Tracking Moving Objects in Image Sequences Using 1-D Trajectory Filter

Soo-Chang Pei, Wen-Yi Kuo, and Wan-Ting Huang

***Abstract*—The three-dimensional (3-D) Fourier transform of the nonaccelerating object in an image sequence lies on a plane passing through the origin of the 3-D frequency domain. Instead of using a 3-D frequency planar filter, we propose a new filter design technique for object tracking using a one-dimensional trajectory filter—in the spatiotemporal domain. This filter can track a moving object with linear or nonlinear trajectory without prior knowledge of the target trajectory. Also, it is easy to design and has a very low computational complexity. More importantly, it can be implemented in real time.**

***Index Terms*—Object tracking, spatiotemporal filtering, trajectory filter.**

I. Introduction

THERE are several filter design techniques for detecting and tracking moving objects in digital image sequences. The conventional discrete Fourier transform (DFT) method [5] involves the three-dimensional (3-D) fast Fourier transform (FFT) of the input image sequence. The disadvantages of this method are that the input sequence must be stored in memory in order for the DFT to be calculated, and the complexity is rather high.

The linear difference equation (LDE) method is proposed in [1]. The continuous 3-D Laplace transform transfer function is designed according to the trajectory of the target. The corresponding discrete 3-D planar-resonant filter is obtained by triple bilinear transformation of the continuous filter. This method has the significant advantage that memory requirements are far less than in the DFT case. However, it is not generally known how to select the coefficients of the LDE to obtain a required stable input–output transfer function. Furthermore, as a result of the triple bilinear transform, the shape of the resonant plane suffers from higher frequency distortion. Thus, the 3-D planar-resonant recursive filter is difficult to design and realize.

Combined DFT and LDE (DFT/LDE), multidimensional filters have been proposed [2], [3]. For the case of 3-D signals in image sequences, the idea of DFT/LDE is only employing DFT filtering over the two-dimensional (2-D) spatial domain and LDE filter on the remaining temporal dimension. They are easier to design than DFT and LDE methods and have advantages in terms of computational efficiency.

Manuscript received May 9, 2005; revised August 9, 2005. This work was supported by the National Science Council, R.O.C., under Contract NSC93-2752-E-002-006-PAE and Contract NSC93-2219-E-002-004. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Pamela C. Cosman.

The authors are with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: pei@cc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/LSP.2005.860550

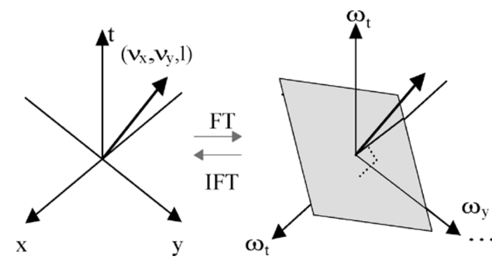

Fig. 1. Fourier transform pair of a line in 3-D space.

In this letter, we propose a new method of filtering moving objects by a one-dimensional (1-D) filter in the spatiotemporal domain. This new method is not involved with Fourier transform and only implemented in one specific direction. It is a simply implemented method and also has very low computational complexity.

II. Objects Moving in Spatiotemporal Domain

A moving object with constant velocity in an image sequence can be viewed as a line signal in the spatiotemporal domain [5]. Define $(v_x, v_y, 1)$ as the linear trajectory vector of the target. The magnitude of the Fourier transform of the trajectory lies in a plane passing through the origin of the 3-D frequency domain with a normal vector $(v_x, v_y, 1)$, as shown in Fig. 1.

Thus, in order to extract the moving object, we need a 3-D planar-resonant filter with a passband enclosing this plane.

III. 3-D Filtering Using 1-D Infinite Impulse Response (IIR) Filter

In Section I, it is mentioned that the LDE filters have higher frequency distortion because of the triple bilinear transformation from s-domain to z-domain. In order to avoid the distortion brought by bilinear transformation, we seek a way to design the desired 3-D planar-resonant filter directly in the z-domain.

Before discussing the 3-D case, we will look at the 2-D case [4]. We begin with a 2-D low-pass filter that varies in one dimension only as

$$H_2(z_1, z_2) = H_1(z_1). \tag{1}$$

Let (z_1, z_2) and $(\hat{z}_1, \hat{z}_2)$ represent original and rotated axes, respectively. Then, consider the transformation

$$z_1 = \hat{z}_1 \hat{z}_2^{\frac{\beta}{\alpha}} \tag{2}$$

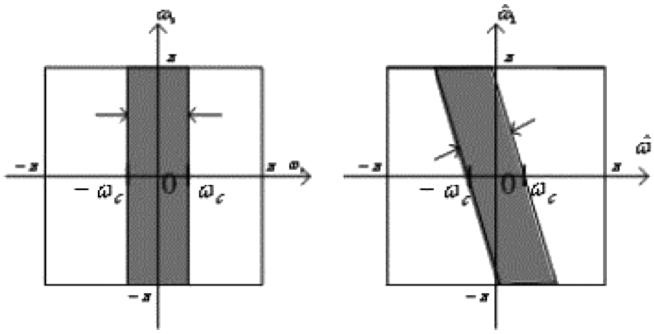

Fig. 2. Frequency responses of low-pass filters of (1) and (5).

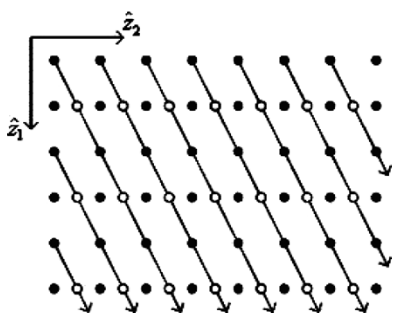

Fig. 3. Recursive direction of transformed filter with $\beta/\alpha = 1/2$.

where α and β are integers. The corresponding frequency transformation is

$$\exp(-j\omega_1 L_1) = \exp(-j\hat{\omega}_1 \hat{L}_1)\exp(-j\hat{\omega}_2 \hat{L}_2) = \exp\left[-j\left(\hat{\omega}_1 \hat{L}_1 + \frac{\beta}{\alpha}\hat{\omega}_2 \hat{L}_2\right)\right] \quad (3)$$

or

$$\omega_1 L_1 = \hat{\omega}_1 \hat{L}_1 + \frac{\beta}{\alpha}\hat{\omega}_2 \hat{L}_2 \quad (4)$$

where L_1 and L_2 are sample intervals, and the angle of rotation is $\theta = \arctan(\beta/\alpha)$.

Now, we have a rotated filter

$$\hat{H}_2(\hat{z}_1, \hat{z}_2) = H_1\left(\hat{z}_1 \hat{z}_2^{\frac{\beta}{\alpha}}\right) \quad (5)$$

which varies in both frequency directions. The responses before and after transformation are shown in Fig. 2. In this figure, $\hat{L}_1$ and $\hat{L}_2$ are assumed to be unity.

In the frequency domain, the effect of the transformation is a rotation of the frequency response with a contraction of the bandwidth as seen in Fig. 2. In the spatial domain, it is equivalent to the rotation of the recursion direction of the digital filter with a new sample interval. The transformed recursion direction scheme of the filter in (1) with $\beta/\alpha = 1/2$ is shown in Fig. 3. Thus, we can use $H_1(z_1)$, which is a 1-D filter to implement 2-D filtering.

Now, let us consider the 3-D case. Following the same procedure in the 2-D case, first we have a 3-D low-pass filter that varies only in one dimension as

$$H(z_m, z_n, z_t) = H(z_m). \quad (6)$$

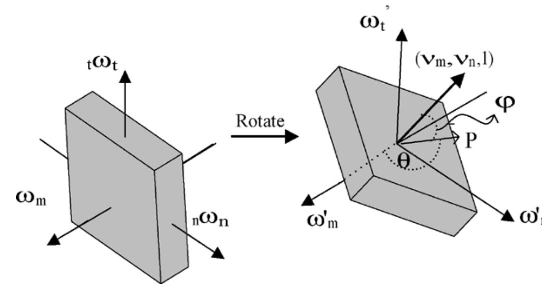

Fig. 4. Rotated 3-D planar-resonant filter in the (z_m, z_n, z_t) plane.

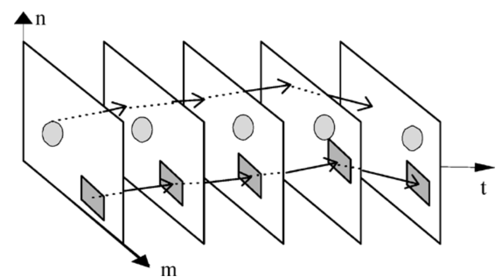

Fig. 5. Set of parallel directional 1-D sequences in the spatiotemporal domain.

If we want to track an object moving in the direction $(v_m, v_n, 1)$, the desired filter can be obtained by rotating the axes of the (z_m, z_n, z_t) plane. Here, we first rotate the filter counterclockwise on the (z_m, z_n) plane with an angle θ and then in the direction of z_t with an angle θ, as indicated in Fig. 4, where P is the projection of $(v_m, v_n, 1)$ on (z_m, z_n) plane, $\theta = \arctan(v_n/v_m)$, and $\phi = \arctan(1/\sqrt{v_m^2 + v_n^2})$. The transformation is shown in the following:

$$\begin{aligned} z_m &= \left(\hat{z}_m^{\cos\theta} \cdot \hat{z}_n^{\sin\theta}\right)^{\cos\phi} \cdot \hat{z}_t^{\sin\phi} \\ &= \hat{z}_m^{\cos\theta .\cos\phi} \cdot \hat{z}_n^{\sin\theta .\cos\phi} \cdot \hat{z}_t^{\sin\phi} \\ &= \hat{z}_m^{\frac{v_m}{\sqrt{v_m^2+v_n^2+1}}} \cdot \hat{z}_n^{\frac{v_n}{\sqrt{v_m^2+v_n^2+1}}} \cdot \hat{z}_t^{\frac{1}{\sqrt{v_m^2+v_n^2+1}}}. \end{aligned} \quad (7)$$

Equivalently

$$z = \hat{z}_m^{v_m} \cdot \hat{z}_n^{v_n} \cdot \hat{z}_t^{1}. \quad (8)$$

$H(\hat{z}_m^{v_m}, \hat{z}_n^{v_v}, \hat{z}_t^{1})$ is a frequency planar-resonant filter with normal vector $(v_m, v_n, 1)$, which matches the frequency response of the moving target. Assuming that the size of the input image is $N \times N$, the filtering can be done by using the original low-pass 1-D filter (6) and by sampling the signal in the direction of $(v_m, v_n, 1)$, i.e., a set of parallel directional 1-D sequences with the same size of the input frame are collected as the parallel input of $N \times N$ 1-D low-pass filters.

As illustrated in Fig. 5, if we want to track the movement of the rectangle, let the recursive direction of the 1-D filter be the trajectory of the rectangle. Thus, after the parallel filtering of every point in the image $N \times N$ 1-D low-pass filters, the filtered output will contain only the rectangle. The circle and the background will be rejected.

How do we design the low-pass 1-D filter we want? It is obvious that the one with sharp cutoff frequency response and low computational complexity will be our choice. The main advantage of an IIR filter [compared to an finite impulse response

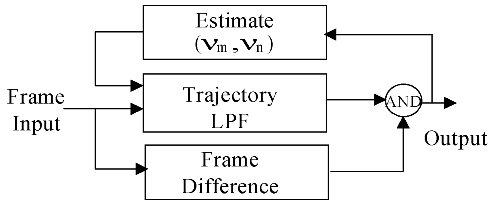

Fig. 6. Proposed 1-D system for object tracking.

(FIR) filter] is a much steeper cutoff frequency response with fewer stages. Thus, with an IIR filter, we can have both sharp cutoff frequency and fewer recursive frames.

The formula of a third-order 1-D IIR filter is

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{i=0}^{3} B_i z^{-i}}{\sum_{j=0}^{3} A_j z^{-j}} = \frac{Y(z)}{X(z)}. \tag{9}$$

With the directional filtering according to the constant trajectory vector, we can write a more compact 1-D difference equation with the input–output as the frame images as follows:

$$\begin{aligned} y(m,n,t) &= \sum_{i=0}^{3} \beta_i x(m - v_m i, n - v_n i, t - i) \\ &\quad - \sum_{j=1}^{3} A_j y(m - v_m j, n - v_n j, t - j), \quad \mathrm{A}_0 = 1 \end{aligned} \tag{10}$$

where $x(m,n,t)$ is the input frame, and $y(m,n,t)$ is the output frame with the desired filtered moving object only. The trajectory $(v_m, v_n, 1)$ is constant here. With the above equation, every point in a frame can be filtered at the same time.

IV. Implementation of 1-D Trajectory Filter in the Spatiotemporal Domain

The proposed system for object tracking in image sequences is shown in Fig. 6. Note that it is a 1-D system where every point at the output is independent of adjacent points and the whole frame can be parallel filtered. This is an advantage for real-time implementation.

The current output is obtained by filtering the current input frame by the trajectory filter proposed above with feedbacks being inputs and outputs of previous frames. A problem arises when there is no previous output, i.e., when $y(m,n,0)$ is desired, what are $y(m,n,-1)$, $y(m,n,-2)$, and $y(m,n,-3)$? The experimental results show that using frame differences as the initial outputs turns out to be a good choice, for example, when it comes to $y(m,n,5)$, let

$$\begin{aligned} y(m,n,2) &= x(m,n,1) - x(m,n,2) \\ y(m,n,3) &= x(m,n,2) - x(m,n,3) \\ y(m,n,4) &= x(m,n,3) - x(m,n,4). \end{aligned} \tag{11}$$

In the above, we have assumed that the targets are moving linearly. Now, we want to extend the system to deal with objects moving with nonlinear trajectories. The main idea is that whether the trajectory is linear or nonlinear, the target between two adjacent frames can be viewed as moving in a piecewise linear trajectory as long as the frame rate is fast enough. Thus, we can simply adapt the trajectory vector $(v_m, v_n, 1)$ by estimating the optical flow of the centroid of the object at every output. Thus, the difference function (10) for our trajectory filter is modified in the following:

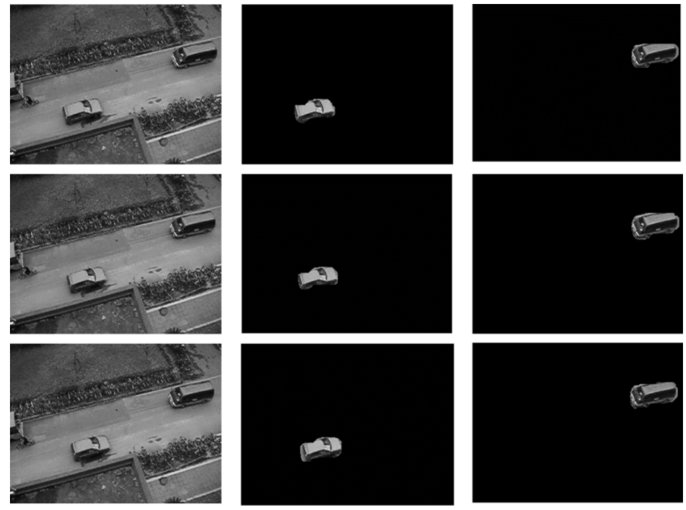

Fig. 7. Filtered inputs and outputs of a sequence with multiple vehicles.

$$\begin{aligned} &y(m,n,t) \\ &\quad = \sum_{i=0}^{3} B_i x\left(m - \sum_{k=0}^{i} v_m(t-k), n - \sum_{k=0}^{i} v_n(t-k), t-i\right) \\ &\qquad - \sum_{j=1}^{3} A_j y\left(m - \sum_{k=0}^{i} v_m(t-k), n - \sum_{k=0}^{i} v_n(t-k), t-j\right), \\ &\qquad A_0 = 1. \end{aligned} \tag{12}$$

In Fig. 6, a block of frame difference is added parallelly to the trajectory filter. Frame difference can be thought of as a high-pass filter that removes the background and the stationary signals of the input frame.

The overall system can be thought of as a low-pass filter in the direction of the target trajectory $(v_{\mathrm{m}}, v_{\mathrm{n}}, 1)$ that operates in parallel with a high-pass filter in the direction (0, 0, 1).

A logical AND is performed after the results of the two blocks. We want the regions that survive through these two filters; if the target stops moving, it will not survive through the high-pass filter and the system will lose the track of the target. Thus, before the AND element, a condition should be added. If the target stops moving, the AND will be neglected.

At the end of the system, the output is obtained by mapping the positions of the output of the two blocks to the original input frame.

V. Experimental Results

In this section, we test the system we proposed with real videos taken from the fifth floor of National Taiwan University's electrical engineering building. All the videos used in this section have frames of size 240 $*$ 320 and frame rate of 15 f/s. A 1-D third-order IIR digital low-pass Butterworth filter with cutoff frequency 0.05π is used as the trajectory filter to track the moving object.

Example 1: The Linear Trajectory Case: We use a real sequence with two cars moving with constant velocities in different directions. In Fig. 7, one is moving from bottom-left to

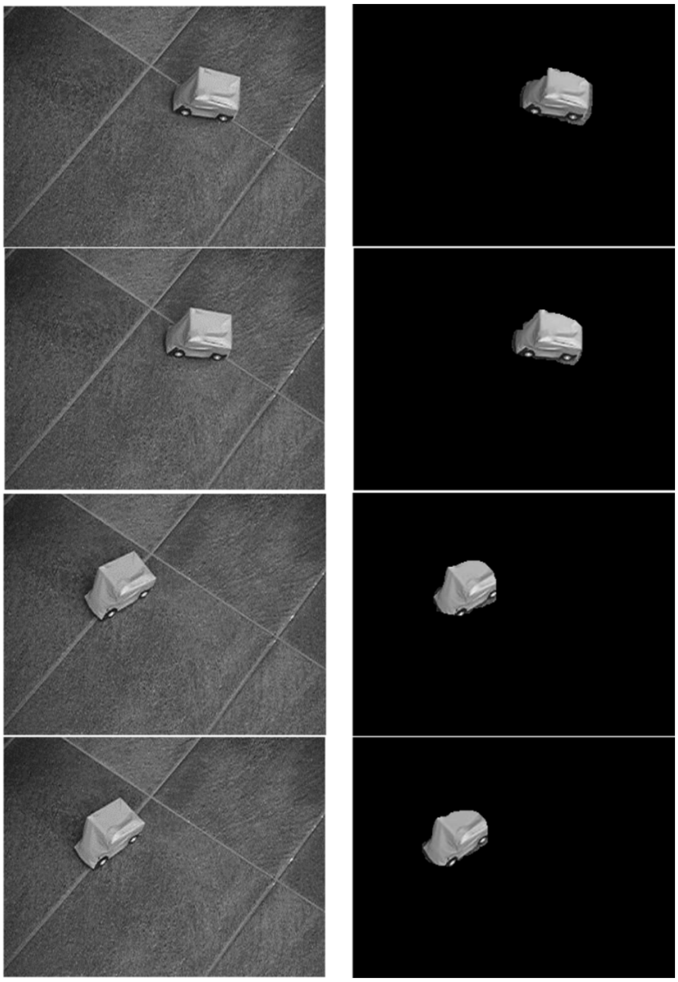

Fig. 8. Filtered inputs and outputs of the object with nonlinear trajectory.

top-right of the frame, and the other is in the opposite direction. Images in the middle column of Fig. 7 are the outputs of filtering along the trajectory of the bottom-left vehicle, and those in the right column are results of the top-left vehicle. It is obvious that the proposed system can successfully track the target and reject the unwanted signals. The edge of the target is a little bit wider, but the overall performance of the system is satisfactory.

Example 2: The Nonlinear Trajectory Case: Another sequence of a vehicle moving with curved trajectory is also tested, and the 9th, 10th, 18th, and 19th frames are shown in Fig. 8 for the reader to observe its curved trajectory; the moving vehicle can still be extracted easily.

VI. Conclusion

In this letter, an effective 1-D trajectory filter for object tracking is proposed. Different from other MD filters, this new method is easy to design and implemented in real time with very low computational complexity. It can be used to track objects moving in linear or nonlinear trajectory without prior knowledge of trajectories. Although the passband of the trajectory filter is only coarsely designed, the frame differences help to improve the accuracy of the system. The experimental results have shown that its performance is satisfactory and useful for real-time practical applications. It is worth mentioning that the space–time approaches recently have been used in computer vision for action recognition [8].

References

[1] L. T. Bruton and N. R. Bartley, "The enhancement and tracking of moving objects in digital images using adaptive three-dimensional recursive filters," *IEEE Trans. Circuits Syst.*, vol. CAS-33, no. 6, pp. 604–613, Jun. 1986.

[2] A. A. Choudhury and L. T. Bruton, "Multidimensional filtering using combined discrete Fourier transform and linear difference equation methods," *IEEE Trans. Circuits Syst.*, vol. 37, no. 2, pp. 223–231, Feb. 1990.

[3] R. W. Issler and L. T. Bruton, "Tracking and enhancement of objects in image sequences using 3-D frequency planar combined DFT/LDE filters," in *Proc. Int. Symp. Circuits Systems*, New Orleans, LA, May 1990, pp. 999–1002.

[4] H. Chang and J. K. Aggarwal, "Design of two-dimensional recursive filters by interpolation," *IEEE Trans. Circuits Syst.*, vol. CAS-24, no. 6, pp. 281–291, Jun. 1977.

[5] B. Porat and B. Friedlander, "A frequency domain algorithm for multiframe detection and estimation of dim targets," in *Proc. IEEE Conf. Computer Vision*, 1987, pp. 591–594.

[6] L. T. Bruton, "Selective filtering of spatio-temporal plane waves using 3D cone filter banks," in *Proc. IEEE Pacific Rim Conf. Communications, Computers, Signal Processing*, vol. 1, Aug. 26–28, 2001, pp. 67–70.

[7] ——, "Three-dimensional cone filter banks," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 50, no. 2, pp. 208–2216, Feb. 2003.

[8] E. Shechtman and M. Irani, "Space-time behavior based correlation," in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, 2005.