Performance evaluation of optical flow estimators: Assessment of a new Affine flow method.

Etienne Grossmann José Santos-Victor
Instituto Superior Técnico & Instituto de Sistemas e Robótica
Av. Rovisco Pais, Lisboa, PORTUGAL
{etienne,jasv}@isr.ist.utl.pt

Abstract

Over the years, computer vision researchers have developed a number of algorithms to solve a large number of problems. However, most of the existing algorithms are not characterized in terms of their performance, accuracy, cost, etc. Consequently, it is hardly ever possible to compare and choose between these various algorithms to tackle a specific problem.

One of the contributions of this paper is the introduction of a framework for evaluating the performance of optical flow estimators, which is based on classical estimation theory criteria, and on considerations about the computation cost. This framework is general, and may be applied to other estimation problems.

The optic flow is widely used in many vision systems. It is a vector velocity field defined on sequences of images. The *Affine Optic Flow* is formed by the optical flow together with its first-order derivatives with respect to image coordinates. As a second contribution, we present two new estimators for the affine flow. We justify theoreticaly their design with hypotheses concerning the input images, which we show to be empirically valid.

Finally, we use the performance analysis framework in order to compare the affine flow estimators with a more classical "differential" method.

1 Introduction

The optic flow is defined as "the apparent motion of brightness patterns" in a sequence of images. It has been given many mathematical definitions, and many methods have been devised to compute it [2].

Very often, the optical flow is used as the input to other vision algorithms. For example, when identified with the motion field (the [optic] projection of the 3D motion onto the image plane), it allows to compute structure and/or motion of the watched scene. The optic flow and its (first) spatial derivatives together form the "*affine flow*", which is an even richer source of information, as shown in [14, 13]. An important and

well-known aspect of the optic flow is that it is ill-defined : it is based on a model that is only *approximately* correct. Knowing the nature of the error in this model is important for estimation algorithms. However, this point is poorly documented in literature, and we attempt to gain some insight about it.

One contribution in this paper, is constituted by two new estimators of the affine flow. Their design may be justified theoretically when one makes some assumptions about the input images. Furthermore, we verify experimentally the validity of these assumptions.

Over the years, many methods have been developed in computer vision (e.g. edge detection, edge linking, feature matching, structure/motion-from-X, etc). These algorithms are usually based on some assumptions about the input data. Very often, it does not appear possible to verify rigorously that these assumptions hold. The good functioning of the proposed method is taken as evidence that they are "sufficiently" valid.

Our approach thus differs, since we present some (limited) numerical justification of our assumptions separately from the results given by our methods. The assertion of the good functioning of a method also is a critical point. It is often done graphically, by the display of results (e.g. of extracted edges, estimated flow). This gives a graphical demonstration of the functioning of the algorithm, and may provide some insight about it. Nevertheless, a need is also felt for more quantitative measures, that allow comparison of different methods, and/or quantitative modelling of the output of the system.

There is no canonical method for asserting the performance of implemented estimators; the second contribution of this paper lies in the definition of such a method, based on the two notions :

- "Accuracy" : Estimation theory provides tools to define the quality of the output of an algorithm. First, an error measure is defined and then its statistical properties are studied. Since analytical study of the performance appears very difficult, we rely on numerical measures. We discuss and justify the methods we used.

- "Cost" : The cost at which estimates are produced. This is crucial in computer vision e.g. in active vision, where there are time-constraints to be met. We discuss the notion of "cost", as it may have many interpretations.

Only recently have these two concepts been studied jointly [1], e.g in [9]. To our knowledge, there are very few results linking the "quality" of an estimator to the "computational

[1] "Experimental design", or "experience planning", goes in that direction, by considering the cost at which one does observations. However, it usually considers the cost of doing computations to be negligible, contrarily to what is the case in computer vision

cost" at which it is implemented. The main point in the methodology we adopted is that we *characterize estimators in terms of the "accuracy" obtainable for a given unit of computational resource.*

The analytical calculation of the performance is a difficult problem, and our approach relies on analysing the results obtained by applying our estimates to various data sets. This approach requires some sort of "ground-truth" to be known. A common problem in computer vision is that no ground-truth is available, for real-world data. For this reason, synthetic data is most often used. However, real-world and synthetic data being different, one may argue about the "significance" of this last approach. We show how substitutes to ground-truth can be used with real-world data (also with limited significance).

We identified two reasons, common to many computer vision problems, for the absence of ground-truth : The ill-definition of the estimated quantity (e.g. edge detection, optic flow), and the approximate nature of the observation model (e.g. image noise).

1.1 Structure of the paper

We briefly introduce the optic flow in Section 2. In Section 3, we present the two new estimators together with a classical one, which is used for comparison purposes.

Section 4 is devoted to the problem of performance evaluation of a given estimator, dealing with the error definition, performance characterization and cost analysis.

The results obtained are described in Section 5. First, we address the problem of verifying the validity of the various assumptions that we have previously formulated. Then, we present performance measures against various parameters, involving both synthetic and real-world data.

Finally, in Section 6 we draw some conclusions and establish further directions of research. Appendix A gives an introduction to the methods of robust regression, which are used at the core of both our estimators.

2 Optical Flow

The optic flow is defined in [7] as "the apparent motion of brightness patterns" in a sequence of images. It is a velocity vector field defined over sequences of images, assuming that the pixels in one image displaced by that field do not change brightness level. The definition may be given the following mathematical interpretation :

$$I(x+u, y+v, t+1) = I(x,y,t) \quad \text{for all } x, y, \text{ and } t \tag{1}$$

where $I(.,.,.)$ is the sequence of images, seen as a function, x and y are pixel coordinates, t is time, and (u,v) [2] is the optic flow.

When x and y are real-valued, one easily sees that this definition is ill-posed, as there generally is no unique vector (u,v) verifying equation (1), but rather, infinitely many or none. There are many ways [7] to define a unique vector field that verifies equation (1) : Some involve regularization techniques (smoothness constraints on the flow), others assume that the flow fits certain models (e.g. local constancy [1]) and there are still other methods [10, 11], which make use of higher order image derivatives.

2.1 Affine Flow

In this paper, we assume that the flow is an affine (vector valued) function (of the image plane coordinates) :

$$\begin{array}{rcl} u(x,y) & = & u_0 + (x-x_0)u_x + (y-y_0)u_y \\ v(x,y) & = & v_0 + (x-x_0)v_x + (y-y_0)v_y \end{array} \tag{2}$$

This is an approximation, since in general there is no affine vector field that verifies equation (1).

The estimators we present in this paper compute the optic flow and its derivatives (with respect to image coordinates) (u_x, u_y, v_x, v_y). Together, these values form the "*affine flow*", defined by the following set of parameters :

$$\Theta = [u_0, u_x, u_y, v_0, v_x, v_y]^T . \tag{3}$$

2.2 Error Modeling

The optic flow, and its affine approximation are not exact models, and a "noise" term has to be taken into account. Knowing the characteristics of this noise term is important, because they determine the choice of the numerical method used for estimating the flow. We temptatively identified the following sources of error in our model :

e_1 - Non exactness in equation (1). It can be violated in the presence of occlusion, shading and other optical effects.

[2] More rigor would require writing $(u(x,y,t), v(x,y,t))$.

e_2 - Use of discrete coordinates (x, y), and discrete grey-level values render unplausible definition (1).

e_3 - Measurement of image value $I(x, y)$ and of derivatives I_x, I_y and I_t are "noisy". Despite its fundamental aspect, and the great development of filtering techniques, this noise is poorly characterized.

e_4 - Non exactness in the affine approximation equation (2). It depends on the optical properties and dynamics of the scene. This term also lacks a good modeling.

Presently, the terms $e_1, ..., e_4$ lack a good statistical characterization, and thus many estimation methods, that require precise knowledge of the noise may not be used adequately.

In the algorithms presented here, we consider the noise terms combined into a unique term, whose characteristics were studied empirically, as described in Section 5.1.1.

3 Anisotropic Affine Flow Estimators

We present two novel estimators for the affine optical flow, that we denominate as **Anisotropic**. The methods are based on a directional differentiation scheme of the images *along* the direction of the current flow estimate. In the following sections we detail the models and obervations to be used by these estimators.

3.1 Discrete Model for the Optical Flow

We used the following equation to define the flow on discrete sequences of images :

$$I(x + u, y + v, t + 1) = I(x, y, t) + Noise \text{ for all } x, y \text{ and } t. \tag{4}$$

To some extent, this interpretation is similar to that of "region-based matching techniques" in two ways. In fact, it involves only image *values*, and not image derivatives. Secondly, in order to estimate the flow, we will assume that it follows locally a given model (affine in our case, but other models could be used, as in [1]). However, the implementation of our algorithm is more closely related to that of "differential techniques", which rely on the following - well known - interpretation of the definition of the optic flow [7]:

$$-I_t(x, y, t) = I_x(x, y, t)u + I_y(x, y, t)v + \nu \text{ for all } x, y \text{ and } t, \tag{5}$$

where I_x, I_y are the spatial derivatives of the image sequence, I_t is the time-derivative, and ν denotes a random noise term.

3.2 Observation Equation

The presented estimators compute the affine flow by solving a system of "observation equations". In this subsection, we will show how these equations are obtained.

We consider the first-order approximation of the image[3] :

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

where $h(.)$ denotes the approximation error of the first-order expansion. This equation is valid for arbitrary values of $(\tilde{u}, \tilde{v})$. A suitable choice of these values will be discussed later in the paper. Combining the definition (4) and the approximation (6), it yields:

$$I(x,y,t) + \nu = I(x+\tilde{u}, y+\tilde{v}, t+1) + [I_x, I_y]\,[u-\tilde{u}, v-\tilde{v}]^T + h(u-\tilde{u}, v-\tilde{v}) \quad (7)$$

which can be re-written as:

$$I(x,y,t) - I(x+\tilde{u}, y+\tilde{v}, t+1) - I_x\tilde{u} - I_y\tilde{v} + h(u-\tilde{u}, v-\tilde{v}) + \nu = I_x u + I_y v \quad (8)$$

If we (somewhat abusively) assume I_x and I_y to be known, then equation (8) is a linear observation equation, where the only unknown terms are ν and $h(u-\tilde{u}, v-\tilde{v})$. These terms constitute the random part of the observation model, and will be referred to as noise.

Notice that, for the special case where $\tilde{u} = \tilde{v} = 0$, our observation equation reduces to the one used by the so-called "differential" techniques (see equation (5)).

We now have the possibility of choosing the values of $\tilde{u}$ and $\tilde{v}$ in order to reduce the variance of the noise term. It can be shown that *under some realistic conditions, replacing* $(\tilde{u}, \tilde{v})$ *by an estimate* $(\hat{u}, \hat{v})$ *of* (u, v) *minimizes the variance of the random term* $h(u-\tilde{u}, v-\tilde{v})$. These assumptions are :

a_1 - The error in the estimate $(\hat{u}, \hat{v})$, the noise terms due to the optic flow model, and image measurements noise are all independent random variables. Although we have not verified experimentally these points, it is a plausible hipothesis.

a_2 - The functions $E|h(., v)|$[4] and $E|h(u, .)|$ have their minimum in 0 and are increasing on $[0, +\infty[$ and decreasing on $]-\infty, 0]$ (for all v and u). This is justified experimentally in Subsection 5.1.2.

[3] In equation (6), $I_x(x+\tilde{u}, y+\tilde{v}, t+1)$, is abbreviated I_x, and similarly for I_y.

[4] $E|h(u,v)|$ is the expectancy, taken over images I and pixels (x, y), of $|I(x,y) - I(x+u, y+v) - I_x u - I_y v|$

.

a_3 - The density of (u, v) is such that the conditional densities of u and v are unimodal (for any value of v and u), with modes $\hat{u}$ and $\hat{v}$. This is a convenient assumption which is not provided theoretically. We believe that weaker sufficient conditions may exist.

It can also be shown that, under the same assumptions, *the variance of the random term diminishes with that of the estimate of* (u, v). The estimate $(\hat{u}, \hat{v})$ is provided either by:

- An intermediate estimate (as explained in Subsection 3.3).
- The (possibly scaled) estimate of the flow in the previous frame.
- $(0, 0)$ at the beginning of the sequence

The direction $(\hat{u}, \hat{v})$ is "privileged" in some way when this value is given to $(\tilde{u}, \tilde{v})$, and we will call that choice the **"Anisotropic"** approach. On the opposite, using $(0, 0)$ will be referred to as an **"Isotropic"** approach.

When one assumes that the flow is affine, as expressed in equation (2), and rewriting equation (8), it yields:

$$-I(x+\hat{u}, y+\hat{v}, t+1) + I(x, y, t) + [I_x \; I_y] \begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} = [I_x \; xI_x \; yI_x \; I_y \; xI_y \; yI_y] \cdot \Theta, \qquad (9)$$

This is the *observation equation* that will be used to solve for the affine parameters Θ. All the terms in the left-hand side may be measured, and constitute the observation. The term $[\, I_x \; xI_x \; yI_x \; I_y \; xI_y \; yI_y \,]$ may also be computed (since we chose x and y, and assumed that I_x and I_y were known), and is called the "regression vector".

3.3 Solving for the Affine Flow Parameters

Since each observation (of values $I(x, y, t)$, $I(x+\tilde{u}, y+\tilde{v}, t+1)$, $I_x(x+\tilde{u}, y+\tilde{v}, t+1)$ and $I_y(x+\tilde{u}, y+\tilde{v}, t+1)$) yields *one* equation in *six* unknowns, at least six observations are needed to define Θ uniquely. In practice, we will use over-constrained systems, with 50 to 3000 observations.

A common method [8] to solve over-constrained systems of linear equations is robust regression, which we describe in appendix. For now, the important point is that this method is *iterative*. It yields a series of estimates $\hat{\Theta}^1$, $\hat{\Theta}^2$, ..., that eventually converges to the "correct" value. In our case, the number of iterations is fixed.

The two algorithms we present differ in the way of alternating observation-making (computing values for the left-hand side of equation (9)) and iterations of robust regression. In the first algorithm, called (plain-)**Anisotropic**, the observations are computed, and a classical robust regression follows. In the second method, the observations are re-computed between the iterations of robust regression. This approach is called **Anisotropic-M** (where the "M" stands for "Modified robust regression").

The "classical" **Isotropic** estimator, that will be used for comparison, is similar to the first algorithm, the difference residing solely in the choice of $(\tilde{u}, \tilde{v}) = (0, 0)$.

The "Anisotropic-M" can be shown to be a "Region-based" matching algorithm. The series of estimates $(\hat{\Theta}^k)_{k=1...}$ converge towards a value $\hat{\Theta}^0$ that "robustly" minimizes the sum of the differences terms $I(x, y, t) - I(x + \tilde{u}, y + \tilde{v}, t + 1)$ (for all (x, y) in which observations are drawn). Thus $\hat{\Theta}^0$ minimizes a robust distance between one image and the previous one, after warping by an affine transformation. This point is detailed in appendix A.

The different estimators, for the same number of observations and of iterations (and all other conditions equal), have different computational costs. The "Isotropic" is the most economical, because of its simplicity, while the "Anisotropic-M", is the costliest. Hence, the computational cost must be taken into account when comparing the performance of these various estimators.

4 Performance Evaluation of an Estimator

We have now to discuss the possible ways of asserting the quality of estimators. There are two important parts in the performance evaluation process :

- One part is entirely described in estimation theory, and proceeds in two steps :
  - Define an error measure for estimates (Section 4.1).
  - Use it in a performance measure for estimators. In practice, this measure is not always analytically tractable, but we may empirically estimate it from data with ground-truth. Moreover, we have somewhat extended the performance evaluation to data without ground-truth (Section 4.2).
- The second part is methodological, and reflects our considerations about computation cost. It consists in comparing performances only when they were obtained "at the same cost". This point may seem trivial, but is often overlooked. In order to

allow performance comparisons, estimators must offer convenient way of controlling their "computational cost". (Section 4.3).

4.1 Measure of Error of an Estimate

There are many possible ways to define the error of an affine flow estimate. We chose to measure the (vectorial) error by the difference between the true value and the estimate:

$$\Theta - \hat{\Theta}.$$

Then, we consider a (scalar) error, the squared norm of this vector, given by:

$$\|\Theta - \hat{\Theta}\|^2.$$

One may argue against this quadratic criterion. For example, once an estimate is all the way wrong, the user does not care if it is ten times worse[5]. Moreover, we know that minimizing a quadratic error criterion can lead to numerical problems (e.g. the well-known numerical instability of least square methods). However, this error criterion facilitates the analytical study of performance.

We must choose a norm on R^6.For example, the usual norm defined by:

$$\|\Theta - \hat{\Theta}\|^2 \triangleq (\hat{u}_0 - u_0)^2 + (\hat{u}_x - u_x)^2 + (\hat{u}_y - u_y)^2 + (\hat{v}_0 - v_0)^2 + (\hat{v}_x - v_x)^2 + (\hat{v}_y - v_y)^2$$

is not satisfactory. An error of e.g. 0.1 would be penalized in the same way if it were committed on u_0, which is typically in the range of $[-3, 3]$, as if it was committed on u_x, which ranges in $[-0.1, 0.1]$.

It is customary to use the norm defined by the (inverse of the) covariance matrix of the quantity one wants to study (which is then seen as a random variable). However, this matrix is not know a-priori. In that case, it can be approximated by the empirical covariance matrix obtained from a population of flow estimates computed on "representative" image sequences.

Since we use *estimates* instead of "true values", the obtained covariance matrix also depends on the characteristics of the estimator (that produced the estimates). We supposed no knowledge of those characteristics, and assumed that the error commited is negligible. However, we believe a more rigorous treatment of the question is feasible.

Each estimate used for estimating the covariance matrix was produced, not by a single run of the estimators, but by the mean of 15 estimates (each produced by a run). We

[5] Whereas the proposed quadratic criterion will penalize hundredfold that tenfold wronger estimate

used two real-world sequences to estimate the empiric covariance matrix, for a total of 40 images, although more images would improve the quality of the estimate. One can see that the approximation of the norm will improve with the number of images used.

The norm matrices obtained by this process are not diagonal. However, if we argue that in general, the different components (u_0, u_x, u_y, v_0, v_x, v_y) of an estimate have no reason to be correlated, one can neglect the off-diagonal terms, and use this diagonal matrix to compute the norm. In what follows, we will always use this norm in R^6.

4.2 Performance of an estimator

Estimation theory has many criteria to express the quality of an estimator. The first and most natural is *bias*, which we do not discuss here, since the estimators we consider are (theoretically) unbiased.

Then comes *covariance*, when it is defined. An estimate of a covariance matrix may give insight on the behavior of the estimator. However, we would like a performance criteria - like a scalar index - which allows the comparison of various estimators. We may consider taking the expected norm of the error, as a measure of quality:

$$\mathcal{E} = E(\|\hat{\theta} - \theta\|^2) \tag{10}$$

The norm is the one defined in Section 4.1. Analytical study of the above expression has not given useful results, for the following reasons :

- The optic flow problem is not, in general, a classical estimation problem, since *the true parameter* is not available, unless one uses synthetic data.
- A covariance matrix may be calculated analytically as suggested in [5]. However, it is the covariance matrix of the limiting value $\lim_{k\to\infty} \hat{\Theta}^k$. In our case we only perform *a few* iterations of the robust regression algorithm. Moreover, the expression involves the "noise" terms in the observation equation (8), which is poorly characterized.

However, when ground-truth data is available (e.g. when synthetically deformed images are used), we may estimate the error criterion defined in equation (10) by:

$$\hat{\mathcal{E}} \quad = \quad \frac{1}{N_e} \sum_{\hat{\theta}} \|\hat{\theta} - \theta\|^2, \tag{11}$$

This estimate is of course greatly dependent on the data used. In our case, we could observe that the performance varied depending on the image sequence.

Transposing this performance measure to the case where no-ground-truth is available is non-trivial. A natural way consists in replacing Θ in equation (11) by a highly reliable estimate Θ^* of the "true" parameter:

$$\mathcal{E}^* = \frac{1}{N_e} \sum_{\hat{\theta}} \|\hat{\theta} - \theta^*\|^2 \tag{12}$$

The significance of doing this kind of substitution greatly depends on the way Θ^* is obtained. In some cases, it is theoretically very well founded, e.g. estimating the variance of a distribution of unknown mean. In our case, we must remain cautious, since we cannot even assume that there exists a "true" affine flow.

Our choice was to compute off-line good estimates of the affine flow (relying on computationally demanding methods), which are then used as ground-truth. Each estimate was the average of 20 estimates produced by "anisotropic-M" estimators, with 2000 observations. We are currently studying the validity of evaluating performances in this way. We justify empirically our method in [3]. Furthermore, when comparing performance on sequences with known ground-truth, the criteria $\hat{\mathcal{E}}$ and $\mathcal{E}^*$ give compatible results.

A more technical approach could involve building confidence intervals for performance measures (for chosen confidence levels). We are currently working on the application of statistical tools to performance estimation.

4.3 Computational Cost

Often, there are time-constraints when using an estimator. For example, in active vision, the existence of a closed loop control frequency imposes a limited time for all the algorithm computations. Hence, computational cost is an important issue. We will characterize estimators in terms of the "accuracy" obtainable (as defined in Section 1) for a given unit of computational resource. Unfortunately, we are not aware of analytic results on that subject[6]. In statistics or mathemathics, computational cost of numerical procedures is usually assumed to be zero, and at most, the order of magnitude of the complexity is mentioned[7].

Definition of the Computational Cost : The concept of "computational cost" deserves some attention, as it may have various definitions. Specifying the complexity seems

[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.

[7] That is, one determines whether the algorithm has e.g. linear or quadratic or exponential complexity.

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).

The way complexity translates into execution time is non-trivial and is also dependent on the machine. To model this dependency analytically requires fine knowledge of the machine, as well as (for estimating mean execution time) precise knowledge of the data flow.

The "empirical mean cpu-time" (for our machine) was taken as criterion. This approach has the disadvantages of depending on the machine used, on the efficiency of the code, and limited control. However, since the different implementations we discuss here are very similar, we consider that the mean cpu-time is a fair measure to compare them.

Setting the Computational Cost : In order to compare implementations at a fixed cost, we must be able to set their cost at a chosen level, and the algorithm must offer a way of doing so.

For each estimator, we modeled the computation cost as function of the number of observations, and of the other tuning parameters. Thus, for a given setting of tuning parameters (not counting the number of observations), one may choose the number of observations to achieve a desired computational cost.

Experimentation confirmed that our model for cost was realistic, and that the computational cost was correctly controlled.

Measuring the Computational Cost : We estimated the parameters of the model using the few available functions under Unix. Since measuring execution time (when it is of the same order of magnitude as the precision of the clock), is not widely described, we must detail the procedure followed:
Time intervals were measured either by

- Counting the number of times a function may be called within a predefined number of timer ticks, or
- Counting the number of ticks before a predefined number of function calls were completed,

whichever occurred last. This, and a few more precautions, guarantees us a minimum level of precision in the measure.

Although no significant change in the estimated model parameters was observed when the load of the machine changed, we believe finer software tools would be welcome. Also, a better adapted concept of complexity requires further analysis.

5 Results

In this section we will present the various results obtained. In a first step we will focus on the validitation of the models and assumptions made throughout the paper. Then, we will study the performance of the various estimators against several parameters. As a side-result, we will show how this framework can be useful to tune some of the intrinsic parameters of an algorithm.

5.1 Validity of Assumptions

The main assumptions made were the affine nature of the optical flow field, together with the characteristics of the error in the affine image approximation. These assumptions will be looked into in the following sections.

5.1.1 Validity of the Affine Flow Model

We here study the validity of both the optic flow definition and of the assumption that the optic flow is affine. We estimate the distribution of the "*Noise*" term in equation (4),

$$Noise = I(x+u, y+v, t+1) - I(x, y, t)$$

when the flow is assumed to be affine over the whole image, that is, when :

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

We used two real-world sequences of 20 images (the first of which are shown in Figure 1) to build an histogram of the values of the "*Noise*" term in equation (4). Additionally, there is an extra "error" term due to the absence of ground-truth data. Obviously, for this study, synthetic sequences could not be used. We estimated the affine flow on these sequences as follows: Each value is the average of 20 estimates produced by "Anisotropic-M" estimators with 1.0 second of allocated cpu-time (which is a large value).

Replacing the unknown flow by the estimates, we could observe the sum of the "Noise" term and of another term due to error in the flow estimate. We took 34000 measures and put them in 500 bins. Since the optic flow (and thus its derivatives) is ill-defined, there is

no ground-truth data. The resulting histogram is shown in Figure 2. By using knowledge about the estimator and about image approximations, the above experiment could be refined, and the effect of the error in the flow estimates could be reduced.

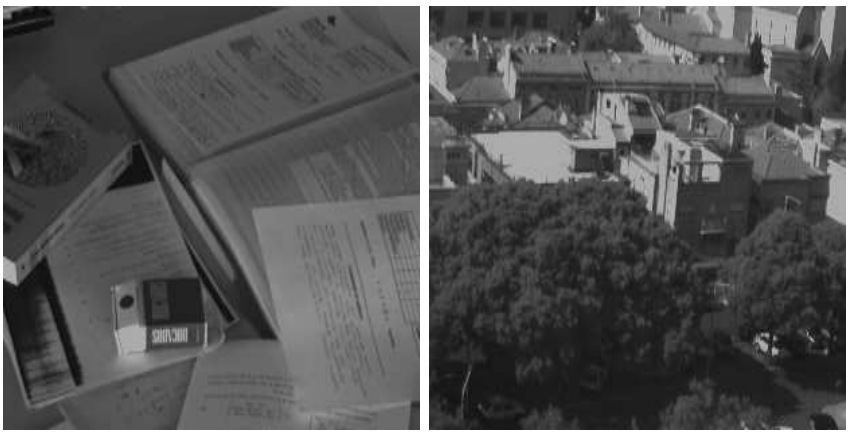


Figure 1: Typical images, used to estimate the noise distribution.

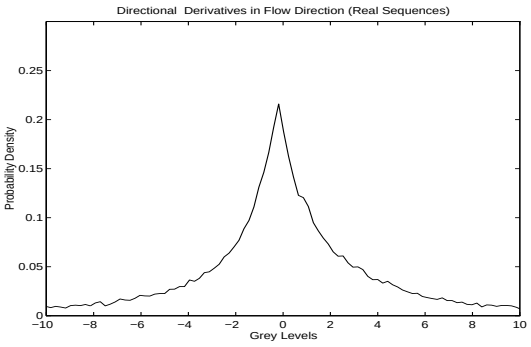


Figure 2: Histogram of the residues in equation (4), using the affine flow approximation, computed from real images. The unknown flow parameters are replaced by a good estimate; thus there is an extra error term. The tails are important : The variance is 39. This histogram was produced using 34000 measures and 500 bins.

The distribution in Figure 2 is clearly non-Gaussian and, moreover, it is long-tailed. The variance of the displayed data set is 39.

The Least Squares method is well known to be inefficient when the noise is non-Gaussian, and in that case, robust regression is usually preferred. Therefore, our choice of using robust regression rather than Least Squares appears justified.

5.1.2 Image Approximations

Here, we show that the assumptions made in Subsection 3 on the function h are justified. Recall that we have defined h as the error committed when doing a first-order

approximation of the image grey-level values:

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

A different "h" function is thus defined for every pixel in the image (and every image), and it is thus more rigorous to call it $h_{I,x,y}$. We want to show that the expectancy, over images and pixels, of the absolute value of h,

$$E(|h_{I,x,y}(a,b)|)$$

is an increasing function of $|a|$ (when b is fixed) and of $|b|$ (when a is fixed).

We have estimated $E(|h(a,b)|)$ from five images, which we consider "representative". On each one, we chose randomly 1000 points, and for each point we computed $|h_{I,x,y}(a,b)|$ for each value of (a,b) in $\{-19,..,19\}\times\{-19,..,19\}$. From these, we computed an estimate of the expectancy of $|h_{I,x,y}(a,b)|$ (one estimate for each value of (a,b)).

Figure 3 shows level contours of an empirical estimate of $E(|h(a,b)|)$. The minimum level is in the center, and increases towards the sides. Our assumption is only approximately correct because the level contours show that the minima (of the function that associates $E_{empirical}(|h(a,b)|)$ to a (resp. b), for a fixed b (resp. a)), are not exactly on the $a=0$ and $b=0$ axes, but rather, slightly shifted. We believe this is due to asymmetry in the images used, and that this feature would disappear if more images are used.

5.2 Factors influencing the performance

Now, we can study how different factors influence the performance of the three considered estimators ("Isotropic", "Anisotropic", and "Anisotropic-M"). All the figures below show the empirical mean error norm plotted against an influencing factor. Whenever synthetic flow was used, its value was obtained either by :

- A random process : At each time instant t, u_0^t is taken as: $u_0^t = b * u_0^{t-1} + \eta^t$, where η^t is a sequence of independent Gaussian random values, and $b \in [0.0, 0.8]$ (u_0^t is a first order auto-regressive process (AR(1)). A similar procedure was adopted for the remaining parameters u_x, u_y, v_0, v_x, v_y.

- Simulating the flow produced by a camera moving while watching a flat surface. The flow is then a second order polynomial in the image coordinates. The expected value of the estimators is the first-order part of the polynomial, which is used as ground-truth.

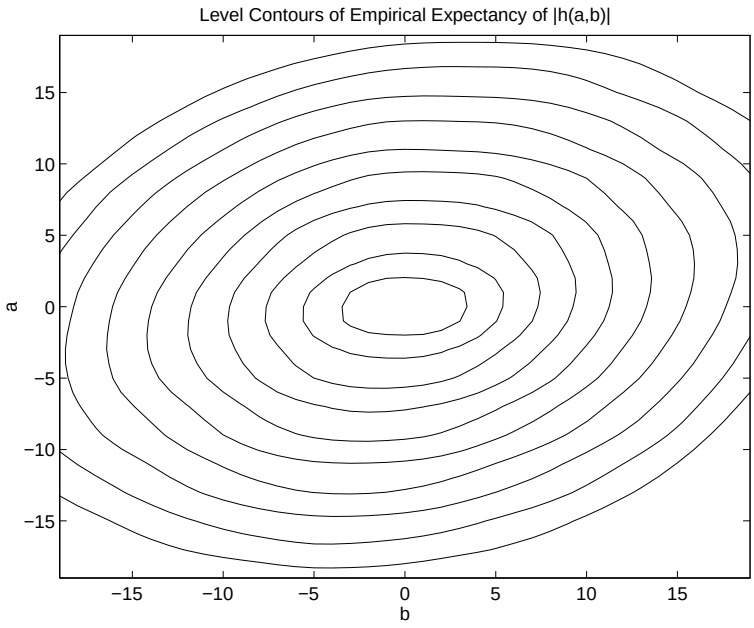


Figure 3: Contour levels of the empirical expectancy of the error committed when doing first-order approximation of images. The minimum is at the center, and the level increases towards the sides. This figure shows that our hypothesis is approximately valid.

We also specify wether each figure plots the performance obtained from a single sequence, or a result of averaging the estimates from many sequences.

5.2.1 Performance as a function of cost

Figure 4 shows that the empirical expected error norm decreases when the computational cost (expressed in seconds) is increased. This justifies comparing estimators that are given the same computational resource. This figure was obtained from one sequence of images, with synthetically generated optic flow. This graph is similar to that of error versus number of observations, since the cost is proportional to the number of observations made.

In theory, the error of robust regression is known to tend asymptotically to $1/N_o$, where N_o denotes the number of observations. The estimators tested have displayed a similar behavior.

5.2.2 Performance as a function of a time-smoothing parameter

The good functioning of many vision algorithms is dependent on the correct setting of some tuning parameters. However, one often has little insight about what a "good value" might be. A partial solution to the problem is to plot the performance as a function of

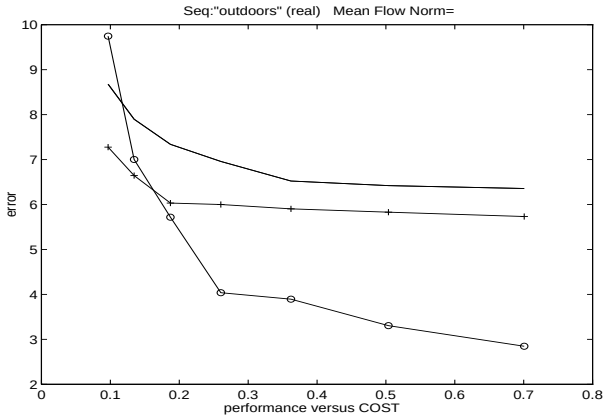


Figure 4: Empirical squared error of the estimator vs. computation cost, for "Isotropic"(plain curve), "Anisotropic"(+) and "Anisotropic-M"(o) methods (All other conditions are equal).

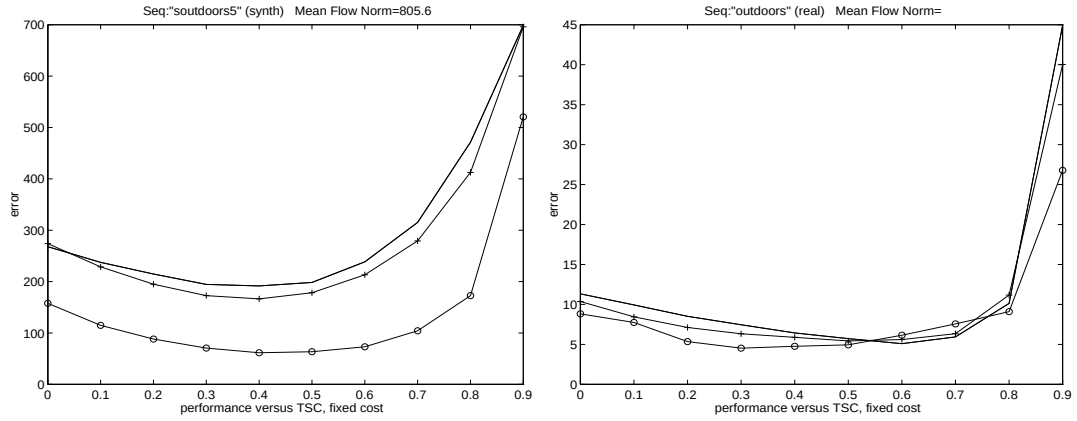
each given parameter. We illustrate here how this can be done.

Figure 5 plots the mean error norm as a function of a "time-smoothing coefficient". The images are time-smoothed by the following Infinite Impulse Response (IIR) filtering scheme :

$$I(t) = aI(t-1) + (1-a)I_0(t) \ ,$$

where $I(t)$ denotes the smoothed image, $I_0(t)$ is the original image (position indexes are omitted). We call a, the "time-smoothing coefficient".

The differences between Figure 5-(a) (from a synthetic sequence) , and Figure 5-(b) (real-world sequence) illustrate the differences between using synthetic and real-world data.



(a) Synthetic data (b) Real-world data

Figure 5: Empirical squared error of the estimator vs. time-smoothing coefficient, for "Isotropic"(plain curve), "Anisotropic"(+) and "Anisotropic-M"(o) methods.

All performance measures presented in Subsection 5.2 are done with a time-smoothing coefficient of 0.35; we set the other tuning parameters (e.g. spatial filter width, various robust regression parameters, etc) to "good" values by the same kind of investigation.

5.2.3 Performance as a function of the number of iterations

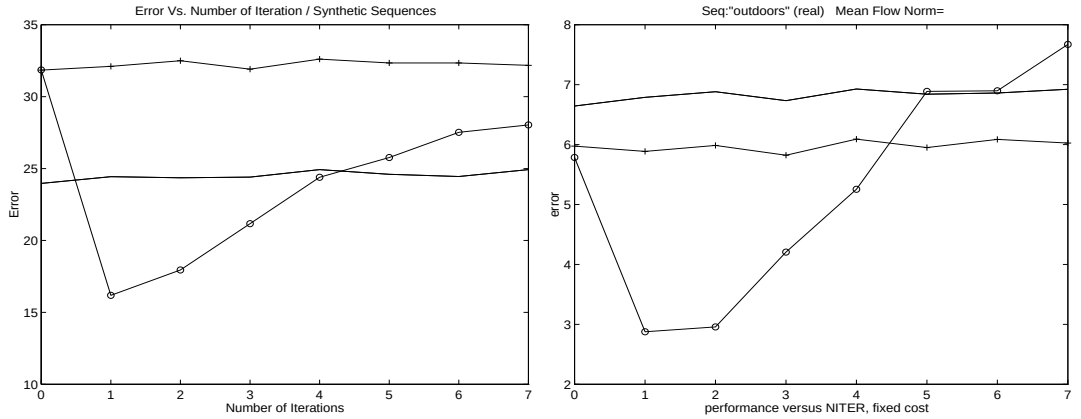
Some parameters in the implementation have an effect on both its performance, and its cost. Such parameters include the width of the image smoothing filters (when Finite Impulse Response filters are used), the number of observations used (which is used to set the computation cost), or the number of iterations of the robust regression. Here, we study the choice of this last parameter.

An important part of the computational cost of the flow estimator comes from the robust regression algorithm. The cost of regression is proportional to both the number of iterations and the number of observations. Thus, if one increases the number of iterations (which should increase the performance) and wishes the cost to remain constant, one must diminish the number of observations (which should decrease the performance). Choosing the number of iterations, in order to maximize the performance is tricky, since we have no theoretical result to help us.

Figure 6 plots the empirical expectancy of the squared error of the three families of estimators, "Isotropic" (solid line), "Anisotropic" (+) and "Anisotropic-M" (o). 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). The performance is the average of performances from multile sequences, when the flow takes random values (AR(1), with null auto-regression coefficient).

Figure 6 shows the effectiveness of the "Anisotropic-M" approach, on both synthetic (Figure 6 (a)) and real-world data (Figure 6 (b)) The flat curves of the "Anisotropic" and "Isotropic" seem to indicate that for a given computation cost, the performance of these methods does not vary with extra iterations.

The result greatly depends on the used sequences. However, the general tendency is that of the above data, except that the performance of the (simple) "Anisotropic" method is usually better than here.



(a) Synthetic data (b) Real-world data

Figure 6: Empirical squared error of the estimator vs. number of iterations, for "Isotropic"(plain curve), "Anisotropic"(+) and "Anisotropic-M"(o) methods.

5.2.4 Is Robust Regression Adapted ?

It is well known that perfectly good numerical methods may misbehave when run in conditions they were not meant for. On some sequences of images, we noticed that flow estimators behaved in an unexpected way. In the following experience, we compare estimators that differ only in the number of robust iterations. The number of observations is held fixed and the cost is therefore variable. One sequence was used, with large flow values (generated by simulated 3D motion of plausible amplitude).

One would expect the empirical error to decrease with the number of iterations. Figure 7 shows that the "Anisotropic-M" behaves according to this idea. However, the "Isotropic" and (simple)"Anisotropic" methods display constant error levels. One may ask if the "Anisotropic-M" method behaves "well" because of the iterations of the robust regression algorithm, or because of the successive amelioration of observations, that in turn allow better estimates.

The choice of robust regression was justified by the assumption that the image derivatives I_x and I_y were known *exactly*. This abusive assumption alone could explain the previous observations. To overcome this limitation, we could use a model for the observation of I_x, I_y and I, and perform maximum likelihood estimation.

Notice however that, in spite of these aspects, we consider the results obtained with the "Anisotropic-M" approach very encouraging.

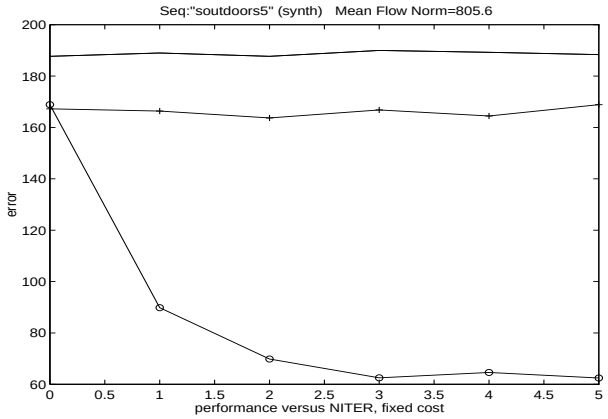


Figure 7: Empirical squared error of the estimator vs. number of iterations, for "Isotropic"(plain curve), "Anisotropic"(+) and "Anisotropic-M"(o) methods (All other conditions are equal). Number of observations is constant.

6 Conclusions

Many of the existing methods in compuer vision lack good characterization and performance evaluation. This lack is preventing the use of computer vision techniques in many indistrial problems. Furthermore, characterizing the performance of the various methods is the only way of providing a fair comparison and analysis between various algorithms.

In this paper, we address the problem of characterizing the performance of optical flow estimators, even though the framework can be extended to other problems. We use the estimation theory framework to define performance measures for various algorithms. First we consider the case where ground turth data is available. The methodology is then extended to deal with the absence of ground-truth data, which is most often the case, in computer vision. Another distinctive aspect is that we consider the cost of an estimator within the performance evaluation framework.

We present two novel algorithms - **Anisotropic** - to estimate the affine optical flow. These algorithms are then compared with a *classic* estimator, using the developed framework. The work described here is grounded in two main ideas:

Good characterization of the input : We have identified (roughly) the nature of the error in our model and made hypotheses concerning properties of the input images. Then, we verified these hypotheses and used them to justify our algorithm. The resulting algorithm compared favorably with the classical - simpler - approach.

Good characterization of the output : Special care must be taken when estimating the performance of the estimators. Since we cannot do so analytically, we proceed empirically.

An important point is that we compare performance measures "fairly", in the sense that the computational cost is also considered. We believe the simple tools we used can and should be complemented in future work. We are currently working in that direction, also hoping to automatize the optimization of algorithms.

We hope these two points will eventually "connect" in the sense that one will be, in the future, able to characterize the expected quality of the output from that of the input. Being able to assemble many vision sub-systems into a greater system of *known characteristics* will undoubtfully mark a big improvement in computer vision.

A Reminder about robust regression

Robust regression may be used to estimate an unknown (vectorial, of dimension p) parameter Θ, when one has the following observation model:

$$Y = X\Theta + \eta$$

where η_i is "noise", Y is an observation (i.e. a scalar measurement), and X is a known (line) vector describing the way the observation was taken. Drawing n observations yields n observation equations

$$Y_i = X_i\Theta + \eta_i \quad i = 1, .., n.$$

When n is equal to p, and the matrix X formed by the lines $(X_i)_{i=1..n}$ is invertible, we may estimate Θ by $X^{-1}Y$; This solution is easily seen to be highly sensitive to noise, and it is preferable to use extra observations (i.e. $n > p$).

Robust Regression Solution When an over-constrained system is considered, we must define the solution $\hat{\Theta}$ that we will compute. Usually, it will be an unbiased estimate of Θ.

Most often, knowledge about the nature of the noise term η can be used. For example, when it is assumed to be Gaussian and (all terms are) independent, and variance is the same for all η_i, the maximum likelihood solution is the least squares solution:

$$\hat{\Theta} = (X^T X)^{-1} X^T Y$$

that minimizes

$$\sum_{i=1..n} (Y_i - X_i\Theta)^2 ,$$

this last expression being the log-likelihood of the unknown parameter taking the value Θ when the observations are $(Y_i, X_i)_i$. Different assumptions on the nature of the noise yield different expressions for the log-likelihood.

However, knowledge about the noise term is often not available. Making wrong assumptions about it may cause unsatisfactory behavior of some maximum likelihood estimators. The goal of robust estimation is to yield satisfactory estimators, even when the noise term in imprecisely known.

A "robust solution" is defined as a minimum of

$$\sum_{i=1..n} \rho(\frac{Y_i - X_i\Theta}{\sigma}) ,$$

for some function ρ, (usually) convex, having a unique minimum in 0; or equivalently, as a zero of

$$\sum_i \psi(\frac{Y_i - X_i.\hat{\Theta}}{\sigma})X_i = 0 \in R^n ,$$

where $\rho' = \psi$. σ is a "scale" factor, a "robust" equivalent to a standard deviation.

This is an *implicit* definition of the solution $\hat{\Theta}$. Huber [8] suggests computing $\hat{\Theta}$ by the following iterations :

$$\hat{\Theta}^{k+1} = \hat{\Theta}^k + (X^T W^k X)^{-1} X^T W^k (Y - X.\hat{\Theta}^k)$$

where W is diagonal, and

$$W^k_{i,i} = \psi(\frac{Y_i - X_i.\Theta^k}{\sigma})\frac{\sigma}{Y_i - X_i.\Theta^k}$$

are the successive *weighs* given to the observations. The value of σ can also be computed iteratively [8]. Notice that when $\rho(x) = x^2/2$, we obtain the least squares solution in just one iteration.

For ψ, we have used either

$$\begin{aligned} \psi(x) &= x \text{ if } \quad |x| < 1 \\ &= |x|/x \quad \text{otherwise,} \end{aligned}$$

yielding a so-called "Huber-estimator", or

$$\psi(x) = \frac{2x}{\pi(x^2 + 1)}$$

with approximately equivalent results. Assympotically, the "Huber" robust estimator can be shown [8] to tend to a normal distribution, where the mean is the true value.

The "Anisotropic-M" algorithm presented in the paper, is not stricto-sensu a robust regression, since the observations are changed between iterations. However, intuition and heuristics seem to indicate that replacing observations by less noisy ones at each iteration still yields a valid estimate, and moreover, the variance seems to be lowered.

Like all iterative algorithms, one has to choose when to stop iterating. Iterating until some convergence criterion is met is commonly done. However, this also make unpredictable the number of iterations (computation cost). In our case we fixed the number of iterations, and thus the computation cost is (nearly) constant.

References

[1] J. Bergen, P. Anandan, K. Hanna and R. Hingorani. "Hierarchical Model-Based Motion Estimation". In Proc. of the 2nd European Conference on Computer Vision, pp. 237-252, Santa Margherite Ligure, Italy, 1992.

[2] J.L. Barron , D.J Fleet and S.S. Beauchemin. "Performance of Optical Flow Techniques". Proc CVAP 1992 or Robotics and Perception Laboratory (Queen's University Kingston, Ontario, Canada , 1992

[3] E. Grossmann and J. Santos-Victor. "Quality evaluation of optic flow estimators: A need for vision-based robotics". In "Advances in Robotics Research", World Scientific Publishing, 1996.

[4] R. 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] R. Haralick. "Propagating Covariance in Computer Vision". In Proc. of the Workshop on Performance Characteristics of Vision Algorithms, pp 1-2, April '96, Cambridge.

[6] B. Horn. *Robot Vision.* M.I.T. Press, 1985.

[7] B. Horn and B. Shunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[8] P. Huber. "Robust statistics". John Wiley and Sons 1981.

[9] H. Liu, T. Hong, M. Herman and R. Chellappa. "Accuracy vs. efficiency trade-offs in optical flow algorithms". In Proc. of the 4th European Conference on computer Vision, Vol. 2, pp. 174-183, Cambridge, UK, 1996.

[10] H. Nagel. Displacement vectors derived from second-order intensity variations in image sequence. *Computer Vision Graphics and Image Processing*, 21:85–117, 1983.

[11] H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on PAMI*, 8:565–593, 1986.

[12] Proc. of the Workshop on Performance Characteristics of Vision Algorithms, April '96, Cambridge.

[13] J. Santos-Victor and G. Sandini. Visual Behaviors for Docking. *Computer Vision and Image Understanding*, 1996.

[14] M. Subbarao. "Interpretation of visual motion: a computational approach". Morgan Kaufmann Publishers, Inc, San Mateo, California