Application of Velocity Filtering to Optical-Flow Passive Ranging

YAIR BARNIV
NASA/Ames Research Center

The sequence of images obtained from a passive forward-looking forward-traveling sensor provides the optical flow on which passive ranging is based. Passive ranging is of interest in the context of nap-of-the-Earth helicopter obstacle avoidance. Ranging can be performed for some chosen features or, in principle, for all points in the field-of-view (field based) of the sensor.

Velocity filtering is a track-before-detect algorithm that can be utilized to perform field-based passive ranging. We expand on the theoretical understanding and the performance analysis of the velocity filtering algorithm as applied to optical-flow-based ranging.

I. INTRODUCTION

Passive ranging is an area of considerable interest for applications such as rotorcraft nap-of-the-Earth navigation and spacecraft landing. A forward-looking imaging sensor, such as a forward-looking infrared (FLIR), typically produces the optical flow information necessary for range computation. The principle of passive ranging using optical flow is akin to triangulation; the optical flow is a sequence of angular measurements collected from different locations along the vehicle's flight path with respect to all points in the field of view (FOV).

There are two basic ranging approaches: feature-based and field-based; the first only deals with some chosen objects, while the second regards the scene as a continuum. Feature-based methods suffer from the need to identify objects between successive frames; this is not required with field-based methods because they track *all* points in the FOV. The general method of velocity filtering has been suggested as a potential field-based method.

The relevance of velocity filtering to optical-flow passive ranging stems from the concept of filtering in the three-dimensional Fourier transform (3D FT) domain (e.g., see [1]). Reed [2] suggested the use of 3D matched filtering (MF) for detecting moving constant-velocity objects. The MF equations and the signal-to-noise (SNR) expressions for the MF output in the frequency domain are developed here.

Porat and Friedlander [3] apply filtering in 3D FT for the purpose of detecting moving objects in mosaic-sensor imagery. They use a bank of directional filters where each filter is defined as a plane in the 3D FT domain. In the same context, Fries [4] develops a matched filter in the hybrid two-dimensional FT/one-dimensional temporal (2D FT/1D temporal) domain. His derivation is very general in including camera jitter noise and nonwhite clutter background.

Stotts [5, 6] implements the MF suggested by Reed [2] and obtains experimental results for the detection of an aircraft-like object. He derives an expression for the filtering SNR and loss as a result of velocity mismatch.

Application of the space-domain implementation of velocity filtering to optical flow was suggested [7] as a field-based method. This work uses without proof the space-domain equivalent of filtering in 3D FT.

Smith [8] applies the principles of optical flow directly to obstacle avoidance (feature-based) without the intermediate step of ranging by determining whether an object is in or out of the "tunnel of safe passage". Error analysis and experimental results with real imagery are presented. The performance of the algorithm seems to be very dependent on the scene contents.

Our purpose is to evaluate the performance of the velocity filtering method as applied to optical-flow

Manuscript received March 19, 1990.

IEEE Log No. 9202244.

Author's address: NASA, Ames Research Center, MS 210-9, Moffett Field, CA 94035.

U.S. Government work. Not protected by U.S. copyright.

0018-9251/92/\$3.00 © 1992 IEEE

Fig. 1. Motion translates to time-augmented space.

passive ranging under real-world constraints. We start by reviewing (in Sections II and III) the theory of 3D FT as applied to constant-speed moving points. In Section IV we derive the space-domain shift-and-add algorithm from the general 3D MF formulation. In Section V we find the 2D velocity passband of the filter and develop a variable-velocity (hyperbolic) filter for the particular use in optical flow. We further calculate in Section VI the depth (range) passband and accuracy of this new filter. In Section VII we address the problems associated with pixel interpolation and object expansion and present experimental results.

II. PRELIMINARIES

A. Motion as Time-Augmented Space

The motion of a physical point in one or two dimensions can be described as a line or a surface in a space which is augmented by the time dimension. Fig. 1 shows an example of a constant-velocity point in the (x, y) plane that describes a straight line in the 3D space (x, y, t) . This line can either be given in a parametric form, i.e., $(x_0 + v_x t, y_0 + v_y t, t)$, or as an intersection of two planes $x = x_0 + v_x t$, and $y = y_0 + v_y t$. The parametric form is easily related to the vector $(v_x, v_y, 1)$ which is parallel to the line.

B. Dirac Delta Function

The 1D Dirac Delta Function (DF) can be defined by the properties

$$\begin{cases} \delta(t) = 0, & t \neq 0 \\ \int_{-\infty}^{+\infty} \delta(t) dt = 1 \end{cases} \quad (1)$$

Higher dimensions of the Dirac DF can be similarly defined. The DF is mostly used because of its sifting

property; in 2D, for example it is given by

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \delta^2(x - x_1, y - y_1) dx dy = f(x_1, y_1) \quad (2)$$

where the superscript "2" is used to differentiate between the 1D and 2D DFs. A $\delta^2(x, y)$ factor in an integrand can be replaced by $\delta(x)\delta(y)$ because the 2D integral becomes 1D separable. This is why we can use the equality (see [9])

$$\delta^2(x, y) = \delta(x)\delta(y). \quad (3)$$

In the sequel we use the concise notation

$$\delta^2(\bar{r}) \equiv \delta^2(x, y). \quad (4)$$

C. 3D Fourier Transform of a Moving Point

A functional relationship between location and time of the form $x(t), y(t)$ describes the trajectory of a moving point. In order to also refer to the amplitude or gray level of the point as dependent on location and time, we use a moving 2D DF. The amplitude function of a unit-amplitude point having a fixed velocity vector $\bar{v} \equiv (v_x, v_y)$ can be described as

$$\begin{aligned} s(x, y, t) &= \delta(x - x_0 - v_x t) \delta(y - y_0 - v_y t) \\ &= \delta^2(\bar{r} - \bar{r}_0 - \bar{v}t) \end{aligned} \quad (5)$$

so that, at any given time, its 2D integral over the (x, y) plane is unity.

The 3D FT of $s(x, y, t)$ is

$$\begin{aligned} S(\bar{k}, \omega) &= \iiint \delta^2(\bar{r} - \bar{r}_0 - \bar{v}t) \\ &\quad \times \exp\{-j\omega t\} \exp\{-j\bar{k} \cdot \bar{r}\} dt d\bar{r} \\ &= 2\pi \exp\{-j\bar{k} \cdot \bar{r}_0\} \delta(\omega + \bar{k} \cdot \bar{v}) \end{aligned} \quad (6)$$

where the vector $\bar{k} \equiv (k_x, k_y)$ denotes spatial frequencies in radians/meter, $d\bar{r} \equiv dx dy$, and dot product denotes vector scalar multiplication. The integrals in (6) and in all subsequent equations are between $\pm\infty$ unless stated otherwise. The result of (6) is composed of a phase term which only depends on the starting point $\bar{r}_0$ at $t = 0$, and a 1D DF which can be written in a more revealing form as

$$\delta(v_x k_x + v_y k_y + \omega). \quad (7)$$

Since the DF is active (goes to infinity) when its argument equals zero, we get

$$v_x k_x + v_y k_y + \omega = 0 \quad (8)$$

which is the equation of a plane in 3D FT that passes through the origin. The coefficients of the plane equation yield the normal vector, $(v_x, v_y, 1)$, which is parallel to the line that describes the moving point in the (x, y, t) domain from which we started.

Fig. 2. 3D FT of moving point is plane.

Thus, if we overlay the (x, y, t) and the (k_x, k_y, ω) coordinate systems as shown in Fig. 2, then the line describing the moving point in (x, y, t) is perpendicular to the plane in the 3D FT domain which represents its FT. The figure also shows the vectorial equation of the plane as obtained from (8) by replacing the first two terms with the scalar vector product $\bar{k} \cdot \bar{v}$. It is seen that the "height", or ω , of any point on the plane is given by $\omega = -\bar{k} \cdot \bar{v}$.

III. 3D FOURIER TRANSFORM OF A FINITE-TIME MOVING POINT

So far we have assumed continuous data that exists between $\pm\infty$ in all three dimensions (x, y, t) . In reality we have to require causality and finite-time duration; this can be expressed by using a rectangular-pulse amplitude function of the form

$$a(t/T) = \begin{cases} 1, & \text{for } 0 \leq t \leq T \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

and assuming that a point having this time-limited amplitude is moving with a velocity $\bar{v}$ from the initial location $\bar{r}_0 \equiv (x_0, y_0)$. This "signal" can be written as

$$s(\bar{r}, t) = a(t/T) \delta^2(\bar{r} - \bar{r}_0 - \bar{v}t) \quad (10)$$

and its FT given by

$$S(\bar{k}, \omega) = \iiint a(t/T) \delta^2(\bar{r} - \bar{r}_0 - \bar{v}t) \times \exp\{-j\omega t\} \exp\{-j\bar{k} \cdot \bar{r}\} dt d\bar{r}. \quad (11)$$

Performing first the spatial and then the temporal integration, we get

$$\begin{aligned} S(\bar{k}, \omega) &= \int_{-\infty}^{+\infty} a(t/T) \exp\{-j\omega t\} \exp\{-j\bar{k} \cdot (\bar{r}_0 + \bar{v}t)\} dt \\ &= T \exp\{-j\bar{k} \cdot \bar{r}_0\} \exp\{-j(\omega + \bar{k} \cdot \bar{v})T/2\} \\ &\quad \times \text{sinc}[(\omega + \bar{k} \cdot \bar{v})T/2\pi], \end{aligned} \quad (12)$$

where

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}. \quad (13)$$

Equation (12) reverts to what we already know, because, when $T \rightarrow \infty$, (12) is the same as (6) except for the temporal phase term in (12) which would disappear for an $a(t/T)$ defined to be symmetrical around $t = 0$. We notice the plane equation in the argument of the sinc function in (12) except that, now, it is not an ideal flat zero-width plane but rather a plane that bulges into a pancake whose thickness is determined by the sinc function.

IV. MATCHED FILTERING APPLIED TO A MOVING POINT

In this section we derive the MF for a moving point in the FT and the spatial/temporal domains.

A. Matched Filtering in FT Domain

An MF is the filter that optimally enhances the amplitude of the signal when it is embedded in Gaussian noise. The well-known form of the MF for a signal embedded in white Gaussian noise is [10, 11]

$$H(\bar{k}, \omega) = \frac{S^*(\bar{k}, \omega)}{N_0} \exp\{-j\omega T\} \quad (14)$$

where N_0 is the constant-spectrum value of the white noise, star denotes complex conjugation, and the MF output is sampled at $t = T$. The temporal phase term ensures that the filter is realizable. Passing the moving-point signal (plus noise) through the MF of (14) amounts to taking the 3D inverse transform of the product,

$$\begin{aligned} G(\bar{k}, \omega) &= S(\bar{k}, \omega) H(\bar{k}, \omega) \\ &= T^2 \text{sinc}^2[(\omega + \bar{k} \cdot \bar{v})T/2\pi] \exp\{-j\omega T\} / N_0 \end{aligned} \quad (15)$$

which yields

$$\begin{aligned} g(\bar{r}, t) &= \frac{T^2}{(2\pi)^3 N_0} \iiint \text{sinc}^2[\cdot] \\ &\quad \times \exp\{j[\bar{k} \cdot \bar{r} + \omega(t - T)]\} d\bar{k} d\omega. \end{aligned} \quad (16)$$

Substituting $x = (\omega + \bar{k} \cdot \bar{v})/2\pi$, we get

$$\begin{aligned} g(\bar{r}, t) &= \frac{T^2}{(2\pi)^2 N_0} \int \text{sinc}^2(xT) \exp\{j2\pi x(t - T)\} dx \\ &\quad \times \iint \exp\{j\bar{k} \cdot [\bar{r} - \bar{v}(t - T)]\} d\bar{k} \\ &= T/N_0 \delta^2(\bar{r} - \bar{v}(t - T)) \cdot \text{tra}(t/T) \end{aligned} \quad (17)$$

Fig. 3. Triangle function.

where the $\text{tra}(t/T)$ is the triangle function shown in Fig. 3.

The output of an MF has to be sampled at $t = T$, or read out of the appropriate frame when the data is discrete in time. At that instant we get a point detection at the origin, $\bar{r} = 0$, in the form of a $\delta^2(\bar{r})$ having amplitude proportional to T . The peak of the MF output is at the origin because we are using spatially noncausal images which exist for both positive and negative values of $\bar{r}$ components. As one would expect, (17) includes the factor T so that the result of MF operation is linearly proportional to the observation time.

B. MATCHED FILTERING IN THE SPATIAL/TEMPORAL DOMAIN

Let us now consider the MF operation in the spatial/temporal domain. Using (12) in (14), we get the 3D impulse response

$$\begin{aligned} h(\bar{r}, t) &= F^{-1}[H(\bar{k}, \omega)] \\ &= \frac{T}{(2\pi)^3 N_0} \iiint \exp\{j[\bar{k} \cdot \bar{r}_0 - \omega T/2 + \bar{k} \cdot \bar{v} T/2]\} \\ &\quad \times \text{sinc}[(\omega + \bar{k} \cdot \bar{v})T/2\pi] \exp\{j(\bar{k} \cdot \bar{r} + \omega t)\} d\bar{k} d\omega \end{aligned} \quad (18)$$

which leads to

$$h(\bar{r}, t) = 1/N_0 \delta^2(\bar{r} + \bar{r}_0 - \bar{v}(t - T)) a(t/T). \quad (19)$$

We now want to understand the operation of convolving any *general* 3D imagery, $I(\bar{r}, t)$ —not necessarily that of a point object—with the impulse response of (19). The resulting imagery can be written as

$$\begin{aligned} I_c(\bar{r}, t) &= I(\bar{r}, t) * h(\bar{r}, t) \\ &= \iiint I(\bar{\rho}, \tau) h(\bar{r} - \bar{\rho}, t - \tau) d\tau d\bar{\rho} \end{aligned} \quad (20)$$

which, in this case, is

$$\begin{aligned} I_c(\bar{r}, t) &= 1/N_0 \iiint I(\bar{\rho}, \tau) \delta(\bar{r} - \bar{\rho} + \bar{r}_0 - \bar{v}(t - \tau - T)) \\ &\quad \times a[(t - \tau)/T] d\tau d\bar{\rho}. \end{aligned} \quad (21)$$

The temporal limits of integration in (21) are determined by the period of existence of the imagery which is assumed to be $(0, T)$. This time window multiplied by the one that appears explicitly in the integrand, makes a combined window:

$$\begin{aligned} W(t, \tau, T) &\equiv a(\tau/T) a[(t - \tau)/T] \\ &= \begin{cases} 1, & \text{for } 0 \leq \tau \leq t \text{ and } 0 \leq t \leq T \\ 1, & \text{for } t - T \leq \tau \leq T \text{ and } T \leq t \leq 2T \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (22)$$

The integral over τ between 0 and t of this window equals $T \text{tra}(t/T)$. Thus, after spatial integration, (21) can be reduced to

$$I_c(\bar{r}, t) = 1/N_0 \int_0^t I[\bar{r} + \bar{r}_0 - \bar{v}(t - \tau - T), \tau] d\tau \quad \text{for } 0 \leq t \leq T. \quad (23)$$

Sampling (23) at $t = T$,

$$I_c(\bar{r}, T) = 1/N_0 \int_0^T I[\bar{r} + \bar{r}_0 + \bar{v}\tau, \tau] d\tau. \quad (24)$$

This is implemented by summing up (integrating) shifted versions of the original images starting with the first at $\tau = 0$, i.e., $I[\bar{r} + \bar{r}_0, 0]$ and ending with the last at $\tau = T$, i.e., $I[\bar{r} + \bar{r}_0 + \bar{v}T, T]$. The last frame has to be shifted towards the origin by the vector $\bar{v}T$ compared with the first one. The additional shift towards the origin of size $\bar{r}_0$, which is common to all frames, is a result of the particular MF we are using which was matched to a point at $\bar{r}_0$. That shift would bring the point $\bar{r}_0$ to the origin. Since we actually want to *preserve* the original location of any general point, $\bar{r}_0$, in the resultant image, we want to shift this point back to where it came from. This is accomplished by discarding the $\bar{r}_0$ term from (19). The final effect is that *all* detected points (pixels) after MF stay in their initial original locations.

The direction of the velocity-related spatial shift, $\bar{v}t$ in (24), is such that, if a pixel goes away from the origin, it is shifted backward as if to cancel out its velocity. The effect of (24) is thus to align the objects having velocity $\bar{v}$ so that they appear *stationary* and collocated in their initial (general) locations in all frames. Once all frames are aligned, they are integrated over the total time T . Only those objects having velocity $\bar{v}$ will integrate coherently; others having different velocities will appear smeared in the resultant image.

So far we have discussed the 3D space as a continuous volume. In reality we deal with finite-size images, say, of $N \times N$ pixels that appear every T_f s. This limits the effective part of the image that we can work with to the intersection of the frame with itself shifted by $-\bar{v}T$.

V. VELOCITY RESOLUTION OF A 3D MATCHED FILTER

A. General Derivation

A general unknown imagery may contain all possible velocity vectors; thus it will be passed through a bank of filters tuned to all the discrete velocity vectors that span the required range. To cover that range economically, we first have to find the resolutions of a single filter in both angle and length of the velocity vector. We start from the MF result of (14), where we use an MF tuned to $\bar{v}$ for an object moving at a different velocity $\bar{v}_0$ for which we use (12) with $\bar{v}_0$ replacing $\bar{v}$. Rewriting (15) for this case,

$$G(\bar{k}, \omega) = T^2 \text{sinc}\{(\omega + \bar{k} \cdot \bar{v}_0)T/2\pi\} \text{sinc}\{(\omega + \bar{k} \cdot \bar{v})T/2\pi\} \times \exp(j\bar{k} \cdot \Delta \bar{v}T/2) \exp(-j\omega T)/N_0 \quad (25)$$

where

$$\Delta \bar{v} \equiv \bar{v} - \bar{v}_0.$$

The 3D inverse FT, as in (16), sampled at $t = T$ is

$$g(\bar{r}, T) = \frac{T^2}{(2\pi)^3 N_0} \int \text{sinc}\{(\omega + \bar{k} \cdot \bar{v}_0)T/2\pi\} \times \text{sinc}\{(\omega + \bar{k} \cdot \bar{v})T/2\pi\} d\omega \iint \times \exp\{j\bar{k} \cdot (\bar{r} + \Delta \bar{v}T/2)\} d\bar{k}. \quad (26)$$

Using Parseval's Theorem,

$$\int f^*(t)g(t) dt = \frac{1}{2\pi} \int F^*(\omega)G(\omega) d\omega \quad (27)$$

and the FT pair,

$$a[(t + T/2)/T] \exp\{-jxt\} \xrightarrow{\text{FT}} T \text{sinc}\{(\omega + x)T/2\pi\} \quad (28)$$

the first integral over ω in (26) is replaced by

$$\begin{aligned} & \frac{1}{2\pi} \int T \text{sinc}[\cdot] T \text{sinc}[\cdot] d\omega \\ &= \int a[(t + T/2)/T] \exp\{j\bar{k} \cdot \bar{v}_0 t\} \\ & \quad \cdot a[(t + T/2)/T] \exp\{-j\bar{k} \cdot \bar{v} t\} dt \\ &= \int_{-T/2}^{T/2} \exp\{-j\bar{k} \cdot \Delta \bar{v} t\} dt \\ &= T \text{sinc}\left(\frac{\bar{k} \cdot \Delta \bar{v} T}{2\pi}\right). \end{aligned} \quad (29)$$

Now (26) reads:

$$g(\bar{r}, T) = \frac{T}{4\pi^2 N_0} \iint \text{sinc}\left(\frac{\bar{k} \cdot \Delta \bar{v} T}{2\pi}\right) \times \exp\{j\bar{k} \cdot (\bar{r} + \Delta \bar{v}T/2)\} d\bar{k}. \quad (30)$$

Fig. 4. Line-segment DF.

Skipping some simple but cumbersome derivation, the final result is

$$g(\bar{r}, T) = 1/N_0 \delta(\Delta v_x y - \Delta v_y x), \quad (31)$$

$$-T\Delta v_x \leq x \leq 0$$

which is a 2D line DF on the bounded segment shown in Fig. 4. The total area under this segment is

$$\int_{-T\Delta v_y}^0 \int_{-T\Delta v_x}^0 \delta(\Delta v_x y - \Delta v_y x) dx dy = T \quad (32)$$

and it is distributed evenly along it. We notice that the dimensions of $\delta(\Delta v_x y - \Delta v_y x)$ are $\delta([m/s][m]) = [s]\delta([m^2])$ which is the same as $t\delta^2(\bar{r})$ in (17), that is, $[s]\delta([m])\delta([m])$ or $[s][m^{-2}]$.

The line DF of (31) reduces to that of (17) evaluated at $t = T$ when $\Delta v_x = 0$ (which implicitly also makes $\Delta v_y = 0$). This means that, because of the velocity mismatch, the point detection of (17) smears over the length of the line segment of (31) which is

$$l = T\sqrt{\Delta v_x^2 + \Delta v_y^2} = T|\Delta \bar{v}| \quad (33)$$

i.e., the moving pixel is now detected in the form of a smeared line segment instead of a single point at the origin as it was for $\Delta \bar{v} = 0$. This causes location uncertainty along with a loss in gain.

Note that, because (33) depends on the absolute value of the speed mismatch, this equation tells us the effect of both: mismatch in *speed* as well as mismatch in the velocity-vector *direction*. For mismatch in direction of size $\Delta\alpha$,

$$|\Delta \bar{v}| = 2|\bar{v}|\sin(\Delta\alpha/2). \quad (34)$$

Also note that it is the absolute value of the velocity vector mismatch that appears in the result of (33) and not the *relative* velocity vector mismatch.

The gain loss is simpler to understand when we think in terms of discrete pixels rather than in terms of continuous spatial distances. For $l = p$ along, say, the x -axis (p denotes the side width of a pixel) the peak smears over 2 pixels so that the loss is by a factor of 2 or -6 db. We can rewrite (33) for the discrete case as

$$l_p \approx N|\Delta \bar{v}_p| \quad (35)$$

where N is the total number of frames (replacing the integration time T), and $\bar{v}_p$ is the velocity measured in pixels/frame. The expression of (35) is inaccurate for small $|\Delta \bar{v}_p|$ values (in the order of $1/N$ pixel/frame)

Fig. 5. Smearing detected pixel in general direction.

because of the pixels' discretization. The derivation for that range of values is given below.

The gain loss results from smearing a 1-pixel area over more than a single pixel as shown in Fig. 5 where a pixel is smeared by shifting it $(\Delta x, \Delta y)$. For gain losses in the order of -3 dB, the peak of the MF output still occurs in the nominal pixel denoted by $(0,0)$. Thus, we want to find the intersection areas between the shifted (smeared) pixel and the nominal pixel at each one of the N frames and sum them up.

Counting frames from zero, i.e., $0 \leq n \leq N-1$, the shift in pixel-width units at frame n is given by

$$\Delta x = n|\Delta \bar{v}_p| \cos \alpha; \quad \Delta y = n|\Delta \bar{v}_p| \sin \alpha. \quad (36)$$

The contribution of frame n into the nominal pixel is $(1 - n\Delta x)(1 - n\Delta y)$ so that the loss (to be denoted by L) can be written as the summation of the contributions of all N frames divided by the total nominal contribution of N pixels to normalize it to unity, i.e.,

$$\begin{aligned} L &= 1/N \sum_{n=0}^{N-1} (1 - n|\Delta \bar{v}_p| \cos \alpha)(1 - n|\Delta \bar{v}_p| \sin \alpha) \\ &= 1/N \sum_{n=0}^{N-1} [1 - n|\Delta \bar{v}_p|(\cos \alpha + \sin \alpha) \\ &\quad + n^2|\Delta \bar{v}_p|^2 \sin(2\alpha)/2]. \end{aligned} \quad (37)$$

For small $|\Delta \bar{v}_p|$ and large N we approximate the above sum by an integral, that is,

$$\begin{aligned} L &\approx 1/N \int_0^N (\cdot) dn = 1 - N/2|\Delta \bar{v}_p|(\cos \alpha + \sin \alpha) \\ &\quad + N^2/6|\Delta \bar{v}_p|^2 \sin(2\alpha). \end{aligned} \quad (38)$$

For a general loss of $L \leq 1$, we solve the quadratic equation

$$l_p^2/6\sin(2\alpha) - l_p/2(\cos \alpha + \sin \alpha) + (1 - L) = 0 \quad (39)$$

where $l_p = N|\Delta \bar{v}_p|$ is the length of smear in pixels as

Fig. 6. Smear length for -3 dB and -6 dB loss as function of smear direction.

defined by (35). The solution is

$$l_p = \frac{1}{2\sin(2\alpha)} [3(\cos \alpha + \sin \alpha) - \sqrt{9 + \sin(2\alpha)(24L - 15)}]. \quad (40)$$

When there is no loss, or $L = 1$, we find from (40) that there is no smear, or $l_p = 0$ as expected.

We can now use (40) to calculate $|\Delta \bar{v}_p|$ for a -3 dB loss by evaluating it for $L = 1/\sqrt{2}$ and dividing the resulting l_p by N . From (35), a large N yields a small $|\Delta \bar{v}_p|$ for any given fixed loss. Fig. 6 shows the relationship between the total smear length during N frames l_p , and the direction of the velocity mismatch $\Delta \bar{v}$ for the cases of -3 dB and -6 dB losses. We see that, allowing a -6 dB loss, the smear length is 1 pixel when its direction is along either the x or the y axes. We already know that, because it corresponds to the simple case where the detection pixel (or the peak) smears evenly over the area of two pixels. For the -6 dB loss, if the smear is in any diagonal direction (45°), its length is only allowed to be 0.9 pixels. For the -3 dB case, the allowed smear length along the axes is 0.58 and, in a diagonal direction, is 0.47 pixel.

When the smear length is larger than 1 pixel, the loss is always larger than 6 dB. This is why we regard the high-loss region of l_p as a *rejection* rather than a *loss* region since we refer to the velocities outside of the tuned velocity filter that we want to reject. An approximate relationship for the rejection capability of the velocity filter in this region can be given by

$$\begin{aligned} \text{loss} &\approx \frac{1}{1 + N|\Delta \bar{v}_p|}, \quad \text{for } |\Delta \bar{v}_p| \leq 1 \text{ pixel/frame} \\ &\approx 1/N, \quad \text{otherwise.} \end{aligned} \quad (41)$$

For example, for $N = 32$ frames, the nominal gain for a perfect velocity match is 32. If the speed changes by 0.1 pixel/frame (from whatever it was), $|\Delta \bar{v}_p| = 0.1$, and $l_p = 3.2$ pixels. The peak will smear over 4.2 pixels, i.e., a loss of 12.5 dB. Looking at it another way,

Fig. 7. Smear lengths shown in polar coordinate system.

the gain separation between pixels that have an 0.1 pixels/frame speed difference is 12.5 dB.

B. Required Number of Velocity Filters

Here we use the allowed smear lengths to estimate the number of velocity filters required to cover any given range of speeds, say, 0 to $V_{\max}$, and all velocity directions.

Fig. 7 shows the allowed smear for the -3 dB and -6 dB cases in a polar coordinate system. For a given number of integrated frames N , dividing the pixel values of the figure by N allows reading the axes in terms of the Δv_x and Δv_y components of the velocity vector mismatch. With that interpretation, the vector $\Delta \bar{v}_p$ can be drawn from the center of the figure to any point along the closed curve. We can thus regard the closed curve as enclosing a 2D area in the *velocity domain* (v_x, v_y). If the velocity mismatch vector falls inside this area, the mismatch loss is equal or less than the one described by the curve.

The above regions (approximated and referred to as circles) can be used to construct a bank of filters so as to cover any required region in the velocity plane in a complementary way. Fig. 8 shows how one could cover the speed range of 0 to $V_{\max}$. The number of circles (each standing for a velocity filter tuned to its center velocity vector) is

$$N_f \approx V_{\max}^2 / |\Delta \bar{v}_p|^2. \quad (42)$$

For example, if $V_{\max} = 1$ pixels/frame, and $|\Delta \bar{v}_p| = 0.05$ pixels/frame (for a 3 dB loss), we need 400 filters to cover the required range of velocity vectors.

VI. APPLICATION TO THE OPTICAL FLOW PROBLEM

A. General

In this section we apply the general theory to the optical-flow problem which, under the assumption of a known focus-of-expansion (FOE), is only a 1D

Fig. 8. Covering velocity plane with complementing filters.

Fig. 9. Overlay of velocity and image planes for velocity-filters coverage.

problem, at least in terms of image coordinates. Thus, we only have to cover a 1D strip in the velocity vector plane that corresponds to a strip in the image plane. In other words, we can divide the frames into some fixed number of angular sectors (all having the FOE as their vertex) and apply a bank of velocity filters that cover a linear strip in the velocity-vector plane which is oriented the same as the sector in the image plane. This method of coverage is shown in Fig. 9 which is an overlay of the velocity and image planes. The -3 dB contours of the passband of the velocity filters are shown as circles.

Each velocity filter in Fig. 9 is applied to all pixels underneath it. The product of number of filters and pixels, which determines the computational load, stays constant if we replace this filtering process by another one in which every pixel has its own dedicated filter centered on it.

The optical flow problem is 1D in the sense that the velocity *direction* is known for each pixel; however, it has one additional dimension we have ignored so far which is the sensor/object range (henceforth referred to as depth). Because a pixel represents an object at some unknown depth, it corresponds to a *range* of velocities in the image plane that needs to be covered with velocity filters. We show that the distance of the pixel from the FOE as a function of time has a hyperbolic profile defined by the unknown depth parameter. Thus, we suggest to use “depth filters” rather than constant-velocity filters. In the sequel we use “filter” to mean “depth filter”.

B. Derivation of the Depth Filter

In the forward-looking optical flow problem each pixel—representing the angular squint from the axis

of the sensor—moves in the image plane radially away from the FOE. Assuming, for simplicity, that the flight velocity vector coincides with the axis of the sensor, let us set up the (x, y, z) coordinate system at the sensor so that the image plane lies in the (x, y) plane, the x -axis parallel to the ground, and the z axis points in the flight direction. We denote the location of the object in the image plane by (θ_x, θ_y) to correspond with the (x, y) axes.

With that, we can express the projection of a stationary object at (x, y, z) onto the image plane by its image-plane coordinates (in units of pixels)

$$\theta_x = fx/z, \quad \theta_y = fy/z \quad (43)$$

where f is the imaging-device focal length in pixels. The time derivatives of the projection coordinates are

$$\dot{\theta}_x = v\theta_x/z; \quad \dot{\theta}_y = v\theta_y/z \quad (44)$$

where v is the speed of the sensor, z denotes the sensor-to-object depth, and $\dot{z} = -v$. Thus, for any θ (irrespective of direction in the image plane)

$$\dot{\theta} = fr/z; \quad \dot{\theta} = v\theta/z \quad (45)$$

where $\theta = \sqrt{\theta_x^2 + \theta_y^2}$ and $r = \sqrt{x^2 + y^2}$. Also, using the initial values z_0 and θ_0 at $t = 0$,

$$\theta = \frac{\theta_0 z_0}{z_0 - vt}; \quad \dot{\theta} = \frac{v\theta}{z_0 - vt}. \quad (46)$$

Based on the above equations we prepare a bank of contiguous filters, each tuned to a hyperbolic time function determined (for a given speed v) by θ_0 and z_0 . The passband of such a filter in terms of depths can be found by writing the difference between two θ s calculated from (46) for a fixed θ_0 and v but with two different values for z_0 ; one is the tuned-for z_0 itself and the other is some general z_b .

We define the passband by requiring that the absolute value of the difference between the above two θ s after $T = N/R_f$ is less than some (positive) l_p , or

$$-l_p \leq \theta_b(T) - \theta(T) = \theta_0 \left[\frac{z_b}{z_b - vT} - \frac{z_0}{z_0 - vT} \right] \leq l_p. \quad (47)$$

When solved for the passband ends in terms of sensor/object depths, this inequality translates to

$$z_{bl} \leq z_b \leq z_{bh} \quad (48)$$

where the low end of the passband is

$$z_{bl} = \frac{z_0(1 + \theta_0/l_p) - vN/R_f}{\frac{z_0 R_f}{vN} + \theta_0/l_p - 1} \quad (49)$$

and the high end is

$$z_{bh} = \frac{z_0(1 - \theta_0/l_p) - vN/R_f}{\frac{z_0 R_f}{vN} - \theta_0/l_p - 1}. \quad (50)$$

Fig. 10. Time-trajectories of pixel receding from FOE for nominal and extreme object/sensor ranges.

As an example, with $v = 10$ m/s, $N = 16$ frames, $R_f = 32$ frames/s, $\theta_0 = 10$ pixels, $z_0 = 50$ m, and $l_p = 0.5$ pixel, we find $z_{bl} = 36.034$ m and $z_{bh} = 86.818$ m. Thus, if we tune a filter for a pixel at distance of 10 pixels from the FOE and for a nominal object depth of 50 m, the filter passband is between 36 and 86.8 m.

Fig. 10(a) shows the time-evolution of the pixel used in the example for the nominal, high, and low object/sensor depths. After 16 frames the pixel separation between the nominal (center) graph and those for the high and low depths are $\pm l_p = \pm 0.5$ pixels. Fig. 10(b) shows a similar case except that $\theta_0 = 100$, and the depth bandwidth of the filter reduces to the range 48.062 m to 52.120 m. It is seen that many more filters are needed to cover any given range of depths for pixels which are farther from the FOE.

In Fig. 11 we show the dependence of the number of filters needed to cover a given range of depths (40–120 m) on the distance of the pixel from the FOE, θ . The upper part of the figure shows horizontal lines with tic marks that mark the centers and passband of all (overlapping) filters. Noticing that the solution of z_0 in terms of z_{bh} from (50) is the same as that of z_{bl} in terms of z_0 in (49), we always start from the maximum depth and apply (49) recursively until the solution falls under the minimum specified depth.

Fig. 11. Number of filters required to cover given range of sensor/object distances.

For the second line from the bottom, as an example, we find two filters centered at 77 m and 46 m and overlapping at 57 m which is their -3 dB points. The total required number of filters is (approximately) half the number of tic marks as shown in the lower part of the figure. The density of the filters increases with θ almost linearly and decreases with the depth. Note that the passband of the filters also represent depth accuracy. As expected, the accuracy improves with increasing projection values (large θ) and decreasing depth.

Preparing figures similar to Fig. 11 but with different parameters (not shown here), we noticed the following general behavior.

- 1) The number of filters increases with the speed v .
- 2) The density of the filters increases as the depth decreases.
- 3) The number of filters decreases as l_p increases because we allow each filter to have a wider passband, thus decreased gain at the passband ends.
- 4) The number of filters increases with the number of integrated frames N , because increasing N makes a higher gain, and thus a narrower filter (for the same l_p).

5) From (49), increasing the rate of the frames, R_f , has the opposite effect of increasing N .

VII. ALGORITHM IMPLEMENTATION

A. Main Routines

The algorithm is implemented as follows. We first use (49) repeatedly for all $1 \leq \theta_0 \leq \theta_{\max}$ to get the depth centers of all filters required to cover a given range of sensor-object depths. Then, we apply the following algorithm.

Do For All Pixels

Get polar coordinates of pixel with respect to the FOE (distance and angle from the horizontal axis) in the image plane.

Do For All Filters Applicable to This Pixel

Do For All Frames

Calculate expected pixel distance from FOE at this frame using current filter (pixel travels along a ray from FOE).

Read value of this pixel at this frame.

Store value to calculate statistics.

End Frames' Loop

Calculate variance of pixels' values which were picked up from the frames according to the filter in use.

Record the minimum variance among all filters up to current one and the corresponding depth for this pixel.

End of Depth Filters Loop

End Of Pixels' Loop

The criterion for choosing the depth for a pixel is that its gray-level must be approximately constant as it shows up in different locations in different frames. Thus, we choose the filter for which the ratio of variance/average over the frames is the minimum.

B. Preprocessing

Practical scenes contain objects composed of many pixels, where each object is defined by a nearly constant gray level. If we try to track any particular pixel found in the inside of an object, we may associate it with other pixels belonging to the same object in the other frames instead of with its own shifted versions because the pixels are indistinguishable due to their similar gray levels. This is why the above algorithm (like others) cannot work directly on the original imagery; there is nothing to track in a featureless region such as the inside of a uniform object.

In order to extract the features out of the images, we chose to use the signed gradient operation which is the spatial derivative along the radius from the FOE; this is the direction along which each filter picks up pixels from the frames. Thus, a "feature" is defined as

a gray level variation in the radius direction. We now consider two methods of gradient-based preprocessing.

Straightforward gradient preprocessing has a crucial drawback when followed by the main algorithm. The problem is that nearby objects are always observed on the background of other objects. The background objects, being farther from the sensor, move slower compared with the nearby object. As a result, the gray-level *differences* at the edges of a close object change with time as *different* background objects may pass behind it. Thus, edges of objects, as obtained from the gradient operation, have a time-varying gray level and cannot be used with any algorithm that assumes constant gray levels.

This is why we consider the alternative of only using the Gradient as a *test* in choosing the edges out of the *original* imagery. We do that by zeroing out all parts of the original imagery having gradient (in absolute value) under some threshold.

C. Expansion of Objects Between Frames

We know that the linear dimensions of an object are inversely proportional to the depth. There are a few things to consider in this respect. First, there is the question of compliance with the sampling theorem. Since the objects in all practical cases can be of any size, there will always be some objects which are smaller than a pixel; thus there is the potential of aliasing. A simple method to avoid aliasing is to blur the optical point-spread-function (PSF) in front of the sensor so that it is larger than, say, two pixels.

Assume, for example, that we have an object with a perfect (step function) edge. Using a PSF of the above size will make this edge show up in roughly two pixels, *irrespective* of the size of the object itself which increases with time. Thus, in such an ideal case, the algorithm does not have to accommodate for expansion of objects between frames. However, if we assume an object having a gradual gray-level transition at its edges which spans a few pixels, the size (or width) of the edges will expand with time. In such a case the algorithm does have to accommodate for expansion of objects between frames.

In reality, there will always be a mix of object edges that fall into the above two categories. Moreover, every nonideal object goes from the category of sharp edges to that of the wide edges as it approaches the sensor. We could think of an edge-width test to determine category, but, for this work, we simply experimented by using each category separately for the whole run.

D. Methods of Interpolation

The issue of interpolation arises because we are dealing with spatial samples (pixels) of the physical scene. Since we must be able to deal with small objects

in the order of 1- to 3-pixels size, we use the pixel readings to describe the gray levels of an underlying physical object rather than trying to estimate them. Two interpolation methods are suggested as follows.

1) *First Method*: If we consider a pixel at frame number 1 as the object itself or a part of the object, then this "object" (pixel) will normally intersect 4 pixels of any later frame as it travels with the optical flow. Thus 4 pixel values are affected by the gray level of the original pixel but also by other pixels whose gray-levels can be thought of as (zero-mean) random variables. Interpolating over these values, we get a random variable having an *average* equal to the tracked pixel gray level and some variance.

We track a pixel-object by summing up its shifted-and-interpolated gray levels over all frames, that is, summing up N (assumedly independent) random variables. The variance of the sum variate (normalized by the average) decreases as $1/N$. Since our test for assigning depth to a pixel makes use of the sample variance (which is itself a random variable), there is some probability of wrong depth assignment associated with the randomness of this interpolation scheme.

Notice that the tracked object-pixel occupies only $1/4$ of the four-pixel area used for interpolation whereas $3/4$ is random. Assuming a larger object, say 2×2 pixels, increases the ratio between average and random area contributions from $1/3$ to $4/5$ because nine pixels have to be observed instead of four. Experimenting with different sizes, we have found that an assumption of 2×2 -size objects yields the best results, probably indicating that the PSF of the camera for our imagery was of that size.

2) *Second Method*: This method avoids, at least in principle, the randomness associated with the first method. We assume, as above, that the minimum-size object at frame 1 is 2×2 pixels. At any later frame this object occludes at least one pixel of the later frame completely. Thus, given the gray-level values of the four frame-1 pixels (the object), we can calculate their contribution into the later-frame pixel using their intersected areas with that pixel.

This method performed quite poorly compared with the first method. The reason may be its sensitivity to the underlying assumptions. Obviously further work is required regarding the interpolation methods.

E. Parameters of the Algorithm

Here we list the parameters of the algorithm. Working with either the gradient images themselves or with the "gradient-as-a-test" method, we have to specify the FOE and the threshold G_{th} under which the gradient value will be set to zero.

For the main algorithm we need to choose the following parameters (in addition to specifying the FOE):

Fig. 12. FSN: Frame 1.

Fig. 14. Original FSN Frame 1 with gradient ≥ 5 .

Fig. 13. FSN: Frame 15.

Fig. 15. Original FSN Frame 15 with gradient ≥ 5 .

- 1) Whether or not to compensate for object expansion (expand = 0 or 1),
- 2) side: side in pixels of assumed object,
- 3) $z_{\max}$ and $z_{\min}$: maximum and minimum depths,
- 4) l_p : width of the depth filter (allowed smearing in pixels),
- 5) R_f : rate of frames in frames per second,
- 6) v : speed of vehicle in m/s,
- 7) N : total number of frames.

F. Data Set Used and Results

Our data consisted of 15-frames imagery taken at the NASA/FSN image processing lab (pencils on a table). The final product of some representative cases is presented in the form of a depth image based on frame number 1.

In order to use a data set which was taken indoors, we need to scale it in terms of vehicle speed, frame rate, and distances. The general formula for the depth scale factor is

$$S = \frac{v}{R_f v_l} \quad (51)$$

where v is the actual speed of the vehicle in m/s, v_l is the laboratory speed of the camera in m/fr, and R_f

is the laboratory frames rate in fr/s which is assumed equal to the actual frame rate. The images were taken every 1.25 cm. Arbitrarily choosing $R_f = 1$ fr/s and $v = 10$ m/s, the lab speed becomes $v_l = 0.0125$ m/fr, and $S = 800$. The above arbitrary choices affect the depth scaling but not the essential results. The depths of relevant objects in this set are between 0.33 and 0.9 m (at the first frame) which translate to 260 and 720 m, respectively.

The original images (first and last) are shown in Figs. 12 and 13. Figs. 14 and 15 show the original first and last frames in the parts that have an absolute gradient values above 5. The rms of the gradient values is 35.

Fig. 16 shows the depth image obtained from processing the gradient imagery with $G_{th} = 2.5$, $z_{\max} = 800$ m, $z_{\min} = 100$ m, $l_p = 1$, side = 2, $v = 10$ m/s, and expand = 1. Darker gray level in the depth images indicates a larger depth. The ground-truth (in parenthesis) and the min/max of measured depths for the 3 pencils and the bracket are shown in the figure. It is seen that the depth errors are in the order of 5 to 10%. The periphery parts of the depth image are blank because all pixels which exit the FOV prior to the last frame were excluded.

Fig. 16. FSN depth image based on gradient imagery.

Fig. 17. FSN depth image based on gradient-as-a-test imagery, expand = 0.

Figs. 17 and 18 show the depth images obtained from processing the original images with "gradient-as-a-test" and expand = 0,1 respectively. The other parameters common to these two cases are $G_{th} = 5$, $z_{max} = 800$ m, $z_{min} = 100$ m, $I_p = 1$, and side = 2. These results are similar to those of the direct-gradient method but there are more obvious erratic errors. Comparing the last two figures, it is seen that compensating for object expansion only had a minor effect; a wide optical PSF (say 2×2) can explain that.

VIII. SUMMARY

In this paper we evaluated the performance of the velocity filtering method as applied to optical-flow passive ranging under real-world conditions. After having reviewed the theory of 3D FT as applied to constant-speed moving points, we derived the space-domain shift-and-add algorithm from the general 3D MF formulation. We then modified the constant-speed algorithm to fit the actual speed

Fig. 18. FSN depth image based on gradient-as-a-test imagery, expand = 1.

encountered in the optical flow application and found the passband of that filter in terms of depth (sensor/object distance) so as to cover any given range of depths. We developed two algorithmic solutions for the problems associated with pixel interpolation and object expansion and presented experimental results.

Since our results here are only based on a single imagery set, they can, at best, serve as a rough indicator for the expected performance in the general case. In [12] we reported on experimentation with another imagery set, but many more cases have to be tested before one can make firm conclusions about performance. It is however clear that performance will strongly depend on the type of scenery; if it is composed of distinct nonoverlapping objects laid out against a uniform background, the algorithm is expected to perform very well. We cannot, at this time, predict performance in other situations.

REFERENCES

- [1] Papoulis, A. (1969) *Systems and Transforms in Optics*. New York: McGraw Hill, 1969.
- [2] Reed, I. S., Gagliardi, R. M., and Shao, H. M. (1983) Application of three-dimensional filtering to moving target detection. *IEEE Transactions on Aerospace and Electronic Systems*, AES-19, 6 (Nov. 1983), 898-905.
- [3] Porat, B., and Friedlander, B. (1987) A frequency domain approach to multiframe detection and estimation of dim targets. Presented at the International Conference on Acoustics, Speech, and Signal Processing, Apr. 6-9, 1987.
- [4] Fries, R. W. (1989) Three-dimensional matched filtering. Code 1050 03, Society of Photo-Optical Instrumentation Engineers, 1989.

- [5] Reed, I. S., Gagliardi, R. M., and Stotts, L. B. (1988)
Optical moving target detection with 3D matched filtering.
IEEE Transactions on Aerospace and Electronic Systems,
24, 4 (July 1988), 327-335.
- [6] Stotts, L. B., Reed, I. S., and Gagliardi, R. M. (1987)
Air vehicle detection by using an ensemble of
moving-picture images.
Technical report TR-1163, NOSC, Jan. 1987.
- [7] Kendall, W. B., and Jacobi, W. J. (1988)
Passive electro-optical sensor processing for helicopter
obstacle avoidance.
Space Computer Corporation, 2800 Olympic Blvd., Suite
104, Santa Monica, CA 90404-4119, Dec. 1988.
- [8] Smith, F. W. (1987)
Passive ranging from a moving vehicle via optical flow
measurements.
SPIE, Vol. 829, *Applications of Image Processing X*, 1987,
310-317.
- [9] Bracewell, R. (1965)
The Fourier Transform and Its Applications.
New York: McGraw-Hill, 1965, 69-90.
- [10] Wozencraft, J. M., and Jacobs, I. M. (1965)
Principles of Communication Engineering.
New York: Wiley, 1965, 229-231.
- [11] Nahi (1969)
Estimation Theory and Applications.
New York: Wiley, 1969.
- [12] Barniv, Y. (1990)
Application of velocity filtering to optical flow
calculations.
Technical manual TM 102802, NASA/Ames FSN Branch,
Moffett Field, CA 94035, Aug. 1990.

Yair Barniv earned his first degree in electrical engineering from the Technion-Israel Institute of Technology. He earned his higher degrees at Carnegie-Mellon University, Pittsburgh, PA, in digital/optical signal processing.

Early in his career he worked on the development of airborne and ground radars for fire-control systems and air-traffic control. Later, in the aircraft industry, he worked on missile guidance design, implementation, integration, and field testing. During that nine year period he also did performance analysis and feasibility/cost-effectiveness studies in a broad spectrum of projects related to electro-acoustics, electro-optics, electronic countermeasures (ECM), radar, and navigation. After graduation from Carnegie-Mellon University he joined Systems Control Technology. During this eight year period he did research on cruise-missile path optimization, closed-loop ECM, synthetic aperture radar (SAR) application to homing, ballistic missiles interception, fleet antimissile defense, and others. His main contribution during that period was an innovative approach to streak detection in IR imagery using dynamic programming. Recently Dr. Barniv has developed new techniques in the areas of nonacoustic ASW, velocity-discriminating SAR, and passive ranging for helicopter nap-of-the-Earth obstacle avoidance based on optical-flow and stereo vision. Currently he is working on biological/neural-nets implementation of passive ranging algorithms.