



# Near-recursive optical flow from disturbance fields

*Emanuele Trucco*

Department of Computing and Electrical Engineering  
Heriot-Watt University, Edinburgh, UK  
mtc@cee.hw.ac.uk

*Federico Viel, Vito Roberto*

Dipartimento di Matematica e Informatica  
Università di Udine, Udine, I  
{viel,roberto}@dimi.uniud.it

## Abstract

We derive a formal link between temporally weighted frame differences, or *disturbance fields*, which carry limited information commonly used for motion detection, and the optic flow, which carries rich information on local image motion. We use this to formulate a novel, simple, near-recursive optic flow algorithm based on a recursive-filter formulation. Most quantities involved are computed recursively, using only data from the current and previous frame. Experimental results with well-known synthetic, ground-truthed test sequences and standard performance metrics indicate good quantitative performance. Test with real sequences suggest similar or better performance than a similar algorithm.

## 1 Introduction

The key idea behind *image differencing* techniques [1, 7] is that sufficiently large intensity changes from a frame to the next indicate significant events in the scene imaged, nearly invariably motion. Although the definition of “sufficiently large” needs some care (see Rosin [7] for an excellent review of thresholding methods for change detection), differencing algorithms tend to be simple and efficient. Therefore they often feature as change detection modules in real-time vision systems, e.g., in surveillance [2], tracking [9], virtual reality and teleconferencing [4, 8]. In a different context, Halevy and Weinshall [1] use temporally weighted image differencing, called *disturbance fields*, to track multiple point-like targets tracking in high clutter. Differencing algorithms give very limited information about local motion. Far more complex algorithms are needed to compute a good-quality estimate of the image motion field, or *optic flow* [6].

This paper presents a closed-form link between temporally weighted frame differences, or *disturbance fields*, and the optic flow. We use the equations expressing this link to formulate a novel, simple, near-recursive optic flow algorithm which takes advantage of previous history. The algorithm is efficient in two senses. First, most quantities are computed recursively using only data from the current and previous frame, although a



variable number of previous frames influences the calculation. Second, as motion detection is an intrinsic part of the OF algorithm, we can compute OF only at pixels where motion is significant, at no additional computational cost.

Experimental results with well-known synthetic, ground-truthed test sequences and standard performance metrics indicate good quantitative performance. Test with real sequences suggest similar or better performance than those achieved with our implementation of Lucas and Kanade's algorithm [5], probably the most similar algorithm to ours in terms of approach and complexity.

This paper is organised as follows Section 2 defines disturbance fields and analyses some temporal properties. Section 3 derives the formal link between disturbance fields and optic flow. Building on this result, Section 4 derives a near-recursive optic flow algorithm. Section 5 presents some experimental results. Section 6 summarises and discusses our work.

## 2 Disturbance fields

Following [1], we define the *disturbance field*,  $D_k$ , as the difference between the current frame  $I_k$  of a sequence and an exponentially-weighted average of the past frames. More recent frames are given more importance. For efficiency,  $D_k$  is expressed as a recursive filter [3]:

$$\begin{aligned} A_k &= (1 - w)I_k + wA_{k-1} \\ D_k &= I_k - A_{k-1} \end{aligned} \quad (1)$$

with  $w \in [0, 1)$  a real number controlling the duration for which a frame influences future values of  $D$ . The smaller and smaller contribution that each past frame gives to the current frames become, in practice, negligible after a certain time interval. This interval, in effect the algorithm's memory span, is an important parameter which must be quantified. To this purpose, we begin by observing that  $A_k$  in Eq. (1) can be written in non-recursive form as follows:

$$A_k = (1 - w) \left[ \sum_{j=1}^k w^{k-j} I_j + \frac{w^k}{1 - w} I_0 \right]$$

so that

$$\begin{aligned} D_k &= I_k - A_{k-1} \\ &= I_k - (1 - w) \left[ \sum_{j=1}^{k-1} w^{k-j-1} I_j + \frac{w^{k-1}}{1 - w} I_0 \right]. \end{aligned} \quad (2)$$

Recalling that  $w < 1$ , we assume that  $w^{j-1}I_{j-1} < wI_j$ , a reasonable assumption *on average*; of course, intensities may not satisfy this locally. We then consider a negligible contribution as one adding an intensity less than 2% of a 255-wide dynamic range, i.e., 5. In these assumptions, the answer to our question is any time interval of width  $\Delta = k - j - 1$  satisfying

$$(1 - w)w^\Delta I_j < 5. \quad (3)$$



Assuming the worst case  $I_j = 255$ , we have

$$\Delta > \log_w \frac{1}{51(1-w)} \quad (4)$$

Figure 1 shows a plot of the right-hand side of Equation (4) against  $w$ , for two values of  $I_j$  (255 and 128). Eq. (4) allows us to estimate the time interval after which a given frame becomes practically insignificant in the computation of the disturbance field. For instance, for  $w = 0.6$  and  $I_j = 255$ , only the most recent 6 frames (approximately one fourth of a second at 25 frame/s) are expected to count.

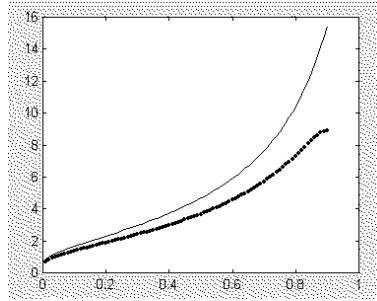


Figure 1: The interval  $\Delta$  (in frames) after which the contribution of a frame on the current disturbance field becomes negligible, plotted against  $w$ . Solid line is solution for  $I_j = 255$ , dotted line for  $I_j = 128$ .

### 3 The relation between disturbance field and optic flow

This section makes explicit the relation between Eq. (1), which measures the intensity changes taking place in time, and the optic flow. We begin by re-writing  $A_k$  in Eq. (1) term by term:

$$\begin{aligned} A_k &= w^{k-1}I_1 + w^{k-2}I_2 + \dots + wI_{k-1} + I_k + \\ &\quad -w^kI_1 - w^{k-1}I_2 - \dots - w^2I_{k-1} - wI_k \\ &\quad +w^kI_0. \end{aligned} \quad (5)$$

We notice that terms in the first and second line can be paired using consecutive frames, i.e.,

$$A_k = w^k(I_0 - I_1) + w^{k-1}(I_1 - I_2) + w^{k-2}(I_2 - I_3) + \dots + w^2(I_{k-2} - I_{k-1}) + w(I_{k-1} - I_k) + I_k.$$

If we regard  $I_{k-1} - I_k$  as a finite-difference, forward time derivative, we can write

$$A_k \approx \sum_{j=0}^k w^{k-j} \frac{\partial I_j}{\partial t} + I_k, \quad (6)$$



where the approximate equality reminds us that the equality holds within the limits of the approximation of the forward derivatives. The disturbance field becomes therefore

$$\begin{aligned}
 D_k &= I_k - A_{k-1} \approx I_k - \sum_{j=0}^{k-1} \frac{\partial I_j}{\partial t} - I_{k-1} \\
 &= - \sum_{j=0}^k w^{k-j} \frac{\partial I_j}{\partial t}; \tag{7}
 \end{aligned}$$

in other words, the disturbance field can be expressed as a difference of the current frame and a weighted sum of the past time derivatives.

If we now make use of the fundamental optic flow constraint, that is,

$$-\nabla I_j^\top \mathbf{d}_j \approx \frac{\partial I_j}{\partial t}, \tag{8}$$

where  $\nabla I_j^\top \mathbf{d}_j$  indicates dot (scalar) product, we can write (within the validity of the above constraint)

$$D_k \approx \sum_{j=0}^k w^{k-j} \nabla I_j^\top \mathbf{d}_j. \tag{9}$$

Equation (9) is the desired closed-form relation between the disturbance field and the optic flow. It says that the disturbance field can be written as a weighted sum of the well-known dot product gradient-optic flow featuring in Eq. (8), the fundamental constraint. As expected, contributions from past frames become less and less important in time.

Equation (7) shows that the disturbance field can be regarded as a linearly filtered version of the time derivative of the intensity. The filter is a non-causal exponential [3], the same written in recursive form in Equation (1). Equation (9) suggests that the disturbance field is linked through the same filter to the normal flow. We shall use this result to get a novel, simple and efficient flow algorithm.

## 4 A near-recursive optic flow algorithm

We assume that the flow at a given pixel,  $\mathbf{d}_j$ , does not change significantly in the most recent  $\Delta$  frames,  $\Delta$  being the time interval in which the influence of a frame on future DFs remains significant (recall that  $\Delta$  is small in practice, as detailed in Section 2). Notice that this assumption can be delicate, and we shall come back to it later. We can therefore take  $\mathbf{d}_j$  out of the sum in Eq. (9):

$$D_k \approx \mathbf{d}_k^\top \sum_{j=k-\Delta-1}^k w^{k-j} \nabla I_j, \tag{10}$$

which contains 2 unknowns in  $\mathbf{d}_k$ , the flow at frame  $k$ , for each pixel. If we, however, make the common assumption that the flow does not change significantly in small neighbourhoods (local constancy of motion), we can write Eq. (10) for a few adjacent pixels (2

would be enough theoretically), resulting in a linear system in the 2 unknowns  $d_x, d_y$ :

$$\begin{aligned}
 D_k(\mathbf{p}_1) &= \mathbf{d}_k^\top \sum_{j=k-\Delta-1}^k w^{k-j} \nabla I_j(\mathbf{p}_1) \\
 \dots &= \dots \\
 D_k(\mathbf{p}_N) &= \mathbf{d}_k^\top \sum_{j=k-\Delta-1}^k w^{k-j} \nabla I_j(\mathbf{p}_N)
 \end{aligned} \tag{11}$$

This system becomes overconstrained for  $N > 2$ , so that the algorithm does not need large support regions to achieve a sufficient number of constraints, and the minimum theoretical computational effort is low. Of course, the minimum size of a support region leading to acceptable results depends on the sequence at hand.

For clarity, we summarise below the three assumptions made on the way. The first two are the usual ones for optical flow algorithms; the third one is needed to incorporate past history.

1. the fundamental optic flow constraint applies;
2. the flow is nearly constant in small spatial neighbourhoods;
3. the flow remains practically constant over  $\Delta(w)$  frames.

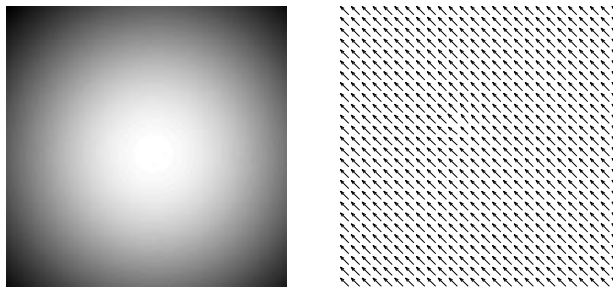


Figure 2: Left: the cosine-shaded Gaussian surface. Right: the calculated flow for the stationary-flow case.

The computational complexity of local calculations is very close to Lucas and Kanade's algorithm [5], probably the OF algorithm most similar to ours. In our experiments, however, CPU times were on average 17% smaller for our algorithm. The extra cost, i.e., the computation of the disturbance field,  $D_k$ , is minimal as all quantities involved are computed recursively, and easily offset for many sequences by the fact that full OF computation can take place only where significant motion is detected. Notice that no extra effort is introduced for this test, as  $D_k$  is computed anyway as part of the OF algorithm.

## 5 Experimental results

We ran experiments with ground-truthed standard sequences to assess quantitatively the algorithm's accuracy and to investigate the effects of  $w$ , and with real sequences (no ground truth) to gain a qualitative appreciation of performance. All experiments were performed in MATLAB on a Sparc Ultra 10 (440 MHz) running Solaris 2.8.



## 5.1 Effect of $w$

Here, we investigated the effect of  $w$ , which controls the system's memory span. Integrating information from previous frames should be beneficial when the motion does not change much over the frames (ideally, the flow is stationary), otherwise worsen results moderately. This is indeed proven by experiments. Figure 2 (left) shows a  $200 \times 200$ , cosine shaded Gaussian surface ( $\sigma = 100$ ). We produced a sequence of 30 frames by translating this image along the main image diagonal by (1, 1) at each frame. The resulting, stationary flow computed by our algorithm ( $7 \times 7$  spatial support window) is shown in Figure 2 (right). Table 1 shows the computed values of the three error measures suggested in [www.cs.brown.edu/people/black/images](http://www.cs.brown.edu/people/black/images); calling  $\mathbf{d}$  and  $\bar{\mathbf{d}}$ , respectively, the estimated and ground-truth flow vectors, and  $\theta_v = |\arcsin \frac{v_x}{\|\mathbf{v}\|}|$  the direction of a vector in the image plane, the errors are the average error vector magnitude  $\|\bar{\mathbf{d}} - \mathbf{d}\|$ , the average angular deviation  $\theta_{\mathbf{d}} - \theta_{\bar{\mathbf{d}}}$ , and the average relative magnitude error  $\frac{\|\mathbf{d}\| - \|\bar{\mathbf{d}}\|}{\|\bar{\mathbf{d}}\|}$ . Averages are computed over the whole frame at each instant, then averaged over the whole sequence. The figures suggest that indeed errors decrease as the memory span  $w$  increases. Conversely, errors increase with the memory span in the case of non-stationary flow. Table 2 shows results computed in the same conditions as the previous experiments, but with accelerated motion. We notice that the algorithm is stable in  $w$ , in the sense that small variations of  $w$  result in small variations of the estimated flows.

## 5.2 Accuracy

Here, we ran the algorithm on two standard test sequences, *rotsphere* and *yosemite*, both available at [www.cs.brown.edu/people/black/images](http://www.cs.brown.edu/people/black/images). The former shows a textured sphere

$w$	avg err vect magn	avg ang dev (deg)	avg rel magn err
0.1	0.0224	0.5581	0.0091
0.5	0.0202	0.5570	0.0077
0.7	0.0188	0.5512	0.0068
0.8	0.0180	0.5280	0.0067

Table 1: Errors for the pattern in Figure 2 moving of constant motion (stationary flow). See text for error definitions and motion parameters.

$w$	avg err vect magn	avg ang dev (deg)	avg rel magn err
0.1	0.5243	12.0373	0.0451
0.5	0.5921	13.2199	0.0594
0.7	0.7616	16.0575	0.0997
0.8	0.9203	17.5172	0.1416

Table 2: Errors for the pattern in Figure 2 moving of accelerated motion (varying flow). Frame-frame displacements in pixels along the co-ordinate axis are [1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3] and [1 2 2 2 2 2 2 3 3 3 3] respectively. See text for error definitions and motion parameters.

rotating around a fixed axis, one frame of which is shown in Figure 3 (left); the motion field is stationary. The second is the well-known Yosemite Valley sequence, not shown for reasons of space. Figure 3 (centre) shows the calculated (stationary) optic flow for

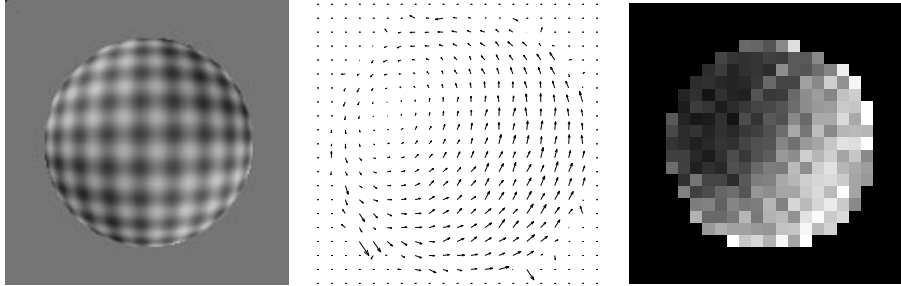


Figure 3: Left: A frame from the *rotspher* sequence. Centre: the calculated (stationary) flow. Right: the total local error over sequence (the brighter the larger).

*rotsphere*; Figure 3 (right) suggests the total local error, by visualising the local average vector magnitude error averaged over the whole sequence. Relative errors (not shown) tends to be larger where the flow is smaller, that is, near the point at which the axis of rotation pierces the sphere. Table 3 summarises quantitative results: errors decrease as the size of the support window increases, and increase slightly as  $w$  increases (nonstationary flow). Performance is clearly similar to that of [5], considering the nonstationary flow.

	win. size	$w$	avg err vect magn	avg ang dev (deg)	avg rel magn err
LK	7	0.3	0.2627	5.6104	0.1067
	7	0.5	0.2721	5.4933	0.1146
	7	0.7	0.2941	5.4436	0.1317
	7		0.1808	3.2866	0.0919
	9	0.3	0.2167	4.3093	0.0943
	9	0.5	0.2268	4.2719	0.1006
LK	9	0.7	0.2541	4.202	0.1201
	9		0.1783	2.8939	0.0905

Table 3: Errors for the sequence in Figure 3. For comparison, the LK line gives the results of our implementation of Lucas and Kanade. See text for error definitions and motion parameters.

Table 4 shows quantitative results for the *yosemite* sequence (nonstationary flow). Again the tendency is that errors decrease with decreasing  $w$  and increasing support window size. Errors are higher at the discontinuity earth-sky, as expected. Notice that the nonstationary flows used here are less than ideal for the algorithm, as frame-frame disparities are at least one pixel. This means that potentially considerable flow variations are considered over the memory span driven by  $w$ , violating the quasi-constant flow assumption 4. Experiments indicate a substantial improvements in accuracy when smaller interframe disparities are used. For instance, we build a subpixel-displacement sequence using the pattern in Figure 2, with 0.25 pixel displacement between frames. Table 5 shows that the



errors computed as per Table 2 are reduced approximately by two orders of magnitude.

### 5.3 Tests with real sequences

We show here a qualitative comparison between our algorithm and our implementation of [5] with a real sequence (no ground truth) of a person walking. Figure 4 shows three frames of the input sequence, and the associated OF fields computed by our algorithm and by [5]. There are two motions in the image, as the camera is following the moving figure, which itself moves while staying approximately in the centre of the image. On average, our algorithm achieves better results on background pixels.

## 6 Conclusions

We have derived a closed-form link between *disturbance fields*, a variety of temporally weighted frame differences, and the normal optic flow. We have used this to derive a novel, efficient optic flow algorithm based on a recursive-filter formulation. Most quantities are computed recursively. The disturbance field is one of these, so that the full algorithm can be applied only to pixels where motion is significant (assuming, as usual, that intensity changes are due to scene motion). The quality of results seems suitable for several tasks not requiring highly accurate estimates of the image motion field, e.g., region segmenting by motion, and comparable (better at times) with that achieved with our implementation of [5], a very similar algorithm in terms of approach and complexity.

algor	win. size	$w$	avg err vect magn	avg ang dev (deg)	avg rel magn err
LK	7		0.4807	9.2749	2.8829
	7	0.3	0.6322	12.4394	3.3296
	7	0.5	0.5920	12.0433	3.414
	7	0.7	0.5965	12.0367	3.9648
LK	9		0.4125	9.041	3.2797
	9	0.3	0.5044	11.3249	3.7966
	9	0.5	0.4844	11.0966	3.9045
	9	0.7	0.4882	10.8859	4.6081

Table 4: Errors for the *yosemite* sequence. For comparison, the LK line gives the results of our implementation of Lucas and Kanade. See text for error definitions and motion parameters.

$w$	avg err vect magn	avg ang dev (deg)	avg rel magn err
0.1	0.0018	0.1376	0.0048
0.5	0.0013	0.109	0.0034
0.7	0.0011	0.1079	0.0027
0.8	0.0011	0.1031	0.0031

Table 5: Errors for the pattern in Figure 2 moving of constant motion (stationary flow) with sub-pixel interframe displacements. See text for error definitions and motion parameters.



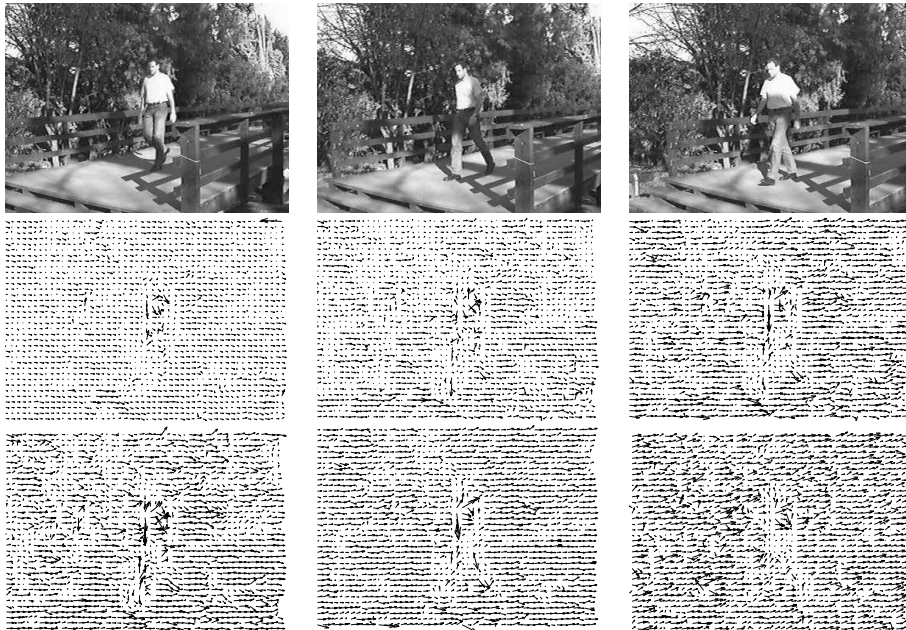


Figure 4: Top row: three frames (11, 21, 30) from a sequence. Middle row: OF fields computed by our algorithm ( $w = 0.5$ ,  $9 \times 9$  support window). Bottom row: the same computed by our implementation of [5].



The recursive formulation introduces a memory effect which can be beneficial with near-stationary flows and worsens errors only slightly with non-stationary flows. Future work includes testing the algorithm for figure-ground segmentation in an immersive videoconferencing environment, and investigating an adaptive  $w$ .

## Acknowledgments

We thank Francesco Isgrò and Alessandro Verri for discussions and comments, and Tiziano Tommasini for precious support with implementation and experiments. This work was partially supported by EU IST grant “VIRTUE” (Virtual Team Environment, [www.cee.hw.ac.uk/virtue/research.html](http://www.cee.hw.ac.uk/virtue/research.html)), and V. Roberto’s Italian Space Agency grant.

## References

- [1] G Halevy and D Weinshall, *Motion of disturbances: detection and tracking of multi-body non-rigid motion*, Proc. IEEE Int. Conf. on Comp. Vision and Patt. Rec. CVPR97, San Juan, 1997.
- [2] I. Haritaoglu, D. Harwood and L. S. Davis: *W4: Real-Time Surveillance of People and Their Activities*, PAMI, vol 22, no 8, 2000, pp. 809–830.
- [3] S. Haykin: *Adaptive filtering*, Prentice Hall (fourth edition), 2002.
- [4] F Isgrò, E Trucco and L-Q Xu: *Towards teleconferencing by view synthesis and large-baseline stereo*, Proc. IEEE/IAPR Int. Conf. on Image Analysis and Processing, Palermo (Italy), 2001.
- [5] B D Lucas and T Kanade: *An Iterative Image Registration Technique with an Application to Stereo Vision*, Proc. 7th Int. Joint Conf on Artif. Intell., Vancouver (CA), 1981, pp. 674–679.
- [6] A Mitiche and P Bouthemy: *Computation and Analysis of Image Motion: A Synopsis of Current Problems and Methods*, Int Journ of Comp Vision, vol 19 no 1, 1996, pp. 29–55.
- [7] P Rosin: *Thresholding for change detection*, Proc. IEEE Int Conf on Computer Vision and Pattern Recognition CVPR’98, pp.274–279.
- [8] E Trucco, K Plakas, N Brandenburg, P Kauff, M Karl and O Schreer: *Real-time disparity analysis for immersive 3-D teleconferencing*, IEEE ICCV’01 Workshop on Video Registration, Vancouver (Canada), 2001.
- [9] Y Ricquebourg and P Bouthemy: *Real-Time Tracking of Moving Persons by Exploiting Spatio-Temporal Image Slices*, PAMI, vol 22 no 8, 2000, pp. 797–808