I.J. Intelligent Systems and Applications, **2012, 9, 1-14**
Published Online August 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2012.09.01

Difference of the Absolute Differences – A New Method for Motion Detection

Khalid Youssef, Peng-Yung Woo
Department of Electrical Engineering, Northern Illinois University, Dekalb, IL 60115, USA
E-mail: pwoo@niu.edu

***Abstract*—** This article presents a new method, which reduces costs and processing time for spatial object motion detection by focusing on the bare-hand motion that mimics computer mouse functions to allow the user to move the mouse pointer in real-time by the motion of his/her hand without any gloves worn, any object carried, or any key hit. In this article, the study of this topic is from the viewpoint of computer vision and image processing. The principals of the difference of the absolute differences (DAD) are investigated. A new method based on the DAD principles, which is conceptually different from all the existing approaches to spatial object motion detection, is developed and applied successfully to the bare-hand motion. The real-time implementation of the bare-hand motion detection demonstrates the accuracy and efficiency of the DAD method.

***Index Terms*—** Bare-Hand Motion, Computer Vision, Difference of the Absolute Differences Method, Human-Machine Interaction, Image Processing, Spatial Object Motion Detection

I. Introduction

Motion detection and further tracking of general spatial articulated objects including the human body and limbs is a research area that has been attracting more and more attention. Recent literature reveals a lot of studies in this area including the proposed wearable target method [1]-[2], the sensor method [3], the electrostatic field method [4], the multiple-camera method [5]-[7] and so on.

Hand motion detection has become a key topic in the research area of spatial object motion due to its potential in human-machine interaction. In this study, a hand gesture is defined as a dynamic movement referring to a sequence of hand postures over a short time span. A hand posture refers to a static hand pose without any involvement of movements. The hand gesture recognition process is realized by “building up out of a group of hand postures in various ways of composition” [8]. According to this definition, hand motion detection can be considered as a hand gesture recognition process.

Hardenberg and Bedard [9] have traced the research on vision-based hand gesture recognition and tracking back to 1991. They claim that up to 2001, the time of their publication, there has not been any dominating hand motion detection and tracking method. Furthermore, it is claimed that most of the proposed systems have problems in case of lighting condition and background clutter changes. Also, none of the presented work provides a robust motion detection and tracking method for a rapid hand motion. Triesch and Malsburg [10] use wavelets. Though robust results in classifying hand postures against complex backgrounds are claimed, the calculations proposed in their approach cannot be performed in real-time. Ware and Balakrishnan [11] use color segmentation. However, the user has to wear special colored gloves. Sato *et al.* [12] use infrared segmentation with expensive hardware equipments. Segen [13] uses contours. Restrictive background conditions are required for their approach. Laptev and Lindeberg [14] use Blob-models. This approach requires an explicit setup stage before starting the tracking. Crowley *et al.* [15], and O'Hagan and Zelinsky [16] use correlation, but a maximum speed of the hand motions is set.

Recently, Garg *et al.* [8] provide a review on the state-of-the-art hand motion detection and tracking studies. None of the existing approaches is believed to solve the problems raised in [9]. Stenger *et al.* [17] propose an approach formulated within a Bayesian framework, which is extremely computationally expensive. Bretzner *et al.* [18], Sanchez-Nietsen *et al.* [19], and Stenger [20] propose the discernment of the skin color of the user. The drawback of this approach is the mistaken discernment of the skin due to other skin-color-like objects as well as lighting condition changes. Lienhart and Maydt [21], Barczak and Dadgostar [22], Chen *et al.* [23], and Wang and Wang [24] work on local invariant features. Though methods based on this seem promising, they are computationally costly which leads to expensive equipments.

A bare-hand human-machine interaction that is essentially composed of the bare-hand motion detection and tracking is important in many areas. The key advantages of bare-hand interaction in the areas such as virtual environment, smart surveillance and medical systems are, just to mention a few, the elimination of physical contact, the reduction of space occupation, the increase in device durability, the reduction of equipment cost, the expansion of user applications and so on. Several recent publications on bare-hand interaction are focusing on its use in video games [7], [25]-[29]. Video game giants such as Nintendo, Sony's Playstation, and Microsoft's Xbox are using this technology as the basis of the new trend in video games.

With the development of the computer technology and applications, the current facilities for human-computer interaction such as the keyboard, mouse, and pen do not meet the increasing demands [8]. Evidently, the human hands used directly as an input device without any gloves worn, any object carried, or any key hit, in particular, provide natural human-computer interaction, and in many cases are much more practical than the conventional input devices. The bare-hand interaction here is essentially composed of the bare-hand motion detection as well as the virtual clicks that are the results of hand gestures. In the bare-hand motion the user moves the mouse pointer, and in the virtual clicks the user performs the similar functions as those performed by the right and/or left clicks of a conventional computer mouse, both in real-time by his/her different hand gestures.

As well-known, an appealing bare-hand motion detection method must satisfy two main requirements, i.e., the ability to run in real-time and the affordability in the sense of less expensive equipments. If these requirements were not satisfied, the method would not be practical or accessible to the average users. There have been some methods proposed that satisfy either one of the two requirements. However, the challenge is to satisfy both of them. All the existing bare-hand motion detection methods are systematically based on the same approach that is to track the hand motion by using consecutive hand postures. Each method tries to achieve hand posture recognition in a different way.

A new approach to spatial object motion detection is proposed in this article, and a new method based on this approach is developed. Instead of achieving object posture recognition, the proposed approach is to detect the motions of the object by determining the motion direction directly without relying on the positions. The developed method encompasses an innovative technique that permits vast enhancements in the processing speed, detection accuracy, and performance robustness by transforming 2D image processing to 1D signal processing. Also, it is extremely cost effective, since it can operate with only one commercial video camera without any additional sensor or other equipment. To overcome the deficiencies in the existing approaches to the bare-hand motion detection, the developed method is applied to a system design with satisfactory results.

The organization of the rest of the article is as follows. The new concepts and theory of the developed method is given in Section II. The graphic tool that works for the new concepts is presented in Section III. In Section IV, the system design of a bare-hand motion detection application based on the developed method is described. Experimental results are presented in Section V, followed by conclusions and observations in Section VI.

II. The Method of the Difference of the Absolute Differences

A. *The difference of the absolute differences v.s. the sum of the absolute differences*

The prevailing way of thinking in object motion detection is to determine the positions of an object at different time instances. The position of the object at a certain time instant can be determined from one frame alone, though, at most of the time, two or more consecutive frames are also used. Once the positions of the object are determined at two time instances, the motion direction between the two time instances can be obtained. However, in a lot of practical object motion detection applications, the exact position of the object is not a matter of concern. For example, the actual physical position of a regular computer mouse on the table is irrelevant to the operation of the computer. It is the change of the mouse positions that directs the pointer on the computer screen. Thus, the key idea in the developed method is to obtain the motion direction directly without determining the positions of the object. The information loss of the exact object position in this method is worthily compensated for by the enhancements in the processing speed, detection accuracy, and performance robustness. The developed method is named the Difference of the Absolute Differences (DAD), which requires three consecutive frames to determine the motion direction of an object in a certain direction. In fact, this is similar to working with the second-order derivative of a function.

In image processing, the method of the Sum of the Absolute Differences (SAD) is widely used for motion detection and video compression [30]-[32]. The SAD method basically performs a pixel-by-pixel comparison between two frames by the calculation of the absolute difference between pixels. This process is repeated for multiple consecutive frames, after which the absolute differences are summed up to create the final SAD 2D matrix. In a typical motion detection application, a number of consecutive frames are compared to a background frame with each frame divided into a number of smaller blocks, and the exceeding of a set threshold in a certain location of the SAD matrix indicates some motion in the corresponding block in the background frame. Though the SAD method is effective for motion detection, it performs poorly when used in a real-time application such as the hand motion detection to determine the position of the hand because of many factors, namely, the dynamic background, high noise, and other moving objects. Some important information such as the motion direction is lost in the process of creating the SAD matrix, regardless of the number of the frames used. Furthermore, for robust position detection, a very high frame rate is needed when using the SAD method, which requires expensive cameras and leads to exhaustive computation, which, in turn, prevents the application from running in real-time on personal computers. The SAD and DAD images of a hand are shown in Fig. 1.

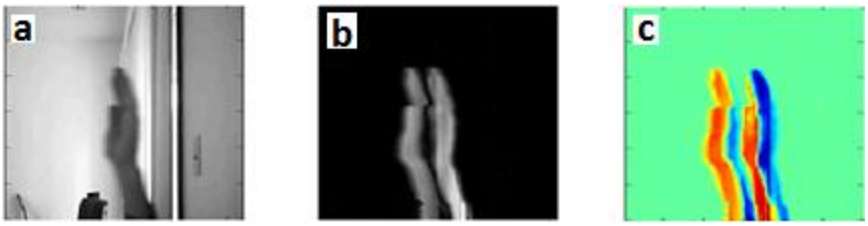


Fig.1 (a) Ordinary image (b) SAD image (c) DAD image

B The concept of the difference of the absolute differences

Consider a gray-scaled digital image, where each pixel in the image varies between 0 (white) and 255 (black). $l(t)$ is the luminance function of a pixel with respect to time. Assuming the time difference between two consecutive frames is a unity, the luminance difference is then effectively the first-order derivative of $l(t)$ between the two frames. Fig. 2 shows the luminance function $l(t)$ of a certain pixel over an interval of 200 frames as well as $l'(t)$, $g(t)$ and $f(t)$, which are defined by (1)-(3), respectively.

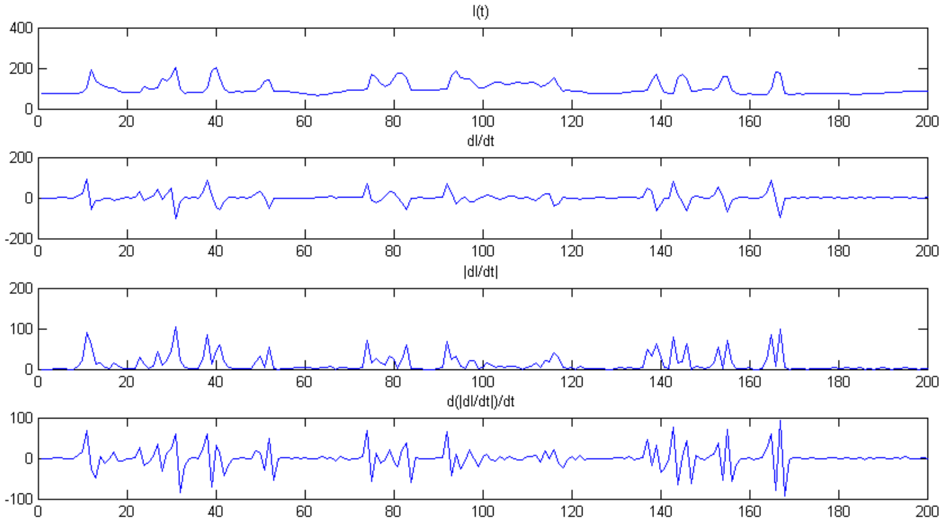


Fig.2 Luminance function $l(t)$ of a pixel and $l'(t) = dl/dt$, $g(t)=|dl/dt|$, and $f(t)=dg/dt$

$$l'(t) = dl/dt = \lim_{t\to 0}\frac{l(t+\Delta t)-l(t)}{\Delta t} \qquad (1)$$

$$g(t) = |dl/dt| = \left|\lim_{t\to 0}\frac{l(t+\Delta t)-l(t)}{\Delta t}\right| \qquad (2)$$

$$f(t) = dg/dt = \lim_{t\to 0}\frac{g(t+\Delta t)-g(t)}{\Delta t} \qquad (3)$$

The function $l(t)$ indicates the luminance in the pixel, where a steady value corresponds to an absence of motion, and an abruptly changing value corresponds to a motion activity. Taking the first-order derivative of $l(t)$ locates its peaks and valleys (local maximums and minimums), which correspond to the abrupt changes in luminance, i.e., the motion activities. The first-order derivative, however, cannot determine the motion direction, since it is unable to differentiate two possible scenarios that might cause abrupt changes in the luminance in a pixel, i.e., an object moving toward and covering the pixel so that the pixel value now corresponds to the object luminosity, and an object moving away and not covering the pixel anymore so that the pixel value now corresponds to the background luminosity. It can be determined from $l'(t)$ that at the location where a motion activity takes place, there is either an abrupt change from a high luminosity to a low luminosity, or *vice versa*. Since the relation between the object luminosity and the background luminosity is not constrained, i.e., the background can have either a higher or a lower luminosity compared to the object, $l'(t)$ cannot determine whether the motion activity corresponds to the first or the second scenario.

To overcome this difficulty, $f(t)$ is used. Since $l'(t)$ determines the locations of the moving object, taking the derivative of $l'(t)$, which is the second-order derivative of $l(t)$, gives the change in the locations of the moving object. Furthermore, since the locations of the moving object are not related to the sign of $l'(t)$, the derivative of the absolute values of $l'(t)$ are then used instead, which is preferred for practical reasons such as keeping the values of $l'(t)$ in the original range of 0 to 255.

The luminance functions of 120 pixels in a full row over an interval of 200 frames are shown in Fig. 3. The functions $f(t)$ of the same pixels over the same interval are shown in Fig. 4. In Fig. 3 and Fig. 4, the 2D graphs on the right are the top view of the 3D graphs. A plot of the values of function $f(t)$ of all pixels in one frame (i.e., at a specific time or when t is a constant) is shown in Fig. 5, which shows the relation between motion activities in adjacent pixels in one frame. This enables the determination of the motion direction at that instant of time.

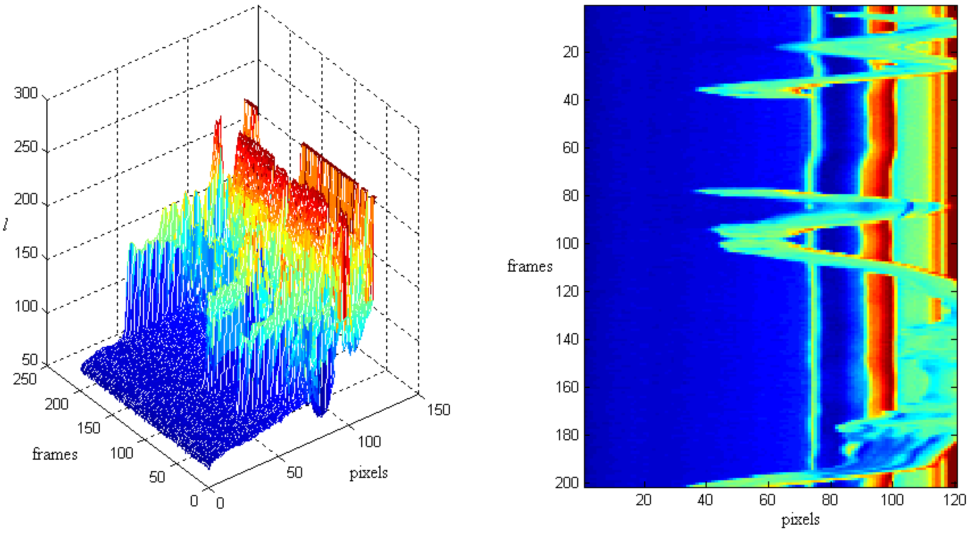


Fig. 3 Luminance functions $l(t)$ of pixels in a full row

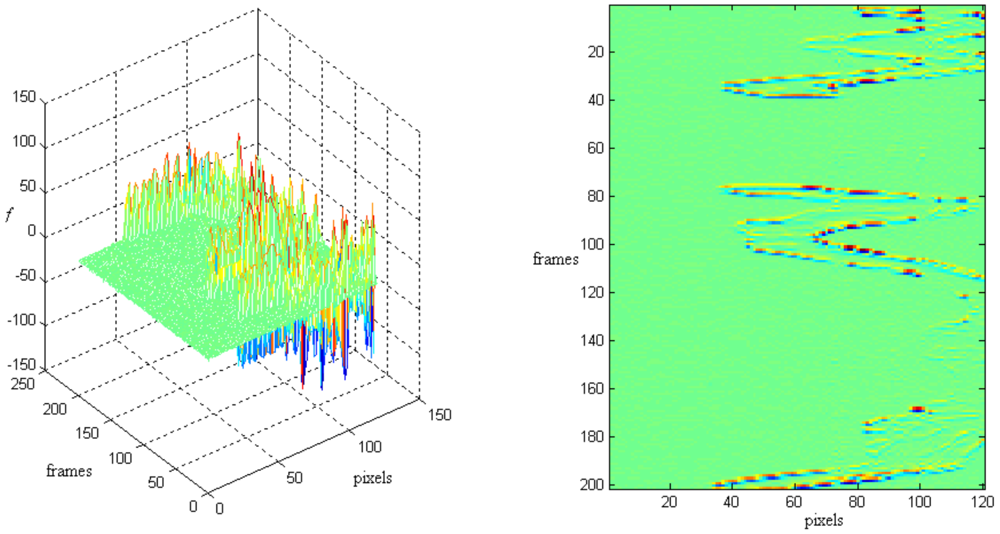


Fig. 4 Function $f(t)$ of pixels in a full row

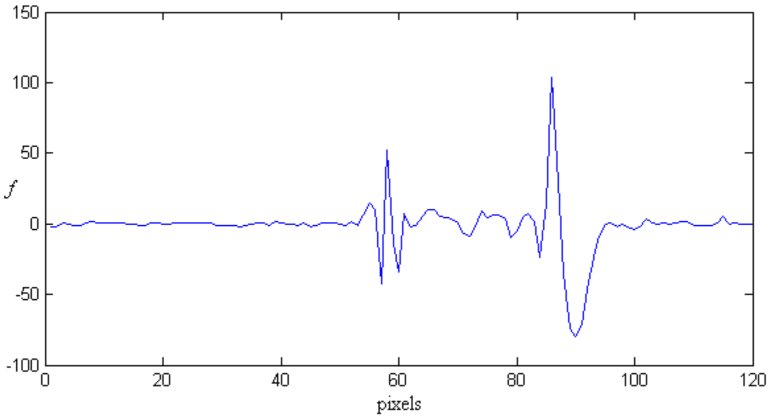
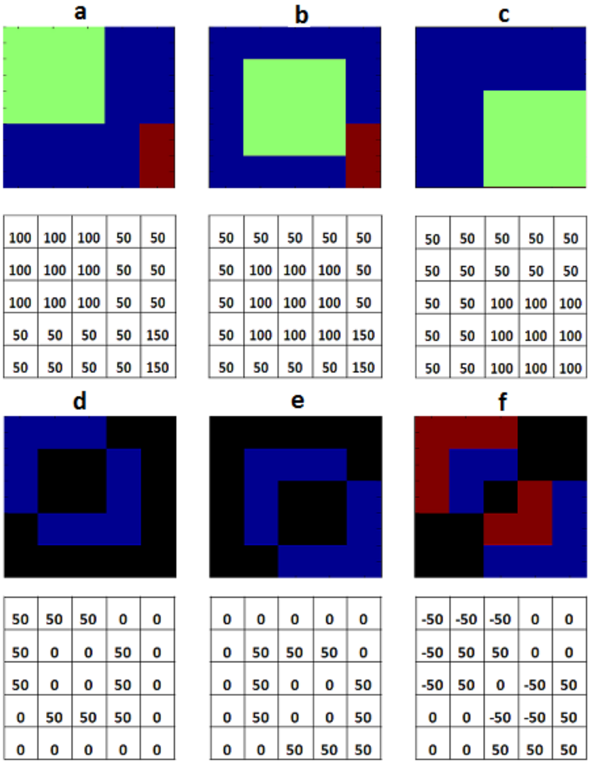


Fig. 5 Values of function $f(t)$ of all pixels in one frame

C. *The generalization of the concept of the difference of the absolute differences*

The concepts above can be generalized to regions composed of more than one row. For example, to approximate the motion of a whole frame in the horizontal direction, the $f(t)$ of each pixel is first calculated, and then summations along the columns in the vertical direction are performed. This is clarified further in Fig. 6, where frame d is obtained by calculating the absolute difference $I'(t)$ between frame b and frame a, while frame e the absolute difference between frame c and frame b. Finally, frame f is obtained by calculating the difference between frame e and frame d. The pixel values in the frames are displayed in tables under them. Note that in the absolute difference frames d and e, there are two types of pixels. The pixels with a zero value indicate that there has been no change in the corresponding pixels between the two frames, while the pixels with a positive value indicate a change. On the other hand, there are three types of pixels in the DAD frame f, namely, zero, positive and negative. A negative pixel in a DAD frame indicates that the object occupying that position in the previous frame is not occupying it anymore, i.e., moving away from that position. The opposite is true for a positive pixel. Thus, it is concluded that the motion direction is from negative pixels to positive pixels in a DAD frame. By adding all the rows in the DAD frame, i.e., performing summations along the columns in the vertical direction, into one vector as shown in Fig. 7, the sequence "–, +, –, +" indicates that the object moves from left to right in the horizontal direction.



100	100	100	50	50
100	100	100	50	50
100	100	100	50	50
50	50	50	50	150
50	50	50	50	150

50	50	50	50	50
50	100	100	100	50
50	100	100	100	50
50	100	100	100	150
50	50	50	50	150

50	50	50	50	50
50	50	50	50	50
50	50	100	100	100
50	50	100	100	100
50	50	100	100	100

50	50	50	0	0
50	0	0	50	0
50	0	0	50	0
0	50	50	50	0
0	0	0	0	0

0	0	0	0	0
0	50	50	50	0
0	50	0	0	50
0	50	0	0	50
0	0	50	50	50

-50	-50	-50	0	0
-50	50	50	0	0
-50	50	0	-50	50
0	0	-50	-50	50
0	0	50	50	50

Fig. 6 Method of the Difference of Absolute Differences (DAD)

-150	50	0	-50	150

Fig. 7 DAD superposition

D. *Mathematical approximation for computation efficiency*

In order to increase the computation efficiency, some mathematical approximation can be taken to simplify the calculation of the difference of the absolute differences for the consecutive frames as described in Fig. 6 and Fig. 7. As is to be shown, when some reasonable conditions are applied, the rows in the original frames a, b and c can be summed into vectors first, respectively, and then the absolute differences and therefore the difference are calculated in one dimension rather than performing calculations in two dimensions first and then the summation. This shuffling of steps enhances the total processing speed.

Two assumptions are made. First, the color of the moving object is assumed to be almost uniform, i.e., pixels representing the object have about same values. Second, since the size of the object is small relative to that of the different regions of the background, the color of each region of the background is also assumed to be almost uniform. In bare-hand motion detection, the first assumption is evidently always true, while the second is true for most cases.

The entries of the image matrices obtained from consecutive frames are mainly composed of two parts that represent the background and the moving object, respectively. For the frame rates that are high enough, the background is almost unchanged between two consecutive frames, and thus the pixels and therefore the matrix entries representing the background are almost identical. Similarly, the pixels and therefore the matrix entries representing the moving object are also almost unchanged, though undergo a position shift. The above results presents the fact that the entries of two consecutive matrices are identical everywhere except at the location of the moving object. Mathematically, for the matrix entries of the two consecutive images, we must have either $b_{ij}=a_{ij}$ or $b_{ij}=a_{i+y,j+x}$, where a_{ij} and b_{ij} are the entries of the image matrices A and B, respectively.

With the conditions above, the approximation of the difference of the absolute differences for the consecutive frames can be achieved. Without loss of generality, this is illustrated mathematically by using 3×3 image matrices, where entries h are the pixels of the moving object, and entries d the pixels of the background. While the object is moving from the left to the right in the horizontal direction, the three consecutive image matrices are, respectively

$$A=\begin{bmatrix}a_{11}&a_{12}&a_{13}\\a_{21}&a_{22}&a_{23}\\a_{31}&a_{32}&a_{33}\end{bmatrix}=\begin{bmatrix}d&d&d\\h&d&d\\h&d&d\end{bmatrix}, \qquad (4)$$

$$B=\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}=\begin{bmatrix} d & d & d \\ d & h & d \\ d & h & d \end{bmatrix}, \qquad (5)$$

$$C=\begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}=\begin{bmatrix} d & d & d \\ d & d & h \\ d & d & h \end{bmatrix}. \qquad (6)$$

where the second steps in the equations are based on the two aforementioned assumptions. The DAD operation done in Fig. 6 and the summation done in Fig. 7 finally give g_{ij}, which are the vector entries in Fig. 7 by using (7).

$$g_{ij}=\sum_i\left(\left|c_{ij}-b_{ij}\right|-\left|b_{ij}-a_{ij}\right|\right)$$

$$=\sum_i\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & |d-h| & |h-d| \\ 0 & |d-h| & |h-d| \end{bmatrix}-\begin{bmatrix} 0 & 0 & 0 \\ |d-h| & |h-d| & 0 \\ |d-h| & |h-d| & 0 \end{bmatrix}\right)$$

$$=\sum_i\begin{bmatrix} 0 & 0 & 0 \\ -|d-h| & 0 & |h-d| \\ -|d-h| & 0 & |h-d| \end{bmatrix}$$

$$=\left[-2|d-h| \quad 0 \quad 2|h-d|\right] \qquad (7)$$

On the other hand, let us calculate a function g_{ij}^{*} as shown in (8).

$$g_{ij}^{*}=\left|\sum_i c_{ij}-\sum_i b_{ij}\right|-\left|\sum_i b_{ij}-\sum_i a_{ij}\right|$$

$$=\left|\left[3d \quad 3d \quad d+2h\right]-\left[3d \quad d+2h \quad 3d\right]\right|-\left|\left[3d \quad d+2h \quad 3d\right]-\left[d+2h \quad 3d \quad 3d\right]\right|$$

$$=\left|\left[0 \quad 2(d-h) \quad 2(h-d)\right]\right|-\left|\left[2(d-h) \quad 2(h-d) \quad 0\right]\right|$$

$$=\left[-2|d-h| \quad 0 \quad 2|h-d|\right] \qquad (8)$$

As seen, g_{ij}^{*} = g_{ij} when the two reasonable assumptions are imposed. This indicates that (8), in which the absolute differences and therefore the difference are calculated in one dimension so that the total processing speed is enhanced, can be used to replace (7) to obtain the same results as shown in Fig. 7.

In practice, even if the second assumption is not completely true, it does not have a noticeable effect on the performance of the system. Noise elimination and thresholding techniques further reduce the adverse effect. For some rarely happening severe occasions, e.g., the object is in a transition between two regions with high brightness difference, the approximation undergoes an elimination effect, and the corresponding motion step is ignored without having a noticeable effect on the overall detecting performance. On the other hand, depending on the nature of the application, there is always the option of not using the approximation at the cost of a slower processing speed. It is worth mentioning that even with the speed reduction due to no use of the approximate calculation (8), the DAD method is still much faster than the other existing methods.

III. The Dad Diagram And the Dad Plot

A. *Introduction of the DAD diagram and the DAD plot*

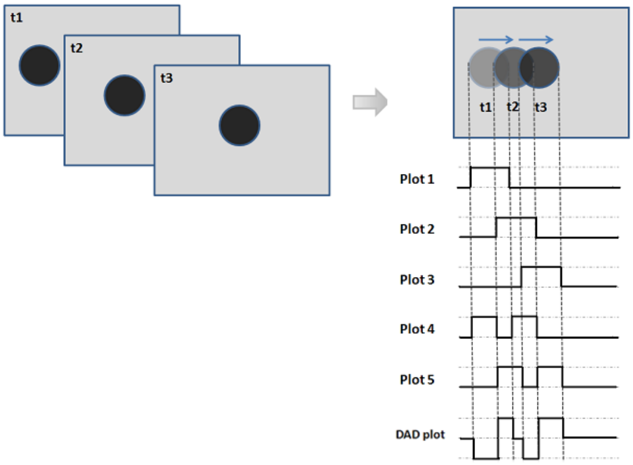


Fig. 8 DAD diagram and DAD plot

The DAD diagram used to study the behavior of a moving object is developed as a graphic tool. It is a graphical illustration that corresponds to (8). The detailed illustration of the basic DAD diagram is shown in Fig. 8. As seen, the DAD diagram is essentially generated from three consecutive frames. In the DAD diagram, the next frame is superimposed on the previous one and lines are extended from the boundaries of the object to obtain plots 1, 2 and 3. The values in plots 1, 2 and 3 are obtained by adding all the rows of frames 1, 2 and 3, respectively. In this basic DAD diagram, a threshold of value one is applied to plots 1, 2 and 3, where the pixel values of the background and the moving object are assumed to be zero and one, respectively. The DAD plot is the difference between plots 5 and 4, which, in turn, are the absolute difference between plots 2 and 1, and that between plots 3 and 2, respectively. The obtained DAD plot helps a 1D study of the motion of an object on a certain axis of the direction.

Fig. 9 gives some examples of the DAD diagrams. As seen, different number of overlaps in the three consecutive frames represents different patterns of the object motion, which present different number of peaks in their DAD plots.

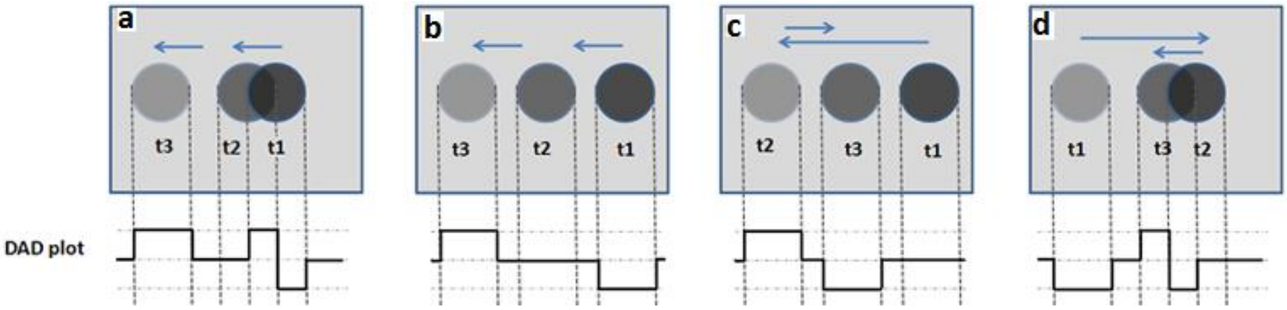


Fig. 9 Examples of the DAD diagram

The values of the variables defined in a DAD plot characterize the different motion patterns of the object in concern and become the key to an accurate motion detection. These variables are defined in Fig. 10, where there is the maximum number of peaks due to two overlaps of the three consecutive frames. Any other motion patterns that have only one or no overlaps give less than four peaks. If a DAD plot has more than four peaks, it indicates there is more than one object moving on the axis of the direction that corresponds to the DAD plot. For a four-peak DAD plot with a threshold applied there are ten main variables of interest. The variables *p1*, *p2*, *p3* and *p4* give the polarities of their corresponding peaks. The polarity is positive if the peak is above zero, while the polarity is negative if the peak is below zero. The variables *w1*, *w2*, *w3* and *w4* are the measures of the width of the corresponding peaks. The variable *d1* measures the length between the beginning of the signal and the beginning of the first peak. The variable *d2* measures the length between the first shift from nonzero to zero that follows the first change in polarity and the second shift from zero to nonzero that precedes the second change in polarity. The variables *Wa*, *Wb* and *D* are three composite variables that can be obtained from a combination of two or more main variables.

B. The use of the DAD plot variables

The variables obtained from the DAD plots are used to find the unique motion patterns of the object in concern, e.g., 1) *Wa* and *Wb* should have similar or close values. 2) *d2* should be less than *Wa* or *Wb*. 3) *Wa* and *Wb* should be in a certain range corresponding to the size range of the object in concern. Any signal that does not have such features is usually just noise.

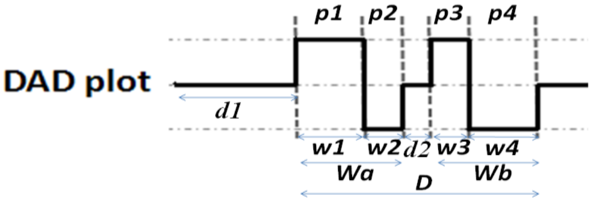


Fig. 10 Variables defined in a DAD plot

A complete presentation of all the possible motion patterns, not including those special cases of complete overlapping between frames, is given in Fig. 11, where column one and column six give valid motion patterns that are motions with no direction change. As seen, for a left-to-right motion there are four possible peak polarity sequences, namely, "-, +", "-, -, +", "-, +, +" and "-, +, -, +", while for a right-to-left motion there are another four, namely, "+, -", "+, -, -", "+, +, -" and "+, -, +, -". Evidently, the right-to-left motion patterns give a reversed order of the peak polarities as those for the left-to-right motion patterns. On the other hand, columns 2, 3, 4 and 5 in Fig. 11 present all the possible motion patterns with direction changes, which are regarded as invalid. Furthermore, in the case of two-peak DAD plots, some of the invalid motion patterns have the same polarity sequence as that of a valid motion pattern. This strongly suggests that in this case, the polarity sequence alone cannot differentiate between the valid and invalid motion patterns. Thus, the width of the peaks as well as the length between the two peaks is used in the determination of the validity of those motion patterns that are described by two-peak DAD plots. It is seen from the figure that those valid motion patterns that are described by two-peak DAD plots have the features 1) $w1 = w2 \equiv w$ 2) $d2 \geq w$. The special cases with complete overlapping between frames have unique polarity sequences that separate them from valid motion patterns. They are very unlikely to occur in practice though.

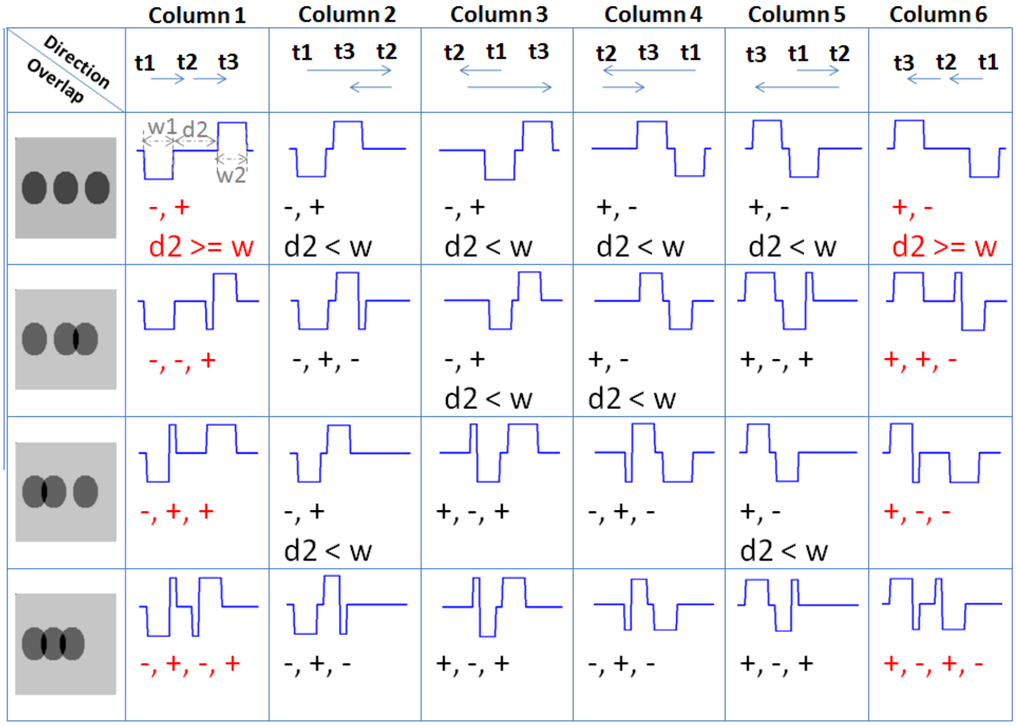


Fig. 11 Motion patterns and their DAD plots, not including special cases with complete overlapping between frames

C. The variations of the DAD diagram

There are variations for the DAD diagram. For example, removing the aforementioned threshold would preserve more information, and using a multi-segment DAD diagram would expose more details in each motion direction. Two variations for the DAD diagram that can be used for object motion detection are shown in Fig. 12, where the left figure shows a variation in which no threshold is applied and the detecting is performed in four axes of the directions (0°, 90°, 45°, -45°), and the right figure shows another variation in which a frame is divided into sectors and the object motion in each sector is plotted separately for two axes of the directions (0°, 90°). Note that in the case of the basic DAD plots shown in Fig. 11, motion patterns can be determined manually. However, for the complex DAD diagram variations, human inspection and determination of motion patterns becomes impossible. Pattern recognition methods are used in practice. One option is to train artificial neural networks to learn all the various motion patterns characterized by the DAD plots. The details of the above procedure are out of the scope of this article.

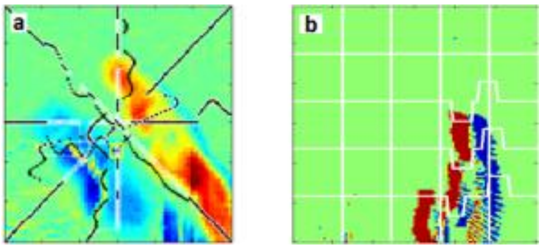


Fig. 12 Variations of the DAD diagram

An important aspect of the DAD method is that it can be viewed as an approach that allows for a trade-off between the processing speed and the accuracy of the object position estimation, whereas the conventional motion detection and tracking methods based on finding the exact position of an object at every frame are regarded as special cases of the DAD method such that its accuracy of the object position estimation attains the maximum in a sense.

In a single-segment DAD diagram, all pixels in one direction are summed to facilitate describing the general motion in the perpendicular direction. The object position is estimated as the region enclosed by the lines perpendicular to the coordinates that locate the beginning and the end of motion activities in two directions. This is demonstrated by an example shown in Fig. 13-a, where the shaded region gives the estimated object position.

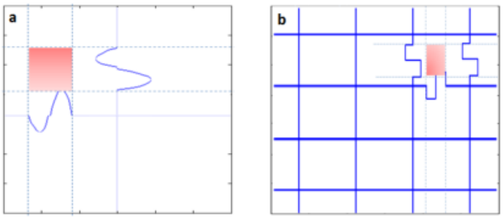


Fig. 13 Object position estimation in a DAD plot

In a multi-segment DAD diagram, the image is divided into several sectors and the motion in each sector is estimated. This gives a better accuracy of the object position estimation as shown in Fig. 13-b. In an $N \times N$ frame, if N sectors in each direction are used, the accuracy of the object position estimation is then the maximum since each sector is now narrowed down to one pixel. However, other complications are to be faced in this situation and it is unnecessary to have such a level of accuracy for most of the cases. Performing an approximation by using a limited number of sectors greatly simplifies the situation and increases the processing speed. Thus, the DAD diagram provides an approach to trading-off the system complexity and the accuracy of the object position estimation.

IV. Application

A. Single-segment DAD diagram v.s. multi-segment DAD diagram

The developed DAD method is successfully applied to a bare-hand mouse application. The basic DAD diagrams as well as their complex variations that demonstrate an advanced level of the DAD method are used individually in this practical application. The single-segment DAD diagrams perform poorly in the bare-hand motion detection, and require a pretreatment of noise reduction and filtering in 2D image processing for an acceptable performance. Though this limits the speed efficiency, the processing speed of the system based on the single-segment DAD diagrams is still faster than those of the existing hand motion detection and tracking methods. On the other hand, if enough segments are used, no 2D image processing is required and all the processing in the system is performed in one dimension. Having multiple segments in each direction greatly increases robustness against noise, which spares complicated noise reduction and filtering techniques. Hence, much higher processing speed is achieved. Also, more information about what is happening in the separate parts of the image becomes available. Since it is unnecessary to maintain the shape information in individual segments, thresholding is applied. The shape information can be induced from the motion activity between the segments. This is illustrated graphically by the example in Fig. 14.

Two example objects are given in Fig. 14, where a circle is in Fig. 14-a1 and a triangle is in Fig. 14-a2. Fig. 14-b1 and Fig. 14-b2 show the plots resulted from adding all the rows in Fig. 14-a1 and Fig. 14-a2 without any threshold applied to the sums, respectively. As seen, shape information of the objects is preserved in these plots. One can easily tell which plot is generated from the circle or the triangle. Fig. 14-c1 and Fig. 14-c2 show the plots resulted from the same operation as before, but with a threshold applied to the sums. In this case, the two plots look identical. One cannot tell which plot is generated from the circle or the triangle. Fig. 14-d1 and Fig. 14-d2 are obtained by dividing the images in Fig. 14-a1 and Fig. 14-a2 into four equal sectors, respectively. Suppose, without loss of generality, that both Fig. 14-a1 and Fig. 14-a2 have 120 rows. Let row 1- row 30 be sector 1, row 31- row 60 sector 2, row 61- row 90 sector 3, and row 91- row 120 sector 4. The rows in the sectors are then added to obtain four separate segments, and a threshold is applied to the sum. As seen now, even though thresholding is applied, shape information is preserved in these plots. One can easily tell which plot is generated from the circle or the triangle. This makes it possible to use segments with discrete values as shown in Fig. 12-b, rather than continuous values as shown in Fig. 12-a.

It should be noted that each of the two example objects given in Fig. 14 is from one frame, while the image shown in Fig. 12-b is the DAD plots obtained from three consecutive frames. In other words, Fig. 14 serves only to explain the concept of "preservation of shape information" in multi-segment DAD plots.

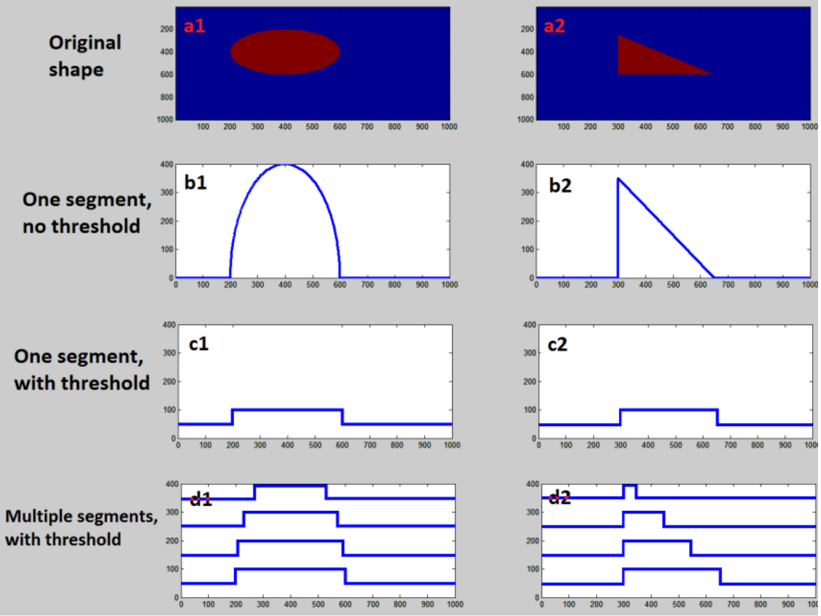


Fig. 14 Preservation of shape information in multiple segments

The relation between the multi-segment DAD plots and the motion direction cannot be easily observed, since the relation between the variables in the DAD plots and the motion direction becomes rather statistical. This makes the problem a perfect candidate for intelligent pattern recognition algorithms. History information can also be used. In object motion detection in real time, variables from the previous DAD plots can be used in addition to the variables from the most recent DAD plot so as to increase the robustness and accuracy of the object motion detection.

B. The DAD method applied to the bare-hand motion detection

In the system designed, the bare-hand mouse moves i the horizontal and vertical directions (0°, 90°), where four segments are designed for each direction, similar to the case shown in Fig. 12-b. Experimental results indicate that four is the minimum segment number that allows the removal of 2D processing while keeping a good system performance. The frame rate reached for this four-segment system exceeds by far 30 frames/second, the standard rate of the commercial web-cameras. Higher accuracy can be achieved by using more than four segments for each direction while still keeping a real-time processing speed on average personal computers. However, with designs of more segments, more variables are introduced and therefore designs with more than ten segments for each direction are not recommended, since this greatly complicates the pattern recognition in the sense of making the processing speed unpractical.

The designed system receives three consecutive 120 × 160 frames as its input and produces two 4 × 120 matrices as its output. Matrices $M^{1}_{120,160}$, $M^{2}_{120,160}$, and $M^{3}_{120,160}$ give the three input frames at *t*-2, *t*-1, and *t*, respectively, where 120 and 160 are the numbers of rows and columns. Matrices $OH_{4,120}$ and $OV_{120,4}$ give the outputs. The overall input/output relation is described by equations (9)-(21), where the matrices and their corresponding matrix entries are denoted by upper-case and their corresponding lower-case letters, respectively. Four row summated vectors $\tilde{H}^{k,s}_{1,160}$ for horizontal motion detection and four column summated vectors $\tilde{V}^{k,s}_{120,1}$ for vertical motion detection are calculated first, respectively, for the four separate equal segments indexed by *s* (*s*=1, 2, 3, 4) in the two directions of the three incoming frames indexed by *k* (*k*=1, 2, 3). The entries of $\tilde{H}^{k,s}_{1,160}$ and $\tilde{V}^{k,s}_{120,1}$ are given by (9) and (10), respectively. The four vectors $\tilde{H}^{k,s}_{1,160}$ for each frame are stacked together to become a matrix $H^{k}_{4,160}$ whose entries $h^{k}_{s,j}$ are shown in (11). So do the four vectors $\tilde{V}^{k,s}_{120,1}$ whose entries $v^{k}_{i,s}$ are shown in (12). (13) and (14) calculate $AH^{k}_{4,160}$ and $AV^{k}_{120,4}$, the absolute differences of the row and column sums, respectively, for the two consecutive frames. (15) and (17) calculate $DH_{4,160}$ and $DV_{120,4}$, the differences between the absolute differences in the two directions, respectively. (16) performs a resize operation on the matrix obtained in (15) so that the resulted $RH_{4,120}$ has the same size as that of $DV_{120,4}$ obtained in (17). (18) and (19) perform a moving window average low pass filter operation on $RH_{4,120}$ and $DV_{120,4}$, respectively. Finally, (20) and (21) calculate the outputs $OH_{4,120}$ and $OV_{120,4}$ by performing a threshold operation. The whole procedure above is illustrated in the flow diagram of the system shown in Fig. 15. The features extracted from $OH_{4,120}$ and $OV_{120,4}$ are then forwarded as inputs to an artificial neural network (ANN). Since the objective of this article is to present the developed DAD method for object motion detection, the component units in Fig.15 are not elaborated here.

$$\tilde{h}^{k,s}_{1,j} = \sum_{i=30(s-1)+1}^{30s} m^{k}_{i,j} \quad s=1,\ 2,\ 3,\ 4;\ k=1,\ 2,\ 3 \tag{9}$$

$$\tilde{v}^{k,s}_{i,1} = \sum_{j=30(s-1)+1}^{40s} m^{k}_{i.j} \quad s=1,\ 2,\ 3,\ 4;\ k=1,\ 2,\ 3 \tag{10}$$

$$h^{k}_{s,j} = \tilde{h}^{k,s}_{1,j} \quad s=1,2,3,4;\ k=1,2,3 \tag{11}$$

$$v^{k}_{i,s} = \tilde{v}^{k,s}_{i,1} \quad s=1,2,3,4;\ k=1,2,3 \tag{12}$$

$$AH^{k}_{4,160} = \left| H^{k+1}_{4,160} - H^{k}_{4,160} \right| \quad k=1,2 \tag{13}$$

$$AV^{k}_{120,4} = \left| V^{k+1}_{120,4} - V^{k}_{120,4} \right| \quad k=1,2 \tag{14}$$

$$DH_{4,160} = AH^{2}_{4,160} - AH^{1}_{4,160} \tag{15}$$

$$RH_{4,120} = decimate[interpolate(DH_{4,160}),3),4] \tag{16}$$

$$DV_{120,4} = AV^{2}_{120,4} - AV^{1}_{120,4} \tag{17}$$

$$fh_{i,j} = \begin{cases} [\sum_{n=1}^{10} rh_{i,(j-10+n)}]/10 & when\ j \ge 10 \\ [\sum_{n=1}^{j} rh_{i,n}]/\ j & when\ j < 10 \end{cases} \qquad (18)$$

$$fv_{i,j} = \begin{cases} [\sum_{n=1}^{10} rh_{(i-10+n),j}]/10 & when\ i \ge 10 \\ [\sum_{n=1}^{i} rh_{n,j}]/\ i & when\ i < 10 \end{cases} \qquad (19)$$

$$oh_{i,j} = \begin{cases} 1 & If\ fh_{i,j} > threshold \\ -1 & If\ fh_{i,j} < threshold \\ 0 & otherwise \end{cases} \qquad (20)$$

$$ov_{i,j} = \begin{cases} 1 & If\ fv_{i,j} > threshold \\ -1 & If\ fv_{i,j} < threshold \\ 0 & otherwise \end{cases} \qquad (21)$$

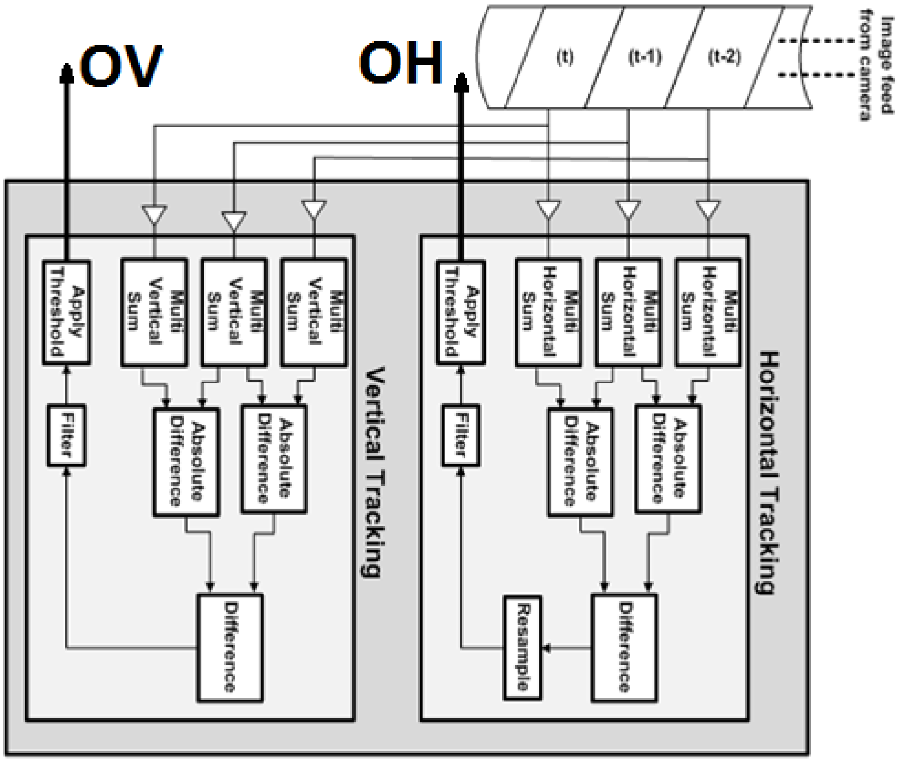


Fig. 15 System flow diagram

C. *Some observations*

In a multi-segment DAD diagram, the DAD plots obtained by the superposition of a number of single vectors are a measure of dimension reduction that greatly reduces the size of the source data while maintaining a unique representation for different motion patterns to a certain extent. As seen in the developed system, since four segments are used for each direction, simpler individual segments are obtained. The values of the variables extracted from the DAD plots are the features that allow the ANN to determine different motion patterns. While the working frame rate falls in the average rate range of the commercial webcams (15 frames/second –> 30 frames/second), the segments with more than two peaks indicate an overlapping of the moving hand in the two consecutive frames so that this extremely slow motion can be safely ignored. One segment from a typical four-segment DAD plot is shown in Fig. 16 as an example. In the case of a multi-segment DAD plot, *d1* gives information about the location of the motion activity on one segment relative to other segments. The values of the six variables are extracted from each segment for each new frame obtained from the webcam. The variables from two previous DAD plots are also used in addition to the variables from the most recent DAD plot to enhance the performance of the ANN. As mentioned already, the values of these variables are used as the inputs of the ANN. This gives a total of 144 features (6 variables × 8 segments × 3 frames) to be sent to the ANN. Since this article focuses on the development of the DAD method for object motion detection, specifically in bare-hand motion detection, the details of the function of the ANN are not discussed here.

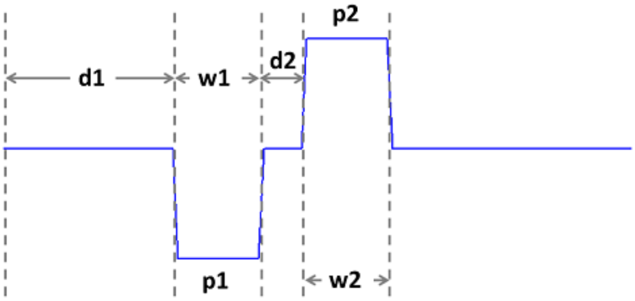


Fig. 16 One segment from a typical four-segment DAD plot

V. Experiment Results

The developed object motion detection method applied to the bare-hand mouse is tested on different platforms, and its processing speed is measured. The results are listed in Table 1. The processor speed of each platform is shown as a metric of reference with an understanding that other platform specifications may influence the processing speed too. As seen, the variety of the platforms presented in Table 1 gives a good idea about the range of the processing speed to be expected in using the developed method. A number of typical processing speeds of the existing similar systems are listed in Table 2 for comparisons [2][9][33].

The developed object motion detection method is also tested in different human-computer interaction systems based on hand motion detection and tracking, including bare-hand mouse, navigation, and game control. The testing is executed in a variety of environments with different backgrounds and lighting conditions. All the tests indicate that the developed object motion detection method fulfills its task requirements in real-time, and features its robustness against noise, immunity to background changes as well as its motion detection accuracy. A video demonstration is available at the web page accompanying this work [34].

Table 1 Experimental results of the processing speed of the developed system on different platforms

Platform	Average desktop	Average laptop	Fast desktop
Processor speed	2.67GHz	1.6 GHz	3.2GHz
Multi-segment system speed	130 frames/second	57 frames/second	195 frames/second

Table 2 Typical processing speeds of the existing similar systems during years 1994 – 2009

Reference	Year	Processor speed	Reported speed
Wang & Popovic	2009	2.4GHz	10 frames/second
Hardenberg & Bedard	2001	Not specified	20 - 25 frames/second
Rehg & Knade	1994	Dedicated Hardware	10 - 15 frames/second

VI. Conclusions And Future Work

A new approach to object motion detection, which is based on the DAD method, is proposed in this article. The experimental results give up to 195 frames/second for a bare-hand mouse system based on multi-segment DAD diagrams that can track a moving hand in horizontal and vertical directions. This frame rate is much higher than the standard frame rates of the commercial webcams, which do not exceed 30 frames/second. Although the system built for demonstration is completely implemented in software, hardware implementation, whose processing speed is limited only by the frame rate of the camera in use, is perfecty realistic. It is worth emphasizing that the approach presented in this article is feasible to all kinds of object motion detection. With extremely high-speed specialized cameras used, processing speeds in the order of hundreds of thousands of frames/second can be reached. Some example applications at a high frame rate are bullet tracking, racing car tracking, missile tracking and so on.

Future works include 3D object motion detection and tracking. One option is to use two cameras that are positioned in a way that one of them detects the moving object in the y-z plane, while the other detects it in the x-z plane. The other option is to use one camera only with the introduction of some basic variations in the DAD diagram. As noticed, the DAD diagrams used in this article serve for the object motion detection in a plane parallel to the camera. 3D motion detection implies that the information of the depth of the object motion is needed. Different patterns obtained from the resulted DAD plots identify the motion in the depth direction. One example case is shown in Fig. 17, where the resulted DAD plot presents a pattern that features uniquely a motion in the depth direction.

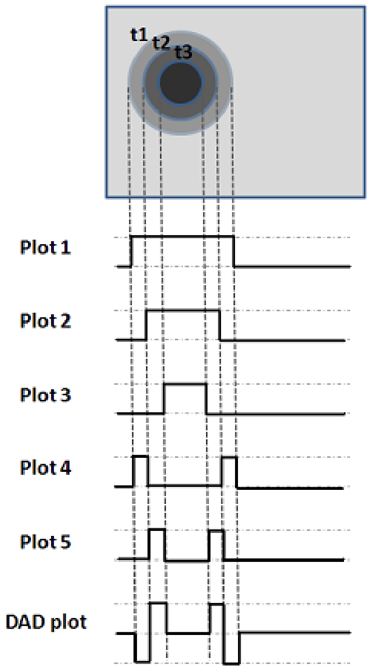


Fig. 17 An example DAD diagram obtained from an object motion detection in the depth direction

References

[1] Wang, R. Y., “Real-Time Hand-Tracking as a User Input Device”, ACM Symposium on User Interface Software and Technology (UIST), 2008

[2] Wang, R. Y. & Popovic, J., “Real-Time Hand-Tracking with a Color Glove”, ACM Transactions on Graphics, 2009

[3] Milanovic, V. & Lo, W. K., “Fast and High-Precision 3D Tracking and Position Measurement with MEMs Micromirrors”, Optical MEMs and Nanophotonics. IEEE/LEOS. 2009

[4] Lee, J. *et al.*, “The 3D Sensor Table for Bare Hand Tracking and Posture Recognition”, Lecture Notes in Computer Science, Springer, 2006

[5] Campos, T. E. & Murray, D. W., “Regression-Based Hand Pose Estimation from Multiple Cameras”, Conference on Computer Vision and Pattern Recognition (CVPR), 2006

[6] Fujiyoshi, H. *et al.*, “Fast 3D Position Measurement with Two Unsynchronized Cameras”, *IEEE* International Symposium on Computational Intelligence in Robotics and Automation, 2003

[7] Schlattmann, M. *et al.*, “Real-Time Bare-Hands Tracking for 3D Games”, IADIS International Conference on Game and Entertainment Technology (GET), 2009

[8] Garg, P., Aggarwa, N. & Sofat, S., “Vision Based Hand Gesture Recognition”, Proceedings of World Academy of Science, Engineering and Technology, Vol. 49, pp. 972-977, 2009

[9] Hardenberg, C. V. & Bedard, F., “Bare Hand Human Computer Interaction”, Proceedings of the 2001 ACM Workshop on Perceptive User Interfaces, 2001

[10] Triesch, J. & Malsburg, C., “Robust Classification of Hand Postures Against Complex Background”, International Conference on Automatic Face and Gesture Recognition, 1996, Killington

[11] Ware, C. & Balakrishnan, R., “Researching for Objects in VR Displays: Lag and Frame Rate”, ACM Transactions on Computer-Human Interaction, 1994

[12] Sato, Y., Kobayashi, Y. & Koike, A. H., “Fast Tracking of Hands and Fingertips in Infrared Images for Augmented Desk Interface”, *IEEE* International Conference on Automatic Face and Gesture Recognition, 2000

[13] Segen, J., “GestureVR: Vision-Based 3D Hand Interface for Spatial Interaction”, ACM Multimedia, 1998, Bristol

[14] Laptev, I. & Lindeberg, T., “Tracking of Multi-State Hand Models Using Particle Filtering and a Hierarchy of Multi-Scale”, Proceedings of the *IEEE* Workshop on Scale-Space and Morphology, 2001

[15] Crowley, J., Bérard, F., & Coutaz, J., “Finger Tracking as an Input Device for Augmented Reality”, Proceedings of the International Workshop on Gesture and Face Recognition, 1995, Zurich

[16] O'Hagan, R., & Zelinsky, A., “Finger Track - A Robust and Real-Time Gesture Interface”, Australian Joint Conference on Artificial Intelligence. 1997, Perth

[17] Stenger, B., Thayananthan, A., Torr, P. & Cipolla, R., “Model-Based Hand Tracking Using a Hierarchical Bayesian Filter”, *IEEE* Transactions on Pattern Analysis and Machine, pp. 1372 – 1384, 2006

[18] Bretzner, L., Laptev, I. & Lindeberg, T., “Hand Gesture Recognition Using Multi-scale Color Features Hierarchichal Models and Particle Filtering”, Proceedings of the International Conference on Automatic Face and Gesture Recognition, 2002, Washington D.C.

[19] Sánchez-Nielsen, E., Antón-Canalís, L. & Hernández-Tejera, M., “Hand Gesture Recognition for Human Machine Interaction”, 12th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG), 2004

[20] Stenger, B., “Template Based Hand Pose Recognition Using Multiple Cues”, Computer Vision – ACCV, Vol. 3852, pp. 551-560, 2006, Springer

[21] Lienhart, R., & Maydt, J., “An Extended Set of Haar-Like Features for Rapid Object Detection”, Proceedings of the *IEEE* International Conference on Image Processing, 2002

[22] Barczak, A. L. & Dadgostar, F., “Real-Time Hand Tracking Using a Set of Co-Operative Classifiers Based on Haar-Like Features”, Research Letters in the Information and Mathematical Sciences, Vol. 7, pp. 29-42, 2005

[23] Chen, Q., Georganas, N. & Petriu, E., “Real-Time Vision Based Hand Gesture Recognition Using Haar-Like Features,” Proceedings of the *IEEE* International Conference on Instrumentation and Measurement Technology, 2005, Warsaw

[24] Wang, C. C. & Wang, K. C., “Hand Posture Recognition Using Adaboost with SIFT for Human Robot Interaction”, Recent Progress in Robotics: Viable Robotic Service to Human, Vol. 370, pp. 317-329, 2009, Springer Berlin / Heidelberg

[25] Kang, H., Lee, C. W. & Jung, K., “Recognition-Based Gesture Spotting in Video Games”, Pattern Recognition Letters, pp. 1701-1714, 2004

[26] Lu, P., Chen, Y., Zeng, X. & Wang, Y., “A Vision Based Game Control Method”, Computer Vision in Human-Computer Interaction, Vol. 3766, pp. 70-78, 2005, Springer Berlin / Heidelberg

[27] Jong-Hyun, Y., Park, J.-S. & Sung, M. Y., “Vision-Based Bare-Hand Gesture Interface for Interactive Augmented Reality Applications”, Entertainment Computing - ICEC 2006, Vol. 4161, pp. 386-389, Springer Berlin / Heidelberg, 2006

[28] Park, H. S., Jung, D. J. & Kim, H. J., “Vision-Based Game Interface Using Human Gesture”, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2006

[29] Song, P., Yu, H. & Winkler, S., “Vision-Based 3D Finger Interactions for Mixed Reality Games with Physics Simulation”, Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry, 2008

[30] Vassiliadis, S., Wong, S., Hakkennes, E., Wong, J. S. & Pechanek, G., “The Sum-Absolute Difference Motion Estimation Accelerator”, Proceedings of the IEEE 24th Euromicro Conference, 1998

[31] Vanne, J., Aho, E., Hamalainen, T. & Kuusilinna, K., “A High-Performance Sum of Absolute Difference Implementation for Motion Estimation”, IEEE Transactions on Circuits and Systems for Video Technology, pp. 876-883, 2006

[32] Rehman, S., Young, R., Chatwin, C. & Birch, P., “An FPGA Based Generic Framework for High Speed Sum of Absolute Difference Implementation”, European Journal of Scientific Research, pp. 6-29, 2009

[33] Rehg, J. & Knade, T., “Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking”, Third European Conference on Computer Vision, pp. 35-46, Stockholm, 1994

[34] http://sites.google.com/site/kyoussefsite/bhtdemo

Khalid Youssef received his M.S. degree in electrical engineering from Northern Illinois University, IL, USA in 2010. During his Master’s education, he did research on robotic control, fuzzy systems, neural networks and image processing, and co-authored international journal articles and international conference papers. Currently, he is a Ph.D. candidate in electrical engineering in University of California (Los Angeles).

Peng-Yung Woo received the B.S. degree in physics /electrical engineering from Fudan University, Shanghai, China in 1982 and the M.S. degree in electrical engineering from Drexel University, Philadelphia, PA, in 1983. In 1988, he received the Ph.D. degree in system engineering from the University of Pennsylvania, PA. He joined Northern Illinois University in 1988 and is currently a Professor of Electrical Engineering. His current research interests include intelligent control, robotics, digital signal processing, neural networks, fuzzy systems and digital image processing. He has authored and co-authored over 50 international journal articles and book sections as well as over 60 international conference papers. He is a senior member of the *IEEE* and the member of the *IASTED* Standing Technical Committee. He has been the member of many International Program Committees for various *IEEE* and *IASTED* International Conferences. He is currently also the Advisory Professor of Tongji University, China.

How to cite this paper: Khalid Youssef,Peng-Yung Woo,"Difference of the Absolute Differences – A New Method for Motion Detection", International Journal of Intelligent Systems and Applications(IJISA), vol.4, no.9, pp.1-14, 2012. DOI: 10.5815/ijisa.2012.09.01