low reconstruction video quality. However, the differences between the l_1 norm and the l_2 norm are slight.

IV. Conclusion

In this work, fast motion estimation algorithms in the l_1 norm and fast motion estimation algorithms in the l_2 norm are studied and compared. All these fast algorithms achieve the same estimation accuracy as the exhaustive search algorithm with a considerably reduced computational load. One modified motion estimation algorithm, which does not rely on the convexity of the motion compensated residual error surface, is also proposed. This modified algorithm can provide a further 40% computational load reduction over the SEA, while keeping almost the same rate-distortion curve. These fast algorithms provide feasible solutions for reducing the cost and improving the rate-distortion-computation tradeoff of a video coder.

Acknowledgment

The authors would like to thank the reviewers for their comments and suggestions.

References

[1] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Processing*, vol. 4, pp. 105–107, Jan. 1995.

[2] H. G. Musmann, P. Pirsh, and H. J. Gilbert, "Advances in picture coding," *Proc. IEEE*, vol. 73, pp. 523–548, Apr. 1985.

[3] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COMM-29, pp. 1799–1808, Dec. 1981.

[4] A. N. Netravali and J. D. Robbins, "Motion compensated television coding: Part I," *Bell Syst. Tech. J.*, vol. 58, pp. 631–670, Mar. 1979.

[5] CCITT Standard H.261, "Video codec for audiovisual services at px64 kbit/s," ITU, 1990.

[6] ITU-T DRAFT Standard H.263, "Video coding for narrow telecommunication channel at (below) 64 kbit/s," ITU, Apr. 1995.

[7] ISO-IEC JTC1/SC2/WG11, "Preliminary text for MPEG video coding standard," ISO, Aug. 1990.

[8] T. Koga, K. Iinuma, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *Proc. Nat. Telecommunications Conf.*, Nov. 1981, pp. G5.3.1–G5.3.5.

[9] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. 38, pp. 950–953, July 1990.

[10] M. J. Chen, L. G. Chen, and T. D. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 504–509, Oct. 1994.

[11] H. M. Jong, L. G. Chen, and T. D. Chiueh, "Accuracy improvement and cost reduction of 3-step search block matching algorithm for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 88–91, Feb. 1994.

[12] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438–442, Aug. 1994.

[13] K. H. K. Chow and M. L. Liou, "Genetic motion search algorithm for video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 440–446, Dec. 1993.

[14] T. M. Apostol, *Mathematical Analysis.* Reading, MA: Addison-Wesley, 1975.

[15] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1991.

[16] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 148–157, Apr. 1993.

Two-Dimensional Matched Filtering for Motion Estimation

Peyman Milanfar

***Abstract*—In this work, we describe a frequency domain technique for the estimation of multiple superimposed motions in an image sequence. The least-squares optimum approach involves the computation of the three-dimensional (3-D) Fourier transform of the sequence, followed by the detection of one or more planes in this domain with high energy concentration. We present a more efficient algorithm, based on the properties of the Radon transform and the two-dimensional (2-D) fast Fourier transform, which can sacrifice little performance for significant computational savings. We accomplish the motion detection and estimation by designing appropriate matched filters. The performance is demonstrated on two image sequences.**

***Index Terms*—Discrete Fourier transform, estimation, image line pattern analysis, image motion analysis, matched filters.**

I. Introduction

The problem of motion estimation from an image sequence has a variety of applications. In particular, the estimation of multiple superimposed translational motions (displacements) has, more recently, received some attention [3], [5], [14], [17], [22]. Traffic monitoring, meteorological monitoring of clouds and storms from satellite imagery, and detection and tracking of airborne or ground-based targets are all practical examples of the need for fast, real-time, multiple motion estimation from video.

Most approaches to this problem can be categorized as working either in the image domain (gradient-based, or region-based), or in the spectral domain (Fourier transform based) [1], [4]. Image (pixel) domain algorithms, typically directed at short sequences of two or three images, use the optical flow brightness constraint to estimate the motion parameters [5], [7]. Spectral approaches are based on the notion that if a sequence of images—thought of as a three-dimensional (3-D) function in two-dimensional (2-D) space and time—contains a linearly moving pattern, then the energy of the 3-D Fourier transform (FT) of this function will be concentrated along a plane through the origin whose orientation is related to the velocity of the moving pattern [6], [9], [19]–[21]. So by computing the 3-D fast FT (FFT) and then finding a plane with strong energy concentration, we can estimate the desired velocity. An important advantage of this technique is that, because of the linear superposition of the FT, the existence of multiple superimposed motions will be manifested in the spectrum simply as energy concentrations along more than one plane through the origin.

In this work, we pursue a spectral technique for estimating translational motion, and we build on previous work in this area [15], [17], [19]–[21]. Specifically, in [17] we presented an algorithm that comprised of 1) projecting the frames in the image sequence onto a pair of orthogonal directions followed by 2) computation of 2-D FT's, and 3) a line-finding algorithm based on array processing [2]. The fundamental difference between the current work and that presented earlier in [17] is that the line-finding technique we adopt here is based on [15] and, while more computationally intensive,

Manuscript received November 30, 1996; revised April 3, 1998. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Steven D. Blostein.

The author is with SRI International, Menlo Park, CA 94025 USA (e-mail: milanfar@unix.sri.com).

Publisher Item Identifier S 1057-7149(99)01558-4.

yields significantly more accurate results even in the presence of considerable noise.

In [15], the authors computed the 3-D FT of the image sequence and detected lines in *slices* of this 3-D spectrum using geometrically defined 2-D matched filters. In fact, these slices can be obtained much more efficiently, and without computation of the 3-D transforms, by first computing projections of the images in the sequence and then computing 2-D FFT's. This idea has been noted and employed before. For instance, in [16] and [21], the frames were projected along the coordinate axes and lines were detected in the 2-D spectra computed from these projections. However, the line detection process was carried out by identifying peaks along the temporal axis of the 2-D Fourier transforms. The innovation presented in this correspondence is precisely in the combination of the projection-based technique and a reliably efficient technique for the detection of the resulting lines in the 2-D spectra. Furthermore, it is demonstrated that this combination is directly applicable to the estimation of multiple superimposed motions. We will also show that when applied locally to image sequences containing motion fields that are adequately approximated as locally translational, the proposed algorithm can produce results that are comparable to the accuracy of the most popular existing motion estimation algorithms in current use.

II. Problem and Solution Outline

If an image is translating according to a motion vector $v = [v_x, v_y]^T$, we can write the image sequence as a convolution:

$$f(x, y, t) = f(x, y) * \delta(x - v_x t, y - v_y t). \tag{1}$$

Computing the Fourier transform of both sides of (1) yields

$$F(\omega_x, \omega_y, \omega_t) = F(\omega_x, \omega_y)\delta(v_x\omega_x + v_y\omega_y + \omega_t). \tag{2}$$

This indicates that all the energy in the 3-D spectrum of the image sequence must be concentrated along the plane given by $v_x\omega_x + v_y\omega_y + \omega_t = 0$. Furthermore, if multiple translational motions are superimposed, then due to the linear superposition property of the FT, the spectrum of the image sequence will simply be concentrated along several planes.

The (direct) least squares optimal approach [17] to estimating the displacement[1] vector(s) v is to compute the 3-D spectrum of the image sequence and detect the plane(s) with strong energy concentration. A different way of describing the planes of interest is to specify a pair of vectors (or lines) that span them. One such pair, for instance, is obtained by considering the intersection of these planes with two other linearly independent planes. An attractive way to accomplish this, and hence to reduce the dimensionality of the problem to two dimensions, is to project each frame along a pair of independent directions, and then to apply 2-D FFT's. In particular, if these projections are taken along the rows and columns of the images, the celebrated projection slice theorem [10] implies that the 2-D FFT's of these projections are slices through the 3-D FFT of the image sequence along the planes $\omega_x = 0$ and $\omega_y = 0$, respectively. Hence, the energy of the resulting 2-D spectra will be concentrated along the *lines* $v_x\omega_x + \omega_t = 0$, and $v_y\omega_y + \omega_t = 0$, instead of along planes in 3-D processing.

We define the (discrete) projections along the rows and columns of the image as follows:

$$p(x, t) = \sum_y f(x, y, t), \tag{3}$$

$$q(y, t) = \sum_x f(x, y, t). \tag{4}$$

[1]In this work, we use the terms *velocity vector* and *displacement vector* interchangeably. Both are assumed to be in units of pixels per frame (ppf).

The shift property of the Radon (projection) transform then implies that as a result of the motion undergone by f, $p(x, t)$ moves v_x samples per frame, and similarly, $q(y, t)$ moves v_y samples per frame. While this observation is not new [16], [17], [20], [21], it is worth pointing out that it can be stated in any coordinate system. That is, for projections $p_\theta(s)$ taken onto any axis (s) forming an angle θ with the x-axis, $p_\theta(s)$ will simply undergo a motion of $v_x\cos(\theta) + v_y\sin(\theta)$ samples per frame. In this paper, however, we treat the case of row and column projections only.

The 2-D algorithm proposed here comprises several simple steps. First, we compute the projection functions $p(x, t)$ and $q(y, t)$. Then, the 2-D spectra, denoted $P(\omega_x, \omega_t)$ and $Q(\omega_y, \omega_t)$, of each of these functions are computed. Next, a matched filter is applied to a function of the magnitude of each of these spectra, producing the outputs $R(v_x)$ and $R(v_y)$. The peak(s) in each output will correspond to the best estimate of the respective components of the motion vector.

Numerous techniques have been introduced in the past to accomplish the line detection task. Among them, the most popular and widely used are based on the Hough transform [8], [11], and more recently a clever array processing technique [2], [17]. The technique that we present here is a version of the Hough transform idea, and has also been suggested in [15]. The proposed technique is a matched filter, which is based on integration of energy over pixels along lines that pass through the origin, whereas the traditional Hough transform approach looks for all possible lines in the image.

To define a matched filter for the detection of lines through the origin in P and Q, we write the expression for the distance between a point and a line L with (unknown) slope $-v_x$, in the *discrete* plane (ω_x, ω_t) as follows:

$$\Delta(v_x) = \frac{|\omega_x(v_x/N_x) + \omega_t(1/N_t)|}{\sqrt{(v_x/N_x)^2 + (1/N_t)^2}} \tag{5}$$

where N_x and N_y denote the dimensions of each image and N_t denotes the number of frames. Given the definition of $\Delta(v_x)$, the set of all pixels satisfying

$$\Delta(v_x) \leq \tfrac{1}{2}w_d \tag{6}$$

is a digital line of width w_d pixels, with slope $-v_x$ and passing through the origin. Defining the average "energy" at all such pixels by $R(v_x)$, we have a parametric description of the matched filter output as follows:

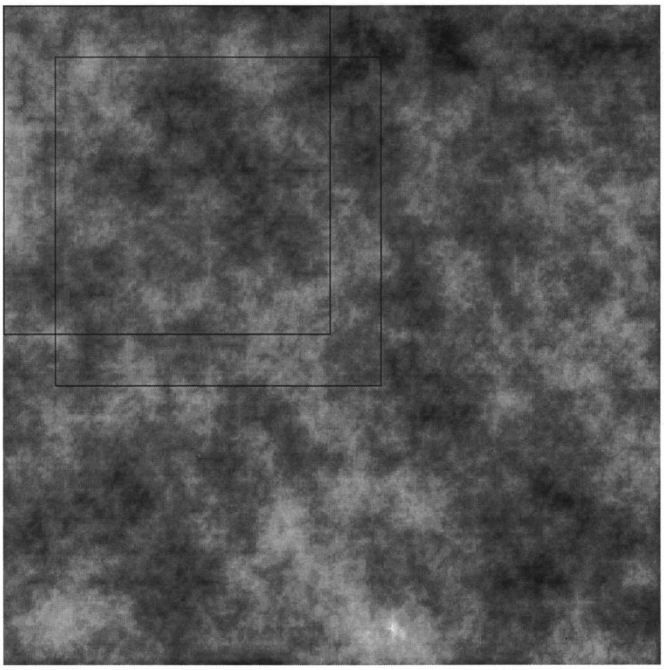
$$R(v_x) = \frac{1}{n(v_x)} \sum_{L(v_x)} |P(\omega_x(i), \omega_t(j))|^\kappa \tag{7}$$

where $L(v_x) = \{(\omega_x(i), \omega_t(j)) \mid \Delta(v_x) \leq \frac{1}{2}w_d\}$ denotes the set of all pixels on the digital line with slope $-v_x$, and $n(v_x)$ denotes the total number of pixels on the same line. The estimate of v_x is then given by

$$\hat{v}_x = \arg \max_{v_x} R(v_x). \tag{8}$$

The above detector is optimal for the detection of straight-line patterns in the same sense as the Hough transform (see [18], for instance), as it is simply a special case of the Hough transform which seeks lines that pass through the origin. The choice of the scaling parameter κ in effect results in a useful form of preprocessing of the magnitude spectrum image. Namely, to estimate v_x accurately, it is important that the peak of $R(v_x)$ be well defined (sharp). To ensure that this happens, we can choose a value of $0 < \kappa \leq 1$ to suppress the typically larger amplitudes at lower frequencies, and expand the usually smaller amplitudes at higher frequencies. This results in a more even distribution of energy across the spatio-temporal frequency plane and hence facilitates the distinction of neighboring

(a)



(b)

Fig. 1. Aerial photo of Washington, DC, and synthetic cloud.

lines with similar slopes. The choice of κ can, in fact, be optimized in accordance with the spectral content of the image sequence. We found that the value $\kappa = 1$ gave the overall most robust results. More formal variations of this idea are also possible. For instance, one may place the line detection problem within a robust estimation framework [12]. Such estimators are less sensitive to outliers and would therefore be useful in the multiple-line detection problem if all but one of the lines is considered as outliers. We leave this extension for future work.

A. Estimation of Multiple Velocities

When two or more peaks are present, we can devise a technique to automatically detect multiple peaks corresponding to motions. We assume that the number of such motions is known *a priori*.[2] We proceed by first locating the highest peak $(\hat{v}_x)$ according to (8). Next, we identify an interval about this peak that contains energy related to the peak at $\hat{v}_x$. Having identified this range, we set all values of R in this range to zero and invoke (8) again. Repeating this process, we recover all the peaks in the plot of $R(v_x)$.

To accomplish the task of peak elimination, we assume that the motion at $\hat{v}_x$ corresponds to a line of unit width in the image $|P(\omega_x, \omega_t)|$. We take the matched filter for any v_x to consist of

[2]Techniques such as minimum description length can be used to estimate the number of motions present.

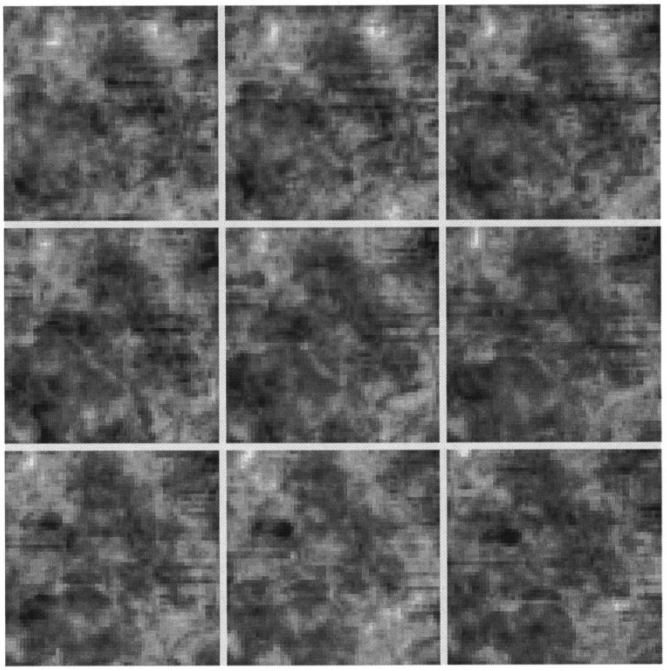


Fig. 2. Left to right, from the top: Frames 1, 5, 10, 15, 20, 25, 30, 35, and 40 of the Washington, DC, sequence.

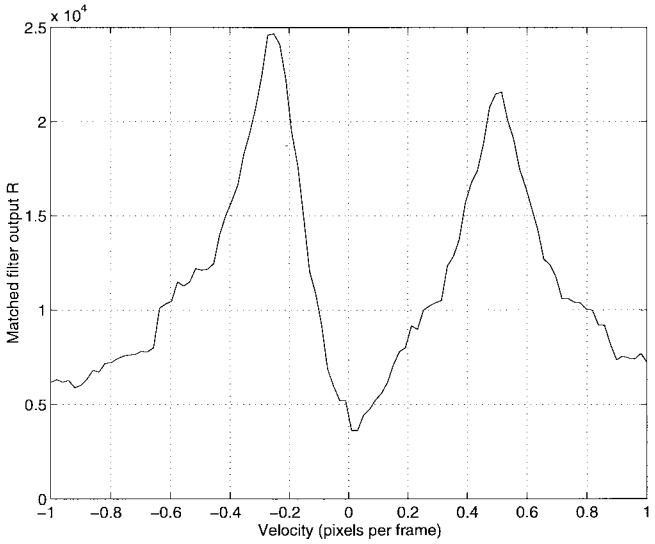
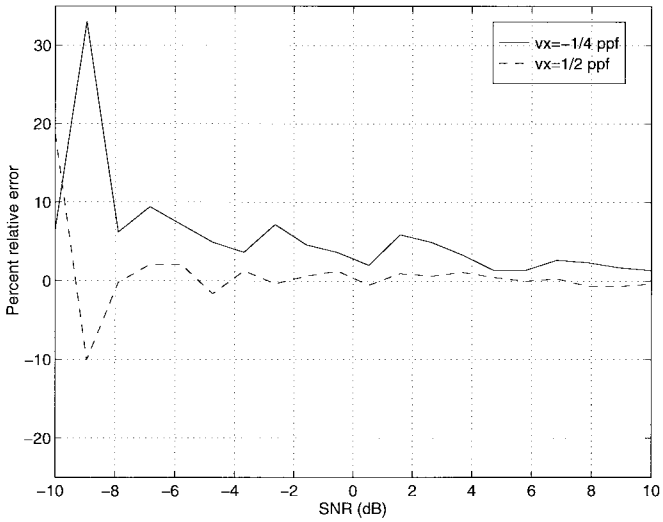


Fig. 3. Matched filter output for x component of Washington, DC, sequence at $\mathrm{SNR} = 10$ dB, with $w_d = 2$.

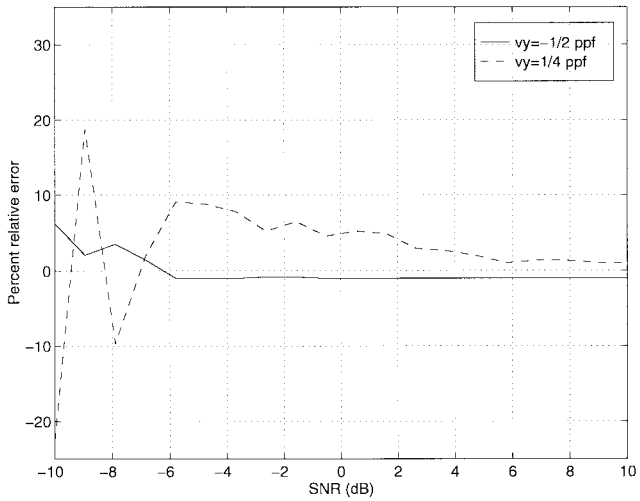
a line of width w_d passing through the origin with slope $-v_x$. The number of pixels in the intersection of the two regions will determine the interval about $\hat{v}_x$ that will be eliminated. It is easily shown that taking $N_x = N_t = N$, the number of pixels l in the intersection of the matched filter and the line we seek is given by

$$l = \begin{cases} w_d/|\sin(\hat{\theta} - \theta)|, & \text{for } \omega_x < N/2, \\ N w_d(1 + |\hat{v}_x|)/\sqrt{1 + \hat{v}_x^2}, & \omega_x = N/2 \end{cases} \tag{9}$$

where $|\hat{v}_x| = |\tan(\hat{\theta})|$ and $|v_x| = |\tan(\theta)|$. The function $l(v_x)$ is symmetric about the peak at $\hat{v}_x$. If we now pick a value $\bar{l}$ such that $l(\hat{v}_x)/\bar{l}$ is equal to some prescribed percentage value ρ, then solving $l(v_x) = \bar{l} = \rho l(\hat{v}_x)$ will yield two values $\underline{v}_x$ and $\overline{v}_x$ symmetric about $\hat{v}_x$ that will determine the radius $r(\hat{v}_x) = (\overline{v}_x - \underline{v}_x)/2$ of the range that is to be eliminated.



(a)



(b)

Fig. 4. Mean x (left) and y (right) relative error curves for Example 1.

B. Velocity Pairing

If the previous steps have produced k horizontal and k vertical velocity component estimates, we denote these estimates as $V_x = \{\hat{v}_x(1), \hat{v}_x(2), \cdots, \hat{v}_x(k)\}$ and $V_y = \{\hat{v}_y(1), \hat{v}_y(2), \cdots, \hat{v}_y(k)\}$. We shall assume that the true velocity components do not have any x or y components in common,[3] (i.e., none of the $\hat{v}_x$'s are equal, and none of the $\hat{v}_y$'s are equal). We need to obtain k displacement vectors $v_1, v_2, \cdots, v_k$ by matching elements of V_x to those of V_y.

In [17], we described a technique based on computing difference frames according to all possible pairings and sequentially picking out the pairings that result in the smallest difference images in the least squares sense. Although this technique is reliable and does not require the use of all available frames, it can be replaced by another technique that relies more directly on the 2-D Fourier transform. The *temporal projection* of the frames can be defined as $s(x, y) = \sum_t f(x, y, t)$. Again, according to the projection slice theorem, the Fourier transform $S(\omega_x, \omega_y)$ of $s(x, y)$ is the slice along the plane $\omega_t = 0$ of the 3-D Fourier transform $F(\omega_x, \omega_y, \omega_t)$.

[3]If this condition is violated, it may then be necessary to pick different projection angles and to repeat the previous steps. This scenario illustrates a potential drawback of the proposed algorithm.

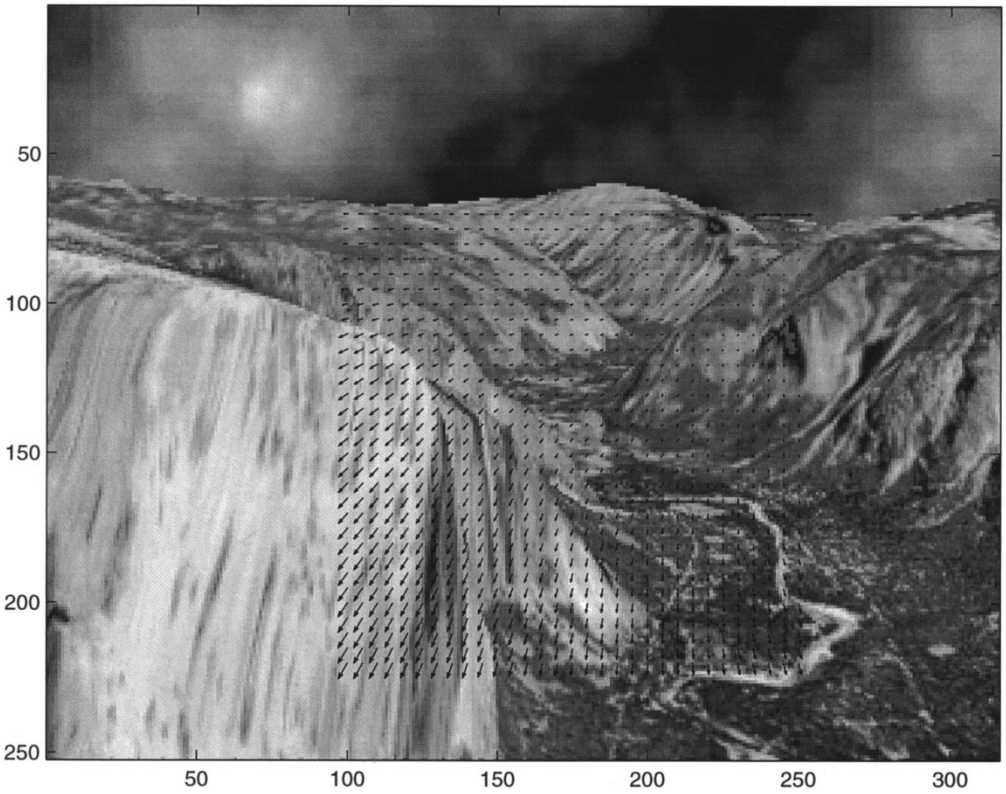


Fig. 5. Frame 8 of the Yosemite sequence, with true motion vectors on pixels of interest for Example 2.

Thus, a motion along the vector (v_x, v_y) will manifest itself as the line $v_x\omega_x + v_y\omega_y = 0$ in this plane. This line passes through the origin and has slope $-v_x/v_y$. Therefore, given k estimated values of v_x and v_y, we can detect k or fewer lines in the plane $w_t = 0$ and compute their slopes. For each estimated $v_x(i)$, we can then find the best matching[4] $v_y(j)$ by comparing the ratio $-v_x(i)/v_y(j)$ to the estimated line slopes from the plane $w_t = 0$. Continuing in this fashion for $i = 1, \cdots, k$, we can match all velocity components.

III. Performance Characteristics

As we discussed in the introduction, the optimum approach to the displacement vector estimation problem will use the 3-D spectrum of the image sequence. The algorithm derived here is therefore suboptimal as it only uses projections of the given 3-D data. In [17], an analytical bound was established for the loss in performance if a 2-D projection-based approach is used instead of the optimum 3-D approach. This bound is based on computation of local approximations to the error covariances for both the 2-D and the 3-D motion estimation approaches under assumptions of high signal-to-noise ratio (SNR) and small motion. Rather than repeat the analysis, we briefly state the results.

An aggregate measure of performance for the 2-D technique is the sum of the variances for the estimates of each motion component. On the other hand, for the direct 3-D approach, the trace of the covariance matrix of the estimate can be used. Denoting these two scalar quantities as C_2 and C_3, respectively, we showed [17] that the relative performance loss is bounded as

$$\frac{C_3 - C_2}{C_3} \leq \frac{D_{xy}^2}{D_{xx}D_{yy}} \tag{10}$$

where

$$D_{xx} = \sum_{x,\,y,\,t} \left(\frac{\partial f}{\partial x}\right)^2, \qquad D_{yy} = \sum_{x,\,y,\,t} \left(\frac{\partial f}{\partial y}\right)^2,$$
$$D_{xy} = \sum_{x,\,y,\,t} \left(\frac{\partial f}{\partial x}\,\frac{\partial f}{\partial y}\right). \tag{11}$$

Note that the right-hand side of this bound (10) is essentially a correlation coefficient. That is, the relative performance loss is small when the *gradients* of the images in the selected (in this case x and y) directions are weakly correlated.

IV. Computational Complexity

Assuming that the matched filters have a velocity resolution of N bins, and that $N_x = N_y = N_t = N$, the dominant terms in the computational complexity of the 2-D algorithm are as follows:

1) the projections, which require $\mathcal{O}(N^3)$ operations;
2) the 2-D FFT's, which require $\mathcal{O}(N^2 \log N)$ operations;
3) the 2-D matched filtering, which requires $\mathcal{O}(N^3)$ operations;
4) velocity component matching which requires $\mathcal{O}(N^3)$ operations.

Therefore, overall, the 2-D approach has complexity that grows as $\mathcal{O}(N^3)$. On the other hand, the 3-D FFT-based approach involving detection of planes requires $\mathcal{O}(N^5)$ computations, which is dominated by the 3-D plane detection task.

As a point of comparison, gradient-based optical flow techniques using *a pair* of $N \times N$ images would require $\mathcal{O}(N^3)$ computations.[5] Scaling this to N frames, we get $\mathcal{O}(N^4)$ complexity. Therefore, on long image sequences, the 2-D spectral technique is roughly an order of magnitude faster than gradient-based techniques.

[4] We assume that $v_y(j) \neq 0$.

[5] Taking a local, say $2 \times 2 \times 2$, stencil for approximating the spatio-temporal gradients.

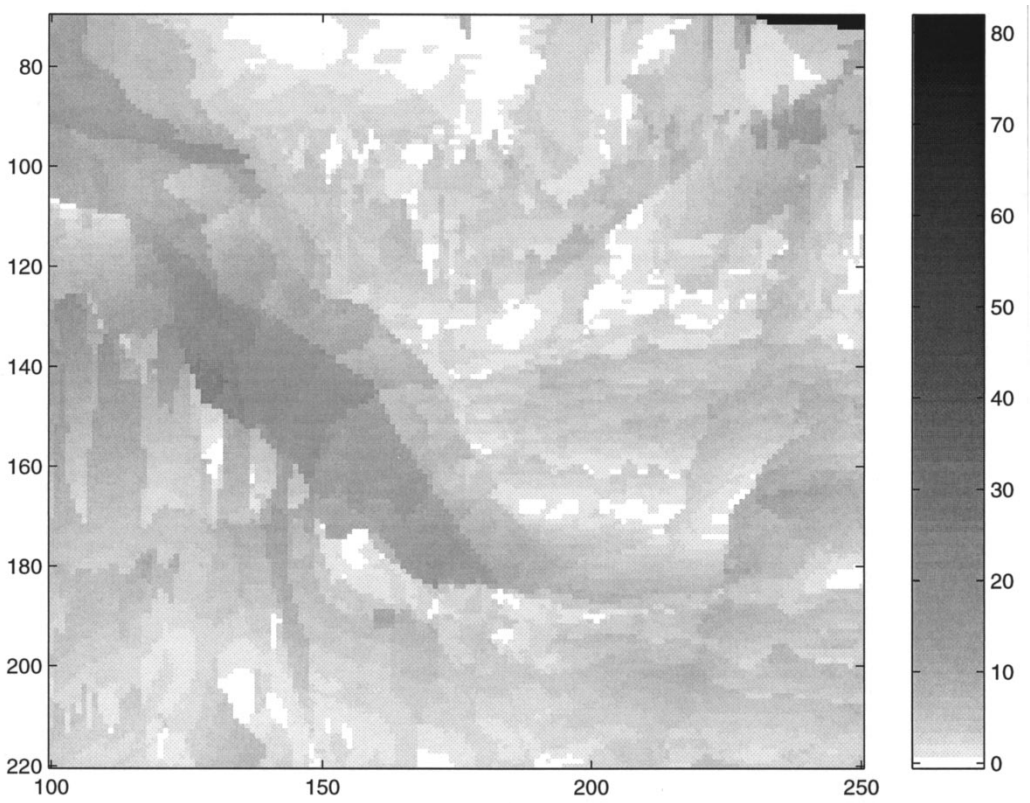


Fig. 6. Directional error in degrees, for Example 2: mean error = 6.1° and standard deviation = 5.5°.

V. Experimental Results

In this section, we present the results of two experiments that demonstrate the performance of the proposed algorithm numerically.

Example 1: We performed an experiment that demonstrated the estimation of two superimposed (added) subpixel motions. To simulate the measured frames in this experiment, we used the 475 × 720 aerial (ortho) photograph of Washington, DC (courtesy of USGS), and the synthetic 512 × 512 cloud image shown in Fig. 1. These images were shifted, added as $I(t) = 0.3I_{\mathrm{D.C.}}(t) + I_{\mathrm{cloud}}(t)$, lowpass filtered, and downsampled to produce 40 frames of a video sequence. Each resulting image has dimensions 60 × 60, and the sequence contains two global superimposed subpixel motion components given by $v_1 = [1/2, -1/2]^T$ and $v_2 = [-1/4, 1/4]^T$, corresponding to the ground and cloud motions, respectively. For more detail on the construction of this image sequence, see [17]. For the experiments, to simulate imperfect sensing conditions, we added Gaussian white noise to each frame to realize a given SNR value.[6] Fig. 2 shows selected (noiseless) frames from the sequence thus generated. Fig. 3 shows the output of the matched filter corresponding to the x velocity of the Washington, DC, sequence at SNR = 10 dB, with a matched filter resolution of 100 bins and a line width of $w_d = 2$.

To quantify the average performance of the 2-D approach, we performed Monte Carlo simulations where the two superimposed motions in the Washington, DC, image sequence were estimated repeatedly for 25 different realizations of noise at each of various SNR values. The mean relative error curves are displayed in Fig. 4. It is clear that the relative errors stay well within 5% of truth for SNR's above 0 dB. The systematic bias, likely due to the finite extent of the data at high SNR's, is on the order of 1%.

Example 2: In this example, we apply the proposed algorithm to the famous Yosemite sequence (obtained courtesy of Baron *et al.* [4]), which is composed of 15 frames each of dimensions 252 × 316. The eighth frame in the sequence, along with the correct flow field on a 151 × 151 pixel subset of the image (where we will compute the flow) is displayed in Fig. 5. For pixels in the subregion of interest, we computed the flow vector by using a 64 × 64 × 15 spatio-temporal window. The projection-based algorithm was then applied locally to each such window; if a complete 64 × 64 window of data was not available, the available data was zero padded. The matched filter search was then conducted over the range of velocities $[-3, 2]$ in each component with a 100-bin resolution, with the matched filter width selected as $w_d = 3$. The error, as measured according to the method outlined in [4], is displayed in Fig. 6. The mean error, over all estimated flow vectors was determined to be about 6.1°, with a standard deviation of 5.5°. We note that the poorest performance appears to occur at points where the true motion field and the image are discontinuous. For instance, in the upper right-hand side of Fig. 6, we can see that the error is large near the boundary of motion between the mountains and the clouds over them. This is likely due to the fact that high-frequency components in the Fourier domain resulting from the abrupt discontinuity corrupt the estimate of the slope of the lines being detected. Furthermore, perhaps the Fourier transform is not sufficiently localized in the spatio-temporal frequency domain to give adequate estimates of motion across discontinuous fields. This problem may be overcome to some extent if instead of the Fourier transform, another time-frequency

[6] The definition of the SNR is

$$\text{SNR (dB)} = \frac{1}{T}\sum_{t=0}^{T-1} 10\log_{10}\left(\frac{\sum_{x,y}(f(x,y,t)-\overline{f}(t))^2}{N^2\sigma^2}\right)$$

where $\overline{f}$ is the spatial average of $f(x, y, t)$, N is the spatial dimension of f, and T is the number of frames. So the SNR is the average SNR per pixel across all frames.

transform such as Wigner–Ville, with better localization properties is used [13]. On the other hand, the interference terms in such transform must be carefully accounted for if any improved performance is to be expected.

In any case, comparing the above results to those errors resulting from the use of other spectral and gradient-based techniques on the same image sequence (displayed in [4, Tab VII]), we can see that the proposed technique compares favorably to these techniques.

VI. Summary and Conclusions

In this paper, we demonstrated a simple spectral technique for the estimation of multiple motions from a (long) sequence of images. In particular, we developed an efficient and accurate algorithm to accomplish this task based on the projection of image frames in a pair of orthogonal directions, followed by the computation of 2-D FFT's and matched filtering. An analytical comparison to the optimum 3-D FFT approach shows that, under favorable conditions, the 2-D approach can perform nearly as well as the optimal 3-D technique, while incurring a significantly smaller computational cost. Furthermore, comparing the performance of the proposed algorithm to that of existing local differential and spectral techniques revealed that this simple algorithm is comparable in performance and presents an improvement in terms of computational complexity.

References

[1] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images-a review," *Proc. IEEE*, vol. 76, pp. 917, 935, 1988.

[2] H. K. Aghajan and T. Kailath, "SLIDE: Subspace-based line detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 1057–1073, Nov. 1994.

[3] ——, "Subspace techniques for image understanding," in *Proc. 28th Asilomar Conf. Signals, Systems, and Computers*, 1994.

[4] J. L. Baron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, pp. 43–77, 1994.

[5] J. R. Bergen, P. Burt, R. Hingorani, and Shmuel Peleg, "A three-frame algorithm for estimating two-component image motion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 886–896, Sept. 1992.

[6] L. Bruton and N. Bartley, "Three-dimensional image processing using the concept of network resonance," *IEEE Trans. Circuits Syst.*, vol. CAS-32, pp. 664–672, July 1985.

[7] P. J. Burt, R. Hingorani, and R. J. Kolczynski, "Mechanisms for isolating component patterns in the sequential analysis of multiple motion," in *Proc. IEEE Workshop on Visual Motion*, Oct. 1991, pp. 187–193.

[8] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, pp. 11–15, Jan. 1972.

[9] D. J. Heeger, "Model for the extraction of image flow," *J. Opt. Soc. Amer. A*, vol. 4, pp. 1455–1471, Aug. 1987.

[10] G. T. Herman, *Image Reconstruction from Projections.* New York: Academic, 1980.

[11] P. V. C. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3069654, Dec. 18, 1962.

[12] P. J. Huber, *Robust Statistical Procedures*, 5th ed. Philadelphia, PA: SIAM, 1989.

[13] L. Jacobson and H. Wechsler, "Derivation of optical flow using a spatiotemporal-frequency approach," *Comput. Vis., Graph., Image Process.*, vol. 38, pp. 29–65, 1987.

[14] D. Kersten, "Computational models of visual processing," in *Transparency and the Cooperative Computation of Scene Attributes.* Cambridge, MA: MIT Press, 1991, pp. 209–228.

[15] A. Kojima, N. Sakurai, and J. Ishigami. "Motion detection using 3-D FFT spectrum," in *Proc. ICASSP*, 1993, vol. V, pp. V213–V216.

[16] S. A. Mahmoud, M. S. Afifi, and R. J. Green, "Recognition and velocity computation of large moving objects in images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1790–1791, Nov. 1988.

[17] P. Milanfar, "Projection-based, frequency-domain estimation of superimposed translational motions," *J. Opt. Soc. Amer. A*, vol. 13, pp. 2151–2162, Nov. 1996.

[18] A. Neri, "Optimal detection and estimation of straight patterns," *IEEE Trans. Image Processing*, vol. 5, pp. 787–792, May 1996.

[19] B. Porat and B. Friedlander, "A frequency domain algorithm for multiframe detection and estimation of dim targets," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 398–401, Apr. 1990.

[20] M. A. Rahgozar and J. P. Allebach, "Motion estimation based on time-sequential sampled imagery," *IEEE Trans. Image Processing*, vol. 4, pp. 48–65, Jan. 1995.

[21] S. A. Rajala, A. M. Riddle, and W. E. Snyder, "Application of the one-dimensional Fourier transform for tracking moving objects in noisy environments," *Comput. Vis., Graph., Image Process.*, vol. 21, pp. 280–293, Feb. 1983.

[22] M. Shizawa and K. Mase, "Principles of superposition: A common computational framework for analysis of multiple motion," in *Proc. IEEE Workshop on Visual Motion*, 1991, pp. 289–295.

Intrinsic Multiscale Representation Using Optical Flow in the Scale-Space

Qing Yang and Song De Ma

***Abstract*—There exists an *optical flow in the scale-space* if the multiscale representation of an image is viewed as an ordinary image sequence in the time domain. This technique can be used to solve the ill-posed tracking problem in the scale-space.**

***Index Terms*—Multiscale representation, optical flow.**

I. Introduction

A critical problem in linear scale-space theory (see the survey by Lindeberg [6]) is that local features in the image may be seriously distorted at large scales. Various nonlinear diffusion equations [1], [7], [8] have been proposed to deal with this drawback. However, in many cases the nonlinear strategy is still not satisfactory because of the essential difficulty in the localization scheme.

A much more thorough method is to track points in the scale-space. Although this idea has been reported in the literature, we seldom perform tracking to obtain a better multiscale representation. It is often believed that this is due to computational complexity. However, we argue that the main reason is that the tracking problem is ill-posed. The procedure of regularization must be introduced. This allows us to define "optical flow in the scale-space" which can be viewed as a standard optical flow.

Manuscript received May 29, 1997; revised May 14, 1998. This work was supported in part by the Natural Science Foundation of China under Grant 69790080. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dapang Chen.

The authors are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China (e-mail: qyang@prlsun6.ia.ac.cn).

Publisher Item Identifier S 1057-7149(99)01562-6.