MVA '96 IAPR Workshop on Machine Vision Applications, November. 12-14, 1996, Tokyo, Japan

Computation of Optical Flow using Dynamic Programming

Georges M. QUÉNOT*
LIMSI-CNRS

Abstract

This paper presents an original algorithm for the computation of optical flow called Orthogonal Dynamic Programming (ODP) as well as several enhancements to it. The principle is to minimize a sum of square differences (SSD) between a pair of images. The originality of the approach is that an optimal matching is searched for entire image strips rather than for pixel neighborhoods. Dynamic programming is used to provide very robust strip alignments and a multiresolution iterative process is used to compute the velocity field. Extensions to the computation of the velocity field for non integer image indexes, to the use of more than two images, and to the search for subpixel velocities, are presented. Results obtained for the Barron, Fleet and Beauchemin performance tests appear to be at least as good as or better than those obtained using classical optical flow detection methods.

1 Introduction

Several techniques have been developed for the computation of optical flow. In a survey and a comparative performance study [1], Barron, Fleet and Beauchemin classify them in four categories: differential, correlation based, energy based, and phase based.

The method proposed here can be considered as a correlation based one since its principle is to minimize a sum of square differences (SSD) between a pair of images. The originality of the approach is that an optimal matching is searched for entire image strips rather than for pixel neighborhoods. Dynamic programming is used to provide very robust strip alignments and a multiresolution iterative process is used to compute the flow field.

This method, introduced in [2] for optical flow detection from a pair of images, has been extended to be able to operate on longer sequences of images. It is also able to compute directly the flow field for any intermediate image within the sequence (with non integer image index).

Compared to classical methods, this one has the advantage to be able to operate on multi-band (color) images and to provide a dense flow field for the whole image (neither holes nor border shrinks). Continuity and regularity constraints enforced by dynamic programming leads to a very good flow field estimation even in homogeneous or aliased areas.

2 The Orthogonal Dynamic Programming algorithm

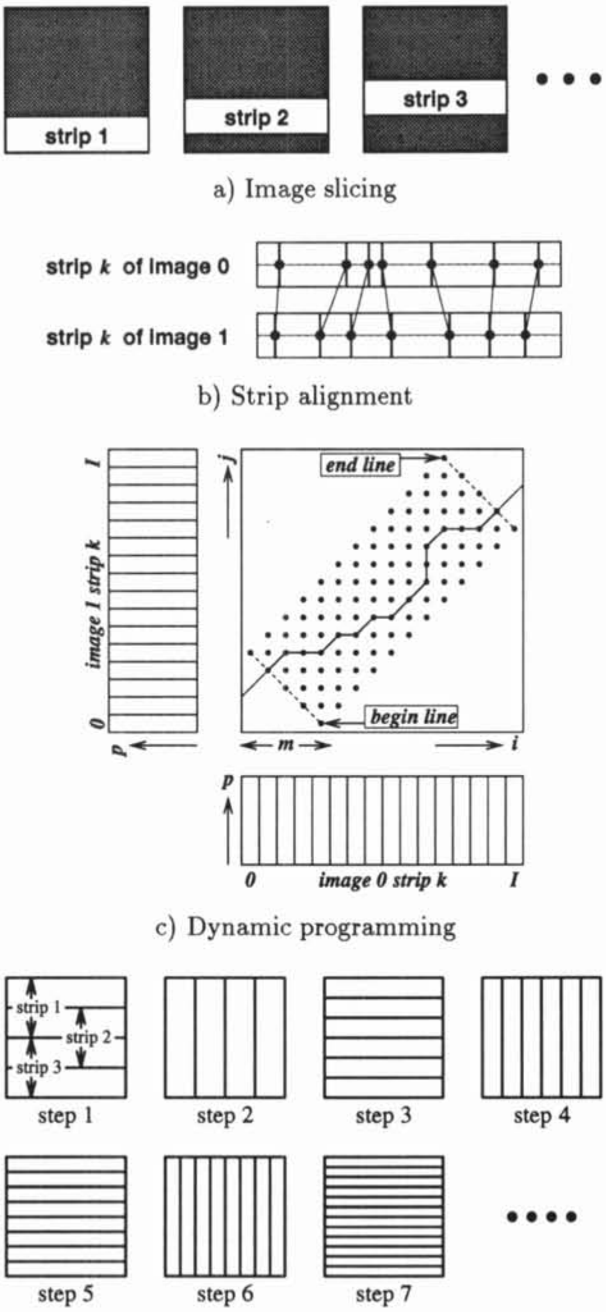
Dynamic Programming is a very robust technique for searching optimal alignments between various types of patterns because it is able to include order and continuity constraints during the search. However, it is applicable only for the search of monodimensional alignments (the reason is that no natural order can be found for a multidimensional set) and uneasy to use directly for image matching although some attempts have been made [3] [4] [5].

The originality of the Orthogonal Dynamic Programming (ODP) algorithm [2] is that it transforms the search problem for two-dimensional alignments into a carefully selected sequence of search problems for monodimensional alignments. It is based on an iterative search for a velocity field that brings the second image over the first one while minimizing the L_1 or L_2 distance between them.

Both images are sliced into parallel and overlapping strips (Figure 1a). Corresponding strips are aligned (Figure 1b) using dynamic programming exactly as 2D representations of speech signal are with the DTW algorithm [6]. An optimal segment to segment alignment is searched for as a continuous and increasing path inside the disparity matrix along which the sum of the local distance (segment to segment L_1 or L_2 distance) is minimum (figure 1c). A dense velocity field is build (or updated) for the whole image by interpolation and smoothing using the relative velocity values obtained for the strips central lines. Two passes are performed using orthogonal slicing directions. This process is iterated in a pyramidal fashion by reducing the spacing and

* Address: LIMSI-CNRS, BP 133, 91403 Orsay CEDEX, France. E-mail: quenot@limsi.fr

width of the strips to refine the accuracy of the matching result (Figure 1d). The width of the search window around the diagonal of the disparity matrix is also reduced during the iterative process.



a) Image slicing

b) Strip alignment

c) Dynamic programming

d) Strip spacing and width reduction

Figure 1: Orthogonal Dynamic Programming

At each iteration, the alignment is searched not between the two original images but between the original image 0 and the image 1 transformed using the velocity field obtained from the previous iterations to bring it over the image 0. In this case, the velocity field is computed in image 0. Computation of the velocity field in image 1 is alternatively possible if the velocity field is used to bring image 0 over image 1.

3 Enhancements

Several enhancements have been made to the original ODP algorithm [2].

3.1 Computation of the velocity field for non integer image indexes

The optimal path inside the disparity matrix (Figure 1c) may be written as: $i_1 = i_0 + v_0(i_0)$ (computation of the velocity field in image 0) or as: $i_0 = i_1 - v_1(i_1)$ (computation of the velocity field in image 1). Il may also be written as: $i_0 = i_\lambda - \lambda.v_\lambda(i_\lambda)$ and: $i_1 = i_\lambda + (1-\lambda).v_\lambda(i_\lambda)$ (computation of the velocity field in image λ with $0 \le \lambda \le 1$).

If $\lambda = 0$ or $\lambda = 1$ we fall in one of the original cases. In the other cases, the values of i_λ for which the optimal path provides values are non integer ones. A simple linear interpolation provides the values v_λ for the integer values of i_λ. During the ODP iterations the images to be sliced and aligned are the images 0 and 1 transformed to the λ image index using respectively the $-\lambda.v_\lambda$ and $(1-\lambda).v_\lambda$ velocity fields.

The velocity field for non integer image indexes is useful for image interpolation using motion information.

3.2 Extension to image sequences

The original ODP algorithm operates using only a pair of images. When a longer image sequence is available a pair of images have to be manually selected around the one for which the velocity field is searched. A natural extension is to select automatically a pair of images inside the sequence. The optimal way to do so is to change the selected pair during the execution of the algorithm. Simultaneously to the reduction of the spacing and width of the strips, the spacing of the images may be increased, from one at the beginning, up to an optimal value that is adaptively determined by a maximum allowed distortion or by the available number of images.

3.3 Search for subpixel velocities

The original ODP algorithm is able to search only for integer velocities at each iteration. Thanks to interpolation and smoothing it is in fact already able to find subpixel velocities. But an improvement is possible through the direct search of subpixel velocities during strip alignment. This is simply done by shrinking around the diagonal line the set of points used for dynamic programming computations (Figure 2). The main effect of this is that the points are moved out of grid. The corresponding segments have non integer and different coordinates. They must be interpolated from the nearest integer coordinate segments of the strips for each point used in the disparity matrix.

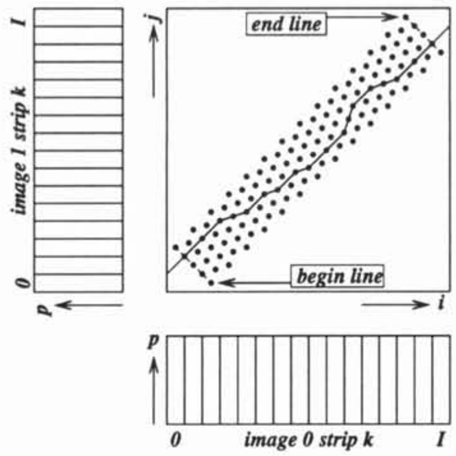


Figure 2: Subpixel resolution search

Subpixel resolution iterations are added after the ones from the original algorithm. Instead of the spacing and width of the strips and the width of the window around the diagonal, from this point this is the distance of the points to the diagonal line of the disparity matrix that is reduced by a $\sqrt{2}$ factor during 10 additional iterations.

Subpixel resolution iterations significantly improve the accuracy of the computed velocity field.

3.4 Use of multi-image distances

When more than two images are available, it is possible to adaptively select a pair of images during the ODP iterations optimizing an accuracy / distortion compromise (section 3.1). When the selected images are separated by one or more intermediate images, it is possible to add the constraint that all the strips of the intermediate images or at least a few of them must also be aligned with the strips of the extreme images (Figure 3). This can be done simply by generalizing the pixel to pixel distance: $\mid p_1 - p_0 \mid$ to a "multi-pixel distance" defined by: $\max(p_i) - \min(p_i)$. Like in the case of subpixel iterations, the segments of the intermediate images generally have non integer coordinates and have to be interpolated from the nearest integer coordinates ones.

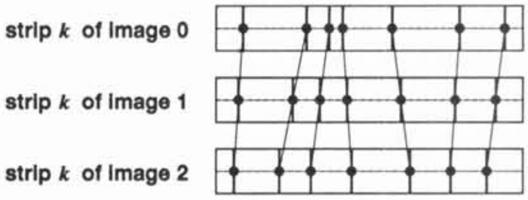


Figure 3: Multi-image strip alignment

4 Experimental results

It has been observed that the ODP algorithm provides a very high quality matching for calibrated patterns as well as for human visual sensation. The results appears to be at least as good as or better than those obtained using classical optical flow detection methods [1]. The choice of the local cost function L_1 or L_2 and of the local recurrent DP equation do not significantly affect the quality of the results. The obtained velocity fields appear to be continuous and differentiable.

4.1 Synthetic image sequences

The results presented in Table 1 have been computed using the image sequences and the error computation procedures provided by Barron, Fleet and Beauchemin [1] so that results can be compared directly. The density measure is taken for the *whole* image (no border offsets have been applied). Like in [1] the error is measured by the angle in degrees between the correct and computed 3-component $(v_x, v_y, 1)$ vectors representing the flow field. The error is displayed as mean $\pm$ standard deviation. Method 1 simply uses adaptive image pair selection, method 2 adds subpixel resolution iterations, and method 3 adds the use of multi-image distances. Methods 1, 2 and 3 take respectively 1, 1.5 and 9 minutes on a 250 MHz R4400 Indigo2 workstation for the yosemite sequence (316×252 images).

4.2 Real image sequences

Figure 4 shows one image and the found velocity field for the four real image sequences provided by Barron, Fleet and Beauchemin [1].

5 Conclusion

The Orthogonal Dynamic Porgramming (ODP) algorithm for the computation of optical flow and several enhancements to it have been presented. This algorithm provides a dense, continuous and differentiable velocity field. It is based on an iterative L_1 or L_2 distance minimization using dynamic programming alternatively on horizontal and vertical image strips while reducing spacing and width. It

Sequence	Method 1	Method 2	Method 3	Count	Density
sinusoid1:	0.57±0.27	0.25±0.27	0.14±0.13	10000	100.0
sinusoid2: (1)	0.00±0.00	0.00±0.00	0.00±0.00	10000	100.0
square1:	0.49±0.54	0.39±0.48	0.32±0.42	10000	100.0
square2:	1.05±0.74	1.07±0.88	1.15±0.83	10000	100.0
treed:	1.01±0.93	0.86±0.93	1.32±1.49	22500	100.0
treet:	0.54±1.07	0.57±1.26	0.20±0.23	22500	100.0
yosemite:	6.41±9.10	5.83±9.39	5.78±9.60	79632	100.0
yosemite: (2)	3.82±7.01	2.97±7.22	2.71±7.04	39684	49.8
yosemite: (3)	3.60±6.87	2.72±6.99	2.41±6.77	15076	18.9

(1) The correct field is a pure translation by a vector with integer components that matches exactly the algorithm search resolution.
(2) Result using a 1D confidence measure (strong intensity gradient).
(3) Result using a 2D confidence measure (strong variation of gradient direction).

Table 1: Performance tests results.

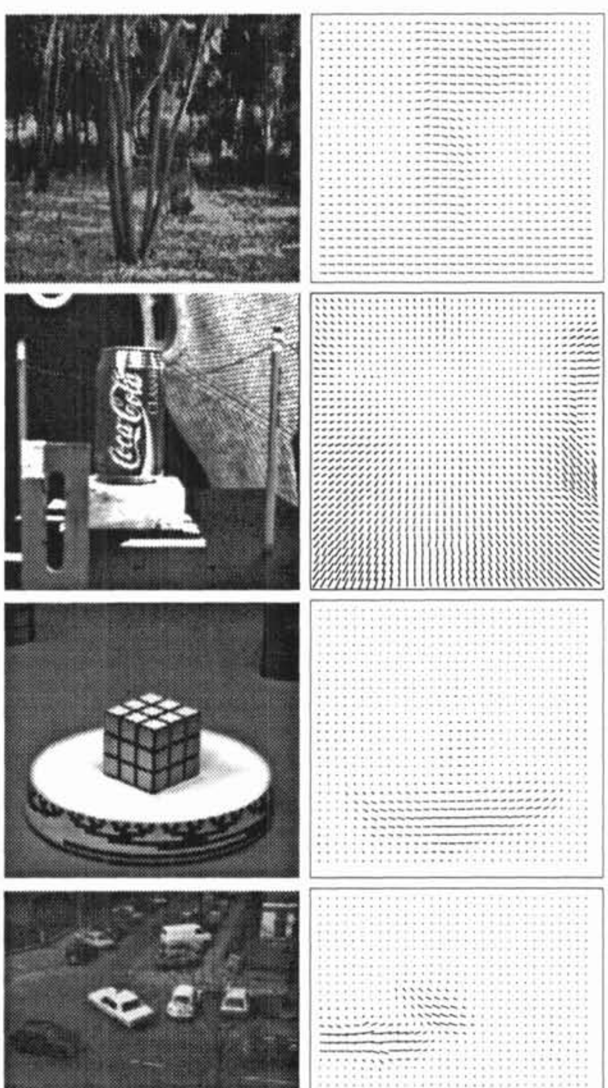


Figure 4: Image and velocity field for real sequences

has been observed that this algorithm has a performance at least as good as or better than classical optical flow techniques. Applications to stereovision and image interpolation have been developed [7].

References

[1] J.L. Barron, D.J. Fleet and S.S. Beauchemin, "Performance of Optical Flow Techniques", *International Journal of Computer Vision*, Vol. 12, No 1, pp. 43-77, 1994.

[2] G.M. Quénot, "The "Orthogonal Algorithm" for Optical Flow Detection using Dynamic Programming", *Proc. IEEE ICASSP-92*, Vol III, pp. 249-252, San-Francisco, 23-36 March 1992.

[3] Y. Otha and T. Kanade, "Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-7, No. 2, pp. 139-154, March 1985.

[4] B. Burg P. Missakian and B. Zavidovique, "Pattern Recognition through Dynamic Programming", *SPIE's 29th Annual Internal Technical Symposium*, San-Diego, July 1985.

[5] P. Adam, B. Burg and B. Zavidovique, "Dynamic Programming for Region Based Pattern Recognition", *Proc. IEEE ICASSP-86*, pp. 2075-2078, April 1986.

[6] H. Sakoe and S. Chiba, "Dynamic Programming Optimization for Spoken Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Proc.*, Vol. ASSP-26, p.43, February 1978.

[7] G.M. Quénot, "Image Matching using Dynamic Programming: Application to Stereovision and Image Interpolation" *Image Communication*, Bordeaux, France, 20-22 mai 1996.