Fast 2-D Moving Target Tracking Algorithm

(Fast 2 차원 동 표적 추적 알고리즘)

金 炅 洙*, 李 商 郁**, 宋 裕 燮***

(Kyeong Soo Kim, Sang Uk Lee and Yoo Seop Song)

要 約

실시간 설치가 가능한 움직이는 물체를 추적하는 알고리즘에 관하여 연구하였다. 본 논문에서는 연산자 표현과 variational 알고리즘[10]에 근거하여 fast 알고리즘을 개발하였다. Variational 알고리즘에 대한 초기 추정치는 상호 상관 계수가 최대가 되는 위치를 directed search를 통하여 구하였다. 개발한 fast 알고리즘은 먼저 영상을 16개의 작은 부영상으로 분해하고, 새로 제시한 움직임 검출 방법과 variational 알고리즘을 각 부영상에 순차적으로 적용하는 알고리즘이다. 따라서 부영상을 단위로 하는 recursive 알고리즘이다. 개발한 알고리즘은[10]과 비교하여 평균 7 대 1 정도의 계산상의 이득을 얻을 수 있다.

Abstract

We have studied on the 2-D moving target tracking algorithm satisfying a real-time hardware implementation requirement. In this paper, a fast algorithm is developed based on the operator formulation and the variational algorithm [10]. Here, we use the directed search for the maximum of the cross-correlation in order to obtain an initial estimate for the variational algorithm and decompose the scene into 16 smaller subblocks and apply the variational algorithm to each subblock sequentially with a new moving area detection method. We call the algorithm subblock based recursive algorithm. Compared with [10], the ratio of the computational savings obtained from the proposed algorithm is 7 on the average.

I. Introduction

Recently, computer analysis of time-varying imagery has gained attention in the field of image understanding. One of the interesting problems in this field is a moving target tracking, which has a numerous applications in teleconferencing, videotelephone, robotic vision system, surveillance, and television and satellite image transmission. Most of the works in the last several years were concerned with rigid bodies moving on the 2-dimensional plane [1-10, 15-17]. Elsewhere, the interpretation of 3-dimensional motion using 2-D motion estimation or correspondence has also been studied [18-20].

Basically, there are two different mathematical approaches in moving target tracking: correlation techniques [15-17] and taylor series expansion [1-10]. In correlation techique, cross-correlation is used to determine the offset of two images that differ by a rela-

*準會員, **正會員, 서울大學校 工科大學 制御計測工學科
(Dept. of Control and Instrumentation Eng., Seoul National Univ.)

***正會員 洪陵機械工業會社
(Hong Neung Machine Depot)

接受日字：1984年 9月 17日

tive geometric distortion [15-16]. J.K. Jain and A.K. Jain [17] also used the cross-correlation to find the displacement vector. Taylor series expansion is commonly used to estimate the 2-D motion of a rigid body. Limb & Murphy [1,2] proposed a simple and somewhat intuitive algorithm which was derived mathematically later using linear regression [4,5]. However, this algorithm operates poorly with low resolution images or images containing thin edges. Cafforio & Rocca [3,4] proposed a linear regression approach based on the first-order terms in taylor series expansion. But because of first-order approximation, the estimation is limited to relatively small displacement. Netravali and Robbins [5,6] discussed a recursive algorithm which seeks to minimize a functional of the motion-compensated prediction error. They discussed the method that can improve estimation accuracy for large displacement by linearizing the intensity function around an initial estimate. Schalkoff and McVey [7,8] presented a mathematical model for tracking translation, rotation, and dilation using 2-D affine transforms. They showed that for small target perturbations the 2-D tracking problem could be approximated as a 1-D time-varying parameter estimation problem. Flachs et al [9] proposed a prototype realtime video tracking system using extensive parallel processing to track a missile-type object. The essence of this algorithm is a Bayesian pixel classifier, aided by fuzzy set logic which forms a binary picture subdivided into target, background, and plume regions.

Recently, Legters and Young [10] developed a mathematical model using an operator formulation for a moving object in a sequence of image. Time-varying translation and rotation operators were derived to describe the motion and variational algorithm in conjunction with predictive Kalman filter was used to track the dynamic parameters of the operator. The operator formulation and variational algorithm will be briefly discussed in section 2. The operator concept is important since it describes the evolution of time-varying target in a simple and mathematical way. The proposed algorithm works quite well in tracking moving target, however the computational burden in real-time implementation is quite enormous. Traditionally, the gradient algorithm utilizes a time-consuming search technique to find minimum or maximum value of a function containing several parameters. Generally, the computational load becomes large as the size of image grows. Furthermore, real-time operating Kalman filter is too complicated and expensive in these days. In order to employ this algorithm in a real-time environment, modification or new algorithm is needed.

The purpose of this paper is to introduce a fast motion tracking algorithm satisfying a real-time implementation requirement. The approach used in this paper is somewhat similar to the variational algorithm in [10], except that here we used directed search algorithm [14,17] instead of Kalman filter to get an initial estimate and decomposed the scene into smaller subblocks and applied the variational algorithm to each subblock sequentially with a new moving area detection method. We shall show that this approach could save a significant computational load because of a smaller subblock size. Two important assumptions were made in this paper to simplify the tracking problem: Target/Background separation (segmentation) problem has already been sloved (i.e., test image used for the simulation is binary image) and changes in the consecutive scenes are gradual enough so that scene differences may be properly expressed using an operator formulation.

The directed search method used for obtaining an initial estimate for the variational algorithm is explained in section 3. Moving area detection scheme and fast motion tracking algorithm are discussed in detail in section 4. The results of moving target tracking experiments and the quantitative computational savings obtained from the proposed algorithm are described in section 5.

II. Operator Formulation & Variational Algorithm

A general two-dimensional time-varying noiseless scene S consists of time varying target O and background scene B which has possibly contain a stationary object. In this section, we review briefly the operator formulation and the variational algorithm discussed by Legters and Young [10]. We will use the algorithm with a modification of the intensity gradient approximation directly in developing a fast algorithm.

1. Operator Formulation

The time evolution of the target scene O is described by an operator T such that

$$O(x,y;t+1) = T(t)O(x,y;t) \tag{1}$$

Note that the operator T is time-varying and it is assumed for convenience that t is nonnegative integral-valued, and the consecutive scenes are separated by one unit of time typically 1/60-1/30 second in TV scenes. From [10], the translation operator is in the form of

$$T(u,v) = \exp\,[-u(\partial/\partial x)-v(\partial/\partial y)] \tag{2}$$

and the rotation operator is

$$R(\theta)=\exp[\,\theta(y-b)(\partial/\partial x)-\theta(x-a)(\partial/\partial y)] \tag{3}$$

On the assumption that segmentation problem has already been solved (i.e., the functions 0 and B will be considered as binary-valued functions), the scene at time t may be modelled as

$$S(x,y;t)=O(x,y;t)+B(x,y)[1-O(x,y;t)] \tag{4}$$

and from (1) and (4), the scene at time t+1 may be expressed as

$$S(x,y;t+1)=S(x,y;t)+[1-B(x,y)]\,[T(t)-1]\,O(x,y;t) \tag{5}$$

Since the moving target of the entire scene is of primary interest and no prior knowledge of the target is assumed, differencing of consecutive scenes will eliminate some uninteresting background information. For instance, the stationary object in the background is eliminated by differencing. Define the difference scene as

$$\begin{aligned} D(x,y;t)&=S(x,y;t+1)-S(x,y;t)\\ &=[1-B(x,y)]\,[T(t)-1]\,O(x,y;t) \end{aligned} \tag{6}$$

It is clear from (6) that difference scenes depend primarily on the moving target only. We assume the difference scenes, similar to the moving targets, can be expressed in terms of operators, as given in (7).

$$D(x,y;t+1)=T(t)D(x,y;t) \tag{7}$$

Also, substituting (6) into (7) yields

$$D(x,y;t+1)=[1-B(x,y)]\,[T(t+1)-1]\,T(t)O(x,y;t) \tag{8}$$

If the changes in the consecutive scenes are gradual enough so that T(t+1) equals approximately T(t), (8) is reduced to (6). This fact justifies the operator formulation to the difference scenes. It should be also noted that if the parameters keep changing rapidly from scene to scene, it may be necessary to use the original scene instead of the difference scene. It should be also noted that if the parameters keep changing rapidly from scene to scene, it may be necessary to use the original scene instead of the difference scene.

2. Derivation of Variational Algorithm

With the operator approach, the error used in deriving a variational algorithm is a quadratic form

$$\begin{aligned} E&=\sum\sum(\hat{D}-D)^2\\ &=\sum_x\sum_y[\hat{T}(t)D(x,y;t)-D(x,y;t+1)]^2 \end{aligned} \tag{9}$$

where the hat indicates the estimate. The variational scheme seeks to find the appropriate parameters to minimize the mean square error defined in (9).

A. Translation

With a translation operator $\hat{T}(u,v)=T(\hat{u},\hat{v})$, the resultant discrete version of the gradient search direction is expressed in the form of (10)[10].

$$\partial E/\partial\hat{u}=-2\sum_x\sum_y[\hat{D}(x,y;t+1)-D(x,y;t+1)]\,(\partial/\partial x)\hat{D}(x,y;t+1) \tag{10}$$

where the discrete version of the intensity gradient is

$$(\partial/\partial x)\hat{D}(x,y;t)= \hat{D}(x+1,y;t)-\hat{D}(x,y;t) \quad (11)$$

Since the segmented image has only the value of 1,0, or −1, (11) has the possibility of missing the negative or positive edge of a target.

$$(\partial/\partial x)\hat{D}(x,y;t)=[\hat{D}(x+1,y;t)- -\hat{D}(x-1,y;t)]/2 \quad (12)$$

We used (12) for the intensity gradient calculation. Throughout the experiment, (12) is found to give more accurate gradient direction than (11).

A similar derivation to the y-direction leads to

$$\partial E/\partial\hat{v}=-\sum_x\sum_y[\hat{D}(x,y;t+1)-D(x,y;t+1)] [\hat{D}(x,y+1;t+1)-\hat{D}(x,y-1;t+1)] \quad (13)$$

The variational scheme adjusts the value of $\hat{u}$ and $\hat{v}$ until the two gradients $\partial E/\partial\hat{u}$ and $\partial E/\partial\hat{v}$ change signs or vanish at the same time, indicating convergence. Clearly, the parameter value must be increased if the gradient has a negative value and decreased if positive value. Since the higher order terms of taylor series expansion is neglected in this approach, the variational scheme may converge to a local minimum or it may diverge. Generally speaking, convergence to a global minimum will depend on the goodness of the initial estimate. Therefore in section 3, we will discuss the method of finding the initial estimate in a straightforward manner that does not include any approximation.

B. Rotation

The tracking of a rotation consists of estimating the angular velocity, θ, and the center of rotation (a,b). The rotation operator R depends on θ, a, and b, and hence the gradients $\partial E/\partial\hat{\theta}$, $\partial E/\partial\hat{a}$, and $\partial E/\partial\hat{b}$ are derived. Replacing T by R and $\hat{u}$ by $\hat{\theta}$ in (10), the resultant gradient for $\hat{\theta}$ is

$$\begin{aligned}\partial E/\partial\hat{\theta}=&-\Sigma\Sigma[\hat{D}(x,y;t+1)-D(x,y;t+1)](x-\hat{a})\\&[\hat{D}(x,y+1;t+1)-\hat{D}(x,y-1;t+1)]\\&+\Sigma\Sigma[\hat{D}(x,y;t+1)-D(x,y;t+1)]\,(y-\hat{b})\\&[\hat{D}(x+1,y;t+1)-\hat{D}(x-1,y;t+1)]\end{aligned} \quad (14)$$

and with a derivation very similar to that of a translational case, we obtain the gradient expressions for the center of rotation (a,b).

$$\partial E/\partial\hat{a}= \Sigma\Sigma[\hat{D}(x,y;t+1)-D(x,y;t+1)]\hat{\theta} [\hat{D}(x,y+1;t+1)-\hat{D}(x,y-1;t+1)] \quad (15)$$

$$\partial E/\partial\hat{b}=-\Sigma\Sigma[\hat{D}(x,y;t+1)-D(x,y;t+1)]\hat{\theta} [\hat{D}(x+1,y;t+1)-\hat{D}(x-1,y;t+1)] \quad (16)$$

Here again, it is noted that a and b should be integer-valued, and the variational scheme follows exactly the same method explained in the translation case using the gradient values of (14), (15), and (16).

III. Initial Estimate for The Variational Algorithm

Extensive simulation has revealed that the variational algorithm described in section 2 is very sensitive to the initial estimate because this algorithm include several sssumptions and approximations. Therefore it is necessary to find the initial estimate for the variational algorithm in the straight forward manner. The Kalman filter approach used in [10] has been proved to provide a relatively good initial estimate, but unfortunately, it is considered too complex and expensive for realizing a real-time tracker. Thus a simple and accurate method to find an initial estimate for the real-time implementation should be developed. In this paper, we use the correlation technique, simplifying the approach of Jain [17] which includes no approximations.

In the translational tracking, the necessary parameters to estimate are u and v. The error expression in (9) can be rewritten as

$$E(u,v) = \sum_x\sum_y[D(x,y;t+1)-D(x-u,y-v;t)]^2 \quad (17)$$

It is noted that the expression $E(u,v)$ is used to emphasize the fact that the error E is a function of the displacements u and v. Now, define a cross-correlation between $D(t)$ and $D(t+1)$ as

$$L(u,v)=\sum_{x}\sum_{y}D(x,y;t+1)D(x-u,y-v;t) \qquad (18)$$

Since different estimates will not change $\Sigma\Sigma D^2(t)$ and $\Sigma\Sigma D^2(t+1)$, minimizing E defined in (17) is equivalent to maximizing the cross correlation L. The directed search with L is more efficient because the calculation of L can be realized in hardware using 'and' logic and accumulations. Thus we search for the maximum correlation L instead of the minimum of error E using 2-D directed search. It is noted that since the initial estimate has only to guarantee the variational algorithm to converge to the real displacement, the exact displacement may not be needed in our case. As a means of obtaining an initial estimate, we crudely measure the displacement using the directed search and use the measurement as an initial estimate for the variational algorithm. Hence we restrict the search area to 5 x 5, thus obtain the submaximum of correlation. In each step, we search five locations which contain the center of the area, and the midpoints between the center and the four boundaries of the area along the axes passing through the center. This procedure continues until the maximum correlation is under the center of the area. The searched displacements, u and v which give the submaximum correlation are used as an initial estimate for the variational algorithm. The algorithm is expressed as follows:

Let's define a set S as
$S=\{(0,0),(2,0),(0,2),(-2,0),(0,-2)\}$
Step 1: (Initialization) $u=v=0$
Step 2: Find $(i,j)\in S$ such that $L(u+i,v+j)$ is maximum.
If $i=j=0$, go to step 4; otherwise go to step 3.
Step 3: $u \leftarrow u+i$, $v \leftarrow v+j$; $S \leftarrow S-(-i,-j)$; go to step 2.
Step 4: $u \leftarrow u+i$, $v \leftarrow v+j$, (u,v) then gives the submaximum correlation.

In the rotational tracking, the necessary parameter to estimate is three. 3-D directed search may be certainly conceivable, but the amount of required calculation to estimate the motion becomes very large. We can avoid this problem by simply assuming that the center of rotation is allowed only on the boundaries of the center of gravity of the target. This means that the initial estimate of the center of rotation is crudely decided as a point near the center of gravity, and only the initial estimate of angular velocity is finely estimated using a 1-D directed (binary) search [14]. Practically, this assumption is appropriate because any motion without deformation can be generated on the assumption above.

IV. Development of the Fast Algorithm

So far we have discussed the motion tracking algorithm using the correlation technique to get an initial estimate and the variational algorithm. Since the estimation process must scan all the interior pixels of the scene, the amount of computation in the estimation process is proportional to the size of the scene. The variational tracking algorithm [10] has been proved to be accurate in motion estimation. Though they use a binary scene to alleviate the computational load, the algorithm is still inappropriate in the sense of real-time hardware implementation because of heavy interations with a large size of scene. It is noted that the amount of computation becomes increasing as the size of the scene grows larger. Hence the computational burden may be decreased if we use a small size scene. However in this situation, the accuracy of the algorithm becomes poor since the spatial resolution is also decreased. Therefore it is necessary to derive a fast algorithm without any sacrifice in the spatial resolution. We will introduce a fast algorithm which brings out considerable savings in the computation and which is easily implementable in hardware.

In the first step, the target in the entire scene is windowed, and only the target region is involved in the computation of the estimation process. The window used here is a simple rectangular window including all the target pixels and simultaneously set up as small as possible. Hence the size of the window must depend on the size of the target. If the target size is large (i.e., the window size is large), a fine spatial resolution of the target is obtained and a good estimation may result, but on the contrary the necessary computation becomes larger. This problem is equivalent to the problem of deciding an appropriate target size in the entire scene. In this paper, we have decided to use 64x64 size window throughout the experiments, and this choice turned out to be a good compromise between the conflicting desire for getting a fine resolution of a target and for lessening the necessary computation.

In the second step, the target region is again decomposed into a few small subblocks. Each subblock has a same size and rectangular shape like a window. Each subblock should be determined whether a moving area or not. Moving area detection is based on the size of a difference scene in each subblock because the size of a difference scene depends on the displacement. Since we assume that the segmentation problem has already been solved, the target region has the value of 1 and 0 elsewhere. Therefore, the difference scene is masked to zero, one, or minus one. The size of a difference scene is defined as the number of pixels that have nonzero values. The moving area decision of a subblock is accomplished by the threshold method. In this paper, we introduce a frame-to-frame adaptive threshold method which is described as follows

$$\text{if } TLL(t) < size(i,j;t) < THH(t), \tag{19}$$

then (i,j)–th subblock is a moving area

with

$$TLL(t) = \max\,[DSIZE(t)/div,\ mins]$$

$$THH(t) = (\text{size of subblock}) - TLL(t)$$

where

DSIZE(t) = size of a difference scene in the target region at t-th frame

size(i,j;t) = size of a difference scene in (i,j)-th subblock where (i,j) indicate the subblock located on i-th row and j-th column (See Fig. 1)

mins = minimum threshold allowed

div = scale factor

DSIZE(t) and size(t) are easily calculated by accumulating the pixel intensities since the intensity has the value either 0 or 1 when the positive part of a difference scene is used, and 0 or -1 when the negative part is used. When both the negative and positive part are used, they are also calculated in the same manner as above because the value of 1 and -1 is equal in the least second bit of twos-complement. Mins is necessary to limit the minimum size of a difference scene in a subblock because too low threshold do not yield a good estimation. Selecting the value of div is somewhat arbitrary, but choose the value such that DSIZE div is distributed around the mins. The value of div adjusts the average number of moving subblocks in one frame. The average number of moving subblocks will be mainly concerned with the average number of iterations in one frame. THH is necessary because the subblock consisted of too many non-zero value pixels rather than zero value pixels (i.e., too large size subblock) may generate a poor result.

In the final step, the variational algorithm is applied to each moving subblock in a sequential manner. We shall call the proposed algorithm subblock based recursive algorithm because the algorithm iterates on the basis of small block. Since the subblocks which are determined stationary by the threshold method

1 (1,1)	2 (1,2)	3	4
5 (2,1)	6	7 (i, j)	8
9	10	11	12
13	14	15	16 (4,4)

Fig. 1. Description of the processing order in subblock based recursive algorithm. In this example the scene is decomposed into 16 subblocks.

may yield a poor estimation result because of few data involved in the estimation process, they must be skipped. In each subblock, moving area decision is first performed and the variational algorithm is applied to the only moving area subblock.

The processing order this recursive algorithm iterates is shown in Fig. 1. Recursion is proceeded in an increasing order of the number shown in each subblock. After recursion is started from the subblock #1, moving area dicision is performed sequentially. Meanwhile, in the first subblock determined moving the initial estimate is calculated using the correlation technique explained in section 3. Then with this initial estimate, the variational algorithm starts the iteration from that subblock to the remaining ones following the processing order. The iteration is performed two times in each subblock because only one iteration may miss the continuity between the subblocks and more than 2 iterations are not necessary due to the presearched initial estimate. If the subblock is determined stationary, the variational algorithm should not be applied, and only preserve the estimation result of the previous subblock since the current moving subblock takes the estimation result of the previous moving subblock as a new initial estimate for the variational algorithm. After all the subblocks are processed, final estimate is obtained. The fast algorithm is summarized as follows and the flow chart is in Fig. 2.

Step 0 : (Initialization) j=0, t=0

Step 1 : j ← j+1

Step 2 : If j-th subblock stationary, go to step 7; otherwise t ← t+1 and go to step 3.

Step 3 : If t=1, the $\hat{X}(0,j)$ ← initial estimate

Step 4 : i=0

Step 5 : (Gradient search)
$S(i+1,j)=\hat{X}(i,j)-\mathrm{sign}(\partial E/\partial \hat{x}(i+1,j))$,
i ← i+1

Step 6 : If i 2, go to step 5.
Otherwise go to step 7.

Step 7 : If j < N, $\hat{X}(0,j+1)$ ← $\hat{X}(2,j)$ and go to step 1; otherwise go to step 8.

Step 8 : $\hat{X}(2,N)$ gives the estimation result.

Where X, E_j, and N denote the estimate vector, the error in the j-th subblock, and the number of decomposed blocks in the entire scene respectively.

In this approach, significant computational saving are obtained because this method iterates on a small subblock rather than the entire scene. The detailed comparison of the developed algorithm with the Legters and Young's [10] will be discussed in section 5.

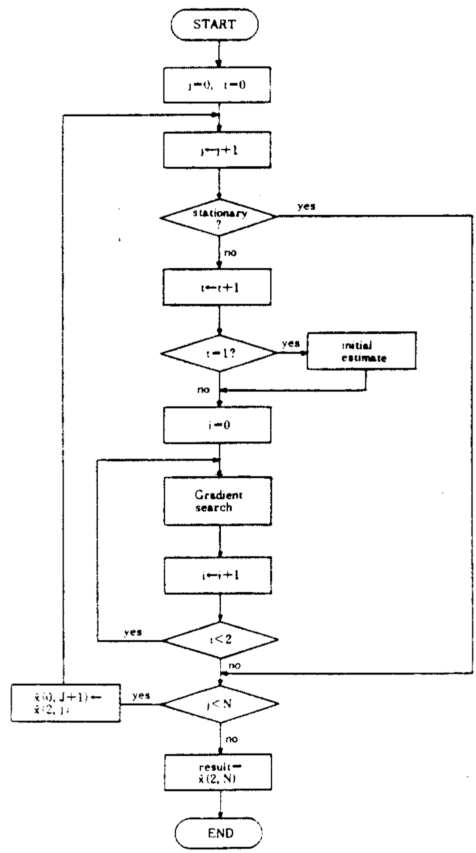


Fig. 2. Flow chart of the sub block based recursive algorithm.

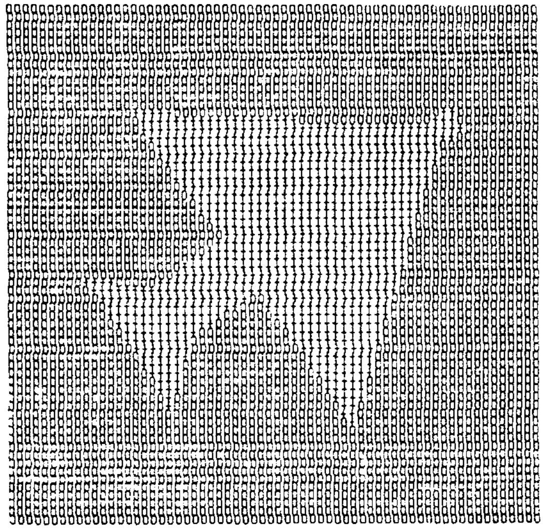
V. Simulation & Hardware Realization

Extensive computer simulation was used to test the proposed algorithm in particular the validity of fast algorithm (subblock based recursive algorithm) using image decomposition. Since segmentation problem was assumed to have already been solved, binary images were used to simulate the tracking algorithm which works with the difference scenes. Fig. 3(a) contains the test image used in simulation. We used 64x64 size target region. If the number of subblock is increased (or decreased) (for example, 32x32 or 8x8 size subblock), the tracking result becomes poor because each subblock pixels involved in the tracking process is too small (or large) to produce a good estimate. 16x16 size subblock (i.e., the number of subblock in the target region is 16) was found to be a good compromise.

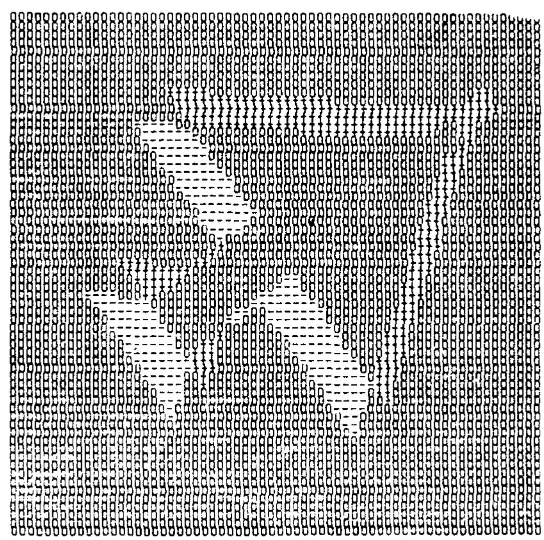
The proposed fast algorithm must include a moving area detection algorithm, which deeply affects the convergence of the algorithm. The false dectection of moving subblock in the recursion process may do the tracking great harm. The inclusion of too small or too large size subblock in the recursion may introduce a diverging direction of gradient, hence result in an false estimation. We select the parameter value of div in (19) to 13 and mins to 10. The selection of parameter values is not critical because if 3-8 subblocks are detected to be moving, it is sufficient to provide the convergence of the algorithm.

The proposed algorithm works with the consecutive difference scene. The difference scene consists of a positive and negative part as shown in Fig. 3(b).

Consecutive scene differences will not have the same shape if the curvature of the trajectory changes. This is due to the unequal displacement of the target in the x and y direction, or due to the unequal rotation along the trajectory of motion. The tracking result obtained from a positive difference scene was observed to be almost equal to the result with a negative scene. This result is expected because of the compatibility between a positive and a negative scene. The same tracking result was also observed when both the positive and negative scenes were simultaneously used. But in the rotational case, poor tracking result was obtained when both the scenes were used, hence either a positive or a negative scene should be used. The tracking result of fast algorithm in the translational motion is shown in Fig. 4. The trajectory of motion used for simulation is not a straight line but a curved line.



(a) Binary scene and



(b) Difference scene

Fig. 3. Test images.

The estimation result shown in the figure reveals that translational motion containing small rotation can be measured by a translational motion tracking only. The initial estimate obtained from the directed search for the maximum correlation was considered to be adequate to the convergence of variational algorithm. The arrow in the figure indicates the converging direction from the initial esti-

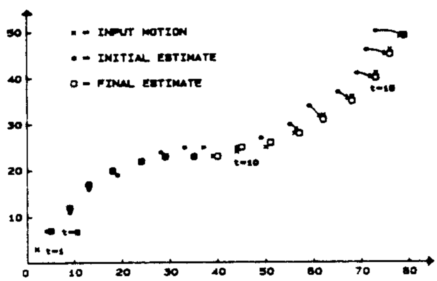


Fig. 4. Results of translational tracking experiments.

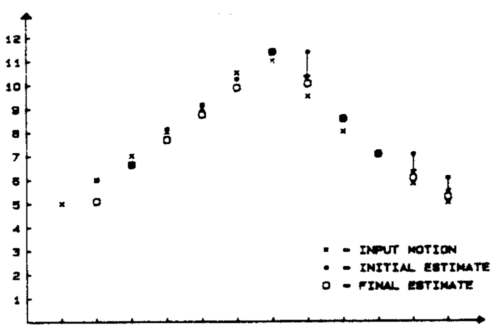


Fig. 5. Results of rotational tracking experiments.

mate toward the final estimate. Fig. 5 shows the rotational motion tracking result. We use only those rotation operators that rotate about the center of gravity of the target because rotation about an arbitrary point can be interpreted as a rotation about the center of gravity followed by translation. Hence we only present the tracking result of angular velocity in the figure. Due to the assumption on the gradual motion in section 2, the algorithm may introduce a relatively large estimation error if the motion of a target changes rapidly. Otherwise the observed estimation error is relatively small (within ±1pixel in translation and ±1° in rotation). The computer simulation confirmed that these results were identical to those obtained using the entire scene. In the tracking process, the searching strategy for the convergence using gradient values is important. The searching strategy becomes complex as the number of parameters to estimate increases. The needed number of parameters is 2 for the translational motion tracking, and 3 for the rotational motion tracking. Fig. 6. shows the flow chart for the tracking strategy.

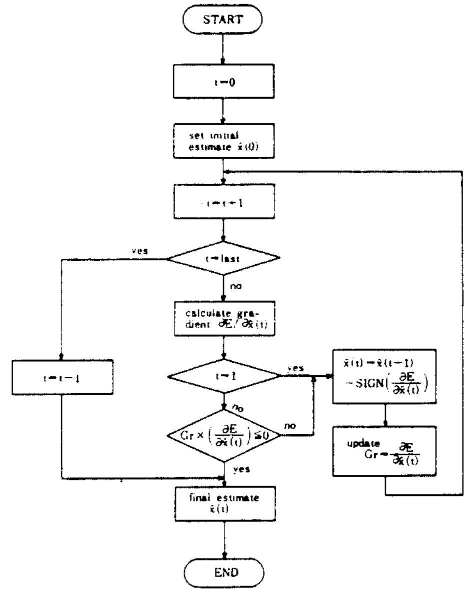


Fig. 6. Tracking strategy in the variational algorithm & represents the displacement vector.

The algorithm developed in this paper can be easily implemented in hardware due to the fact that binary images are used and the gradient calculation can be implemented using TTL logic and accumulator. But in the rotational case, the implementation using TTl logic is impractical because gradient expression for angular velocity in (14) contains the non-binary values. In that case, a high speed digital signal processor such as NEC 7720 or TMS 320 can be used for implementation. But such tragets as airplane, ship, missile, etc. are completely trackable without applying rotational motion tracking algorithm when they are in normal motion, since translational motion tracking only can estimate a motion with a small rotation. Fig. 7 shows the proposed hardware architecture for the translational motion tracking system when wither a positive or a negative scene is used. Finally, the comparison of the proposed algorithm with the non-decomposed one is tabulated quantitatively in table 1. The iteration number shown in the table is induced from the extensive simulation. It is noted that since the

amount of the computation for moving detection is at best half compared with the others, the weighting of 0.5 is included in the number of the moving detection iteration. From the table, the achieved ratio of total computational savings is from
(64x64x5+64x3)/(16x16x16+16x16x16x0.5) = 4 up to
(64x64x8+64x64x8)/(16x16x5+16x16x6+16x 16x16x0.5)=13.5, and on the average
(64x64x5+64x64x5)/(16x16x5+16x16x10+ 16x16x16x0.5) = 7.

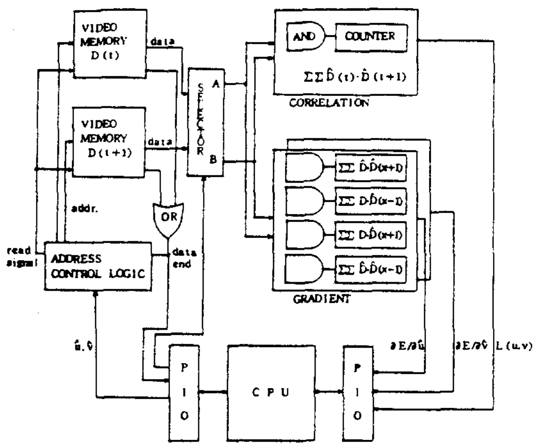


Fig. 7. Hardware architecture for realizing the translational motion tracking.

Table 1. Comparison of the proposed algorithm with the non-decomposed case. (the number in the parentheses denotes the average number of iterations.)

	non-decomposed		proposed	
	size	iteration	size	iteration
correlation (initial estimate)	64x64	5 or 8	16x16	5 or 8
moving detection	X	X	16x16	16
variational algorithm	64x64	3–8(5)	16x16	6–16(10)

VI. Conclusion

We have presented a mathematical approach for a 2-D moving target tracking problem without any assumptions on the trajectory of motion and the shape of a target. Fast algorithm was introduced using a gradient-type approach with the operator formulation. Correlation technique based on a 2-D directed search for the maximum correlation was used to provide an initial estimate for a variational algorithm. Computer simulation using binary images to represent the target texture confirms the validity of the fast algorithm presented herein. The algorithm appears to be a TTL-implementable with a CPU control and efficient approach to the realistic real-time moving target tracking problem. Changes currently taking place in digital technology coupled with research in real-time pattern recognition/image processing algorithm are now making possible highly sophisticated hardware for recognition and tracking purposes.

References

[1] J.O. Limb and J.A. Murphy, "Measuring the Speed of Moving Objects from Television Signals," *IEEE Trans. COMM.*, pp. 474-478, April, 1975.
[2] J.O. Limb and J.A. Murphy, "Estimating the Velocity of Moving Images in Television Signals," *Computer Vision, Graphics, and Image Processing*, vol. 4, Dec., 1975.
[3] C. Cafforio and J. Rocca, "Methods for Measuring Small Displacements of Television Images," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 573-579, Sept., 1976.
[4] C. Cafforio and F. Rocca, "Tracking Moving Objects in Television Images," *Signal Processing*, pp. 133-140, Apr., 1979.
[5] A.N. Netravali and J.D. Robbins, "Motion-Compensated. Television Coding: Part 1," *BSTJ*, pp. 631-670, March, 1979.
[6] A.N. Netravali and J.D. Robbins, "Motion-Compensated Television Coding: Some New Result," *BSTJ*, pp. 1735-1745, Nov. 1980.

[7] R.J. Schalkoff and E.S. McVey, "A Model and Tracking Algorithm for A Class of Video Targets," *IEEE Trans. PAMI,* June, 1979.

[8] R.M. Inigo and E.S. McVey, "CCD Implementation of a Three-Dimensional Video-Tracking Algorithm," *IEEE Trans. PAMI,* vol. PAMI-3, pp. 230-240, March, 1981.

[9] Alton L. Gilbert et al., "A Real Time-Video Tracking System," *IEEE Trans. PAMI,* vol. PAMI-2, pp. 47-56, Jan., 1980.

[10] George R. Legters and Tzay Y. Young, "A Mathematical Model for Computer Image Tracking," *IEEE Trans. PAMI,* vol. PAMI-4, pp. 583-594, nov. 1982.

[11] Duda & Hart, *Pattern Classification and Scene Analysis,* J. Wiely & Sons, 1973.

[12] W.K. Pratt, *Digital Image Processing,* J. Wiely & Sons, 1978.

[13] Ernest I. Hall, *Computer Image Processing and Recognition,* AP, 1979.

[13] D.E. Knuth, *Searching and Sorting, vol. 3, The Art of Computer Programming.* Reading, MA: Addison-Wesley, 1973.

[14] H. Mostafavi and F. Smith, "Image Correlation with Geometric Distortion, Part I: Acquisition Performance," *IEEE Trans. Aerosp. Electron. System,* May, 1978.

[16] H. Mostafavi and F. Smith, "Image Correlation with Geometric Distortion, Part II: Effect on Local Accuracy," *IEEE Trans. Aerosp. Electron, System,* pp. 494-500, May, 1978.

[17] J.R. Jain & A.A. Jain, "Displacement Measurement and Its Application in Interframe Image Coding," *IEEE Trans. COMM.* vol. 19, Dec., 1981.

[18] T.D. Williams, "Depth from Camera Motion in A Real World Scene," *IEEE Trans. PAMI,* vol. PAMI-2, pp. 511-516, nov. 1980.

[19] R.Y. Tsai & T.S. Huang, "Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch," *IEEE Trans. ASSP,* pp. 1147-1152, Dec., 1981.

[20] J.W. Roach & J.K. Aggarwal, "Computer Tracking of Objects Moving in Space," *IEEE Trans. PAMI,* pp. 127-135, Apr., 1979.