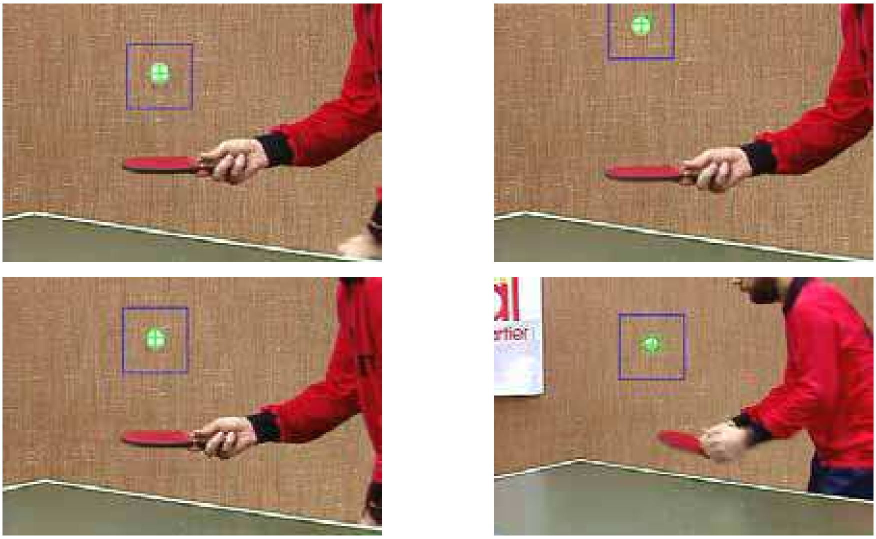


Fig. 4. Tracking the ball with the WM Algorithm

Figure 5 shows, for the same four frames, the ball tracking results from the WM+K algorithm, its tracking window (yellow) is shown together with the now supported WM searching window (green) for comparison. The initial WM+K searching window can be observed on the first frame. For this sequence with well defined environment, both algorithms have a good tracking performance.

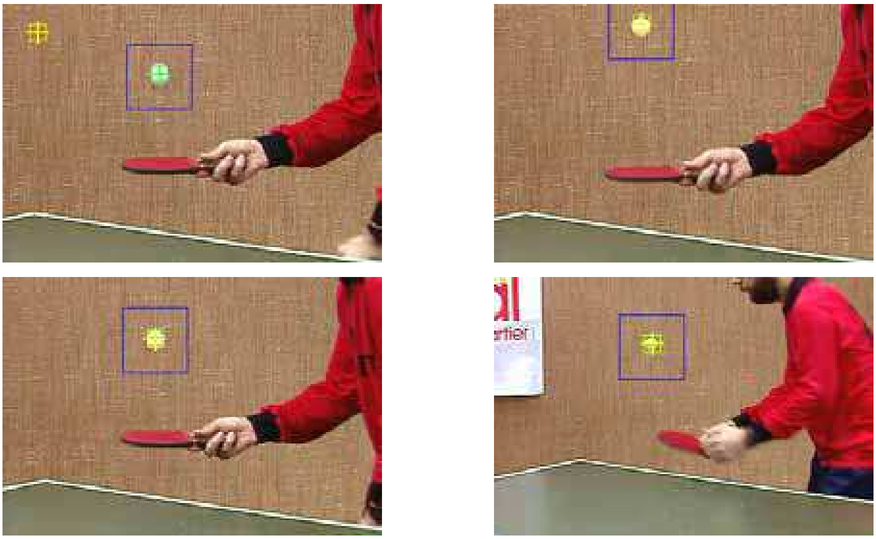
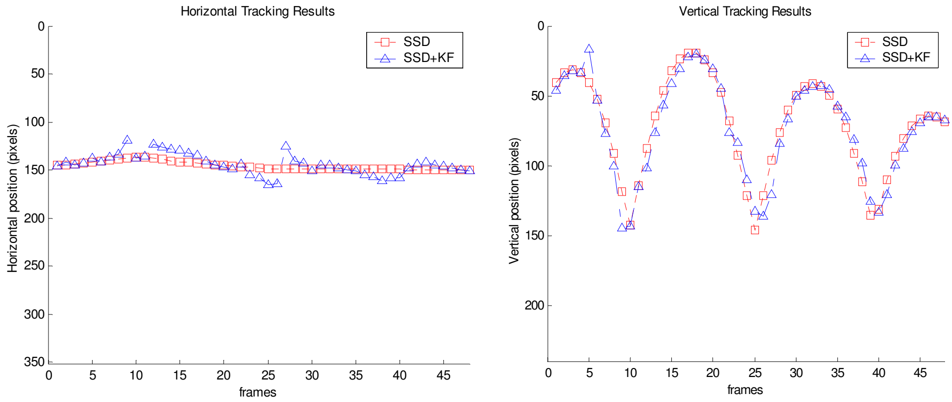


Fig. 5. Tracking the ball with the WM+K algorithm

The ball position (horizontal and vertical) along the sequence of image frames as obtained from both algorithms is shown in Figure 6. There is no much variation along the horizontal direction but is very noticeable along the vertical direction as expected.



(a) horizontal (b) vertical

Fig. 6. Ball estimated position from tracking algorithms: □ supported WM algorithm; Δ WM+K algorithm

5.2 An urban traffic sequence[2]

Figure 7 shows the initial, two intermediates and the final frames of an urban traffic sequence. The scene is a typical urban crossroad with vehicles and pedestrians, commonly found in big cities. The problem now is tracking a particular vehicle. The objects in the scene perform *3D* motion along the vertical and horizontal directions and depth variations. Image size is *320×240* pixels in JPG format. For this sequence the subregion and searching windows were 50×50 pixels and 15×15 pixels respectively.

The choice of the region of interest (ROI) is defined accordingly to the moving objects to be tracked. Other factors that influence the size of the subregions are: the degree of ambiguity, sudden illumination changes, direction, depth, etc. The effects of window size enlargements are: increase in computational cost, window drift caused by ambiguities on ROI and low contrast environments (Giacchetti, 2000). Precisely, the addition of a Kalman filtering stage to a window-matching approach minimizes the effect of these factors. The choice of the window size for the applications in this work was made accordingly to a developed motion detection algorithm (Vidal & Casanova, 2005).

As before, the WM and WM+K algorithms were applied to this sequence. To guarantee a non-biased algorithm on this type of motion, the Kalman, system noise and measurement covariance matrices were randomly initialized. Figure 8 shows, for the same four frames in Fig. 3, the vehicle tracking results from the WM algorithm (green contour). Figure 9 shows the searching results from the WM+K algorithms (yellow) and the now supported WM searching window (green). Here, both algorithms are also capable of tracking the vehicle.

---

[2] Downloaded from http://i21www.ira.uka.de/

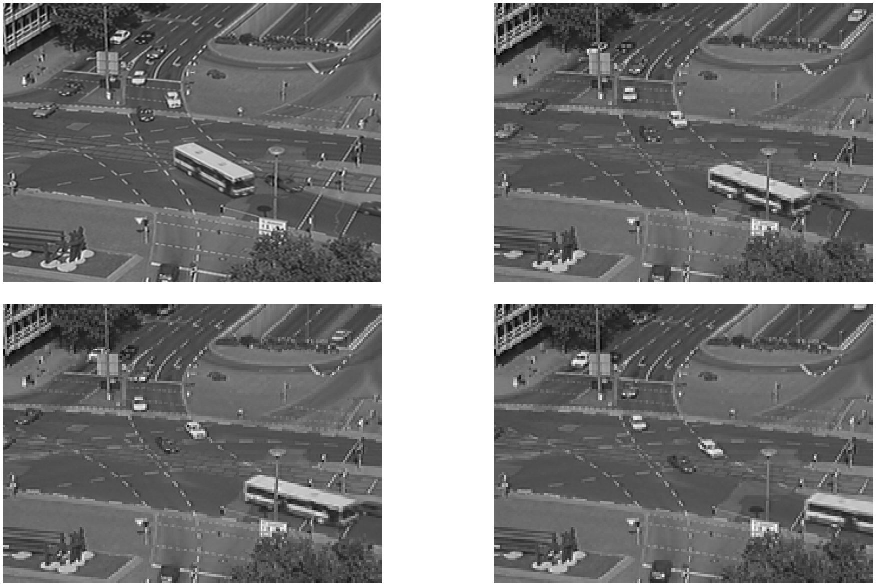


Fig. 7. Vehicle in an urban traffic sequence: initial, two intermediates and final frames

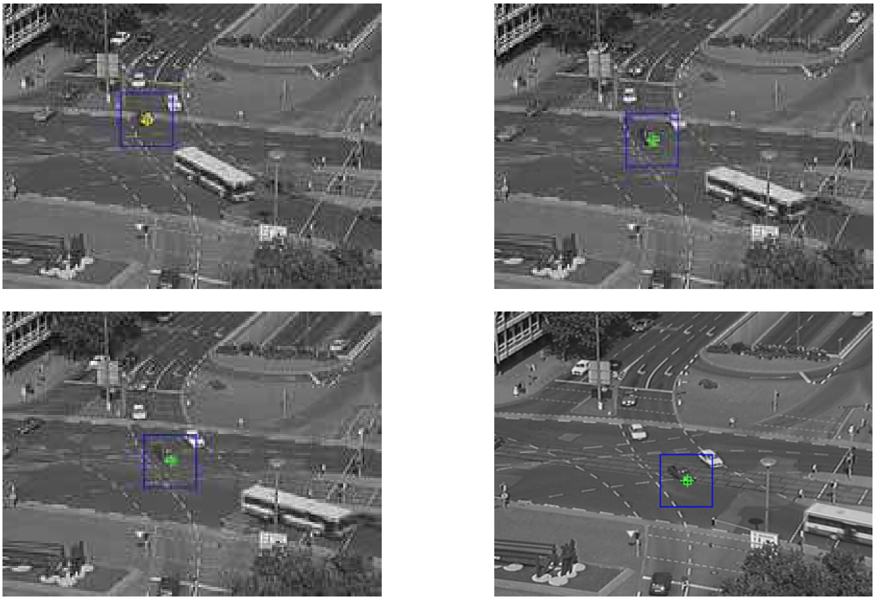


Fig. 8. Tracking a vehicle with the WM algorithm

Figure 10 shows the vehicle position (horizontal and vertical) along the sequence of image frames as obtained from both algorithms. It can be observed that the WM+K algorithm cannot track the vehicle during the first frames (triangular marks). This is due to the random initialization of the Kalman covariance matrices, but as soon as the WM stage interacts with the filter the algorithm converges quickly.

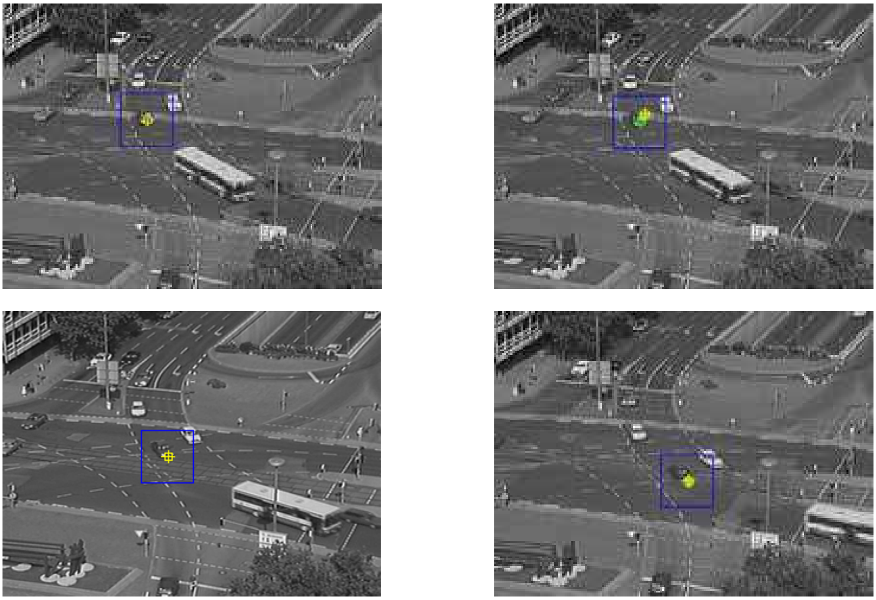
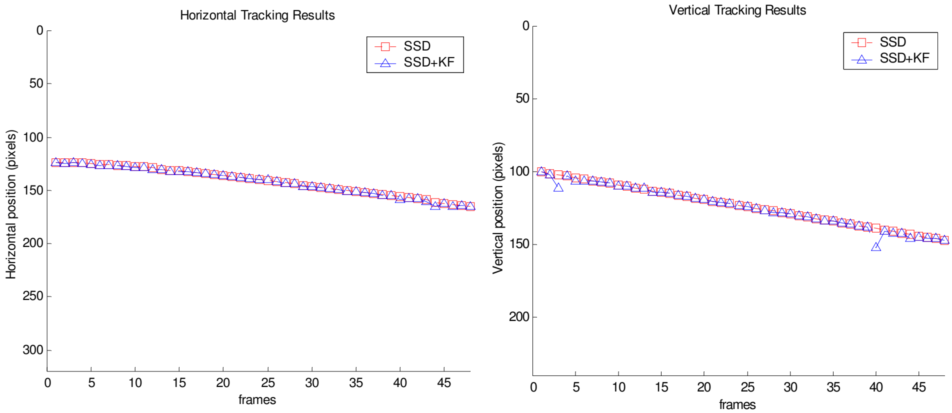


Fig. 9. Tracking a vehicle with the WM+K algorithm



(a) horizontal (b) vertical

Fig. 10. Vehicle estimated position from tracking algorithms: □ supported WM algorithm; Δ WM+K algorithm

5.3 A People Meeting and Walking Sequence[3]

Two cases of a meeting and walking sequence involving people will be considered here. In the first one, two men approach each other and then walk together. Figure 11 shows the initial, two intermediate and final frames of the meeting and walking sequence. This situation is a common scene in public buildings where there is a surveillance system installed. The scene contains problems like photometric distortions, noise, ambiguity and change of scale. For this sequence the image size is *348×288* pixels and the images are in JPG format. A subregion window size of *40×40* pixels and a searching window size of *10×10* pixels were found to be adequate. Figure 12 shows the tracking results of the WM algorithm (red line window), where it can be seen that the algorithm is not capable of tracking the human body, it got lost. On the other hand, Figure 13 shows the searching results (yellow line window) from the WM+K algorithm together with the output from the WM searching window (red line window), now supported by the Kalman filtering stage. Then in spite of the cluttered environment the combined algorithm is able to track the human body till the end of the sequence.

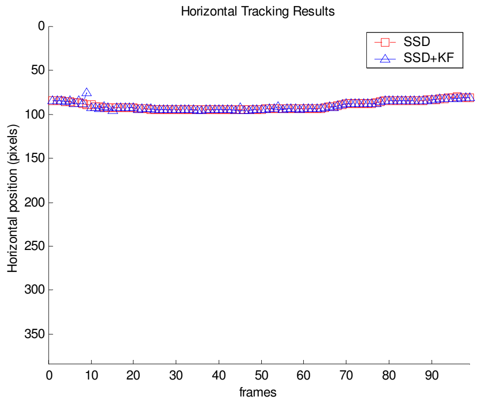
Figure 14 shows the human body estimated position (horizontal and vertical) along the sequence of image frames as estimated by the algorithms. The added Kalman filtering stage then contributes to the robustness of the tracking process.

Fig. 11. People meeting/walking sequence: initial, two intermediates and final frames

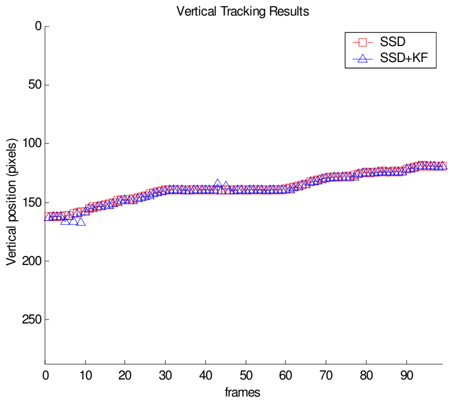
[3] Downloaded from http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

Fig. 12. Tracking people (1) with the WM algorithm

Fig. 13. Tracking people (1) with the WM+K algorithm



(a) horizontal



(b) vertical

Fig. 14. Human body estimated position from tracking algorithms: □ supported WM algorithm; Δ WM+K algorithm

A second tracking people problem is considered. Figure 15 shows four frames of a sequence where two-people walk side-by-side along a corridor. In this case there are some environmental conditions but not photometric distortion, but there will be a partial occlusion of the tracked human body. The image size is *348×288* pixels and the images are in JPG format. Tracking results are shown in Figures 16 and 17 for the same subregion and searching window sizes as the previous people tracking case.

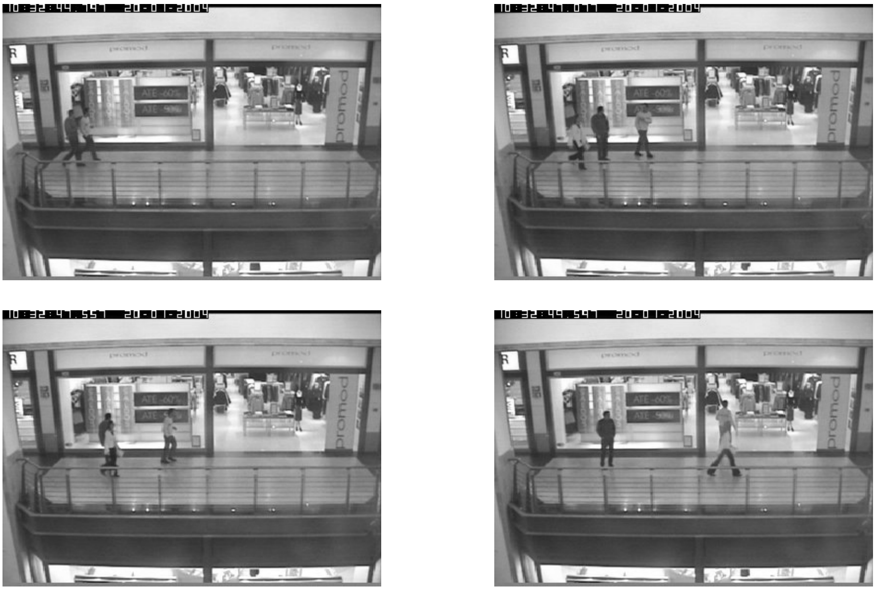


Fig. 15. Two-people walking sequence: initial, two intermediate and final frames

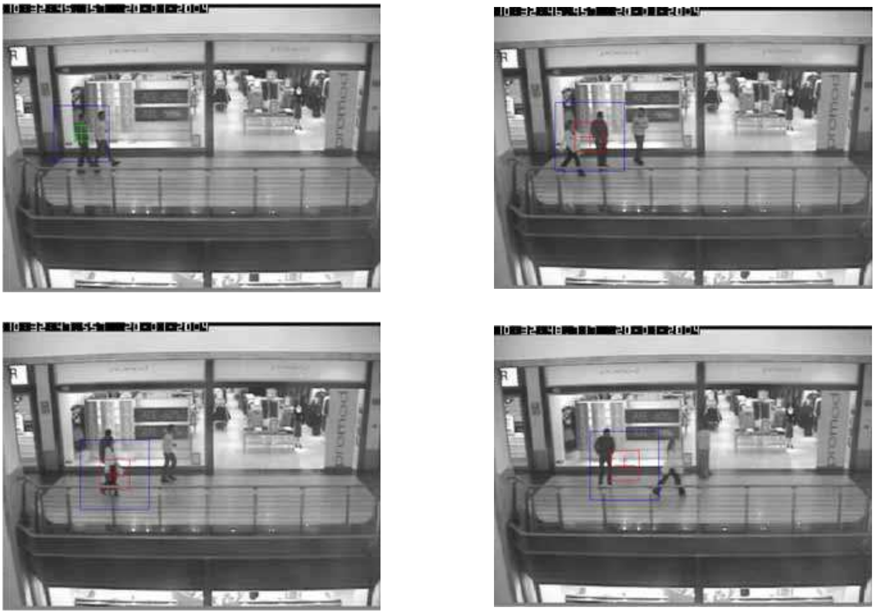


Fig. 16. Tracking people (2) with the WM algorithm

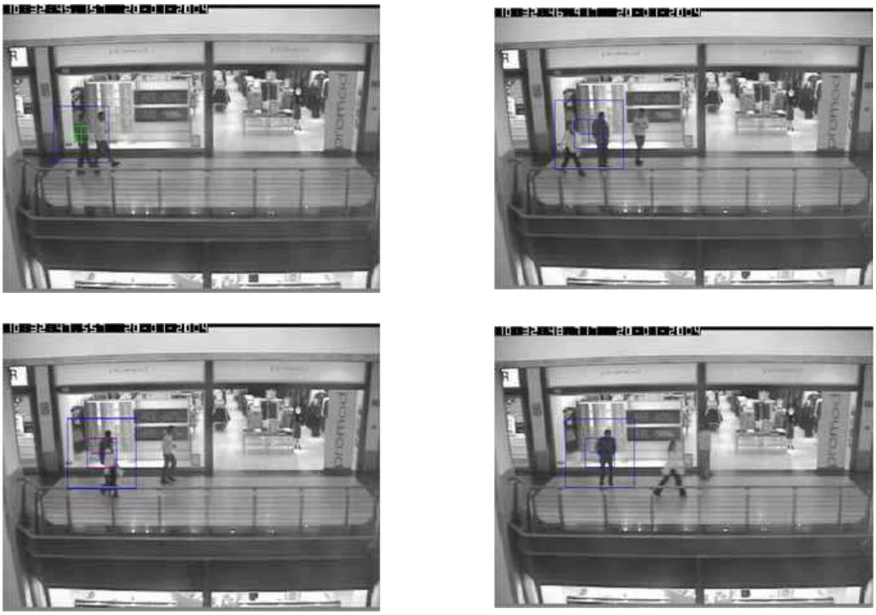
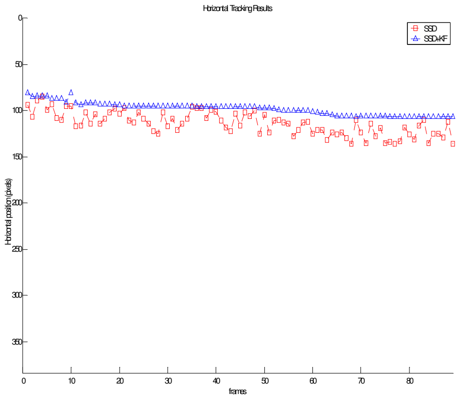
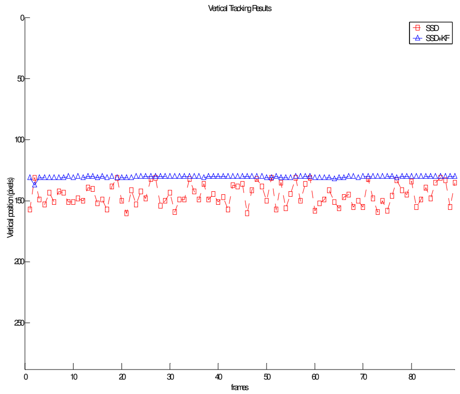


Fig. 17. Tracking people (2) with the WM+K algorithm

Results show that the WM algorithm (red) alone looses the target due to the partial occlusion (frame 3). But the WM+K algorithm (blue) keeps tracking the target in spite of the occlusion, showing that the stochastic filtering adds robustness to the process. The estimated position from the supported WM and the WM+K algorithm are shown in Figure 18.



(a) horizontal



(b) vertical

Fig. 18. Human body estimated position from tracking algorithms: □ supported WM algorithm; Δ WM+K algorithm

5.4 A Bottle floating on the Sea[4]

This sequence shows a bottle on the sea surface, the same four frames are shown in Figure 19. This is sequence is very particular in the sense that there is a random non-rigid movement with depth variations, blurring effect, scale changes and a high degree of ambiguity. These characteristics demand a robust algorithm to keep tracking the object.
The WM algorithm was applied to the bottle sequence and the tracking results for the four frames (Figure 19) are shown in Figure 20. It is clear that the WM algorithm was not able to keep tracking the bottle, mainly due to the ambiguity with the background. On the other hand the WM+K algorithm keeps tracking the ball despite the particular characteristics of the scene; these tracking results are shown in Figure 21.
Finally, Figure 22 shows the estimated position of the bottle as given by the WM+K algorithm. The bottle trajectory was of random nature and the Kalman stage delivered a smoother trajectory, mainly due to minimization of the *drift* effect.
For the sequences here considered, the larger differences on tracking estimation occurred when the elements of the main diagonal of the error covariance matrix change signs. These changes are caused by the system and measurement noises introduced into the Kalman filter (Jwo, 2007). The proposed WM+K algorithm was capable of tracking targets in these sequences. Thus, it could be said that a WM+K tracking algorithm, consisting of a window matching stage generating measurements for a Kalman estimation filter, produces better tracking results and offers robustness to the object tracking process.

---

[4] Downloaded from
http://www.cs.bu.edu/groups/ivc/data/DynamicBackgrounds/ICCV2003/water/object7/

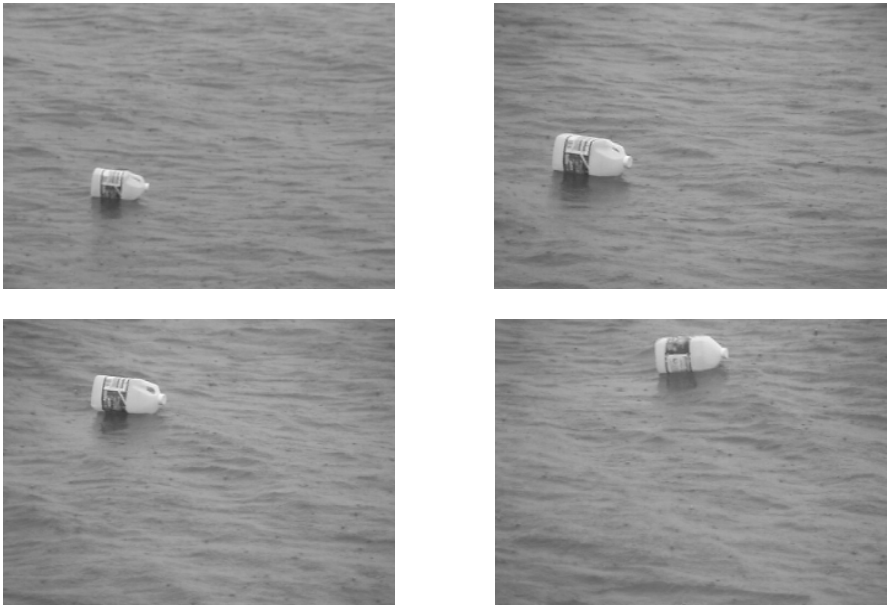


Fig. 19. Bottle floating on the sea sequence: initial, two intermediates and final frames

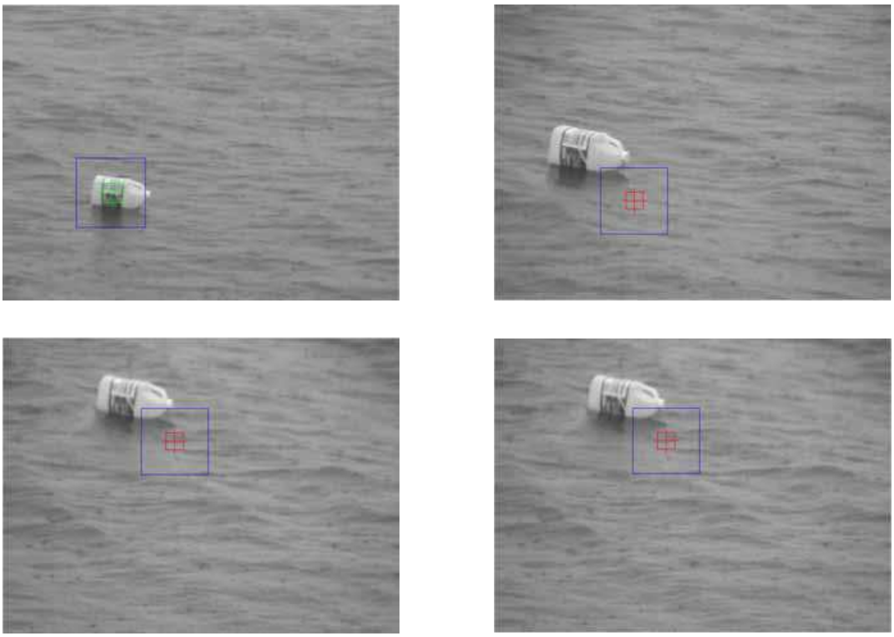


Fig. 20. Tracking a bottle with the WM algorithm

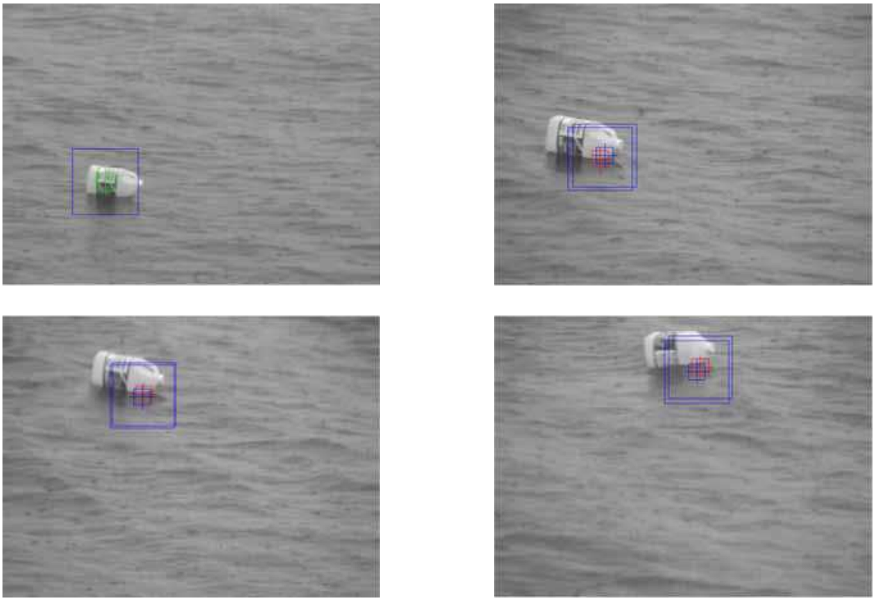
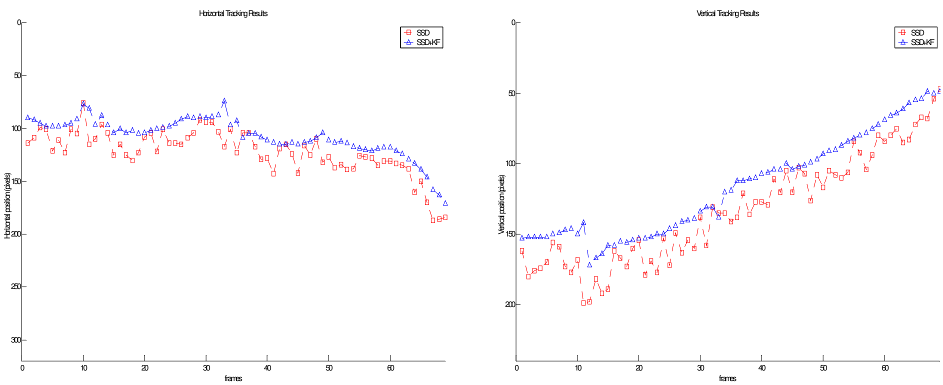


Fig. 21. Tracking a bottle with the WM+K algorithm



(a) horizontal (b) vertical

Fig. 22. Bottle estimated position from tracking algorithms: □ supported WM algorithm; Δ WM+K algorithm

6. Conclusions

This work presented an algorithm for tracking objects from a sequence of images. The algorithm is based on a window matching approach that uses as a similarity measurement the sum of the square differences (SSD). In order to improve the tracking performance under disturbances a Kalman filtering stage was incorporated. This joint operation increases the tracking robustness. The algorithm was implemented within the Matlab environment to take advantage of its developing facilities. Assigning scanning subregions contributed to increase the processing speed without compromising the tracking performance. The developed tracking algorithm was applied to track: a) the ball in a table tennis game; b) a vehicle in an urban traffic situation; c) people meeting and walking in buildings; scene; and d) a bottle floating on the sea. The approach presented would provide improvements for visual tracking due to the fact that the tracking is independent of the motion type and of the object shape. The algorithm also offers flexibility in situations where there is no previous information about the object to be tracked. A further work is the algorithm implementation in a high level language and its operation in real time. Also information from additional video cameras could be considered to attack the problem of object occlusion.

7. References

Anandan, P. (1989). A Computational Framework and an Algorithm for the Measurement of Visual Motion, *International Journal of Computer Vision*, Vol. 2, pp. 283–310.

Babu, R. B. ; Perez, P. & Bouthemy, P. (2007). Robust Tracking with Motion Estimation and Local Kernel-based Color Modeling, *Image and Vision Computing*, Vol. 25, No. 8, pp. 1205–1216, ISSN 02628856.

Barron, J.; Fleet, D.J. & Beauchemin, S.S. (1994). Systems and Experiment Performance of Optical Flow Techniques, *International Journal of Computer Vision*, Vol. 12, No. 1.

Chui, C. K. & Chen, G. (1986). *Kalman Filtering with Real-Time Applications*, 3rd edition, Springer, ISBN 038754013X.

Giachetti, A. (2000). Matching Techniques to Compute Image Motion, *Image and Vision Computing*, Vol. 18, pp. 247–260, ISSN 02628856.

Hager, G. D. & Belhumeur, P. N. (1998). Efficient Region Tracking with Parametric Models of Geometry and Illumination, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 10, pp. 1025–1039, ISSN 01628828.

Hutchinson, S., Hager, G.D. & Corke, P. (1996). A Tutorial Introduction to Visual Servo Control, *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, pp. 651-670.

Jwo, D.J. (2007). Remarks on the Kalman Filtering Simulation and Verification, *Applied Mathematics and Computation*, Vol. 186, No. 1, pp. 159–174, ISSN 00963003.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems, *Journal of Basic Engineering*, pp. 35–45, ISSN 02730979.

Papanikolopoulos, N.P. & Khosla, P.K. (1992). Shared and Traded Telerobotic Visual Control, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 878–885, ISBN 0818627204.

Rostamianfar, O. ; Janabi-Sharifi, F. & Hassanzadeh, I. (2006). Visual Tracking System for Dense Traffic Intersections, *Proceedings of the Canadian Conference on Electrical and Computer Engineering - CCECE '06*, pp. 2000–2004, ISBN 1424400384.

Shen, H. ; Nelson, G. ; Kennedy, S.; Nelson, D. ; Johnson, J. ; Spiller, D. ; White, R. H. & Kell, D. B. (2006). Automatic Tracking of Biological Cells and Compartments using Particle Filters and Active Contours, *Chemometrics and Intelligent Laboratory Systems*, Vol. 82, No. 1-2, pp. 276–282, ISSN 01697439.

Sumi, Y. ; Ishiyama, Y. & Tomita, F. (2007). Robot-vision Architeture for Real-time 6-dof Object Localization, *Computer Vision and Image Understanding*, Vol. 105, pp: 218–230.

Tan, S. S. & Hart, D. P. (2005). A Fast and Robust Feature-based 3D Algorithm using Compress Image Correlation, *Pattern Recognition Letters*, Vol. 26, No. 11, pp. 1620–1631, ISSN 01678655.

Trucco, E. & Verri, A. (1998). *Introductory Techniques for 3-D computer Vision*, Prentice Hall, ISBN 013261108-2, New Jersey, USA.

Urtasun, R.; Fleet, D. J. & Fua P. (2006). Temporal Motion Models for Monocular and Multiview 3D Human Body Tracking, *Computer Vision and Image Understanding*, Vol. 104, pp.157–177, ISSN 10773142.

Veeraraghavan, H. ; Schrater, P. & Papanikolopoulos, N. (2006). Robust Target Detection and Tracking through Integration of Motion, Color and Geometry, *Computer Vision and Image Understanding*, Vol. 103, No. 2, pp. 121–138, ISSN 10773142.

Vidal, F.B. & Casanova Alcalde, V.H. (2005). Motion Segmentation in Sequential Images based on the Differential Optical Flow. *Proceedings of the International Conference on Informatics in Control, Automation & Robotics, ICINCO*, Barcelona, Spain, pp. 94–100.

Vidal, F.B. & Casanova Alcalde, V.H. (2007). Window-Matching Techniques with Kalman Filtering for an Improved Object Visual Tracking, *Proceedings of the IEEE Conference on Automation Science and Engineering, IEEE CASE 2007*, Scottsdale, AZ, USA, pp. 829–834, ISBN 1424411548.

Yau, W. G.; Fu, L-C. & Liu, D. (2001). Design and Implementation of Visual Servoing System for Realistic Air Target Tracking, *Proceedings of the IEEE International Conference on Robotics and Automation - ICRA*, Vol. 1, pp. 229–234, ISBN 0780365793.

Kalman Filter
Edited by Vedran Kordic

ISBN 978-953-307-094-0
Hard cover, 390 pages
Publisher InTech
Published online 01, May, 2010
Published in print edition May, 2010

The Kalman filter has been successfully employed in diverse areas of study over the last 50 years and the chapters in this book review its recent applications. The editors hope the selected works will be useful to readers, contributing to future developments and improvements of this filtering technique. The aim of this book is to provide an overview of recent developments in Kalman filter theory and their applications in engineering and science. The book is divided into 20 chapters corresponding to recent advances in the filed.