AN ABSTRACT OF THE THESIS OF

Jeongmin Lee for the degree of Master of Engineering in Electrical and Computer Engineering presented on March 19, 2003.

Title : Motion Detection Algorithm using Wavelet Transform.

Abstract approved Redacted for privacy

Wojtek J. Kolodziej

This thesis presents an algorithm that estimates motion in image sequence using wavelet transform. The motion detection is performed under unfavorable conditions of background movement, change of brightness, and noise. The algorithm is tolerant to brightness changes, noise, and small movement in the background. The false alarm rate of motion detection is reduced as compared to standard techniques. Using wavelet transform is numerically efficient and the storage requirements are significantly reduced. Also, a more accurate motion detection is achieved. Tests performed on real images show the effectiveness of this algorithm. Practical results of motion detection are presented.

Motion Detection Algorithm using Wavelet Transform

by
Jeongmin Lee

A THESIS

submitted to

Oregon State University

In partial fulfillment of
The requirements for the
Degree of

Master of Science

Presented March 19, 2003

Commencement June 2003

Master of Science thesis of Jeongmin Lee presented on March 19, 2003.

APPROVED:

Redacted for privacy

Major Professor, representing Electrical and Computer Engineering

Redacted for privacy

Head of the Department of Electrical and Computer Engineering

Redacted for privacy

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for privacy

Jeongmin Lee, Author

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

Figure Page

LIST OF TABLES

Motion Detection Algorithm using Wavelet Transform

1 Introduction

The motion detection is one of the most important issues in surveillance system, image compression and object tracking. Especially in a surveillance system, the detection of objects is more important than the compression or tracking. The basic method of the motion detection is to take the difference between successive frames and apply a Gaussian filter.[13][14] This method is not suitable to detect motion under brightness changes, noise, and background movement, as these disturbances increase the false alarm rate. We compare here one of the standard motion detection methods with the proposed algorithm.

There are many motion detection methods based on the wavelet transform and pattern classification by matching process.[5]-[10] These kinds of approaches need preprocessing to generate the reference patterns. Thus the resulting algorithm is data-based, and can detect only reference patterns. Letang et al.[5] use the reference pattern and the current image instead of the two successive images. This method works well under frequent illumination changes, or small parasitical camera motion, but it needs large reference binary maps, which increase storage and preprocessing requirements.

We use discrete wavelet transform with Daubechies' orthogonal wavelet[5][15][18]. Daubechies' orthogonal wavelets are convenient because of their finite support. The wavelet transform is known for being efficient in edge detection and noise cancellation.[17]

One of the basic methods of edge detection is the use of derivatives.[11][12] Wavelet transforms define the edge well, as do derivatives in image processing. Near the object edge in the image, there are big changes in pixel values. Detection of object edges is closely related to the object detection and hence allows tracking of object motion.

The proposed here detection technique is based on the novel idea of the wavelet template that defines the shape of an object in terms of a subset of the wavelet coefficients of the image. This method is invariant under changes in color and texture and can be used to robustly define a rich and complex class of objects.[7]

We have tested proposed method under different illuminations and background motion, which can occur due the surveillance camera motion. The tests included also noisy image, which may results for example, from wireless data transmission.[2]

We used the MATLAB wavelet toolbox for simulation and Web cameras (Logitech ClickSmart 510 (lab1), Logitech QuickCam Express (lab2)) were used for capturing the images.

The organization of this thesis is as follows. In section 2, we introduce the wavelet transform. In section 3, we present the algorithm based on the wavelet transform. In section 4, we discuss the algorithm test results. In sections 5 and 6, we present general results and conclusions.

2 Wavelet Transform [1][4][7][8][10][12][13]

The wavelets, first mentioned by Haar in 1909, have a compact support, which means that the wavelets vanish outside of the finite interval. However Haar wavelets are not continuously differentiable and thus not very suitable for gradient based methods. The wavelets with an effective algorithm for numerical image processing were introduced by (Gabor, 1946). These wavelets were based on function that can vary in scale and is conservative when computing the functional energy. Between 1960s and 1980s, mathematicians such as Grossman and Morlet (1985) defined wavelets in the context of quantum physics. Mallat (1989) gave a boost to digital signal processing by inventing the pyramidal algorithms and orthonormal wavelet bases. Later Daubechies (1990) used Mallat's algorithm to construct a set of wavelet orthonormal basic functions that are the cornerstone of wavelet applications today.

The class of functions that can be represented by the wavelets are those that are square integrable. This class is denoted by $L^2(R)$.

$$f(x) \in L^2(R) \Rightarrow \int_{-\infty}^{+\infty} |f(x)|^2 dx < \infty \qquad (1)$$

The set of functions that are generated in the wavelet analysis are obtained by dilating (scaling) and translating (time shifting) a single prototype function called the mother wavelet. The wavelet function $\Psi(x) \in L^2(R)$ has two characteristic parameters called dilation (a) and translation (b), which vary continuously. A set of wavelet basis functions $\Psi_{a,b}(x)$, is given by

$$\psi_{a,b}(x) = \frac{1}{\sqrt{|a|}}\psi(\frac{x-b}{a}) \qquad a,b \in R; a \neq 0 \tag{2}$$

Here, the translation parameter, "b", controls the position of the wavelet in time. The "narrow" wavelet can access high frequency information, while the more dilated wavelet can access low frequency information. This means that the parameter "a" varies for different frequencies. The continuous wavelet transform is defined by

$$W_{a,b}(f) = < f, \psi_{a,b} > = \int_{-\infty}^{+\infty} f(x)\psi_{a,b}(x)dx \tag{3}$$

The wavelet coefficients are given as the inner product of the function being transformed with each wavelet basis function.

Daubechies (1990) invented one of the most elegant families of wavelets. They are called Compactly Supported Orthonormal Wavelets and are used in Discrete Wavelet Transform (DWT). In this approach, the scaling function is used to compute the ψ. The scaling function $\Phi(x)$ and the corresponding wavelet $\Psi(x)$ are defined by

$$\phi(x) = \sum_{k=0}^{N-1} c_k \phi(2x-k) \tag{4}$$

$$\psi(x) = \sum_{k=0}^{N-1} (-1)^k c_k \phi(2x+k-N+1) \tag{5}$$

Where N is an even number of wavelet coefficients c_k, k = 0 to N-1. The discrete representation of an orthonormal compactly supported wavelet basis of $L^2(R)$ is formed by dilation and translation of signal function $\Psi(x)$, called the wavelet

function. It is assumed that the dilation parameters "a" and "b" take only the discrete values: $a = a_0^j$, $b = kb_0a_0^j$, where k, j $\in$ Z, $a_0 > 1$, and $b_0 > 0$. The wavelet function may be rewritten as

$$\psi_{j,k}(x) = a_0^{-j/2}\psi(a_0^{-j}x - kb_0) \tag{6}$$

and the Discrete-Parameter Wavelet Transform (DPWT) is defined as

$$DPWT(f) =< f,\ \psi_{j,k} >= \int_{-\infty}^{+\infty} f(x)a_0^{-j/2}\psi(a_0^{-j}x - kb_0)dx \tag{7}$$

The choice between dilations and translations is made on the basis of the power of two, the so-called dyadic scales and positions, thus making the analysis efficient and accurate. In this case, the frequency axis is partitioned into bands by using the power of two for the scale parameter "a." Considering samples at the dyadic values, we set $b_0 = 1$ and $a_0 = 2$, and then the discrete wavelet transform becomes

$$DPWT(f) =< f,\ \psi_{j,k} >= \int_{-\infty}^{+\infty} f(x)\{2^{-j/2}\psi(2^{-j}x - k)\}dx \tag{8}$$

Here, $\Psi_{j,k}(x)$ is defined as

$$\psi_{j,k}(x) = 2^{-j/2}\psi(2^{-j}x - k), \qquad j,k \in Z \tag{9}$$

3 Algorithm

The proposed algorithm consists of three main parts, cascade of 3 wavelet transforms $(DWT)^3$, extended pseudo color matrix scaling, and high pass filtering. In Figure 1, we see the block diagram of the algorithm.

We used "Daubechies 1" wavelet, which is usually used for Discrete Wavelet Transform (DWT). The Daubechies wavelets form orthonormal basis of $L^2(R)$.

The "caltrain" images already have small amount of background motion. Due to this, we can have detection errors if only one stage of DWT is used. Therefore we repeat DWT. The results of multiple stages of DWT are shown in figure 2.

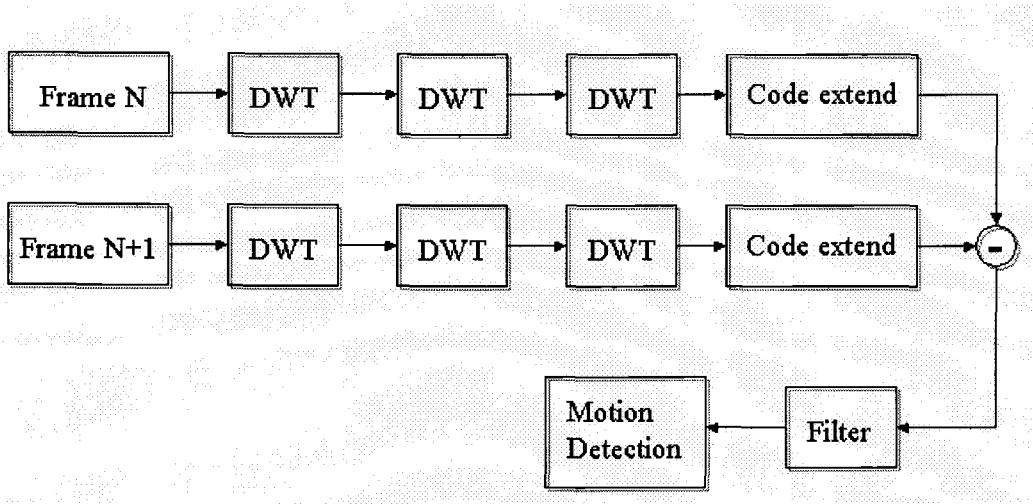


Figure 1. Block diagram of the algorithm

From figure 2, we see that three stage DWT has the best performance, as measured later by low error rate. The reason for using of $(DWT)^3$ is purely experimental. We found that $(DWT)^n$ for $n < 3$ causes more detection errors due to background motion and low resolution. For $n > 3$, the motion detection is less sensitive.

We used extended pseudo color matrix scaling because it displays a rescaled version of data (between 1 to 255) leading to a clearer presentation of the details and providing a better approximation. Together this yields a more accurate motion

detection.[4] In figure 2, (b) is rescaled from 45 – 418.5 to 1 – 255,(c) from 114.5 – 790.75 to 1 – 255,(d) from 236.5 – 1575.625 to 1 – 255, and (e) from 537,1875 – 3112.75 to 1 – 255.

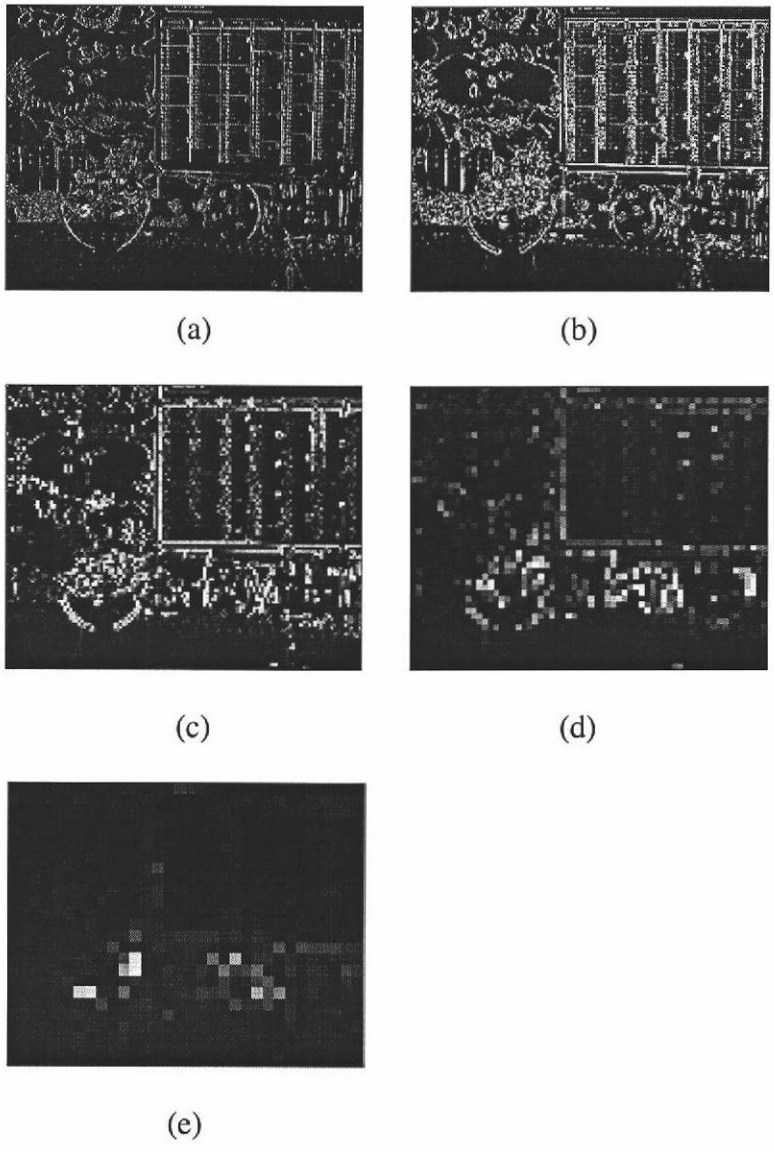


Figure 2. (a) The difference of two original images, (b) The difference after one stage of DWT, (c) The difference after two stage of DWT, (d) The difference after three stage of DWT, (e) The difference after four stage of DWT

The high pass filter has pre-selected threshold value of half of the standard deviation of the original image data. This threshold value is a critical parameter in motion detection.

We used MATLAB to simulate this algorithm. The MATLAB code is presented in the appendix A.

4 Test

We tested the proposed motion detection algorithm with several image frames shown in figure 3, which are "caltrain", "tabletennis", and "lab1" (images captured at OSU laboratory). We used "lab2" to test illumination changes. We tested the images under object motion, illumination change, background motion, and additive noise. The "caltrain" and "lab2" images are used to compare a basic motion detection method with the proposed here algorithm. The "caltrain" and "tabletennis" images are black and white, and the "lab1" and "lab2" images are in color. We applied the basic method to only two frame sequences since this is sufficient to make a comparison.

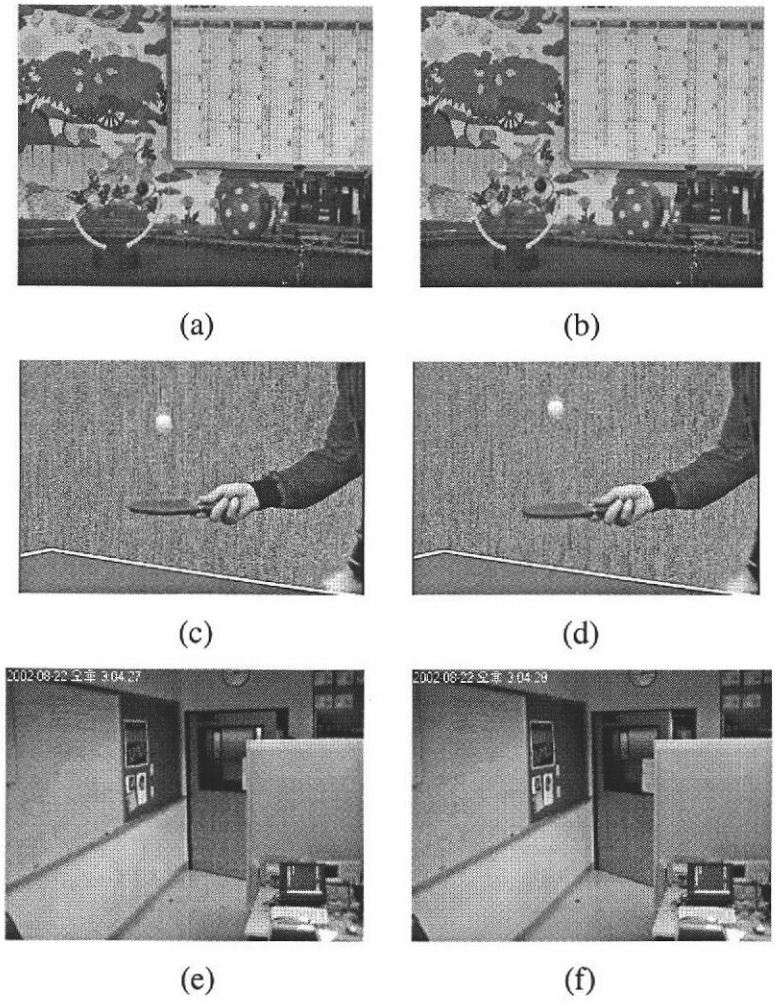


Figure 3. Tested image frames (a), (b) caltrain (512 x 400, 8bits), (c), (d) table-tennis (720 x 480, 8bits), (e), (f) lab1 (320 x 240, 24bits)

4.1 Motion detection

We tested the proposed algorithm using the frame sequences of "caltrain", "tabletennis", and "lab1". The frame sequence of "caltrain" was used to test the basic method.

4.2 Illumination change

We recaptured images under different illuminations (Figure 4) to test the detection algorithms for an illumination change sensitivity. The illumination change test was also used to compare the basic method with the proposed one.

(a) (b)

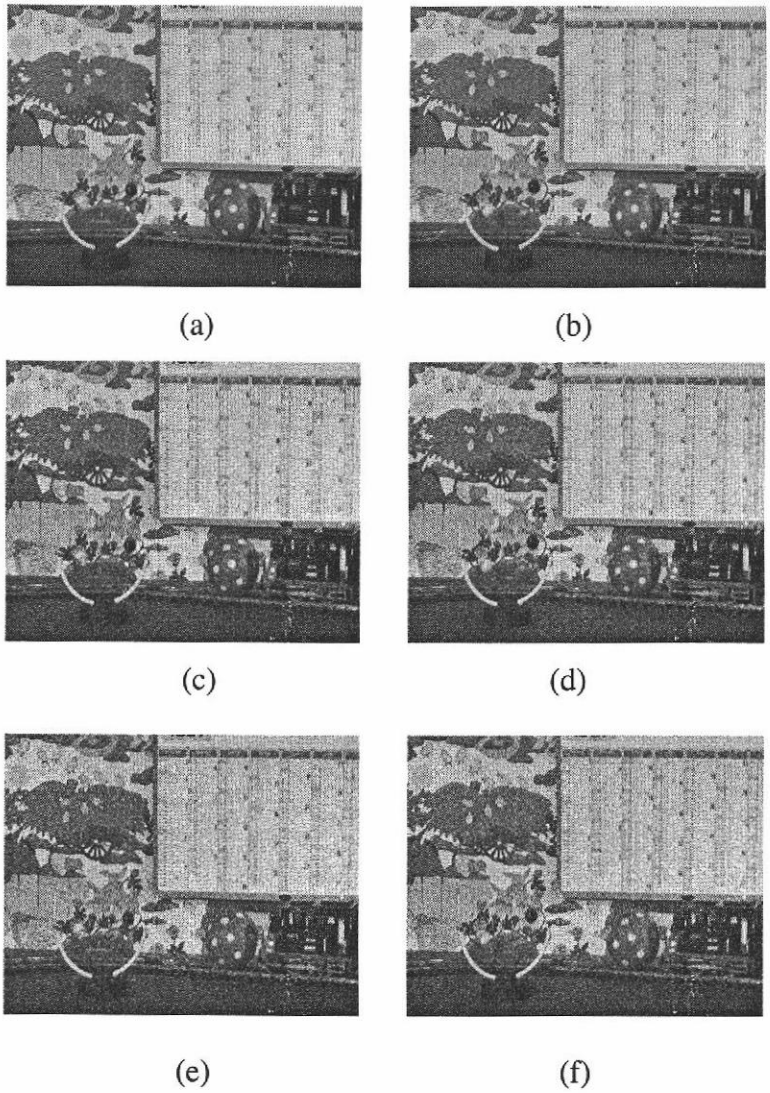
Figure 4. The lab2 with different illumination (a) more bright, (b) less bright, (176 x 144, 24 bits)

4.3 Background motion

The "tabletennis" and "lab1" sequences were used to test for background motion. We generated new images, which have 1 and 2 pixel x-axis differences, 1 and 2 pixel y-axis differences, and a combination of x and y-axis differences, to test for background motion. We did not test the "caltrain" sequence since the two "caltrain" frames (figure 3 (a) and (b)) already have 1 pixel x-axis and 1 pixel y-axis differences.

4.4 Low resolution (noise images)

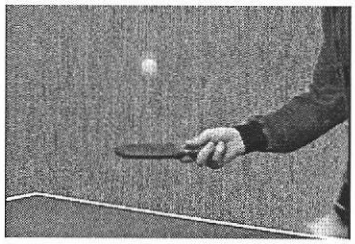
We tested the "caltrain", "tabletennis", and "lab1" sequences by adding the three levels of noise of 10dB, 5dB, and 3dB S/N ratio, to each of the images in figure 1. Each of the images has 8bits of pixel depth, so we added ± 40(10dB), ± 72(5dB), and ±91(3dB) intensity value noise. Figure 5 shows the noisy images.



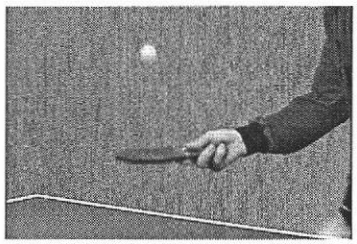
(a) (b)

(c) (d)

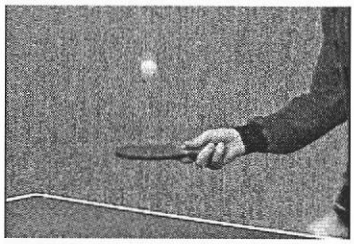
(e) (f)



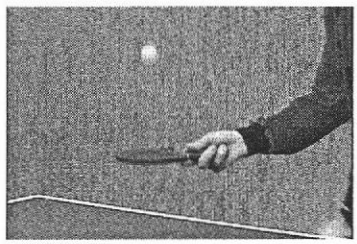
(g)



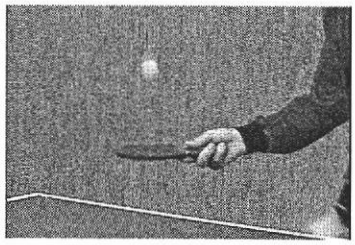
(h)



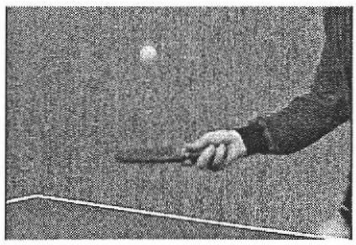
(i)



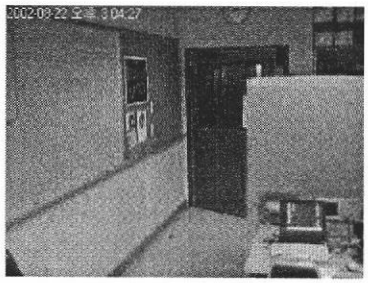
(j)



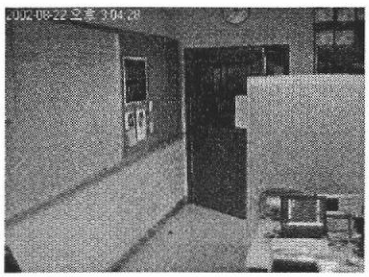
(k)



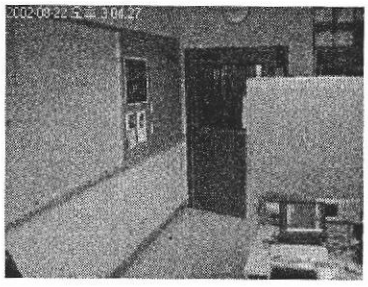
(l)



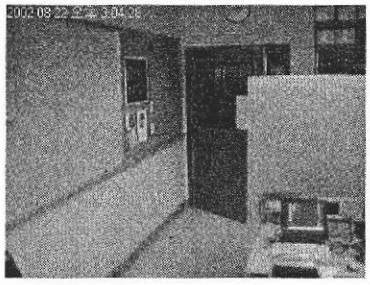
(m)



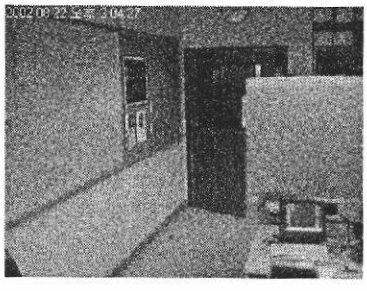
(n)



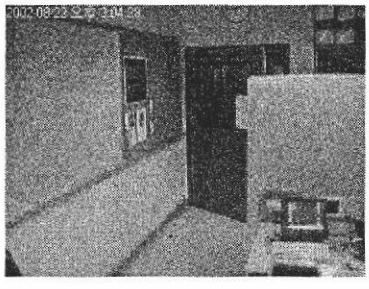
(o)



(p)



(q)



(r)

Figure 5. Noise added images (a), (b) caltrain with 10dB noise, (c), (d) caltrain with 5dB noise, (e), (f) caltrain with 3dB noise, (g), (h) tabletennis with 10dB noise, (i), (j) tabletennis with 5dB noise, (k), (l) tabletennis with 3dB noise, (m), (n) lab1 with 10dB noise, (o), (p) lab1 with 5dB noise, (q), (r) lab1 with 3dB noise.

5 Results

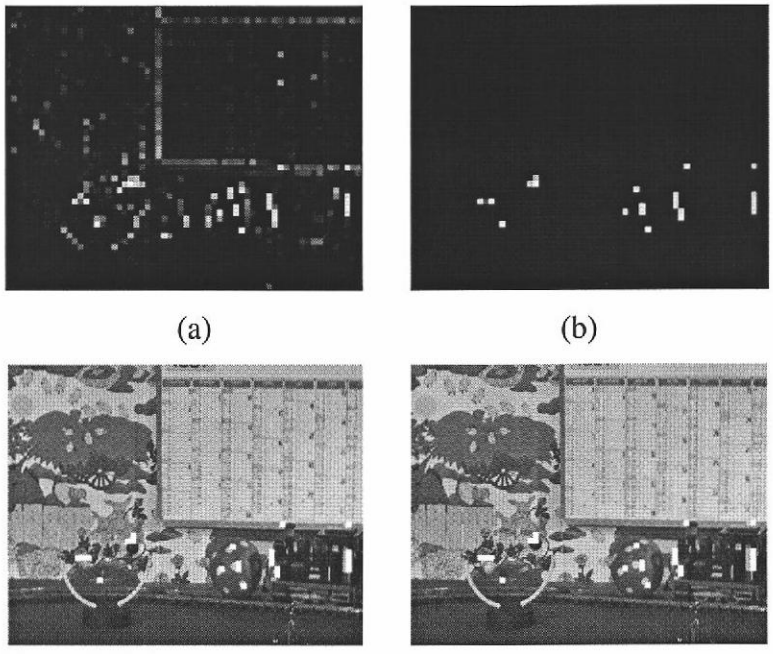
5.1 Motion detection

To illustrate this algorithm performance, each figure below shows the four images. The first image shows the difference of the coefficient after three stages of DWT. The size of this image is 1/64 of the original image. The second image is the image after filtering. The third and fourth image show the motion detection results as seen in the first two original frames. In the second image, the bright part is the detected motion.

Figures 6, 7 and 8 show the results of motion detection of "caltrain", "tabletennis" and "lab1" images, respectively.

The threshold values of figure 6, 7 and 8 are 25.40, 19.70 and 27.12, respectively.

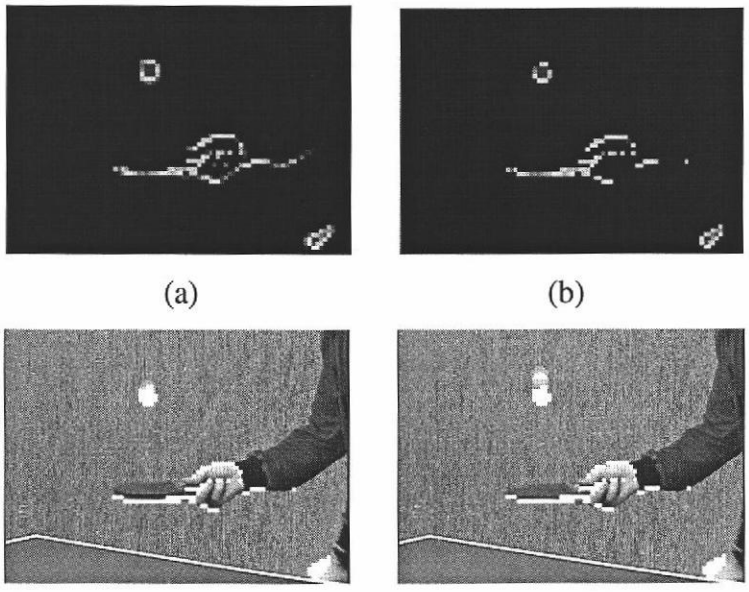
Figures 6, 7, and 8 show that the proposed algorithm works well and detects only the true object motion. In the case of "caltrain", the detection of the movement of objects is a difficult task for human eye. In figure 7, we can see that there is a movement of the ball because there are shadows of the ball. Figure 8 show a big movement of the door, so the movement can be easily detected.



(a) (b)

(c) (d)

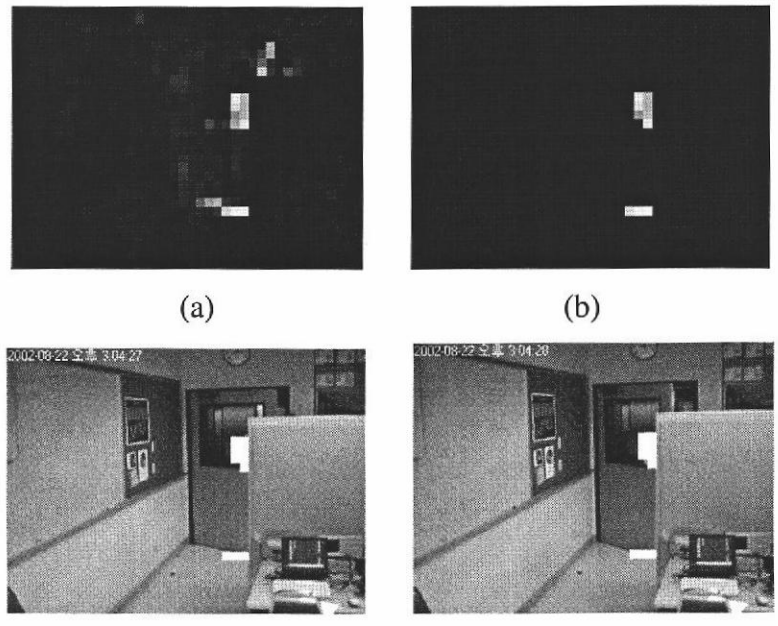
Figure 6. The result of motion detection at the caltrain (a) difference after three stage DWT (64 x 50), (b) after filtering (64 x 50), (c) applied detected motion to the first frame (512 x 400), (d) applied detected motion to the second frame (512 x 400)



(a) (b)

(c) (d)

Figure 7. The result of motion detection at the tabletennis (a) difference after three stage DWT (90 x 60), (b) after filtering (90 x 60), (c) applied detected motion to the first frame (720 x 480), (d) applied detected motion to the second frame (720 x 480)



(a) (b)

(c) (d)

Figure 8. The result of motion detection at the lab1 (a) difference after three stage DWT (40 x 30), (b) after filtering (40 x 30), (c) applied detected motion to the first frame (320 x 240), (d) applied detected motion to the second frame (320 x 240)

Figure 9 shows the result of a simple detection method applied to "caltrain" sequence. This method takes the difference between two successive frames and then applies a Gaussian filter. We can easily see that this method detects many not necessarily real motions in the image.

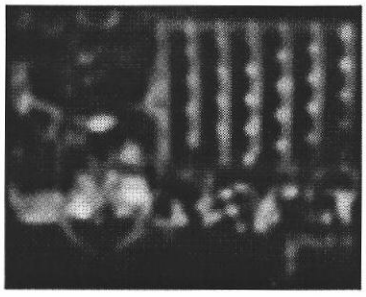


Figure 9. The result of motion detection by simple method at the caltrain (512 x 400)

5.2 Illumination change

Figure 10 shows that the results of illumination changes in the images of figure 4. Here, the threshold value is 27.17. There is no motion detected, showing that the proposed algorithm works well under illumination change.

Figure 11 shows the results of "lab2" images processed with the basic motion detection method. It is easy to see that the basic method is sensitive to illumination changes, erroneously identifying them as the objects motion and causing false alarm.

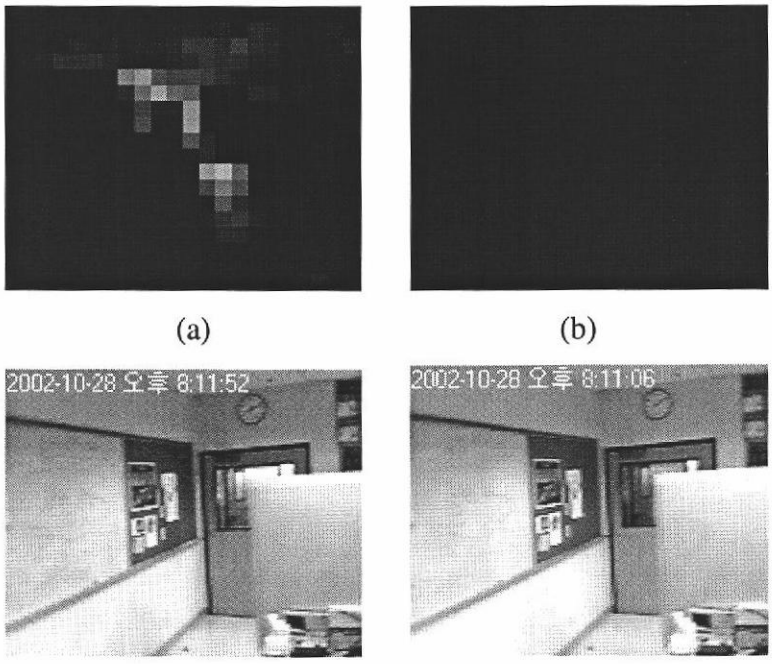


Figure 10. The result of motion detection under different illumination (a) difference after three stage DWT (22 x 18), (b) after filtering (22 x 18), (c) applied detected motion to the first frame (176 x 144), (d) applied detected motion to the second frame (176 x 144)

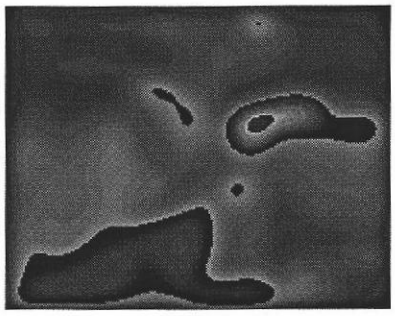
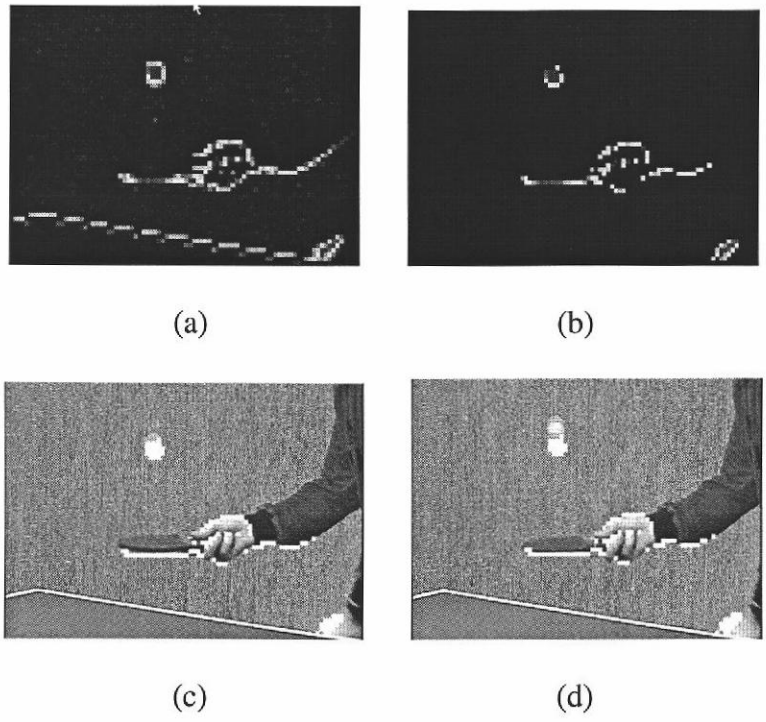


Figure 11. The result of motion detection by the simple method at the lab2 (176 x 144)

5.3 Background motion

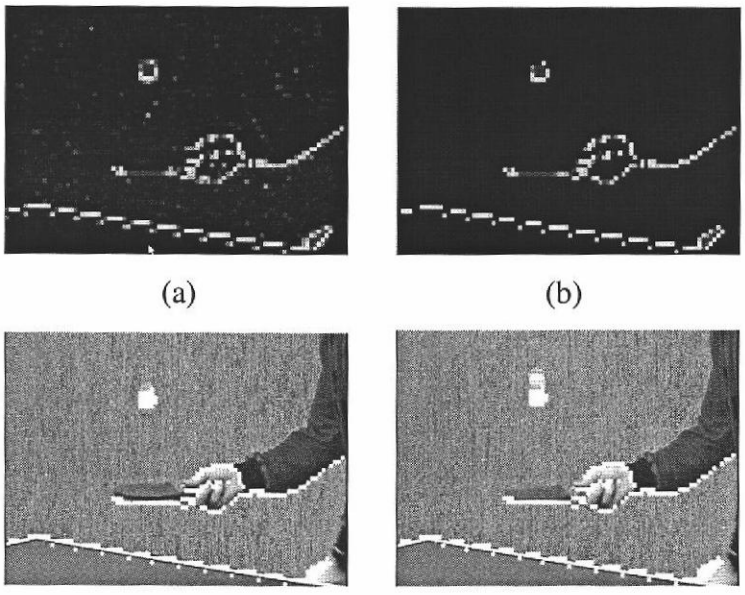


(a) (b)

(c) (d)

Figure 12. The result of motion detection at tabletennis images with 1 pixel y-axis background motion (a) difference after three stage DWT (90 x 60), (b) after filtering (90 x 60), (c) applied detected motion to the first frame (720 x 480), (d) applied detected motion to the second frame (720 x 480)

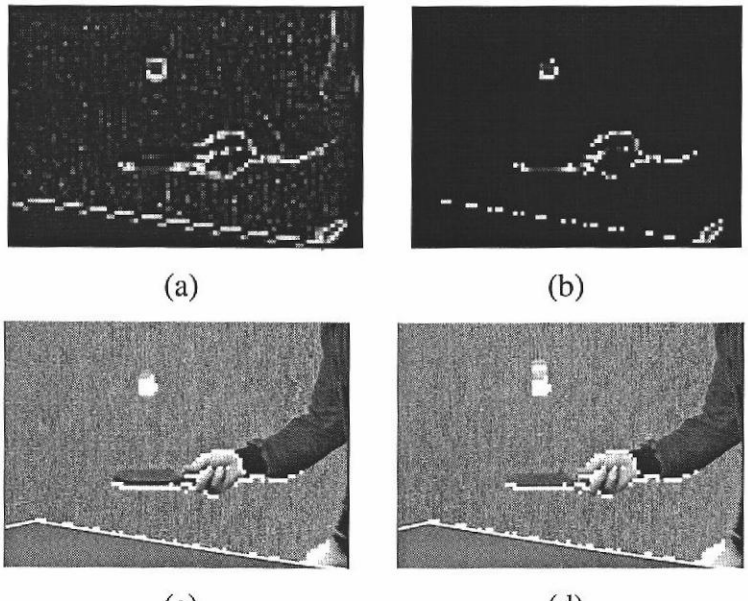
The threshold value of figures 12, 13, 14, 15, and 16 is 18.02. The threshold value of figures 17, 18, 19, 20, and 21 is 21.79. From figure 12 through 21, we can see that the proposed algorithm works well only for 1 pixel x-axis or y-axis background motions. Larger than 1 pixel x-axis or y-axis background motions results in motion detection errors.



(a) (b)

(c) (d)

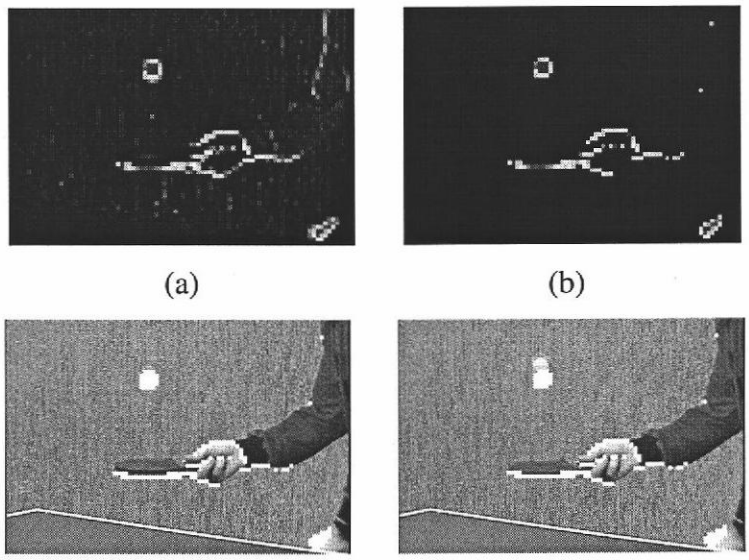
Figure 13. The result of motion detection at tabletennis images with 2 pixel y-axis background motion (a) difference after three stage DWT (90 x 60), (b) after filtering (90 x 60), (c) applied detected motion to the first frame (720 x 480), (d) applied detected motion to the second frame (720 x 480)



(a) (b)

(c) (d)

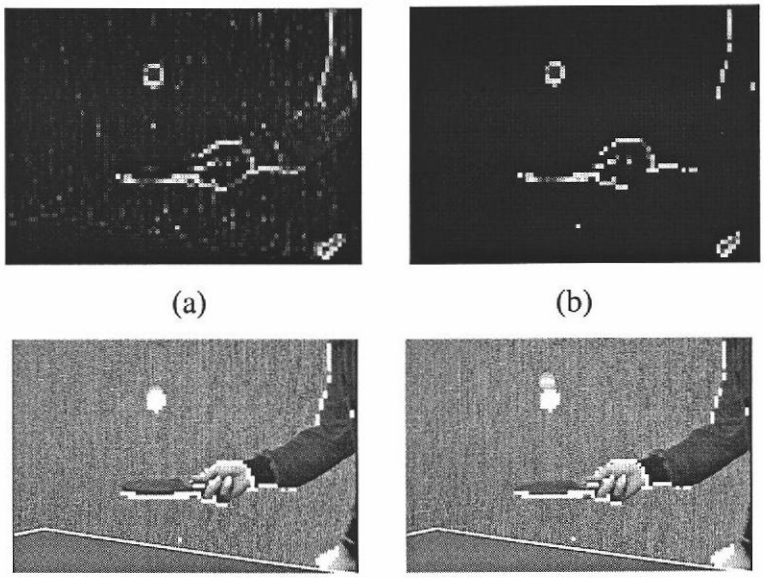
Figure 14. The result of motion detection at tabletennis images with 1 pixel x-axis and 1 pixel y-axis background motion (a) difference after three stage DWT (90 x 60), (b) after filtering (90 x 60), (c) applied detected motion to the first frame (720 x 480), (d) applied detected motion to the second frame (720 x 480)



(a) (b)

(c) (d)

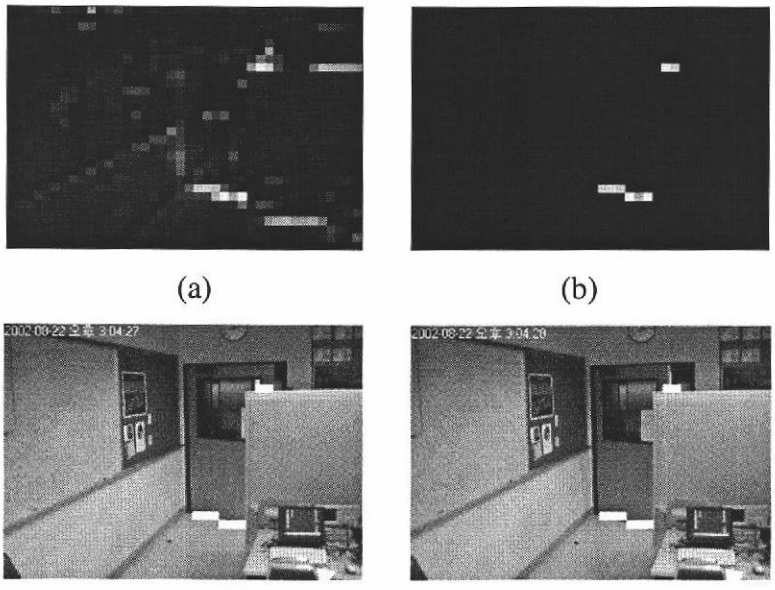
Figure 15. The result of motion detection at tabletennis images with 1 pixel x-axis background motion (a) difference after three stage DWT (90 x 60), (b) after filtering (90 x 60), (c) applied detected motion to the first frame (720 x 480), (d) applied detected motion to the second frame (720 x 480)



(a) (b)

(c) (d)

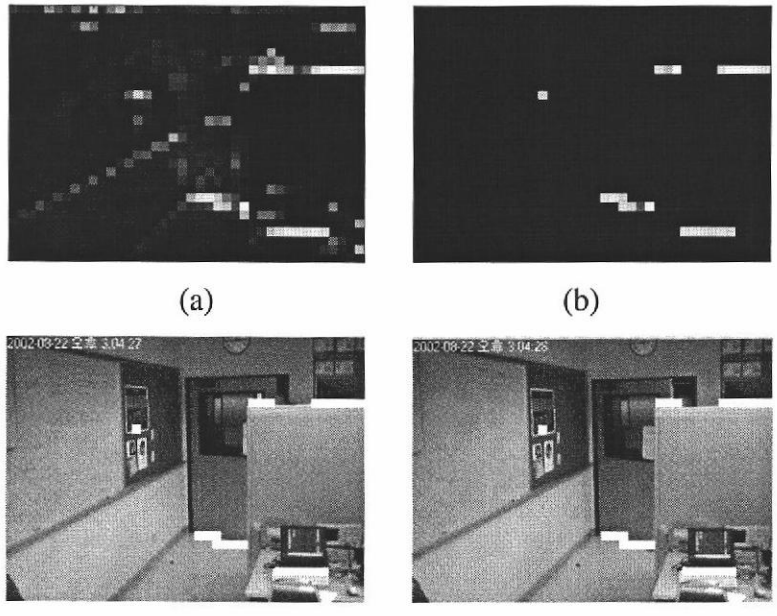
Figure 16. The result of motion detection at tabletennis images with 2 pixel x-axis background motion (a) difference after three stage DWT (90 x 60), (b) after filtering (90 x 60), (c) applied detected motion to the first frame (720 x 480), (d) applied detected motion to the second frame (720 x 480)



(a) (b)

(c) (d)

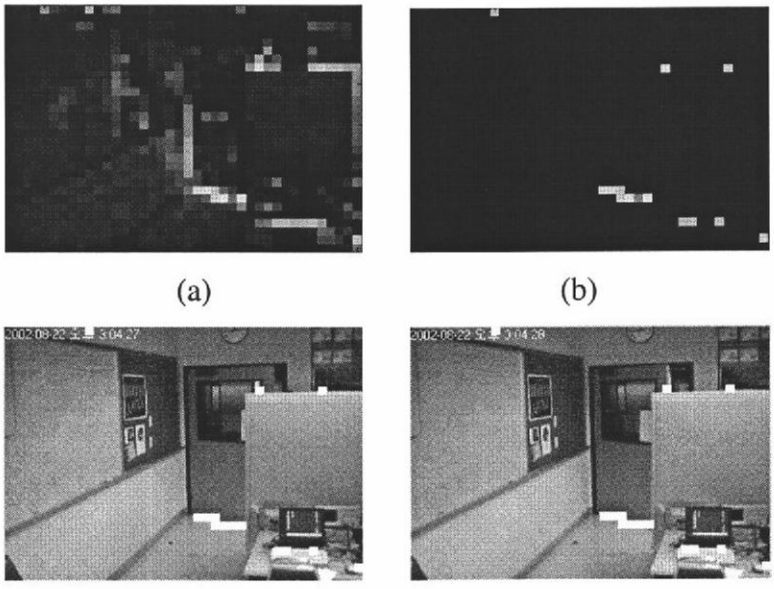
Figure 17. The result of motion detection at lab1 images with 1 pixel y-axis background motion (a) difference after three stage DWT (40 x 30), (b) after filtering (40 x 30), (c) applied detected motion to the first frame (320 x 240), (d) applied detected motion to the second frame (320 x 240)



(a) (b)

(c) (d)

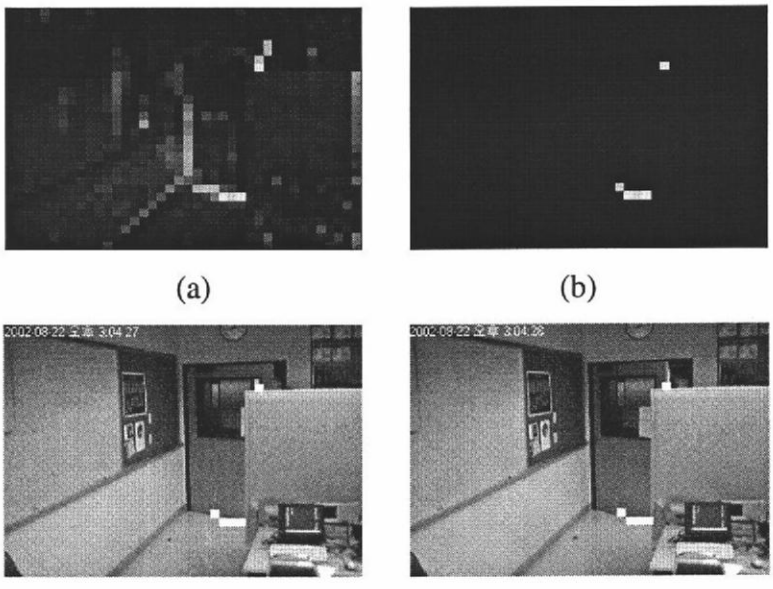
Figure 18. The result of motion detection at lab1 images with 2 pixel y-axis background motion (a) difference after three stage DWT (40 x 30), (b) after filtering (40 x 30), (c) applied detected motion to the first frame (320 x 240), (d) applied detected motion to the second frame (320 x 240)



(a) (b)

(c) (d)

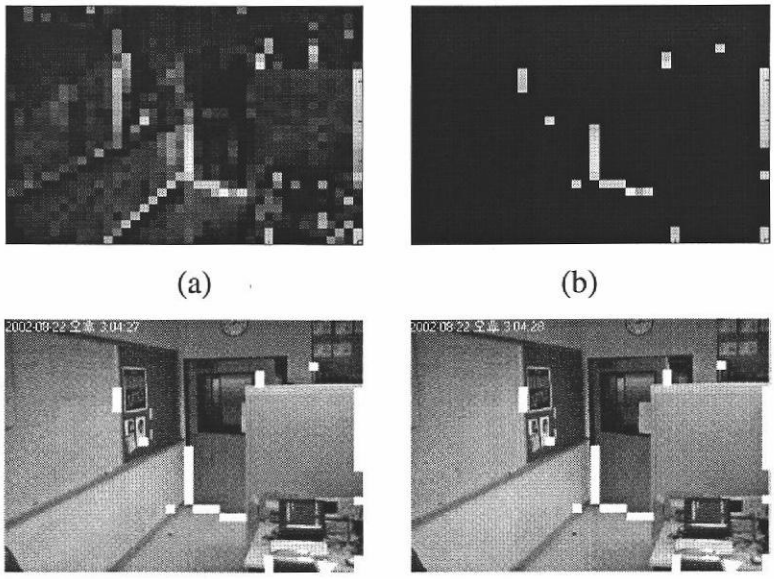
Figure 19. The result of motion detection at lab1 images with 1 pixel x-axis and 1 pixel y-axis background motion (a) difference after three stage DWT (40 x 30), (b) after filtering (40 x 30), (c) applied detected motion to the first frame (320 x 240), (d) applied detected motion to the second frame (320 x 240)



(a) (b)

(c) (d)

Figure 20. The result of motion detection at lab1 images with 1 pixel x-axis background motion (a) difference after three stage DWT (40 x 30), (b) after filtering (40 x 30), (c) applied detected motion to the first frame (320 x 240), (d) applied detected motion to the second frame (320 x 240)



(a) (b)

(c) (d)

Figure 21. The result of motion detection at lab1 images with 2 pixel x-axis background motion (a) difference after three stage DWT (40 x 30), (b) after filtering (40 x 30), (c) applied detected motion to the first frame (320 x 240), (d) applied detected motion to the second frame (320 x 240)

Table 1. Summarizes the experimental results shown in figures 12 to 21
(O : no error, X : error)

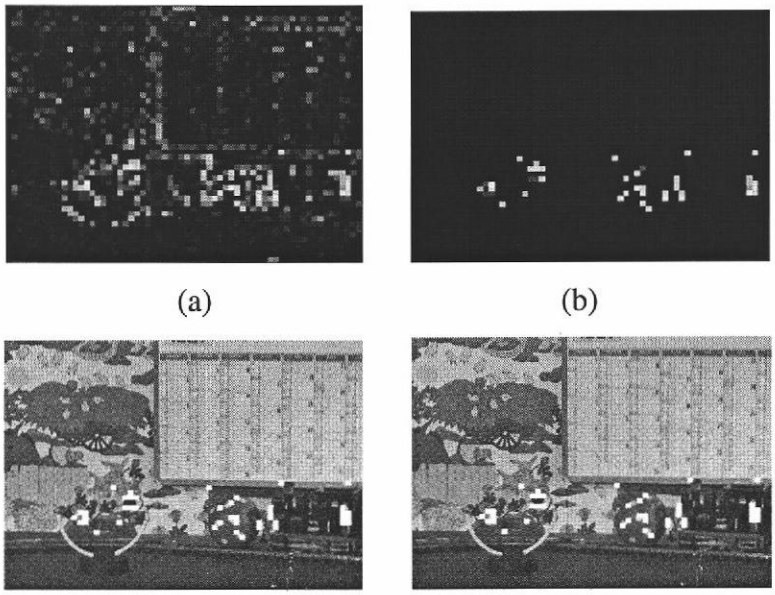
		Images	
		tabletennis	lab1
Background motions	1 pixel y-axis	O	O
	2 pixel y-axis	X	X
	1 pixel x-axis, 1 pixel y-axis	X	X
	1 pixel x-axis	O	O
	2 pixel x-axis	X	X

5.4 Low resolution (noise images)

In figures 22 through 30, the threshold value depends on the noise. The threshold value of figure 22 to figure 24 is approximately 25.80. The threshold value of figure 25 to figure 27 is approximately 19.33. The threshold value of figure 28 to figure 30 is approximately 22.80.

We can see that the proposed algorithm works well even for low S/N ratio (see figure 22 through 30). In case of the "caltrain", the algorithm detects object motion well for less than 10 dB SNR. For 5dB and 3dB SNR of "caltrain", there are errors, because the "caltrain" images include the motion of object, background motion, and noise. These results show that noise is reduced very effectively by the wavelet transform. To further improve these results, more pre-processing of the images is needed.

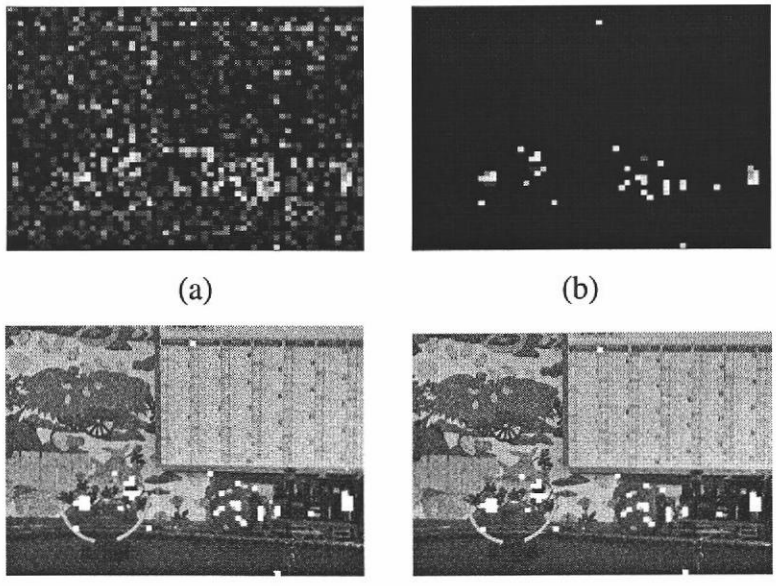
In the case of "tabletennis" and "lab1", the proposed algorithm worked well under 3dB of noise level. These results follows the wavelet transform noise cancellation properties.



(a) (b)

(c) (d)

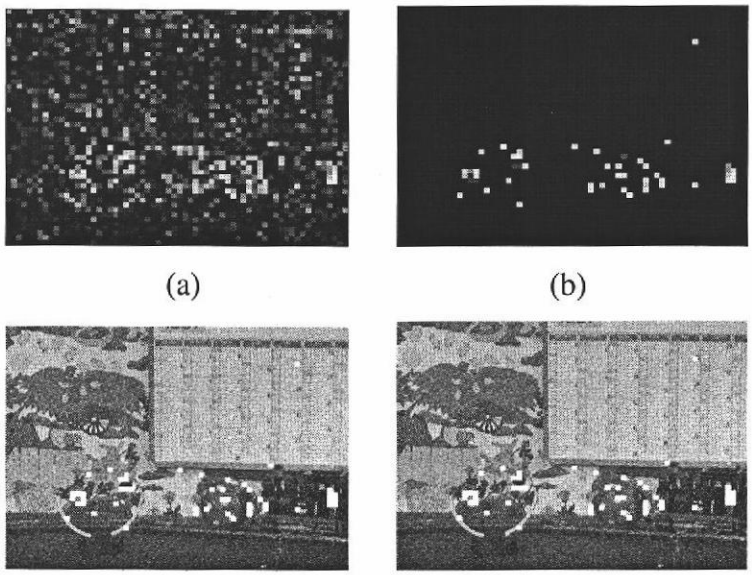
Figure 22. The result of motion detection at the caltrain with 10dB of noise (a) difference after three stage DWT (64 x 50), (b) after filtering (64 x 50), (c) applied detected motion to the first frame (512 x 400), (d) applied detected motion to the second frame (512 x 400)



(a) (b)

(c) (d)

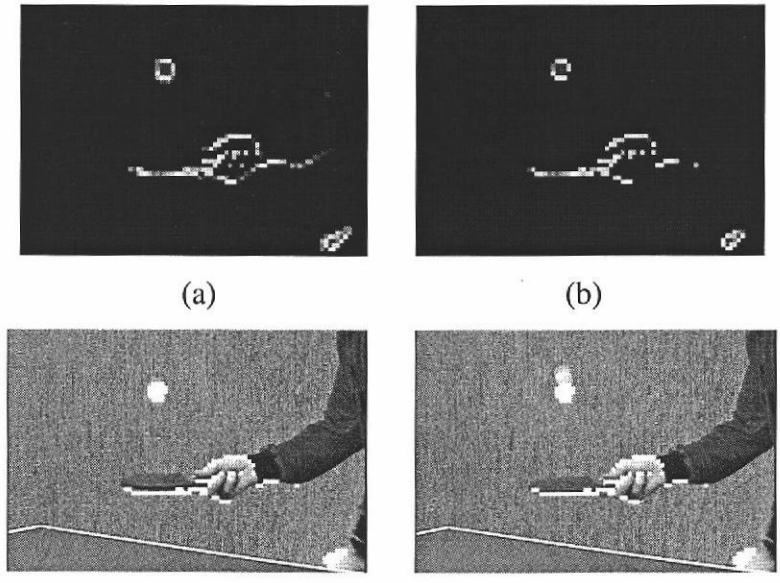
Figure 23. The result of motion detection at the caltrain with 5dB of noise (a) difference after three stage DWT (64 x 50), (b) after filtering (64 x 50), (c) applied detected motion to the first frame (512 x 400), (d) applied detected motion to the second frame (512 x 400)



(a) (b)

(c) (d)

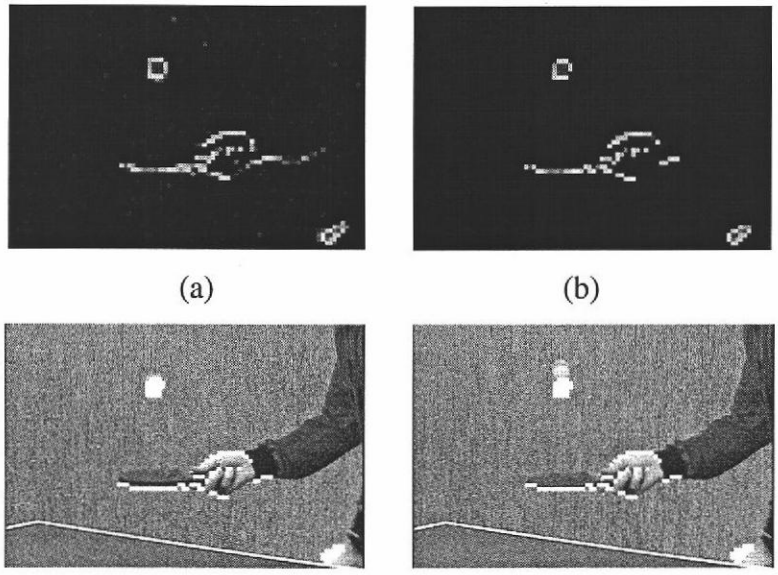
Figure 24. The result of motion detection at the caltrain with 3dB of noise (a) difference after three stage DWT (64 x 50), (b) after filtering (64 x 50), (c) applied detected motion to the first frame (512 x 400), (d) applied detected motion to the second frame (512 x 400)



(a) (b)

(c) (d)

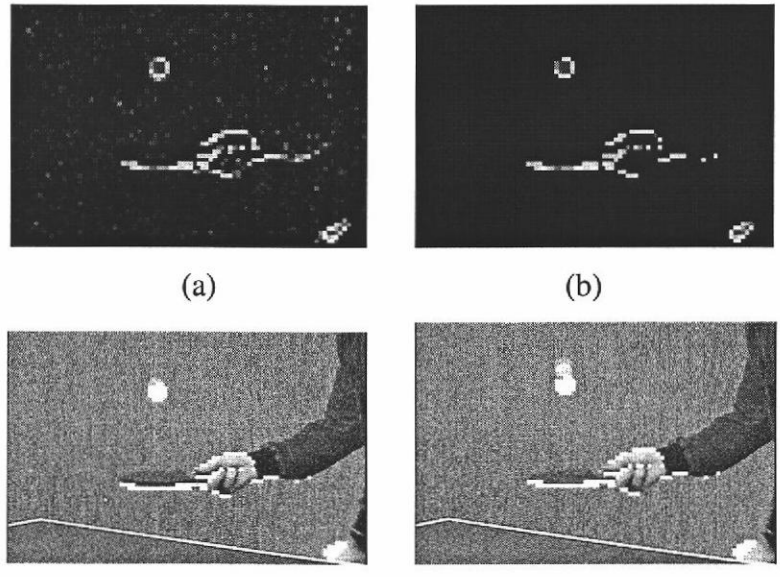
Figure 25. The result of motion detection at the tabletennis with 10dB of noise (a) difference after three stage DWT (90 x 60), (b) after filtering (90 x 60), (c) applied detected motion to the first frame (720 x 480), (d) applied detected motion to the second frame (720 x 480)



(a) (b)

(c) (d)

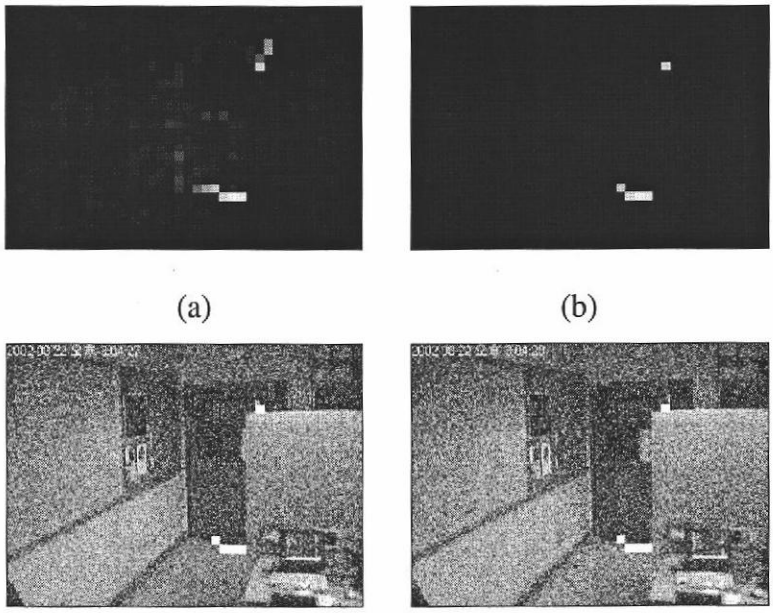
Figure 26. The result of motion detection at the tabletennis with 5dB of noise (a) difference after three stage DWT (90 x 60), (b) after filtering (90 x 60), (c) applied detected motion to the first frame (720 x 480), (d) applied detected motion to the second frame (720 x 480)



(a) (b)

(c) (d)

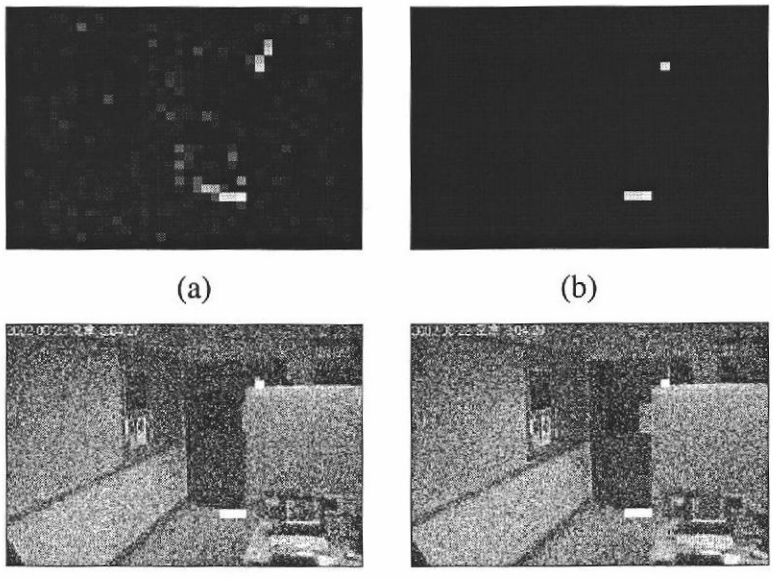
Figure 27. The result of motion detection at the tabletennis with 3dB of noise (a) difference after three stage DWT (90 x 60), (b) after filtering (90 x 60), (c) applied detected motion to the first frame (720 x 480), (d) applied detected motion to the second frame (720 x 480)



(a) (b)

(c) (d)

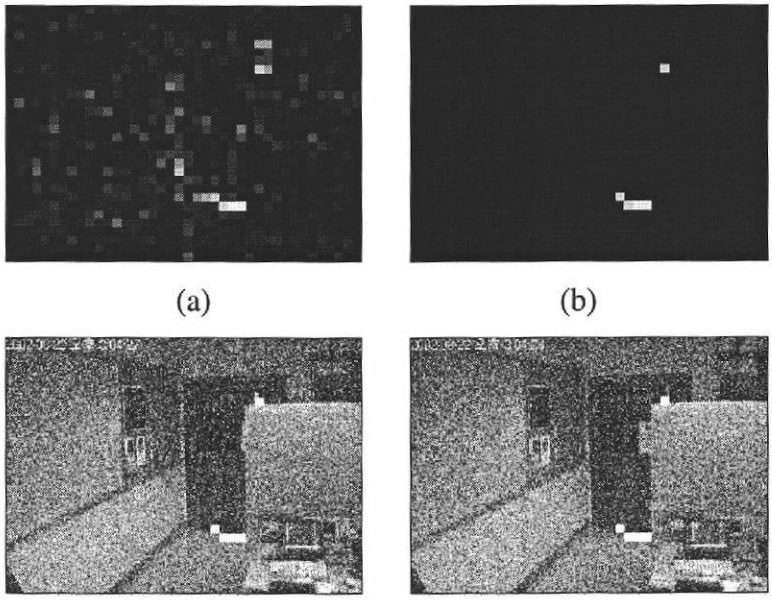
Figure 28. The result of motion detection at the lab1 with 10dB of noise (a) difference after three stage DWT (40 x 30), (b) after filtering (40 x 30), (c) applied detected motion to the first frame (320 x 240), (d) applied detected motion to the second frame (320 x 240)



(a) (b)

(c) (d)

Figure 29. The result of motion detection at the lab1 with 5dB of noise (a) difference after three stage DWT (40 x 30), (b) after filtering (40 x 30), (c) applied detected motion to the first frame (320 x 240), (d) applied detected motion to the second frame (320 x 240)



(a) (b)

(c) (d)

Figure 30. The result of motion detection at the lab1 with 3dB of noise (a) difference after three stage DWT (40 x 30), (b) after filtering (40 x 30), (c) applied detected motion to the first frame (320 x 240), (d) applied detected motion to the second frame (320 x 240)

Table 2. Summarizes the motion detection results presented in figures 22 to 30
(O : no error, X : error)

		Images		
		caltrain	tabletennis	lab1
Noise level (S/N)	10 dB	O	O	O
	5 dB	X	O	O
	3 dB	X	O	O

6 Conclusions

The tests performed in this project illustrate the proposed algorithm robustness with respect to camera motion, small background motion (1 pixel) and illumination change. Also this algorithm exhibits very good noise cancellation properties following the well known insensitivity to noise of the wavelet transform. According to Tables 1 and 2, the proposed algorithm is sensitive to background motion, but not sensitive to the noise.

For the difficult test of the "caltrain" sequence, which has many moving objects, background motion and additive noise, this algorithm detects the true object motion well.

A simple motion detection method generates many errors compared to the proposed algorithm. For example Figure 9 shows that a simple detection can't distinguish between object and background motion. Figure 11 shows how simple detection signals object motion even if none is present.

Based on experimental results, we can conclude that the proposed algorithm has better performance than a standard motion detection method.

The proposed algorithm can be used to detect motion efficiently as a part of video surveillance systems under unfavorable circumstance like low SNR, small camera movement, and changes in illumination.

Bibliography

1. Osman N. Ucan, Serhat Seker, A. Muhittin Albora, and Atilla Ozmen, 'Separation of magnetic fields in geophysical studies using a 2-D multi-resolution Wavelet analysis approach', Journal of The Balkan Geophysical Society, Vol. 3, No. 3, August 2000, p. 53-58.

2. K. Subramaniam, S. S. Dlay, and F. C. Rind, 'Wavelet Transforms for Use in Motion Detection and Tracking Application', Image Processing and its Applications, Conference Publication No. 465, IEEE 1999.

3. Mokoto Kobayashi and Michio Ohta, 'On detection of Motion by Wavelet Transform for an Acoustic Visual Aid System', Proceedings of the 35th Conference on Decision and Control, Kobe, Japan, December 1996.

4. MATLAB Wavelet Toolbox User's Guide.

5. Letang, J.M., Rebuffel, V., and Bouthemy, P., 'Motion detection based on a temporal multiscale approach', Pattern Recognition, 1992 . Vol.1. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on , 30 Aug-3 Sep 1992, p. 65 -68

6. Papageorgiou, C.P., Oren, M., and Poggio, T., 'A general framework for object detection', Computer Vision, 1998. Sixth International Conference on, 4-7 Jan 1998, p. 555 -562

7. Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., and Poggio, T., 'Pedestrian detection using wavelet templates', Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on, 17-19 Jun 1997, p. 193 -199

8. Papageorgiou, C. and Poggio, T., 'A pattern classification approach to dynamical object detection', Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, Volume: 2, 1999, p. 1223 -1228 vol.2

9. Junwen Wu, Xuegong Zhang, and Jie Zhou, 'Vehicle detection in static road images with PCA-and-Wavelet-Based classifier', Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE, 2001, p. 740 -744

10. Hong-Ying Zhang and Bao-Zong Yuan, 'A multiresolution image matching algorithm for moving object detection by wavelets', Speech, Image Processing and Neural Networks, 1994. Proceedings, ISSIPNN '94., 1994 International Symposium on , 13-16 Apr 1994, p. 276 -279 vol.1

11. Yahampath, P., Chen, H., and Kinsner, W., 'Detection of moving objects in facial image sequences', Electrical and Computer Engineering, 1998. IEEE Canadian Conference on, Volume: 1, 24-28 May 1998, p. 369 -372 vol.1

12. Yansun Xu, Weaver, J.B., Healy, D.M., and Jian Lu, 'Wavelet transform domain filters: a spatially selective noise filtration technique', Image Processing, IEEE Transactions on, Volume: 3 Issue: 6, Nov 1994, p. 747 -758

13. Yu-Fei Ma and Hong-Jiang Zhang, 'Detecting Motion Object by Spatio-Temporal Entropy', Microsoft Research, China, http://research.microsoft.com/asia/dload_files/group/mcomputing/ICME01ma-4th.pdf

14. http://ai-depot.com/Robotics/108.html

15. A. K. Peters, 'The world According to Wavelets', Wellesley, 1996.

16. P. Wojtaszcyk, 'A Mathematical Introduction to Wavelets', Cambridge University Press, 1997.

17. Edward Aboufadel and Steven Schlicker, 'Discovering Wavelets', John Wiley & Sons Inc. 1999

18. Y. Y. Tang, L. H. Yang, J. Liu, and H. Ma, 'Wavelet Theory and Its Application to Pattern Recognition', World Scientific Publishing Co. 2000.

Appendix

Appendix A. MATLAB code

```
clear all; close all;

nfig0 = 1;

a1 = imread('caltrain000', 'tif');   %400 x 512 = 204800
a2 = imread('caltrain001', 'tif');   %400 x 512 = 204800

[A1,map1] = gray2ind(a1, 255);
[A2,map2] = gray2ind(a2, 255);

A11 = double(A1) +1;
A21 = double(A2) +1;

nbcol1 =  size(map1,1);
nbcol2 =  size(map2,1);

cod_X1 = wcodemat(A11,nbcol1);
cod_X2 = wcodemat(A21,nbcol2);

% Perform one step decomposition.
[ca11,ch11,cv11,cd11] = dwt2(A11,'db1');
[ca21,ch21,cv21,cd21] = dwt2(A21,'db1');

% Image coding.
    cod_ca11 = wcodemat(ca11,nbcol1);
    cod_ch11 = wcodemat(ch11,nbcol1);
    cod_cv11 = wcodemat(cv11,nbcol1);
    cod_cd11 = wcodemat(cd11,nbcol1);
    figure(nfig0);nfig0=nfig0+1;
    image([cod_ca11,cod_ch11;cod_cv11,cod_cd11]);

    cod_ca21 = wcodemat(ca21,nbcol2);
    cod_ch21 = wcodemat(ch21,nbcol2);
    cod_cv21 = wcodemat(cv21,nbcol2);
    cod_cd21 = wcodemat(cd21,nbcol2);
    figure(nfig0);nfig0=nfig0+1;
    image([cod_ca21,cod_ch21;cod_cv21,cod_cd21]);

% Perform second step decomposition :
% decompose approx. cfs of level 1.
    [ca12,ch12,cv12,cd12] = dwt2(ca11,'db1');
    [ca22,ch22,cv22,cd22] = dwt2(ca21,'db1');

% Code ca2, ch2,cv2 and cd2.
    cod_ca12 = wcodemat(ca12,nbcol1);
    cod_ch12 = wcodemat(ch12,nbcol1);
    cod_cv12 = wcodemat(cv12,nbcol1);
    cod_cd12 = wcodemat(cd12,nbcol1);
    figure(nfig0);nfig0=nfig0+1;
```

```
    image([[cod_ca12,cod_ch12;cod_cv12,cod_cd12],cod_ch11;cod_cv11,cod_cd11]);

    cod_ca22 = wcodemat(ca22,nbcol2);
    cod_ch22 = wcodemat(ch22,nbcol2);
    cod_cv22 = wcodemat(cv22,nbcol2);
    cod_cd22 = wcodemat(cd22,nbcol2);
    figure(nfig0);nfig0=nfig0+1;
    image([[cod_ca22,cod_ch22;cod_cv22,cod_cd22],cod_ch21;cod_cv21,cod_cd21]);

% Perform third step decomposition :
% decompose approx. cfs of level 2.
    [ca13,ch13,cv13,cd13] = dwt2(ca12,'db1');
    [ca23,ch23,cv23,cd23] = dwt2(ca22,'db1');

    cod_ca13 = wcodemat(ca13,nbcol1);
    figure(nfig0);nfig0=nfig0+1;
    image(cod_ca13);
    cod_ca23 = wcodemat(ca23,nbcol2);
    figure(nfig0);nfig0=nfig0+1;
    image(cod_ca23);

%% ==================================

    cod_cadi1 = abs(cod_ca11 - cod_ca21);
    cod_cadi2 = abs(cod_ca12 - cod_ca22);
    cod_cadi3 = abs(cod_ca13 - cod_ca23);

% DWT2 first and then difference
    figure(nfig0);nfig0=nfig0+1;
    image(cod_cadi1);
    figure(nfig0);nfig0=nfig0+1;
    image(cod_cadi2);
    figure(nfig0);nfig0=nfig0+1;
    image(cod_cadi3);

% filtering
    k= 1;
    for j = 1 : 64      % 64 128 256
      for i = 1 : 50    %50 100 200
        if cod_cadi3(i,j) <25.4
          cod_cadi3(i,j) = 0;
        end
        if cod_cadi3(i,j) >5        % Find motion area
          p(k) = i ;
          q(k) = j ;
          k = k + 1;
        end
      end
    end

% Applied detected motion to original image
    if k ~= 1
      for l = 1 : k -1
```

```
      for j = (q(l) * 8 - 7) : (q(l) * 8)
         for i = (p(l) * 8 - 7) : (p(l) * 8)
            A11(i,j) = 255;    % white
            A21(i,j) = 255;    % white
         end
      end
   end
end

figure(nfig0);nfig0=nfig0+1;
image(A11);
figure(nfig0);nfig0=nfig0+1;
image(A21);

B1 = uint8(A11 - 1);
b1 = ind2gray(B1, map1);
figure(nfig0);nfig0=nfig0+1;
imshow(b1);

B2 = uint8(A21 - 1);
b2 = ind2gray(B2, map2);
figure(nfig0);nfig0=nfig0+1;
imshow(b2);
```