Robust Affine Flow Estimation with Controlled Computation Cost

Etienne Grossmann* **José Santos-Victor**

Instituto de Sistemas e Robótica,
Instituto Superior Técnico
Torre Norte - Piso 7 - Av. Rovisco Pais, 1
P 1096 Lisboa Codex - PORTUGAL
{etienne,jasv}@isr.ist.utl.pt

Abstract. This paper addresses a classical problem -optical flow- for which we propose novel estimation algorithms based on robust regression. Another important aspect of our work is that we are concerned with the computational efficiency of our estimators, in the sense of maximizing its performance, for a chosen computation cost. This concern about efficient estimation appears often in active vision. It poses a much harder problem than estimation alone, and there seem to be very few theoretical results. After discussing the issue of performance assessment, we present a practical method for comparing the performance of different estimators, and the results obtained when comparing our algorithms with a more classical one.

1 Introduction

In vision applications, one often has an estimation problem as a subgoal (motion estimation, localization, etc). Satisfactory solutions for these problems can often be computed off-line but there is the need to know how well a solution can be obtained in a fast procedure. Achieving the desired velocity is usually sought through the amelioration of hardware and/or software.

Both approaches are rewarding : The first has greatly benefited of the technological improvements brought by the ever-spreading use of computers, and of constant research in the relevant fields; the second being mostly driven by research in vision, resulting in improved or altogether new algorithms.

We are mainly concerned by the algorithmic part, and more specifically, by the trade-off between the precision of an estimator and the computational cost at which it can be implemented. From that point of view, estimation problems have not benefited of breakthroughs like other domains (like multigrid methods for numerical partial differential equations) have. The methods of estimation theory were greatly developed in statistics, where not loosing information from the sample it is more important than sparing computing resource.

Despite its importance for obvious practical reasons, this trade-off is a question rarely encountered in literature, and we have very few analytical tools to help us. There are two main contributions in this paper. The first one regards a general discussion of performance analysis and characterization of vision algorithms which only recently has captured the attention of researchers. The second contribution is a novel algorithm for computing the affine optic flow. It is based on a modified robust regression algorithm, and is most closely related to so-called "differential" flow-computing techniques; yet we show in Section 3.5 that it can be linked to "region-based matching" methods. Experiments indicate improvements over more traditional techniques.

We compare the results of different affine estimators when all are given the *same computational resource*, and we discuss how being able to build estimators of a given computational cost can help in maximizing the efficiency. An interesting study of how the performance of optic flow estimators varies with the available computational resource is given in in [8]. As opposed to that paper, we do not consider procedures that compute the optic flow in every pixel, but the *affine flow* (the flow and its first derivatives).

After a brief introduction to optic flow, in Section 2, we present our approach in Section 3, detailing the points where it differs from other methods. In Section 4, we discuss the various criteria to assess the quality of an estimator, and their limitations. Section 5 is devoted to the experimental results and establishes directions of future work.

* This work is partially funded by the HCM-European Robotics Network (ERNET), and by a previous fellowship from the European Commission Human Capital Mobility (HCM) Programme. Thanks to the reviewers for suggestions. Special thanks to César Silva for his help in troubleshooting.

2 Optical Flow

The definition of the optical flow is well known (see [5]), and it is widely accepted that the optic flow estimation is an ill-posed problem. A review of different approaches for optic flow estimation can be found in [1]. The optical flow is a vector field (u, v) defined over an image sequence I[2], such that for every point (x, y) in the image, at any time instant t, the following equation is fulfilled :

$$\mathcal{D}_{(u,v,1)} I = 0 \tag{1}$$

where $\mathcal{D}_{(u,v,1)} I$ is the directional derivative of I in the direction $(u, v, 1)$. This means that along the lines of the 3D vector field $(u, v, 1)$, the brightness is constant.

In computer vision literature, Equation (1) is most often expressed as :

$$I_x u + I_y v + I_t = 0, \tag{2}$$

where I_x, I_y and I_t are the partial derivatives of the image sequence. If we consider the image as having continuous support, both definitions are equivalent. Using discrete images changes the situation. It is widely recognized that the model for the optical flow is only *approximately* valid. These two points will be addressed in Sections 3.1 and 2.2.

We consider the optical flow as the projection of the 3D motion in the watched scene onto the image plane. As such, it is a rich source of information about structure and motion.This has been studied in papers by Koenderinck and Doorn,in [7], Longuet-Higgins and Pradzny in [9], or more extensively in Subbarao's book [11]. For example, knowledge —in a single pixel— of the flow and its derivatives gives information about both the relative motion of the camera and the surface "seen" in that pixel, and the orientation of that surface. A highly accurate estimate of the flow is, of course, preferable.

2.1 Affine Optical Flow When observing a flat surface, the optic flow at the center of the image can be expressed as a degree-2 polynomial of the image coordinates, and one can show that the amplitude of the constant and linear terms is somewhat greater than that of the others. In many cases [10], *we can approximate the flow by an affine model*, which is easier to estimate, and sufficient for many active-vision applications.

We will consider this simplified model and study the effect of the approximation. The affine flow model results from a first-order expansion of the flow $(u(x, y, t), v(x, y, t))$ around a point (x_0, y_0), with flow (u_0, v_0). The affine flow model can then be expressed as :

$$\begin{bmatrix} u(x,y,t) \\ v(x,y,t) \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \tag{3}$$

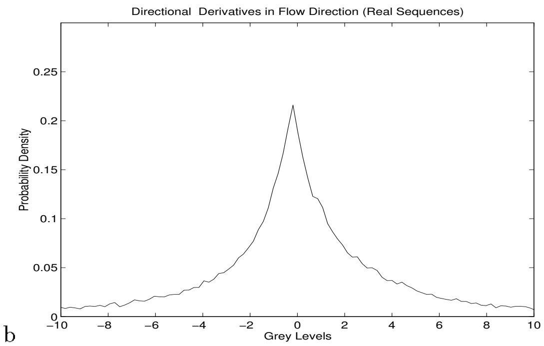
The purpose of the system we present here is to compute the parameters $\boldsymbol{\theta} = [u_0, u_x, u_y, v_0, v_x, v_y]^T$ when the point (x_0, y_0) is the center of the image, $(0, 0)$. The generalization to any other point is trivial.

2.2 Error Modeling Since we use a simplified flow model, we have to consider the various sources of uncertainty (noise), which can be identified as :

- e_1 Non exactness in Equation (1) which defines an ill-posed mathematical problem. It can be violated in the presence of occlusion, shading and other optical effects.
- e_2 Non exactness in the affine approximation Equation (3). It depends on the optical properties and dynamics of the scene, lacks a good statistical modeling, and is likely to produce non-independent noise terms.
- e_3 Error in the estimation of the image values, due to imperfection of the sensor and discretization. Despite its fundamental aspect, and the great development of filtering techniques, this noise is poorly characterized. Often, it is considered to be independent and/or Gaussian, both assumptions being more convenient than realistic.

Even if in practice, these noise terms are hard to characterize separately, we succeeded in gaining some insight on the characterization of the sum of all the terms. Figure 1 shows that the noise cannot be assumed to follow a Gaussian distribution, or even a "slightly contaminated" Gaussian.

[2] An image sequence is a function taking as arguments a point (pixel) in the image plane p (also written (x, y)), and a time instant t. $(u(p, t), v(p, t))$ is then a "vector in the 2D image plane", or, similarly, $(u, v, 1)$ is a "vector in the 3D support of the image sequence" (we often omit the arguments (p, t)) for brevity.



b

Fig. 1.: Histogram of the residues in Equation (1), using the affine flow approximation, computed from real images. The unknown flow parameters are replaced by a good estimate (computed off-line, allowing big computation resource). The tails are important : The variance is 39. This histogram was produced using 34000 measures and 500 bins. The residue is the sum of the noise terms (e_1), (e_2) and (e_3), and a term due to the error in the estimate.

3 The Proposed Approach

This section addresses the estimation of the affine optic flow. We first define the optic flow when discrete images are used.

3.1 Discrete Model for the Optical Flow : Since it is one of the core aspects in our work, we must explain how to transpose the definition of optic flow to discrete sequences of images. We further assume that, x, y, t are integer numbers, and I has discrete support. Then, Equation (1) can be transformed into :

$$I(x, y, t) = I(x + u, y + v, t + 1) \tag{4}$$

in any point and at any time instant. Equivalently, we can write the relation :

$$\mathcal{D}^*_{(u,v,1)} I = 0,$$

where

$$\mathcal{D}^*_{(u,v,1)} I(x, y, t) = I(x + u, y + v, t + 1) - I(x, y, t) \tag{5}$$

We define the partial derivatives I_x and I_y, with the symmetric difference scheme, and I_t with the backward difference scheme :

$$\begin{aligned} I_x(x, y, t) &= [I(x + 1, y, t) - I(x - 1, y, t)]/2 \\ I_y(x, y, t) &= [I(x, y + 1, t) - I(x, y - 1, t)]/2 \\ I_t(x, y, t) &= [I(x, y, t + 1) - I(x, y, t)] \end{aligned}$$

A first-order approximation of $I(x + u, y + v, t + 1)$ in a neighborhood of $I(x + \tilde{u}, y + \tilde{v}, t + 1)$, for any (x, y) and $(\tilde{u}, \tilde{v})$ can be written as :

$$I(x + u, y + v, t + 1) = I(x + \tilde{u}, y + \tilde{v}, t + 1) + [I_x, I_y]\,[u - \tilde{u}, v - \tilde{v}]^T + h(u - \tilde{u}, v - \tilde{v}), \tag{6}$$

where h is the "error" in the approximation.

3.2 Observation equation : The problem now is to derive an observation equation depending on the unknown flow values (u, v), by using Equation (4) and keeping the error as small as possible. For a *chosen* vector $(\tilde{u}, \tilde{v})$, we may write for any point (x, y), the following equations, by using (5) and (6) :

$$-\,\mathcal{D}^*_{(\tilde{u},\tilde{v},1)} I(x, y, t) + [I_x, I_y]\,[\tilde{u}, \tilde{v}]^T - h(u - \tilde{u}, v - \tilde{v}) = [I_x, I_y]\,[u, v]^T \tag{7}$$

where, apart from the error term $h(u - \tilde{u}, v - \tilde{v})$, all the remaining terms in the left hand side can be computed from image data.

The choice of $(\tilde{u}, \tilde{v})$, is then an important decision since it determines the magnitude of the approximation error. We have studied two possibilities.

- **"Anisotropic" approach :** If (u, v) is a random-variable with known expectancy $(\hat{u}, \hat{v})$, we can choose $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) = (\hat{\mathbf{u}}, \hat{\mathbf{v}})$ This choice minimizes —under certain conditions— the error in the observation equation. The (noiseless) observation equation is then :

$$-\mathcal{D}^*_{(\hat{u},\hat{v},1)} I(x,y,t) + [I_x, I_y]\,[\hat{u}, \hat{v}]^T = [I_x, I_y]\,[u, v]^T \tag{8}$$

- **"Isotropic" approach :** The most frequently encountered choice is $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) = (\mathbf{0}, \mathbf{0})$. The (noiseless) observation equation is appealingly similar to Equation (2) :

$$-I_t = [I_x, I_y]\,[u, v]^T = I_x u + I_y v, \tag{9}$$

The "anisotropic" method minimizes the expectation of $|h(u - \tilde{u}, v - \tilde{v})|$, if the following assumptions are verified :

(a) The random variables "u knowing v", and "v knowing u" are integer-valued, unimodal, with modes $\hat{u}$, and $\hat{v}$;
(b) Equation (4) is exact;
(c) The functions $E|h(.,v)|$ and $E|h(u,.)|$ [3] have their minimum in 0 (for all v and u).

Assumption (a) states that one already has an unbiased estimate of the flow, which is the case, and that (u, v) is "well-behaved". Assumption (c) formulates a property of the input images which is easily validated experimentally. Although we have not formally extended this result when u and v take non-integer values, we believe that the rounded values of $\hat{u}$ and $\hat{v}$ are a good choice.

When using the "anisotropic" observation equations, the locations in which the observations are done "follow" the optical flow; i.e. at each frame, they are displaced by the current estimate of the (local) flow. This, in conjunction with the use of autoregression time-smoothing, further justifies the term "anisotropic"; the term "steerable" is used with a somewhat similar meaning to qualify spatial image filters that can be oriented according to the image gradient [2].

3.3 Solving for the Affine Flow Parameters Here we discuss how to solve for the six parameters that define the affine flow. We have a sequence of images, and will successively estimate the value of the parameters at all instants. The goal is then to use observations of the new image to obtain a an improved estimate. If we make our observation using the "Anisotropic" approach, together with the affine flow model Equation (3), we get :

$$-\mathcal{D}^*_{(\hat{u},\hat{v},1)} I + [I_x \;\, I_y]\,[\hat{u} \;\, \hat{v}]^T = [I_x \;\, xI_x \;\, yI_x \;\, I_y \;\, xI_y \;\, yI_y]\,\boldsymbol{\theta} \tag{10}$$

Similarly, using "Isotropic" measurements, one obtains the more commonly seen equation

$$-I_t = [I_x, xI_x, yI_x, I_y, xI_y, yI_y]\,\boldsymbol{\theta} \tag{11}$$

A natural choice for estimating the parameters of the affine flow, would be a maximum-likelihood approach. However, it would require the characterization of each noise term separately, which is difficult in practice. Additionally, it would lead us to a non-linear minimization problem. We plan to address the problem in future work.

3.4 Robust Regression Estimation Here, we have simplified the problem by summing up all the noise terms into a single random variable. We are then faced with a linear regression problem in which the left hand sides of Equations (10) or (11) are the observed values. Since the noise term does not appear to be Gaussian, we opted for robust regression estimators (which are not necessarily maximum-likelihood). Robust regression techniques are suitable when the error follows a slightly contaminated Gaussian distribution. Figure 1 shows that the contaminant may be greater than that. Nevertheless, we used robust regression because it appeared to give reasonably good results, and we assume that the resulting estimates are unbiased.

The robust estimate of the flow parameters, is computed iteratively, through a sequence of estimates which -hopefully- converges to the solution, as shown in Figure 2. We implemented the estimators in two ways : One follow closely the suggestions in [6], and is illustrated on the left side of Figure 2. The second

[3] The expectation is taken over all x, y and images I.

includes an extra step inserted between iterations : We re-compute the left hand side of Equation (10) with our latest estimate, $(\hat{u}, \hat{v})$ (right hand side of Equation 2). According to what we saw in Section 3.1, this should reduce the variance of the noise in the observation. However, doing the extra observations increases the computational cost, and we must decide whether it is worthwhile. The choice is not trivial, as mentioned in the next section.

All together, we now have three different ways of estimating the flow :

– **"Anisotropic"** approach, making observations only once per image. (Classic robust estimator).
– **"Anisotropic-M"** approach, making observations at each iteration of the (Modified) robust estimator.
– **"Isotropic"** approach. Classic robust estimator with observations made with the "isotropic" method.

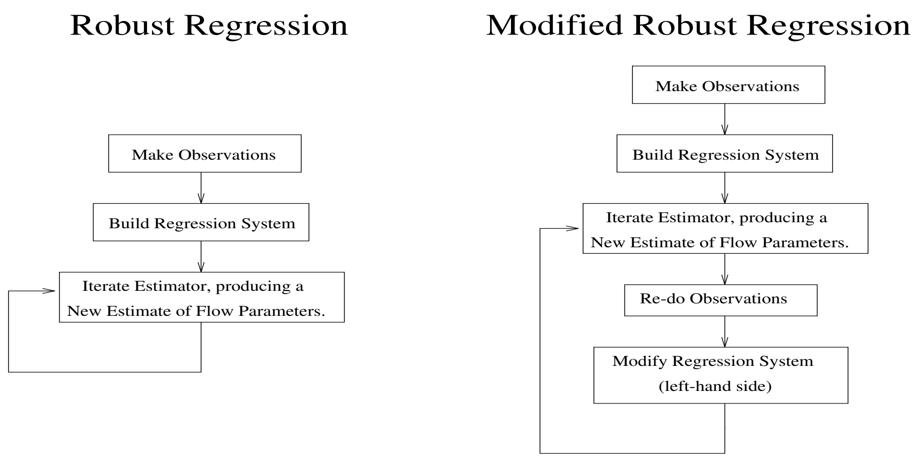


Fig. 2.: Block Diagram of the Robust Estimators : The left diagram corresponds to making one observation per image. In the right diagram, the observations are replaced at each iteration.

3.5 Robust Region-Based Image Matching

A point worth mentioning is that using the "anisotropic" method can be seen to be equivalent to finding the affine deformation[4] of the image plane that "best fits" the image sequence. The so-called "region-based matching techniques" define the optic flow as the vector field that minimizes the difference between the image at a given "time-instant", transformed by the vector field, and the image at the next time-instant. The difference measure is usually, either the correlation, or the "sum of squared differences" (SSD) defined by :SSD $= ||I(x,y,t) - I(x+u, y+v, t+1)||^2$. In our case, a "robust" version of the SSD is minimized.

There are different variants of robust regression [6], the choice of which, determines the cost function minimized by the estimate. This function may be, for example :

– $\sum_{(x,y)} \left[\text{Deform}_{\boldsymbol{\theta}} I(x,y,t) - I(x,y,t+1)\right]^2$, where $\text{Deform}_{\boldsymbol{\theta}} I(x,y,t)$, is the image deformed by the affine flow field $\boldsymbol{\theta}$. This choice corresponds to classical regression, and yields the usual SSD.

– $\sum_{(x,y)} \sigma \; \rho(\; \left[\text{Deform}_{\boldsymbol{\theta}} I(x,y,t) - I(x,y,t+1)\right] / \sigma \;)$, when one uses the so-called "Huber" regression estimator. $\rho(r) = a + r^2/2$ if $|r| < 1$, and $\rho(r) = |r| + a - 1/2$ otherwise; $a > 0$, and σ is an appropriately chosen robust equivalent of the variance.

– $\sum_{(x,y)} \sigma \; \rho(\; \left[\text{Deform}_{\boldsymbol{\theta}} I(x,y,t) - I(x,y,t+1)\right] / \sigma \;)$, where $\rho(r) = log(1+r^2)$, when one uses a robust estimator adapted to "Cauchy"-distributed noise. This is the robust estimator we used in the tests below.

[4] Two images I_1 and I_2 are said to be affine transformations one of another if there are parameters $(u_0, u_x, u_y, v_0, v_x, v_y)$ such that $I_1(x+u, y+v) = I_2(x,y)$, where u, v are defined as in Equation (3)

4 Performance analysis

An important choice is the criterion for the quality of an estimator : It must be meaningful for whoever plans to use the output of the estimator and yet it must facilitate theoretical analysis so as to provide guidelines to the designer of the estimator.

One possible choice is to take the variance as the quality criterion. Then, quality will be dependent on statistics of the input which are rarely known (in our case, the characteristics of the noise terms e_1, e_2 and e_3). Often, simulation substitutes the analytic study, making the the assumption that the synthetic data is "representative enough" of real-world situations.

Another possibility is to consider that our system is to be embedded in a larger system, whose performance is easier to evaluate (like a robotic system with a precisely defined goal). Then, the performance analysis can be done relative to this larger system, hence allowing the comparison of different estimators. However, the performance assessment is complicated (and costlier), since in order to test the estimator, one has to test a greater system. This method is usually reserved to final development tests.

Few options are left between these two extremes, and this is certainly a research issue deserving attention. For a general discussion on performance analysis in computer vision, see [4].

When choosing a criterion on the quality of an estimator, one usually starts by addressing the quality of an estimate. As often in computer vision, this choice is difficult because of the non-rigorous definition of the desired output, and many quality measures can be encountered in the literature.

As a measure of quality of an estimate, when ground truth is known, we use the squared norm of the error. The norm we used reflects the difference in magnitude of the components of the estimates. We used the empirical expectancy of this measure to assess the quality of an estimator [5]. This quality measure was partly extended in [3] to the case when no ground truth is available.

The computational cost of the implementation of an estimator is often a crucial consideration, for example in active vision. Thus, being able to characterize the performance in terms of computational cost (or the contrary) appears a central issue. Unfortunately, we are not aware of analytic results on that subject[6].

The concept of "computational cost" deserves some attention, as it has no universal definition. Complexity seems natural, but there are many variants (worst-case, mean). It may depend not only on the algorithm, but also on the machine running it (it may be parallelizable), and the way complexity translates into execution time is non-trivial. Our criterion is the "empirical mean cpu-time" for our machine. This has many disadvantages : Dependency on the efficiency of the code, and limited control.

By appropriately choosing the number of observations used by an estimator, we may specify the cpu-time of its implementation. A study of the algorithm allowed us to identify a model for the execution time, as a function of different parameters in the estimator. Based on this model, we can determine the number of observations needed to achieve the desired computation time, when all other parameters are fixed. This method is rather crude, and we believe new software tools would be welcome, along with a better adapted concept of complexity.

However, it allowed us to experimentally observe the effect of the tuning parameters on the performance, while keeping its computational cost fixed.

5 Results and Discussion

As discussed before, we can consider three families of estimators : "Isotropic", "Anisotropic" and "Anisotropic-M". The computation cost of all the compared estimators is the same. Within each family, the estimators differ by the number of iterations performed in the robust regression. Accordingly, an estimator that performs a larger number of iterations takes a smaller number of observations.

All the results in this section were obtained with synthetically deformed images (Figure 3 shows one of them), and therefore, the ground truth flow is known. We used two methods to generate sequences of flow parameters (and from these, the image sequences). We could either simulate the motion of a camera

[5] These two choices may be criticized : Once an estimate is all the way wrong, we do not care if it is ten times more so (we have a "minimax" error measure in mind). However, the error measure (squared norm) will keep on increasing quadratically. The empirical mean of errors is non-robust : A single wrong estimate yield a high error measure of an otherwise "good" estimator.

[6] An exception is when, in a regression problem, the noise is Gaussian; otherwise, we may have asymptotic results, which are less useful; or no result other than experimental evidence.

observing a tilted flat surface, or use randomly generated autoregressive time-series (six appropriately scaled time series specify the six flow parameters). The CPU time is fixed to 0.3 seconds, on a DEC ALPHA 3000-500, using the "cc" compiler, with debugging option and no optimization. The number of observations is shown in Table 1.

Method	Min	Max	$Min.error$
$Isotropic$	1203	1539	-
$Anisotropic - M$	997	1529	1424
$Anisotropic$	1127	1415	-

Table 1.: Number of observations (minimum and maximum) used for the various estimators. For the *Anisotropic* we also have the number of observations corresponding to the minimum error estimator.

Figure 4 plots the empirical expectancy of the squared error of the three families of estimators, "Isotropic" (solid line), "Anisotropic" (circles) and "Anisotropic-M" (pluses). The number of iterations of the estimator is in abscissa (not counting the initial estimate : 0 corresponds to the (initial) least-squares solution, 1 indicates that the robust estimator has been iterated once, etc).

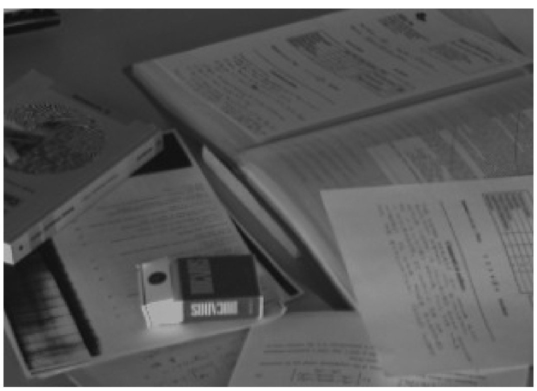


Fig. 3.: One of the used images, taken from a synthetically produced sequence. We used flow values corresponding to either the motion of a camera observing a tilted flat surface, or random (AR(1)) time-series of appropriate amplitude.

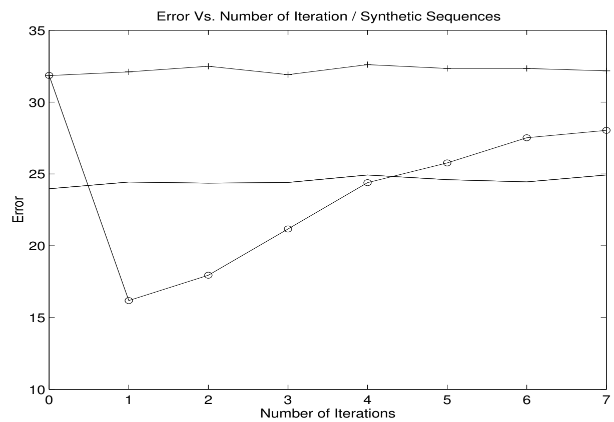


Fig. 4.: Empirical squared error of the estimator vs. number of iterations, for "Isotropic"(plain curve), "Anisotropic"(o) and "Anisotropic-M"(+) methods (All other conditions are equal).

Figure 4 show the effectiveness of the anisotropic approach when observations are renewed at each iteration of the estimator ("Anisotropic" curve). The flat curves of the "Anisotropic-M" and "Isotropic" seem to indicate that for a given computation cost, the performance of these methods does not vary with extra iterations. Other experiments with these estimators show that the performance stays nearly constant when one varies only the number of iterations (and the number of observations is fixed), and that performance also remains nearly constant when the number of observations is varied like here, between 1000 and 1500.

Differences in the implementations of the "Anisotropic" and "Isotropic" methods explain that the number of observations differs when the number of iterations is zero, i.e., when all the methods compute the same solution (namely, the least-squares, from which the robust estimators are started). The difference in the number of observations of the "anisotropic" and "anisotropic-M" estimators, for 0 iterations illustrates imprecision in the time-cost setting procedure.

The use of other sequences has led to different results, depending greatly on the amplitude of the optic flow; however, the general tendency is that of the above data, except that the performance of the "Anisotropic" method is often better than here.

Other choices concerning algorithms (.e.g. variants of robust regression algorithms) and parameters (e.g. smoothing coefficients) were settled by this kind of comparison.

Our approach has several limitations :

– The number of coefficients one has to tune (for example, we still have other ways of iterating a robust estimator, we should set filter widths, perform outlier rejection, etc...) makes the exploration of their effect on performance a lengthy process. In practice, it takes time and experimentation to optimize an estimator, for a given time-cost. An automated procedure would be welcome, but it would be very time-costly (computationally); unless new theoretic tools help us improve its efficiency.
– Being able to compare algorithms and parameter setups on specific data does not give us much insight on the reasons of their performance.

An interesting study of "performance versus time-cost" appeared recently in [8]. In our case, this is equivalent to "performance versus number of estimates" (when all other parameters are fixed). In turn, one can show theoretically [6] —and observe in practice— that the error is inversely proportional to the number of observations. Still more interesting would be a study of "performance versus time-cost" when for each given time-cost value, all the parameters are set to optimize performance (instead of being fixed). This could be an application for an automated tuning procedure.

Future work should aim at a better characterization input data, as it often guides the analytic study of the problem : For example, once one has noticed that the error term $h(u,v)$ in Equation (6), verifies the (c) assumption in Section 3.2, the "anisotropic" algorithm naturally follows. Similarly, a statistical characterization of the validity of the definition of optic the flow (Equation (1)) in the design of flow estimators.

References

1. J.L.Barron , D.J Fleet, S.S.Beauchemin, "Performance of Optical Flow Techniques" , Proc CVAP 1992 or Robotics and Perception Laboratory (Queen's University Kingston, Ontario, Canada , 1992
2. William T. Freeman and Edward H. Adelson, "Steerable filters for image analysis", TR 126, MIT, Media Laboratory, 1990.
3. Etienne Grossmann, José Santos-Victor, "Quality evaluation of optic flow estimators : A need for vision-based robotics", in "Advances in Robotics Research", World Scientific Publishing, 1996.
4. Seven short texts by R.M. Haralick; L.Cinque, C.Guerra and S.Levialdi; J.Weng and T.S.Huang; P.Meer; Y. Shirai; B.A.Draper and J.R. Beveridge : Dialogue on "Performance characterisation in computer vision", in CVGIP on Image Understanding v.60 n.2 pp.245-265, '94.
5. Berthold Klaus Paul Horn, "Robot vision", M.I.T. Press, 1985.
6. Peter J. Huber, "Robust statistics", John Wiley and sons 1981.
7. Jan J. Koenderinck and A.J. Doorn "Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer" Optica Ccta 22, pp 773-791, 1975.
8. Hongche Liu, Tsai-Hong Hong, Martin Herman and Rama Chellappa, "Accuracy vs. efficiency trade-offs in optical flow algorithms", Proc. of the 4th ECCV, vol. 2, pp. 174-183, Cambridge, UK, 1996.
9. H.C. Longuet-Higgins and K.Prazdny "the interpretation of a moving retinal image", proceedings of the royal society of London, B 208, pp 385-397, 1980
10. Shahriar Negahdaripour and Shinhak Lee, "Motion recovery using only first order optical flow information" , IJCV 9:3, pp 163-184 , 1992.
11. Muralidhara Subbarao "interpretation of visual motion: a computational approach", Morgan Kaufmann publishers,Inc, San Mateo, California

This article was processed using the TEX macro package with SIRS96 style