Mode based optical flow estimation.

Jose Otero Rodríguez (jotero@lsi.uniovi.es)
Adolfo Otero Rodríguez
Luciano Sánchez Ramos

Keywords: Computer vision,motion analisys, optical flow, robust estimation

ABSTRACT. *In this paper a new robust estimator of the optical flow field is shown. Arising from the techniques based on the O.F.C. (Optical Flow Constrain), we develop an estimation that takes several measures around a given pixel and discards the erroneous ones, using a voting schema. In this way, velocities that have more accumulated votes are taken as the correct ones. A preliminary comparison with other well known techniques is shown, and clearly our approach performs quite better, even in the cases where two or more movements are present in the analyzed pixels.*

1 Introduction.

There are basically three ways to perform the calculation of the optical flow field:

- Gradient based techniques.
- Correlation based techniques.
- Frequency based techniques.

Gradient based techniques use the well known "optical flow constrain" in order to compute the optical flow [7]. This equation makes the assumption that intensity changes in a sequence of images are due only to the movement of the objects in the scene: a single pixel will have constant brightness in the different positions that it takes during the sequence. Mathematically this is expressed as shown in equation 1.

$$\frac{dI(x,y,t)}{dt} = 0 \tag{1}$$

As it can be seen, the spatiotemporal derivatives must be estimated. This estimation is noise sensitive, so the optical flow estimation has the same problem. Later in this paper we will analyze this expression in depth.

Correlation based techniques [1] [2] try to minimize a measure of similiraty (or disimiliraty) between patches taken from two consecutive frames centered in a given pixel. The displacement that maximizes (or minimizes) the selected measure, divided by the interval between the acquisition of the frames, is the velocity of the pixel. Unfortunately, this approach is computationally expensive, and the complexity grows with the square of the maximum displacement searched. There are approaches that solve this problem using a bidimensional LUT instead of perform floating point calculations, but the gray scale depth must be limited to maintain the LUT in a razonable size [11].

Frequency based techniques use a set of tuned spatiotemporal filters to search for the velocity of a pixel [6]. Each one of the filters will give a response to the stimuli of the data (the sequence of images), the filter with the maximal response will be tuned with the velocity that we are looking for, so identifying the filter we are identifying the velocity. May be this is the most precise approach, but it is very expensive in terms of computational cost.

2 Overview of gradient based techniques.

The optical flow constrain 1 can be expanded and written in the form shown in 2.

$$-\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} = \frac{\partial f}{\partial x}u + \frac{\partial f}{\partial y}v = \nabla(f)c \tag{2}$$

Where $\nabla(f)$ is the spatial gradient of the image, c is the velocity of the pixel and $\frac{\partial f}{\partial t}$ is the temporal derivative of the sequence in that pixel.
The previous expression shows how the spatiotemporal derivatives are related to the observed velocities.
Unfortunately, there are two unknowns in this equation, so it is not possible to solve it in order to recover the full motion vector. This is known in the literature as the "aperture problem" [8] meaning that there is no way to recover the complete optical flow vector using only local (one pixel) information. With local information we can only compute the motion in the direction of the gradient (known as "normal flow").
Many authors try to solve the aperture problem with the incorporation of some kind of global information, involving a process of regularization. This is: given a measure of some error, the process of regularization consists on applying extra restrictions to the sequence of images, searching for a minimum in the measure of the error. One of the first algorithm found in the literature using this technique is the one by Horn and Schunk [7]), in the early 80's. They apply a restriction that consists in maximize a measure (a global smoothness criteria) and minimize another one (the error given by the fit to OFC); these restrictions are applied to all the image.
Some researchers use a estimation of the velocities with a confidence measurement, so for each measure we now how reliable it is [1].
There are authors that use other invariants than pixel intensity like the zero crossings of the laplacian of the gaussian [5]. The motivation is that they are closer to the biological facts in animals visual systems.
Finally, there is another alternative that analyzes the measures in the space of the velocities, that is, it tries to find a robust estimation of the velocity performing an analysis of the results of many systems of OFC equations, each one applied to a pair of pixels (see figure 1) or it tries to fit the data to a model in order to estimate the velocity. In this way, the analysis is performed directly in the data domain that we want to recover, that is, the u, v space [10][12][4].

3 Robust estimation of Optical Flow

Usually in real imaginery, there are many independent moving objects. In this situation, the problem of occluding surfaces arises: places where the velocity changes suddenly from one pixel to another. In this case it's not possible to use approaches like [7] because the global smoothness criteria doesn't holds in those boundaries. Other authors like Nagel [9] propose relaxing the smoothness criteria in places where the gradient is high, smoothing the flow along the contour but not thru the contour. But the problem remains: through the fact that the OFC has two unknowns, it's necessary to take measurements from at least two pixels. If the pixels are chosen from objects with different velocities, the solution to the system of

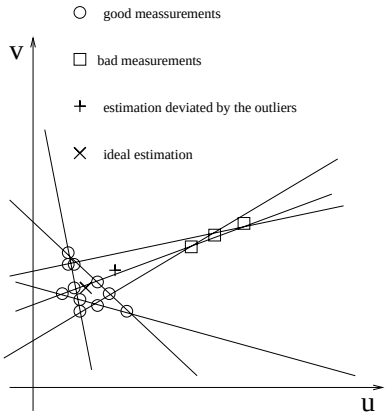

Figure 1: Clusters of intersections of OFC's pairs. Some of them belong to a dominant cluster, a few ones can deviate the estimation if they are not rejected.

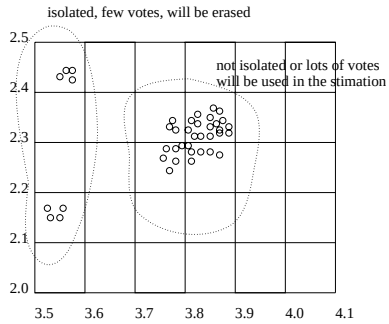

Figure 2: Grid where the voting is performed. As can be seen, the results in this stage are discrete. Later the solutions in left side of the figure will be erased.

equations obtained will give an erroneous velocity. It is necessary the analysis of the velocity distributions presented in a neighborhood of each pixel, in the direction addressed in [12] to determine which velocities are supposed to be correct and which one is the dominant velocity in the neighborhood. In the next section our approach in this direction is explained.

3.1 Vote cumulation for velocity selection

Our first approach consists on the following steps:

- For each pixel we consider a neigbourhood of a given size. For each possible couple of pixels, we solve the system of equations given by the OFC applied to the couple of pixels.
- We discard the solutions whose module exceeds a given constant as errors. We discard also the solutions of systems whose condition number is too high.
- After this, we build a data matrix which elements are the number of times a velocity lies on a given range, in x and y separately. That is, each possible velocity from a set is voted if the data lies in its range. For instance, the velocities in the range form 3.8 to 3.9 vote for 3.85 value.
- We have now a sort of image or surface with brightness (or height) proportional to the accumulated votes. We filter the data with a gaussian kernel and after this we search

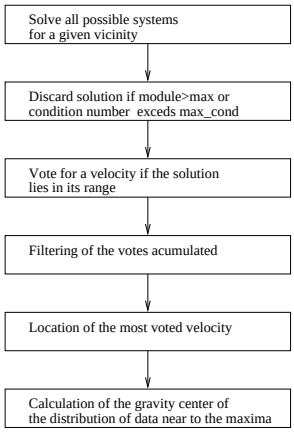

Figure 3: Schema showing the different stages of the method explained in 3.1.

for the maximum in it. We do this to erode the local and isolated maxima.

- The velocity that corresponds to the maximum is an estimation by itself, but to improve the precision we perform another estimation. For every data that is closer to the maximum than a given percent of it, we search for the gravity center of them, that is, we consider that the velocities have associated a weight and we calculate the gravity center of that distribution. This is our final estimation.

In figure 3 the process is summarized in a schema.
The idea of this process is to avoid that a wrong estimation like the one in figure 1 could happen. In that figure, it can be seen that a few data can corrupt a traditional estimation like minimum least squares. Our approach perform well in this cases because it eliminates that wrong data in two stages, first because of the filtering and second when we compute the gravity center of the remaining data.

3.2 Mode as an estimator of Optical Flow

The approach explained in previous section gives good results but it is expensive in terms of computational cost. The results are also discrete in the sense of being in a finite set of values, the grid over the vote accumulation is performed. If the grid is smaller, the number of possible values is greater but the computational cost grows. Because of this two reasons, we develop another method equivalent to the former, based on the use of the mode as an estimator of the most reliable velocity, the result of the previous approach is an approximation of the mode.
As in the previous method, we compute the solutions of every possible pair of OFC's (except the systems whose condition number is to high) for a given neighborhood. We discard the solutions whose module is greater than a threshold as errors. The solutions to the system of OFC's solved are arranged into an array with three columns. In the first column we store the velocity component in x direction, in the second column we store the velocity component in y direction as shown in 4(the array is rotated in the figure).
The array is sorted using the following criteria: the components in x direction are compared first and then if they are equal, the components in y direction are compared. Now for each velocity we count how many velocities are closer than a threshold to it (d in figure 4). Each velocity close to the former increases one unit the third column of the row where

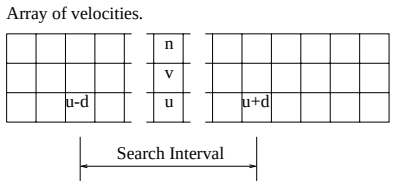

Figure 4: This is the array where the velocities are sorted in order to improve the performance of mode calculation. Because the sorting criteria, we only need to examine the velocities that belong to the search interval.

that velocity is stored, so in that field we store the number of neighbors that each velocity has, closer than the mentioned threshold. Because the array is sorted as we mentioned above, only the velocities such that the difference between component in x direction with the velocity being analyzed is smaller than the threshold must be considered, avoiding the need to compute the SSD for each pair of velocities. In each iteration, the velocity with the highest number of neighbors is stored, and for each neighboorhod centered in the pixel whose velocity is being calculated, that velocity with the maximum number of neighbors is taken as an estimation of the former. The idea is to find which velocity is the one who has more velocities closer to it, in a given interval, in order to find the biggest cluster of velocities for the whole set of solutions calculated. This cluster is supposed to be representative of the neighboorhod being studied and then this is assumed to be an estimation of the velocity of the central pixel of that neighboorhod. The ratio between the number of neighbors of the velocity taken into account in the estimation and the number of velocities computed is a measurement of the reliability of the estimation itself, the bigger the ratio is more reliable the estimation is.

4 Results

In this section we will show some preliminary results of the algorithm. In the left side of figure 5 it can be seen how the solutions of OFC pairs are distributed when the whole region being studied has the same velocity. There is some dispersion because the erroneous calculations of spatiotemporal derivatives but the velocities that have more votes are clustered in a defined group.

In the right side of figure 5 it is shown what kind of results are obtained when in a given region coexist pixels with different velocities. For pairs of OFC applied to pixels with different velocity we obtain a cluster of solutions. For pixels with the same velocity we obtain a cluster too, one for each velocity in the region being studied. The problem is how to find the cluster that correspond to the velocity of most of the pixels of the image. Our approach can perform this task easily because the voting schema applied. In the discretized space of velocity, each possible solution has a number of votes associated. After the filtration step, the most voted velocity is found. Then the gravity center of the solutions with a number of votes close to the maxima is taken as an estimation of the velocity.

In right side of figure 7 it is shown the optical flow field obtained using a least squares estimation over a $9x9$ neighborhood from the sequence of images in figure 6, where two sinusoidal patterns move with velocities $(1, -1)$ (pattern in up and left conner) and $(-1, 1)$

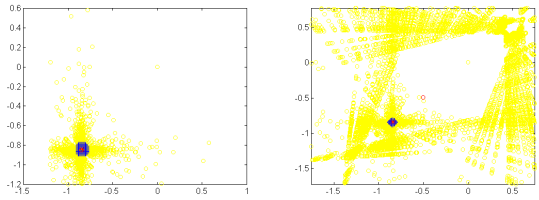

Figure 5: In the left side of this figure can be seen how the intersections of OFC's obtained from pixels with similar velocity are grouped together in a well defined cluster. In the right side is shown what happens when two velocities are present.

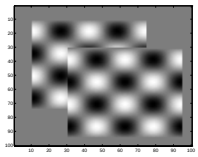

Figure 6: Test sequence. Two overlapped sinusoidal patterns move in opposite directions.

(pattern in down and right conner). The optical flow shown in right side of figure 7 is zoomed between rows 50 70 and columns 20 40 in order to show the boundaries of the movement, where the estimation is very bad, because the pro-mediation between different velocities give as result an inexistent velocity. There is also an effect of undesirated smoothness, the change from one velocity to another takes place slowly.
In left side of figure 7 we show our estimation of the optical flow field from the same sequence of images shown in figure 6, zoomed between rows 50 70 and columns 20 40, so it can be seen the same detail than in right side of the same figure.
It can be seen that the estimation only fails in two cases:

- The central pixel of the neighborhood lies on a surface that moves in different way than the most of the other pixels in the neighborhood. This happens in the corners at 90 degrees, for instance.
- The central pixel of the neighborhood belongs to one surface in first frame and to another, occluded by the former, in the second frame.

But apart from these situations, the estimation of the velocity is much better than the one

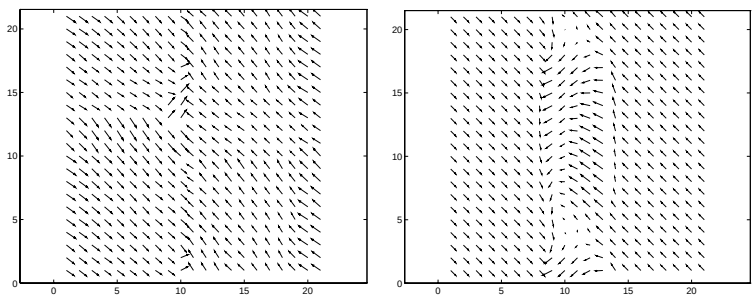

Figure 7: In the left side of this figure is shown the optical flow obtained with our method. In the right side a estimation with least squares is shown. As can be seen our approach preserves the boundaries of the movement.

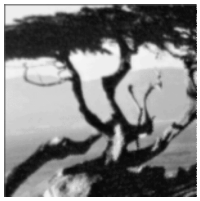

Figure 8: A frame from the "traslating tree" sequence. The sequence is semysintetic an shows a tree moving from left to right.

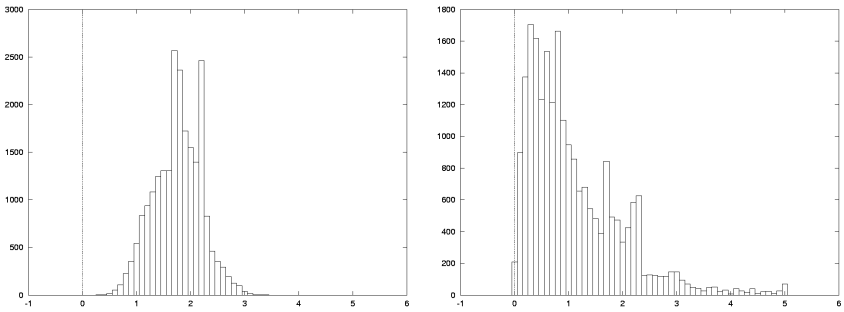

Figure 9: Left side of this figure shows the histogram of the module error using Anandan's algorithm the error is 3.9809e+04 in total. In the right side is the error histogram using our algorithm the total error is 2.4720e+04 in total.

obtained from LS estimation, this is because our approach discards the measures obtained from couples of pixels that:

- Have different velocity.
- Their OFC's are almost parallel, so the solution of the system is not reliable.

Finally, in figure 9 a quantitative comparision betwen Anandan's [1] algorithm and our approach is performed. The test sequence used in this experiment was the well known "traslating tree" proposed as benchmark in [3]. It is a semi-syntetic sequence where a tree moves in front of a camera from left to right.

We define the error as 3 and the total error as 4.

$$\sqrt{(u_{correct} - u_{obtained})^2 + (v_{correct} - v_{obtained})^2} \tag{3}$$

$$\sum_{In\ the\ image} \sqrt{(u_{correct} - u_{obtained})^2 + (v_{correct} - v_{obtained})^2} \tag{4}$$

The histogram in left side of figure 9 is obtained from Anandan's algorithm, in this case, the total error was 3.9809e+04. In the right side the histogram obtained from our approach, the total error fall to 2.4720e+04.

5 Conclusions and future work

The work presented here is in a preliminary stage of development, more test with the sequences used in [3] must be performed, as well as an analysis based on numerical quantities,

not only in qualitative properties. But the approach presented here seems to be promising because the robust estimation is an imperative requisite in real imaginery, where the presence of noise implies erroneous measurements. These erroneous measurements are rejected in this approach, because the searching of a velocity that is the center of a big cluster of similar values has more probability to be an accurate estimation of the velocity than isolated values.

References

[1] Anandan, P. "A computational framework an an algorithm for the measurement of visual motion". International Journal of Computer Vision, 2, pp 283-310, 1989.

[2] Nicola Ancona y Tomaso Poggio. Optical Flow from 1D Correlation: Aplication to a simple Time-To-Crash Detector. Massachusetts Institute of Technology, Artificial Intelligence Laboratory and Center for Biological and Computational Learning.

[3] S.S. Beauchemin and J.L. Barron. The computation of Optical Flow. Dept. of Computer Science. University of Western Onterio. London, Ontario, Canada.

[4] D. Ben-Tzvi,Del Bimbo, P. Nesi. Optical Flow from Constrain Lines Parametrization , Pattern Recognition, Vol. 26, pp. 1549-1561, Nov. 1993.

[5] J H Duncan and T-C Chou. On the detection of motion and the computation of optical flow. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 14, N 3 March 1992.

[6] D. J. Heeger. Optical flow using spatiotemporal filters. Intl. J. Comp. Vis., 279-302, 1988.

[7] Berthold K. P. Horn y Brian G. Schunk. Determining Optical Flow. A.I. Memo N572. Massachusetts Institute of Technology, Artificial Intelligence Laboratory 1980.

[8] David w. Murray y Bernard F.Buxton. Experiments in the Machine Interpretation of Visual Motion. Massachusetts Institute of Technology 1990.

[9] H. H. Nagel. Displacement Vectors Derived from Second Order Intensity Variations. Computer Vision, Graphics, and Image Processing 21, 1983.

[10] P. Nesi. Real-Time Motion Estimation. Department of Systems and Informatics, Faculty of Engineering, University of Florence, 1994.

[11] Jose Otero Rodriguez, Jose A. Cancelas, Rafael C. Gonzalez. Optical Flow Calculation Using Look-Up Tables. IFAC ICV 98.

[12] Brian G. Shunck. Image Flow Segmentation and Estimation by Constrain Line and Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, pp. 110-1027, Oct. 1989.