
International Journal of Computer Vision 29(1), 59–77 (1998)

Robust Optic Flow Computation

ALIREZA BAB-HADIASHAR AND DAVID SUTER
Intelligent Robotics Research Centre, Department of Electrical and Computer Systems Engineering, Monash University, Clayton Vic. 3168, Australia
ali.bab-hadiashar@eng.monash.edu.au
david.suter@eng.monash.edu.au

Received January 10, 1996; Accepted September 18, 1997

Abstract. This paper formulates the optic flow problem as a set of over-determined simultaneous linear equations. It then introduces and studies two new robust optic flow methods. The first technique is based on using the Least Median of Squares (LMedS) to detect the outliers. Then, the inlier group is solved using the least square technique. The second method employs a new robust statistical method named the Least Median of Squares Orthogonal Distances (LMSOD) to identify the outliers and then uses total least squares to solve the optic flow problem. The performance of both methods are studied by experiments on synthetic and real image sequences. These methods outperform other published methods both in accuracy and robustness.

1. Introduction

Despite at least two decades of research, the best methods for the extraction of optical flow are relatively inaccurate and non-robust. By non-robust, we mean that the accuracy, in particular parts of the image, is often considerably worse than the general accuracy attainable over much of the rest of the image. The degradation in accuracy is due to a number of factors such as larger noise in that region and/or failure of the underlying image motion model.

This paper describes and compares two robust methods (methods which are not sensitive to minor perturbation of data) for computing optic flow.

For the first method, a highly robust estimator known as Least Median of Squares (LMedS) is modified and used to find an initial estimate. This initial estimate is then used to classify each equation into two groups: "inliers" and "outliers". Finally, the inlier group is solved using the least square (LS) technique. The resulting technique is known as Weighted Least Squares (WLS).

For the second method, a novel and effective solution is presented for the problem of finding a robust solution to a system of over-determined linear equations with all the data matrices containing both noise and outliers. This technique is called Weighted Total Least Squares (WTLS). The weights for this method are computed using a new robust statistical method named the Least Median of Squares Orthogonal Distances (LMSOD).

Unlike the total least squares (TLS) approach which is only robust to noise, the WTLS method is robust to both noise and outliers and can tolerate outlier contamination in up to 50% of equations in the system. The proposed weighting method is fast and the total computation remains inexpensive.

A major feature of both approaches is that we introduce a final *validation* procedure to detect remaining unreliable estimates.

To demonstrate the performance of the proposed algorithms, the well-known optic flow problem is solved and the results show that the proposed methods, despite being very simple and straightforward, are likely to outperform any other method appearing in the literature. It will also be shown how our validation procedure can detect regions where the flow cannot be reliably estimated (e.g., motion model failure).

The outline of the rest of this paper is as follows. In Section 2 a brief introduction to motion estimation methods is presented. Some of the methods of others, which aim to improve the robustness of optic flow calculation, are also reviewed in this section. The purpose is not to give a comprehensive survey (that is, beyond the scope of this paper) but to contrast the approaches developed here with that of the major competing methods. The motivation for our methods comes from comparing the relative advantages and disadvantages of some standard formulations for solving a set of over-determined linear equations. In particular, we derive the basis for our approaches by comparing ordinary and total least squares approaches with closest point and LMedS reformulations of the problem. Section 3 describes our new method: we adapt some methods for fast (approximate) LMedS in Linear Regression problems, so as to be able to perform outlier detection for WLS and WTLS problems. Section 4 presents results of experiments that verify the effectiveness of our proposed approach. Section 5 concludes the paper.

Since the key to our approach, is to solve an over-determined system of linear equations in a robust manner, we expect the approach will be more widely applicable to other areas (to other problems in computer vision and to other problems in unrelated areas that also reduce to the solution of an over-determined linear system). Thus, the present paper should be considered as the first demonstration of a basic robust scheme for solving linear equations, where we have chosen to apply the method to optic flow calculation. It is not claimed that the current implementation is the best one can do with our basic method. Some suggestions for improving the implementation, for optic flow calculations; as well as suggestion of other areas of application of our techniques, are contained in the conclusion to the paper (Section 5).

2. Optic Flow and Robustness

Differential optic flow techniques try to relate local changes in image intensity (expressed as spatial and temporal derivatives of the image brightness function) to the optic flow (Horn, 1986). Differential techniques usually perform faster than image matching or phase-based techniques and lead to a simple set of linear equations. We base our robust approach in this class of approaches and therefore, the basic idea behind this class of methods needs to be explained before it will be expanded upon.

2.1. Differential Based Optic Flow Methods

The fundamental assumption behind all the optic flow techniques is the fact that the brightness intensity function of a moving object remains approximately constant at least for short duration of time (Horn and Schunck, 1981). The differential technique for calculating the velocity in TV and video signals was formulated by Horn and Schunck (1981) based on the earlier works of Limb and Murphy (1975), Cafforio and Rocca (1976), and Fennema and Thompson (1979). They derived a linear constraint, in the space spanned by horizontal and vertical components of the velocity vectors, known as the Optic Flow Constraint (OFC):

$$\frac{\partial I}{\partial x}u_x + \frac{\partial I}{\partial y}u_y + \frac{\partial I}{\partial t} = 0. \tag{2.1}$$

This constraint relates the spatial ($\partial I/\partial x$ and $\partial I/\partial y$) and temporal ($\partial I/\partial t$) derivatives of the image brightness function at each point, to the optic flow (u_x and u_y), at that point. Since there is only one equation in two unknowns, this equation cannot be solved for both horizontal and vertical components of the optic flow without additional assumptions or information (the well-known aperture problem).

Put pictorially, (and in a way that we will make use of later), a single equation produced by the OFC only constrains the optic flow (u_x, u_y) to lie on a constraint line in 2D $u_x - u_y$ space: we need at least one other nonparallel constraint line to uniquely determine the flow. In other words, using just the information we have so far, the problem is ill-posed. Various alternative strategies to make the problem well-posed (regularise the problem) have been suggested. These include: minimise a functional derived from the OFC and a smoothness penalty term (Horn and Schunck, 1981), assume constant or affine variation in the optic flow (e.g., Burt et al., 1989, Bergen et al., 1991, 1992), and differentiate the OFC to obtain more than one constraint (e.g., Nagel, 1987).

Regardless of the strategy for overcoming the aperture problem, one usually arrives at a set of linear equations to solve for the optic flow at each point:

$$Av = d, \tag{2.2}$$

where v is a two-component vector, $v = (u_x, u_y)$, (the optic flow we wish to derive), A is an $p \times 2$ matrix (whose coefficients are the spatial derivatives of the image brightness function) and d is a p-component

vector (temporal derivatives of the image brightness function). The rank of A should be greater than 2 (otherwise, we still cannot solve for the two components of the flow). Note, although only two independent equations are needed to estimate the flow, it has to be recognised that the derivatives can only be approximated, thus any two independent equations will not necessarily give the best estimate of the true optic flow. So we must, in general, use more equations than we have unknowns. In other words, we inevitably arrive at a set of over-determined linear equations.

2.2. Robust Optic Flow Methods

"Robustness" implies that the method is less sensitive (compared with non-robust methods) to the influence of outliers in input data. Roughly speaking, a robust approach should not produce estimates that are wildly wrong. Nor should a robust approach avoid making erroneous estimates simply through discarding potentially useful information, just to avoid the possibility of the result being erroneous: that is, a method that is overly conservative and produces no estimate almost anywhere in an image, is not a robust method.

2.2.1. Rationale for Robust Methods. Although we concentrate on optic flow methods based upon the differential optic flow constraints, much of what we say about the need for robust approaches still applies to the other classes of methods. In particular, in some form or other, all methods require some form of the following two basic assumptions:

- **Intensity Coherence.** The image brightness of a point imaged in two successive images is (strictest assumption) constant or (weaker assumption) nearly constant.
- **Motion Coherence.** The motion of points nearby in an image is the same (strictest assumption) or slowly varying (weaker assumption).

One common form of the last assumption is to allow the image velocity to locally vary in some linear or affine fashion.

There are a number of reasons why particular methods of optic flow produce erroneous or inaccurate results. It is useful to categorise these sources according to:

- failure of the image/motion model
  - — failure of the brightness consistency (weak or strong forms)
  - — failure of the motion consistency (weak or strong forms)
- noise (e.g., sensor noise, poor approximation of derivatives in a differential based scheme).

We argue that this classification is useful because failure of the image model, as we define it, has some important practical distinguishing features. Firstly, it usually occurs in particular regions that are associated with geometric or physical properties of the scene. For example, a moving specular surface may destroy the accuracy of the brightness constraint, the boundary between two regions moving with different motion will destroy the motion consistency assumption. Secondly, this class of errors tends to lead to more severe errors, and, in some sense, errors that are less random than the errors from the other sources that we have lumped together as "noise". For this reason, we argue that the first class of errors can be detected by robust statistical methods to detect outliers. The second class of errors, if also severe can be similarly detected as outliers. On the other hand, if errors from this class are less severe, and cannot be considered to be outliers, they can perhaps be reduced in effect by simple processes such as least squares solution. In this way we can take maximum advantage of the particular features of some standard robust and non-robust reformulations of the optic flow problem, see Section 3.1.

Thus, the essential point is that, we see robust methods as ones that can avoid producing wildly wrong estimates for the optic flow, *particularly where the source of the likely breakdown in performance is a failure of the image motion consistency or image brightness consistency assumptions*. Such breakdowns commonly occur, with non-robust methods, around the edges between differently moving objects. An extreme case of the latter is the situation where we have "motion transparency" (for example, a reflection on a transparent surface moving differently to the material beyond that surface; or, where there are two populations of movement interspersed, such as a flock of birds against a background of clouds)—an example of which is shown in the next section.

2.2.2. An Example of Motion Transparency and Outliers. To demonstrate the effects of noise and of breakdowns in the underlying motion model, we take

Figure 1. Snapshot from Hamburg Taxi sequence with selected region.

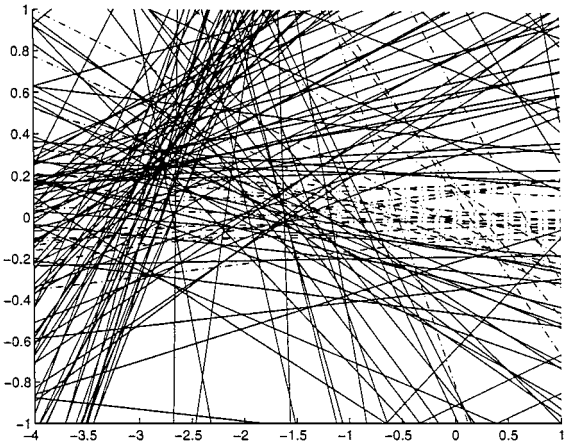

Figure 2. OFC lines for the Hamburg Taxi sequence (selected region). There are 169 lines, one for each pixel in the rectangle. The lower axis represents the horizontal velocity, u_x and the left-hand axis represents the vertical velocity, u_y.

a real image sequence: the famous Hamburg Taxi sequence (see Fig. 1). A van in the lower right corner is moving to the left at approximately 3 pixels/frame. We selected a small window, centred on this van (13 pixels square and centred at the point (220, 130)). For each pixel, we calculated the spatial and temporal derivatives of the image brightness to yield a single OFC for each pixel. We then plotted the OFC lines (see Fig. 2). If, in this small rectangle, we had a single underlying translational motion, and the data were noiseless, we would expect all constraint lines to pass through the point representing that motion, $(-3, 0)$ in this example. However, even though there is some tendency for many of the lines (shown solid) to cluster about this point, there are many lines (shown dashed) that nearly intersect at the point (0, 0). This is mainly due to the branches of the tree that intersect the rectangle from which we have drawn our pixels (the pixels in this rectangle, therefore, are a mixture of those belonging to the van, and those belonging to the tree: the former have approximately uniform motion to the left and the latter are stationary). This is a phenomenon sometimes referred to as "transparent motion" since it is similar to the situation where a reflection, in a pane of glass, for example, has different motion to other objects behind the glass. In addition to the breakdown in the motion consistency or coherence assumption (i.e., the motion model) we have other noise effects, due to a variety of other sources, that cause a spread in the intersections of these two populations.

The point we wish to make, with this example, is that a non-robust method that tries to treat all of the constraint lines as being valid data (for a single underlying motion), will produce poor estimates of the velocity. We wish to have a robust method that will reject the influence of any data that either belongs to another population (two or more independently moving object in a window), or is so badly corrupted by noise as to be unreliable. After briefly surveying other attempts to achieve this end (next section), we devise such robust methods (Section 3).

2.2.3. Previous Robust Approaches. The explicit use of robust statistics for recovering the visual motion dates back to early this decade. Darrel and Pentland (1991) considered using M-Estimators for 3D translations with constant depth. For 2D optic flow, Black and Anandan (1991, 1993, 1996) and Odobez and Bouthemy (1995) developed methods using M-Estimators for a correlation and differential formulations of the optic flow problem. The main difference between these two approaches is the use of different minimisation techniques. However, M-estimators have very low "breakdown points" (Hample et al., 1986)—the percentage of contaminated data that can cause the estimator to give an estimate far from the true estimate. In fact, the breakdown point of the proposed M-estimators is at most $1/(1+p)$ where p is the number of estimated parameters (Rousseeuw and Leroy, 1987). This shows that for an affine model used in (Odobez and Bouthemy, 1995), with 6 parameters, the breakdown point is only 14%. The first work was further modified by adding an Incremental Graduated Non-Convexity (Black, 1994) minimisation to the robust framework of Black and Anandan (1993). This technique can improve the motion estimates over time and adapts to scene changes. The most recent robust method for calculating the optic flow is presented by Ju et al. (1996). This method (known as Skin and Bones) can also be regarded as a modification to the robust

framework of Black and Anandan (1993). Although the performance of this method is exceptionally good for some cases (like the Yosemite image sequence), it still suffers from the limitations of M-estimators as mentioned earlier (see Stewart (1996) for detail discussion on the use of M-estimators for computer vision problems).

A similar approach for estimating the flow field, using the Horn and Schunck (1981) type regularisation technique has also been proposed by Iu (1995). In this method, the motion is estimated by minimising the energy function of a globally smooth motion model. A rank ordering technique is used to reject the local outliers of the globally smooth model and the resulting minimisation problem is solved using a stochastical optimisation technique (simulated annealing). Aside from computational burden of such a minimisation technique, the proposed kind of outlier rejection is likely to become very unstable for centrally located data. Since there is no measure on quantitative performance presented, it is difficult to assess the performance of this method.

In order to cope specifically with motion boundaries, Fennema and Thompson (1979) proposed a clustering method using the Hough transform. This method is computationally very expensive. Schunck (1989) modified this method, to reduce the computational cost, by clustering constraint lines along the OFC produced by the central pixel in a patch. However, such a method is sensitive to perturbations in the data that determined the central constraint line.

To overcome the problem rising from sensitivity to the perturbations in the central OFC, Nesi et al. (1995) proposed a clustering method based on the Combinatorial Hough Transform. This approach, however, still remains expensive. Moreover, the clustering methods just discussed, solve the problem as a version of the Closest Point Problem which unnecessarily discards a natural and useful weighting of the data (see Section 3.2 and Bab-Hadiashar, 1996).

Another similar approach is the method presented by Jepson and Black (1993, 1995). This method relies on using probabilistic mixture model to represent multiple motions. The EM-algorithm is used to retrieve the parameters of the various motions that exist in the scene. The advantage of this method over the clustering technique (Schunck, 1989) is that the later method models multiple motion and thus can retrieve several motions in a multiple motions scenario.

Another approach to optic flow computation using robust statistics is the use of Robust Hough Transform (Bober and Kittler, 1994a, 1994b). In this approach, an affine model of motion is introduced and the flow is calculated by Hough Transform. This Hough transform is based on the parameter space of the affine motion model, and "voting" uses the error between the brightness values of corresponding (using the affine mapping) pixels in sequential frames, weighted by a kernel function belonging to the M-estimators family. The Median of Absolute Deviation is used to scale the residual before applying the kernel function. Since a good match should lead to a low-error function (and thus a low vote in Hough transform space), the problem then reduces to an optimisation problem: finding the minimum in Hough space. The resulting optimisation problem is then solved using the "deepest descent" method. This algorithm has serious limitations due to the robustness limitation of M-estimators (see previous discussion of Black and Anandan's method). Moreover, for the optimisation scheme to work, the support function has to be a well-behaved function. As explained by the authors, the function is only well-behaved in the region with a valid Taylor expansion. So, in the regions with motion boundaries, transparency etc, the optimisation scheme is likely to fail. The poor results presented for the Yosemite Image sequence (Bober and Kittler, 1994b) confirms our intuition about the limitation of this algorithm.

The TLS method has been frequently employed to solve many different computer vision problems. Providing a comprehensive list of all these attempts is beyond the scope of this paper but a number of relevant works are briefly reviewed. Even though such methods can improve the results (basically by considering noise in both the coefficients and the right-hand side of equations such as (2.2)) these methods *are not robust methods* as the break down occurs even at one bad data sample. This deficiency will be addressed in Section 3.4.

Chu and Delp (1989) have suggested using TLS for solving the set of over-determined equations resulting from optic flow formulation. Their study addresses the rank deficient problem (where the data matrix A in the final over-determined set of linear equations is rank deficient) but such a approach fails to address the problem of having discontinuities in either the image brightness function or the optic flow itself (which commonly happens in any practical applications).

Chaudhuri and Chatterjee (1991) presented a performance analysis of the TLS method for 3D motion estimation. In this study, using synthetic data with additive uncorrelated Gaussian noise, they conclude that the

TLS outperforms LS method in deriving both the motion of deformable objects from range data, and the motion of a rigid object under perspective projection.

The TLS was also used by Wang et al. (1992) to recover the smooth flow where the chances of having outliers are limited. However, it is important to note, that assuming the flow is smooth is not sufficient to ensure there are no outliers in the final equations. Secondly, the assumption of smooth flow, over a predefined area, is too conservative to be useful in practical applications. Weber and Malik (1993, 1995) also presented a method for estimating the optic flow based on the TLS method. In this work, the authors allowed the outliers to corrupt the results when they solved the linear equations, but they rejected the final results based on some weighting scheme of the singular value decomposition (SVD) of the augmented data matrices (see Section 4.2, step 5). This approach appears to be too conservative. Indeed, this method under utilises the available information by allowing the outliers to contaminate the estimate in the first place and then attempts to reject the bad results when the damage is irreversible (see Section 5.2 for performance comparison).

The use of an iterative TLS method as a means of extracting straight lines was also proposed by Van Mieghem et al. (1995). This method, similar to those mentioned above, does not have the capability to reject the local outliers and would only work if there were no bad *leverage points* (points with their independent coordinates being far from the bulk of the data, Rousseeuw and Leroy, 1987).

Although Meer et al. (1991) uses LMedS in other problems drawn from the computer vision field, it seems that no previous work has used Least Median of Squares to solve the optic flow problem. Mitiche (1994) has mentioned the possibility of using LMedS, but appears not to have explored this suggestion with an algorithm or experimental results. We base our new methods, in part, on the LMedS technique.

3. Methods for Solving Linear Equations in the Presence of Noise

Before describing our approaches in detail, we need to introduce various abstract problem formulations: Ordinary and Total Least Squares, Closest Point and Standard Regression and Least Median of Squares. For the sake of clarity and simplicity, the 2D version of these problems are described here.

3.1. Ordinary and Total Least Squares Problem

Consider the problem of solving an over-determined set of linear equations $Av = d$, where $v = (x, y)$, A is the matrix with ijth entry a_{ij} and d is a (column) vector with row d_i. Suppose we have p linear equations in two unknowns (x and y):

$$a_{i1}x + a_{i2}y = d_i, \quad i = 1, \ldots, p \tag{3.1.1}$$

In case of LS, we seek the (x, y) that minimises:

$$E_{\mathrm{LS}} = \sum_{i=1}^{p} (a_{i1}x + a_{i2}y - d_i)^2 \tag{3.1.2}$$

This formulation has an explicit solution (simply obtained by differentiating E_{LS}, with respect to the unknowns, and equating these derivatives to zero):

$$v_{\mathrm{LS}} = (A^T A)^{-1} A^T d \tag{3.1.3}$$

The solution v_{LS} exactly satisfies the equation $Av = d + \Delta d_{\mathrm{LS}}$, where Δd_{LS} is a column vector. An attractive feature of the LS solution is that the LS produces an estimate, assuming that all the elements of matrix Δd_{LS} are uncorrelated random variables with equal variance, that has the smallest variance amongst the solutions that are linear in the data d *when there are no systematic errors* (VanHuffel and Vandewalle, 1991). The emphasis on the last qualifying phrase is placed there as this is the key weakness of the LS approach in optic flow calculations; particularly in regions, such as occluding boundaries, where the underlying model is no longer valid. This observation is one of the foundations of our proposed robust methods.

In case of TLS, we seek the (x, y) that minimises (sum of square orthogonal distances):

$$E_{\mathrm{TLS}} = \sum_{i=1}^{p} \frac{(a_{i1}x + a_{i2}y - d_i)^2}{(a_{i1} + a_{i2})^2} \tag{3.1.4}$$

This formulation also has an explicit solution:

$$v_{\mathrm{TLS}} = (A^T A - \kappa^2 I)^{-1} A^T d \tag{3.1.5}$$

where κ is the smallest singular value of the augmented matrix $[A \quad d]$ (VanHuffel and Vandewalle, 1991). The solution v_{TLS} exactly satisfies the equation $[A + \Delta A_{\mathrm{TLS}}]v = d + \Delta d_{\mathrm{TLS}}$ where ΔA_{TLS} and Δd_{TLS} are column vectors.

It should be noted here that the TLS is often preferred over LS because TLS solution, unlike the LS solution which is only consistent where the observation matrix d is error free, remains consistent even when all the data matrices are noisy. Being consistent means that the estimated solution converges to the true solution as the number of equations tends to infinity (based on the assumption that all the elements of all the matrices starting with Δ are uncorrelated random variables with equal variance). It is also important to note that this property of the TLS solution does not depend on any assumed distributional form of the errors (VanHuffel and Vandewalle, 1991).

However, the TLS, as well as the LS problem, is extremely sensitive to the influence of any outliers. Indeed, its breakdown point is only 0%. This means that even one contaminated element, in either of the data matrices, can result in a solution arbitrary far from any solution consistent with the rest of equations.

3.2. The Standard Regression (SR) and the Closest Point (CP) Problem

Much of the robust statistics literature is concentrated on finding a robust solution for the SR problem. As it will be explained in this section, there is subtle difference between the LS based solution to the SR (or CP) problems and the LS based solution to a system of over-determined linear equations. In order to clarify the differences between our robust approaches and the ones proposed in robust statistics literature (e.g., Rousseeuw and Leroy, 1987), it is essential to reiterate the SR and CP problems.

In case of the SR, we have a number of data points, to which we wish to fit a line. That is, given a set of p points: $\{(x_i, y_i), i = 1, \ldots, p\}$, we wish to find the line, parameterised by (m, n), i.e., $y = mx + n$, such that the sum of squared residuals (squared vertical distances to the line from the points) is minimised. That is, to find (m, n) that minimises

$$E_{\mathrm{SR}} = \sum_{i=1}^{p} (mx_i + n - y_i)^2. \tag{3.2.1}$$

In case of CP, given a set of p lines: $y = m_i x + n_i$,; $i = 1, \ldots, p$, the problem is to find the point that has the minimum sum of squared vertical distances to the lines. That is, to find the (x, y) that minimises:

$$E_{\mathrm{CP}} = \sum_{i=1}^{p} (m_i x + n_i - y)^2. \tag{3.2.2}$$

Although, there is a one-to-one relationship between the formulation of the CP and SR problem, neither of those has one-to-one relationship to the formulation of the LS problem. Comparing Eqs. (3.2.2) (or (3.2.1)) with (3.1.2) reveals the difference between these approaches. The main distinction arises from the fact that, even though every equation in the LS formulation can be regarded as a line, the residual associated with each line in the LS formulation will be weighted differently in the final minimisation problem based on the absolute magnitude of the coefficients of that equation. This is because, in order to map a formulation such as (3.1.2) onto (3.2.1) (or (3.2.2)) one has to divide the residual by a constant line dependent factor (to make the coefficient of one unknown, n in (3.2.1) and y in (3.2.2), have unit magnitude).

This distinction is important since our equations often carry a natural scale. For example, the OFC, Eq. (2.1), produces coefficients that are scaled by the gradient of the image brightness. This has the natural and useful consequence that those constraints resulting from parts of the image with high contrast will carry more weight, in a least squares formulation (an observation commonly made in the literature—see, for example, Shi and Tomasi (1994) or Jähne (1993), p. 122). To arbitrarily rescale such natural weights, as one would have to do to reformulate the problem as an instance of the CP (or SR) problem, is likely to be a retrograde step (see Bab-Hadiashar (1996), for details).

3.3. LMedS and WLS

The LMedS problem (Rousseeuw, 1984; Rousseeuw and Leroy, 1987) is a reformulation of the standard regression problem. Instead of finding the line that has the smallest squared vertical distances from the data points, the LMedS approach identifies the narrowest strip (bounded by two parallel lines) that contains one more than half of the data points: the LMedS line then runs down the middle of this strip (Steele and Steiger, 1986). Mathematically, the LMedS is given by:

$$\underset{\theta}{\text{Minimize}} \underset{i}{\text{med}}\ r_i^2 \tag{3.3.1}$$

where r is the vertical distance from every point to the fitted line and θ is the set of estimated parameters (Rousseeuw and Leroy, 1987). It is usual, for efficiency, to only approximately solve the LMedS problem and use this approximate solution in defining outliers. The WLS final solution is then obtained by weighted least squares.

The breakdown point of the LMedS is 50%—it can tolerate up to half of the data points being arbitrarily bad! As long as the majority ($p/2+1$, in our examples) are "sensible", in some sense, the solution will be "sensible". As we described before, our preferred formulation for the optic flow problem is a LS or a TLS formulation. The LMedS method is a robust solution to the SR problem: therefore, we have reformulated the LMedS and the WLS to apply to the LS and TLS problem. In the sequel, when we refer to the LMedS approach, we refer to the approach as modified for our purposes.

3.4. LMSOD and WTLS

In this section, the Weighted Total Least Square (WTLS) method is introduced for solving an over-determined set of linear equations $Av \approx d$ where the data matrices A and d contain both noise and outliers without A being rank deficient. The proposed method differs from the WLS method in two ways. Firstly, unlike the WLS, the outliers (of Eq. (2.2)) in this method are detected using a method we call the Least Median of Square Orthogonal Distances (LMSOD) technique. The LMSOD seeks an approximate solution v which exactly satisfies the $(A+\Delta A)\ v = (d+\Delta d)$ while minimising the median of squares of the orthogonal distances between solution v and the geometrical entity represented by every equation in the original set. The second major difference is that the WTLS method, unlike the WLS which uses ordinary least squares to solve the inlier group, uses the total least squares technique to solve the remaining system of over-determined linear equations (after rejecting the outliers, see Section 4.3).

It is important to note here that this method has all the advantages of the TLS method without being sensitive to the influence of outliers. Comparing the LMSOD to the LMedS, it is trivial to show that the LMSOD method also has a breakdown point of 50%.

3.5. Summary and Comparison of the Formulations

The CP and SR problems are essentially dual formulations of the same problem. The LS formulation is related to the CP and SR formulations but not an identical reformulation. The LS and TLS are all fast to compute; however, they are all non-robust. The LMedS and LMSOD are very robust; but are expensive to compute. Indeed, no closed form solution of the LMedS (or LMSOD) formulation is known. The fastest known method for computing the exact LMedS solution in 2D space has $O(p^2)$ time and $O(p)$ space complexity (Edelsbrunner and Souvaine, 1990).

4. Proposed Optic Flow Methods

We now have all the ingredients (and background rationale) for constructing our methods. Given a set of linear OFC equations (Eq. (2.1)), we would, ideally have a single unique solution. However, in reality, we are faced not only with a situation in which most lines do not go through the real solution point, but that there are severe outlier lines. To overcome the problem of having both noise and outliers, we propose and investigate two different optic flow methods. The first approach is based on LMedS and utilises the WLS for computing the optic flow. The second method is based on using the LMSOD for outlier rejection and utilises the WTLS for computing the optic flow field.

To these basic strategies we need to add one further ingredient. The whole basis of the LMedS and LMSOD methods is that there must be a population that is in the majority. This can, of course, breakdown (if we have, say, three populations roughly equal in size). Moreover, since we are using approximate LMedS and LMSOD solutions, and because there may be other noise (perhaps there simply is not enough texture in the region to give any reasonable constraints); we need to be able to validate our final answer (and reject estimates that are clearly erroneous). We shall propose a measure of reliability (Section 4.4) for this purpose. Thus our scheme requires just two thresholds: a threshold for outlier detection (Section 4.3) before applying WLS or WTLS, and a threshold for reliability to discard any final answers, after WLS or WTLS, that are still unlikely to be accurate (Section 4.4). The precise methods for choosing these thresholds are not essential to the proposed scheme and their details are given in Sections 4.3 and 4.4. One of these threshold is always set (Section 4.3) to a fixed value, determined experimentally (Rousseeuw and Leroy, 1987); and the other is set by a user-defined value.

4.1. Ordinary Least Squares Based Optic Flow Technique (WLS)

For purposes of clarity, we list the steps of our WLS based algorithm here:

1. Estimate the spatio-temporal derivatives of the image brightness function. The precise form of estimate, whilst important for accuracy, is not essential to our proposal. We choose to use, for our experiments, convolutions with derivatives of Gaussian functions (as is customary in many approaches, see, for example, Nagel (1995)).
2. Select a patch of the image, over which we are going to assume some motion consistency. The precise form of the motion consistency is not essential: we are simply assuming a single or dominant population (we only recover the dominant population if there is more than one—our method can be elaborated to remove the dominant population and resolve for any secondary populations).
3. Use an approximate LMedS solution to obtain a temporary estimate of the motion. Again, the precise method to approximate the LMedS solution can vary. We use here an algorithm adapted from Rousseeuw and Leroy (1987). Whereas that method was defined for a SR problem, it is modified to suit the LS problem. One simply chooses some fraction of the constraint equations, contain n equations where n is the number of rows in the solution matrix v of Eq. (2.2), and for each set, calculates the exact solution. For this solution one can calculate the square residuals for all other geometrical entities (lines, planes, hyperplanes, etc.) represented by the rest of the equations and, in turn, calculate the median of these square residuals. The solution with the smallest median of square residual is chosen as the approximate LMedS solution. In a naive approach to approximating the LMedS solution, one would try to use every possible combination of all the equations (lines, planes, etc.—depending upon the motion model dimensions—a combinatorially explosive situation) and evaluate the median of the residuals produced by the intersection of each combination. However, one can use a very small fraction of the possible combinations and the probability is high that the subsample will produce a solution that belongs to the majority population (using approximate LMedS for outlier detection, we need only one such good intersection). Indeed, if one chooses m equations of n unknown, from the p constraints we have in each patch, then the probability of this sample giving a good estimate for the LMedS solution is (Rousseeuw and Leroy, 1987):

$$1 - \left(1 - (1 - \varepsilon)^n\right)^m \qquad (4.1.1)$$

where ε is the fraction of samples that do not belong to the majority population. From this formula, it can be seen that one can choose a very small population m, and still such that the probability is close to 1. We often use m as small as 30.
4. Reject outliers using a method of outlier rejection, based upon the temporary estimate of motion. Details of this step is given in Section 4.3.
5. Solve the WLS problem resulting from the previous step. In our experiments, since we use weights 0 (reject) or 1 (accept), this is simply a matter of removing the rejected equations from Eq. (2.2) and solving by LS, the smaller system, according to Eq. (3.1.3).
6. Examine the result, using a measure of reliability and do not produce any estimate if the result is judged to be still unreliable (see Section 4.4 for details).

In our experiments, we repeat the above process for a patch centred upon every pixel, to yield the estimated of the flow at that pixel.

4.2. Total Least Squares Based Optic Flow Technique (WTLS)

For the sake of clarity, the proposed algorithm is described in four steps (the first two steps are exactly the same as the WLS algorithm and will not be repeated, here):

3. Use a fast and very robust approximate LMSOD solution to obtain a temporary estimate of the solution v (free from the influence of any existing outlier). Here, a method to approximate the LMSOD is proposed in a manner similar to Rousseeuw and Leroy's (1987) approximate LMedS. The method starts by randomly choosing a number of sample equations (the sample must satisfy the same condition as explained in previous section). By solving every such set of equations and finding the median of the squared orthogonal distances between this solution and the geometrical entities represented by the rest of the equations in the original set, one can find the solution which approximately satisfies the LMSOD. Similar to the LMedS case, one needs to choose only one sample belonging to the majority in order to return the approximate solution associated with the majority (see step 3 of Section 4.1). Therefore, limited number of samples is often required.

4. Having found the approximate LMSOD, one can weight the different equations based on the scaled orthogonal distances between the LMSOD solution and the geometrical entity represented by every equation in the original system of equations. The details of the weighting scheme is described in Section 4.3. After identifying the outliers, the outlier equations are eliminated (weight them by zero) to arrive at a new system of over-determined linear equations $A_s\ v = d_s$ in which the number of equations are now less than or equal the original set (subscript s refers to the inlier group).
5. The final solution can be obtained by solving the new system of over-determined linear equations using total least squares technique (Section 3.1):

$$v = \left(A_s^T A_s - \kappa^2 I\right)^{-1} A_s^T\, d_s \tag{4.2.1}$$

   where κ is the smallest singular value of the augmented matrix $[A_s \quad d_s]$ and I is the identity matrix. There are cheaper ways to calculate the total least squares solution than calculating the solution from above formula but their description is beyond the scope of this paper (see VanHuffel and Vandewalle (1991) for more details).
6. The last step in this algorithm is to examine the result, using a measure of reliability and not producing any estimate if the result is judged to be still unreliable (see Section 4.4 for details).

Similar to the first algorithm, the above process is repeated for a patch centred upon every pixel, to yield the estimate of the flow at that pixel.

As stated by Weber and Malik (1995), the TLS is very sensitive to non-iid (independently and identically distributed) noise. In fact, the TLS may perform worse than LS if the noise in different parameters of the system are not uncorrelated. To demonstrate this effect and the performance of our proposed estimator, a simple though demanding example is solved using our method. In this experiment, a set of 81 linear equations is considered: where a narrow majority of them (41 lines) satisfies the solution (2, 3) and the rest are consistent with the solution $(-1, -2)$. Then, we add 10% normal random noise to all the elements of the data matrices associated with the majority (parameters of all the lines passing through (2, 3) are contaminated with normal noise). Figure 3 shows the resulting equations plotted in Cartesian coordinates. The black arrow shows the WTLS solution (1.825, 2.811) and the white arrow points to the LS solution (0.110, 0.848). The TLS solution to this set of equations $(-73.045, 66.238)$ is out of range and has not been included in Fig. 3. This indeed demonstrate the high sensitivity of the TLS solution to non-iid noise and outliers.

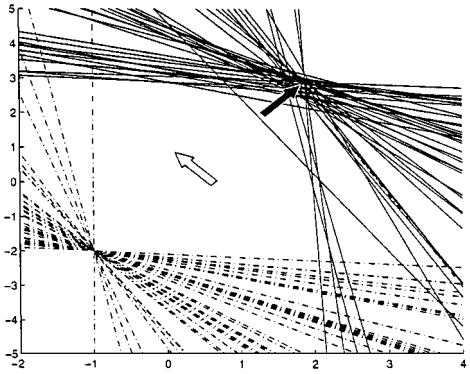

Figure 3. Robust total least square example. The black arrow shows the WTLS solution (1.825, 2.811) and the white arrow points to the LS solution (0.110, 0.848).

4.3. Outlier Threshold

In our method, having obtained an approximate solution, based on an approximate LMedS or LMSOD; we wish to assess the reliability of each constraint equation.

Rousseeuw and Leroy (1987) give a good recipe for detecting outliers. We first calculate, for each constraint a residual r_i, then we calculate a scale factor s^0 according to:

$$s^0 = 1.4826\left(1 + \frac{5}{p-n}\right)\sqrt{\underset{i}{\text{med}}\, r_i^2}. \tag{4.3.1}$$

We then, for every constraint, associate a binary weight (w_i) so that the weight is 0 for any constraint whose residual r_i is such that $|r_i/s^0|$ is greater than 2.5. Rather then using these weights, to directly reformulate the problem now as a (weighted) LS or TLS problem, we go through one more step of scaling. This is because the original weights were chosen, according to Eq. (4.3.1), using the median involving the outliers. Since we now have a better idea of which are truly outliers, we calculate:

$$s^* = \sqrt{\frac{\sum_{i=1}^{p} w_i r_i^2}{\sum_{i=1}^{p} w_i - n}} \tag{4.3.2}$$

and we, finally, reject those constraints for which the associated value $|r_i/s^*|$ is greater than 2.5 (see Rousseeuw and Leroy (1987) for details).

4.4. Measure of Reliability

Although the LMedS and LMSOD techniques have the highest possible breakdown point (50%) of all known robust estimators, it has the potentially fatal flaw in that is still produces an estimate, even if the number of outliers is more than 50%. Moreover, there are extreme cases where an image patch may not contain sufficient data (lack of texture) or data so badly corrupted (aliasing, for example) for any estimate to be valid.

Thus, we still need to validate the estimate produced by our method. A tool for the validation process can be modelled on "the coefficient of determination" (Kvalseth, 1985). The coefficient of determination, denoted R^2, has been defined for the Standard regression problem in at least nine different ways. The most well known of all the definitions is:

$$R^2 = 1 - \frac{\sum_{i=1}^{p}(y_i - \hat{y})^2}{\sum_{i=1}^{p}(y_i - \bar{y})^2} \qquad (4.4.1)$$

where $\hat{y}$ is the estimate of y provided by the regression fit and $\bar{y}$ is the mean of all of the data points y_i.

For the WLS technique, we have found the following statistic, inspired by the form of the above measure, experimentally satisfactory:

$$R^2_{\text{WLS}} = 1 - \frac{\sum_{i=1}^{p} w_i(d_i - \hat{d}_i)^2}{\sum_{i=1}^{p} w_i(d_i - \bar{d}_i)^2} \qquad (4.4.2)$$

where $\bar{d}_i$ is the average value of the d_i and $\hat{d}_i$ is such that $a_{i1}\hat{u}_x + a_{i2}\hat{u}_y = \hat{d}_i$ for the estimates of flow $(\hat{u}_x, \hat{u}_y)$ from our WLS technique.

However, for the WTLS technique, we want to ensure that the Frobenius[1] norm of the perturbation matrix $\Delta = [\Delta A_s \quad \Delta d_s]$ is small enough for the solution to be acceptable. Since it has been shown that κ (the smallest singular value of the augmented matrix $[A_s \quad d_s]$) is equal to the Frobenius norm of the perturbation matrix Δ for the calculated v (VanHuffel and Vandewalle, 1991), we propose the following R^2 statistic:

$$R^2_{\text{WTLS}} = 1 - \frac{\kappa^2}{\sum_i (d_{si} - \bar{d}_{si})^2} \qquad (4.4.3)$$

where d_{si} represents the different elements of vector d_s and maximum number of i is set by the number equations regarded as inliers.

5. Experimental Validation

The quantitative performance of the proposed algorithms have been measured by applying the algorithms to image sequences for which the true flow fields are known. We provide quantitative performance figures for both synthetic and real image sequences. The synthetic image sequences contain a controlled number of the features, exhibited in real image sequences, that violate the basic model assumptions (motion consistency and image brightness consistency): thus they provide very optimistic (upper) bounds on the performance one can expect. However, synthetic image sequences do have two experimental advantages which make them useful: one can easily calculate the true motion field (indeed there are relatively few real sequences with known motion fields in general use in the vision community), and one can control the ways in which the basic model assumptions are violated. In our examples, the synthetic sequences basically violate just one model assumption: there is a motion discontinuity (and a simple and precisely known one at that) that allows us to investigate the performance of the algorithm in quantitatively identifying the failure of the simple motion model (and correctly reporting one of the two motions in a patch near the boundary).

All of the derivatives of the image brightness function are calculated by constructing the appropriate derivative of a 3D Gaussian function (with equal spatial and temporal standard deviation). Each sequence is then convolved with these derivative of Gaussian functions. We report the results for different sized spatial rectangular windows within which the motion is assumed to be constant or affine. The number of unknowns in both approaches (number of columns in matrix A of Eq. (2.2)) depends on the model of motion in every patch of the image. Constant (2 unknowns) and affine (6 unknowns) motions are the most common models of motion proposed in the optic flow literature (see Szeliski and Coughlan (1994) for detail description of different motion models) . To keep the computation minimum, in both approaches, first the LMedS or LMSOD is solved for all the OFC contained in a square window with constant model of motion. Then, the weights are calculated for every OFC based on its residual with respect to the initial estimate. The intuitive idea behind this is very simple. The outliers

contaminating the OFC (due to multiple motions, transparency, etc.) are independent of the motion model and by rejecting the outliers using constant motion model, the computational time is reduced. It is important to note that this argument is only justified for small windows where the chance of disregarding good points at the tail of the affine model by the robust solution calculated using constant model is negligible. One of course, may achieve slightly better results by using the affine model of motion in both steps.

The constraints whose their scaled residual is above some threshold are simply rejected. The final step in estimating the flow field is to solve the new system of over-determined linear equations using LS or TLS and compute the associated R^2 statistics. To investigate the effect of different motion models on the accuracy of estimated flow field, the weighted set of OFC is solved using both the constant (2 unknowns) and the affine motion model (6 unknowns).

The error analysis is performed using Barron's (Barron et al., 1994) software and the errors are reported in "degrees". This measure is the angle between the true and estimated motions when each is expressed in homogeneous coordinates. We refer to the cited paper for full details. As stated by Otte and Nagel (1994), the values of the errors reported by this measure should be treated with some suspicion as estimates that have the same magnitude of error may provide vastly different angular errors. For the sake of comparison, the results of most accurate optic flow techniques were also quoted (where available) from the literature.

Before the experimental results are detailed, the computational cost will be briefly discussed. Properly comparing the computational cost of algorithms is, of course, a difficult procedure. Various optimisations can dramatically change the time it takes an algorithm to run. Moreover, the speed can be affected by various parameter settings: in our approach, two significant settings are the size of the patch, and the number of pairs of lines we use to approximate the LMedS (or LMSOD) result in that patch. For these reasons, since our code (and usually the code of other researchers) is written more for correctness than speed, we give only rough, indicative times. Running on an SGI Indy (SC 4600 at 132 MHz) our code for LS based method (WLS) takes approximately 10 min to calculate the flow for the Yosemite image sequence (it simultaneously computes results for constant and affine model of motion and also perform the validation procedure for both cases), using patches of size 15×15 around each pixel, and using 30 pairs of lines in each patch to approximate the LMedS. We can get similar results faster, by using only 10 pairs of lines per patch, for example, and the running time is roughly halved. The computation time for the WTLS method is slightly higher and depends on the method used for solving the TLS problem (see VanHuffel and Vandewalle (1991) for more details). The Fleet and Jepsons (1990) method takes about an hour to compute the flow for this sequence on the same machine. See Liu et al. (1996) for detail discussion on computation time of various optic flow techniques.

5.1. New-Sinusoid1 Image Sequence

We created a sinusoidal image sequence similar to Sinusoid1 of (Barron et al., 1994). The sequence contains images having the same spatial frequencies as Sinusoid1. However, in contrast to that sequence, which had spatially constant motion across the whole image, our sequence has motion boundaries. This was achieved by creating a stationary square (length of side 50 pixels) in the middle of each image. Figure 4 shows a sample image from the sequence and Table 1 presents the error statistics for this sequence. The result for the Fleet and Jepson (1990) method is created using the software developed by Barron et al. (1994). From this table, it is seen that our method clearly outperforms the Fleet and Jepson method in both accuracy and density of the points for which estimates are provided.

Without the validation procedure, to detect motion estimates that do not fit our image model well, we still have some erroneous estimates along the boundaries of the stationary rectangle.

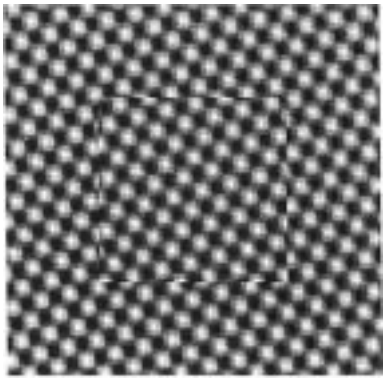

Figure 4. One frame taken from New-Sinusiod1 sequence.

Table 1. Error analysis using New-Sinusoid1 image sequence.

Technique	Avg. error (degree)	Std. dev. (degree)	Density (%)
Fleet and Jepson ($\sigma = 2.5, \tau = 1.25$)	7.39	10.84	43.4
Fleet and Jepson ($\sigma = 2.5, \tau = 2.5$)	1.41	3.65	46.0
WLS2 ($\sigma = 1.0, 5 \times 5, m = 30$, without check)	1.56	7.12	100
WLS2 ($\sigma = 1.0, 5 \times 5, m = 30, R^2 = 0.9999$)	0.05	0.06	84.6
WLS6 ($\sigma = 1.0, 5 \times 5, m = 30$, without check)	1.51	5.86	100
WLS6 ($\sigma = 1.0, 5 \times 5, m = 30, R^2 = 0.9999$)	0.05	0.06	83.5
WTLS2 ($\sigma = 1.0, 5 \times 5, m = 30$, without check)	2.82	8.82	100
WTLS2 ($\sigma = 1.0, 5 \times 5, m = 30, R^2 = 0.9999$)	0.05	0.06	76.1
WTLS6 ($\sigma = 1.0, 5 \times 5, m = 30$, without check)	1.51	6.23	100
WTLS6 ($\sigma = 1.0, 5 \times 5, m = 30, R^2 = 0.9999$)	0.08	0.22	88.4

Note: The first column of entries determines the method applied to generate the row of error statistics. In our methods (WLS and WTLS) the numbers 2 and 6 represent the constant and affine motion models, respectively. The numbers in brackets depict the size of the Gaussian smoothing (σ is the standard deviation of the filter), the size of local patch used (p), the number of pairs of lines used to approximate the LMedS or the LMSOD (m), and the reliability threshold (R^2), in that order.

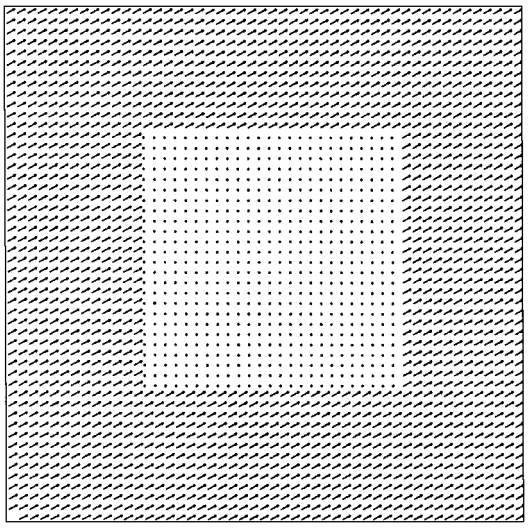

Figure 5. The correct flow of the New-Sinusoid1 sequence. Small $*$ symbols denote zero velocities at those positions.

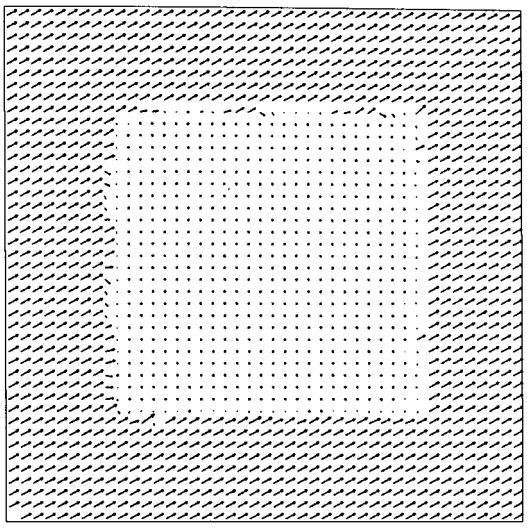

Figure 6. New-Sinusoid1 sequence—Flow calculated with WLS without validation.

From Figs. 5–7, one can clearly see that our validation procedure correctly removes the unreliable motion estimates (and, in this simple case, the unreliable motion estimates are those around the boundary of the central rectangle: where the image motion model breaks down). The size of the improvement can be judged by comparing successive rows in Table 1 (WLS without check or validation compared with WLS using same size patch, etc., but with validation using $R^2 = 0.9999$). The validation procedure reduces the average error greatly but still retains a very high density of reported motion estimates. Since the spatial and temporal derivatives for this sequence are fairly accurate, there is no significant difference between the LS and TLS based approaches. Moreover, the underlying motion is simple translation which can be modelled nicely

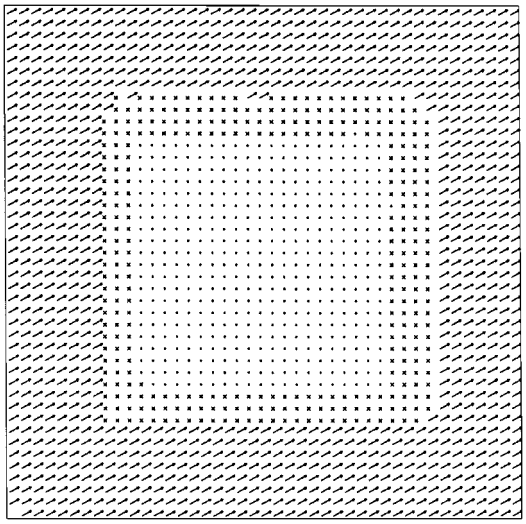

Figure 7. New-Sinusoid1 sequence—Flow calculated with WLS with validation. Representative example of the result of applying the validation procedure ($R^2 = 0.9999$) to remove unreliable motion estimates. The position marked with a $\times$ are those where the estimates were judged to be too unreliable. As expected, these unreliable estimates are clustered around the motion boundaries.

by both the constant and affine model of motion and therefore the final results for both approaches are very similar.

5.2. Yosemite Image Sequence

The Yosemite sequence is one of the most complicated synthetic sequences that is widely used in the research community. The sequence was generated from digital terrain data of the Yosemite valley and the sequence depicts a simulated "fly-through". The motion is mainly divergent, while the clouds drift towards the right with a speed of 1 pixel/frame. Since the fractal texture of the cloud changes in different images of the sequence (Heeger, 1997), its true motion is not simply related to the image brightness changes (Fig. 8). Thus, we only provide the error statistic for the cloudless sequence. The sequence is poorly sampled in time and the larger motions are, therefore, subject to bad temporal aliasing. The results of using this sequence in our experiments are shown in Tables 2 and 3. We also depict the true flows (Fig. 9), flows recovered by our method without validation (Fig. 10) and with validation (Fig. 11).

The first group of error statistics (Table 2) is calculated using the entire image while the second group (Table 3) is calculated by clipping the top 70 rows of the image and computing the results for the rest of the image. Since the horizon appears as a source of discontinuity, the second group (without the horizon) contains more accurate results.

Figure 8. Yosemite Sequence—with cloud.

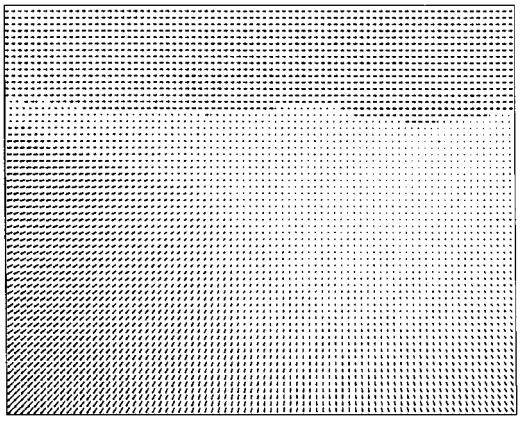

Figure 9. Yosemite Sequence Correct Flow—with cloud.

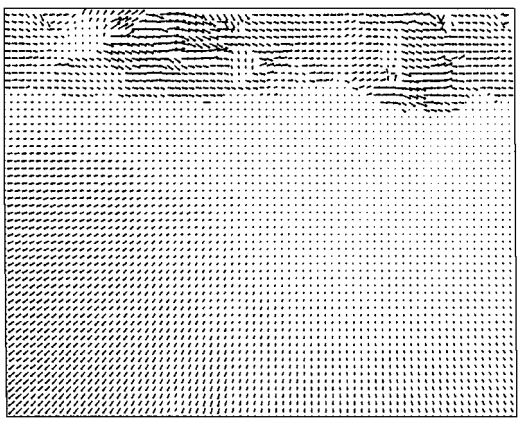

Figure 10. Yosemite Sequence WLS Flow (no validation)—with cloud.

Table 2. Error analysis using Yosemite image sequence (cloud region included).

Technique	Avg. error (degree)	Std. dev. (degree)	Density (%)
Fleet and Jepson ($\sigma = 1.5, = 1.25$, 1990)	4.95	12.39	30.6
Fleet and Jepson ($\sigma = 1.5, = 2.50$, 1990)	4.29	11.24	34.1
Weber and Malik (1993)	3.42	5.35	45.2
Szeliski and Coughlan (1994)	3.06	7.54	39.6
Weber and Malik (1995)	4.31	8.66	64.2
Giachetti and Torre (1996)	2.82	6.98	70.79
WLS2 ($\sigma = 2.0, 15 \times 15, m = 30$, without check)	3.17	6.46	100
WLS2 ($\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.99$)	3.13	7.07	76.2
WLS6 ($\sigma = 2.0, 15 \times 15, m = 30$, without check)	2.86	6.76	100
WTLS2 ($\sigma = 2.0, 15 \times 15, m = 30$, without check)	2.49	3.08	100
WTLS2 ($\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.99$)	2.14	2.58	81.6
WTLS6 ($\sigma = 2.0, 15 \times 15, m = 30$, without check)	2.05	2.92	100
WTLS6 ($\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.99$)	2.01	2.81	96.3
WTLS2 ($\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.999$)	1.74	2.37	72.0

Note: Same as in Table 1.

Table 3. Error analysis using Yosemite image sequence (cloud region excluded).

Technique	Avg. error (degree)	Std. dev. (degree)	Density (%)
Black (1994)	3.52	3.25	100
Black and Jepson (1994)	2.29	2.25	100
Black and Anandan (1996)	4.46	4.21	100
Ju et al. (Skin and Bones, 1996)	2.16	2.00	100
WLS2 ($\sigma = 2.0, 15 \times 15, m = 30$, without check)	2.51	2.57	100
WLS6 ($\sigma = 2.0, 15 \times 15, m = 30$, without check)	2.02	2.05	100
WTLS2 ($\sigma = 2.0, 15 \times 15, m = 30$, without check)	2.56	2.34	100
WTLS6 ($\sigma = 2.0, 15 \times 15, m = 30$, without check)	1.97	1.96	100

Note: Same as in Table 1.

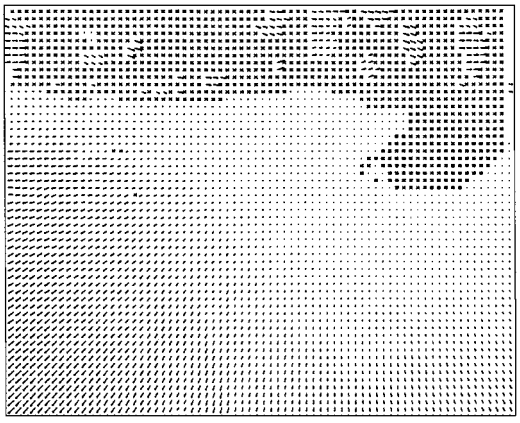

Figure 11. Yosemite Sequence WLS Flow $R^2 = 0.9$—with cloud. The validation procedure has correctly identified many of the motion vectors in the cloud region as being unreliable.

Considering the results for this sequence, we see WLS and WTLS are clearly superior to any of the published methods used in the comparison. One can also see, from these results, that the WTLS based method is generally more accurate than the WLS method. Since the actual motion in this sequence is a "fly-through" motion, the affine model of motion performs better than the constant model of motion.

5.3. Otte Image Sequence

This sequence is a real image sequence, recorded using a camera, which translates toward a scene. The objects in that scene are stationary, except for a Marble block, which is translating towards the left. A snapshot taken from this sequenceis shown in Fig. 12—for more

Table 4. Error analysis using Otte image sequence.

Technique	Avg. error (degree)	Std. dev. (degree)	Density (%)
Fleet and Jepson ($\sigma = 2.0, \tau = 1.25$)	2.08	3.77	50.6
Fleet and Jepson ($\sigma = 2.0, \tau = 2.50$)	2.56	4.08	57.1
Fleet and Jepson ($\sigma = 2.5, \tau = 1.25$)	2.05	3.85	55.8
Fleet and Jepson ($\sigma = 2.5, \tau = 2.50$)	2.53	4.25	62.2
Giachetti and Torre (1996)	5.33	—	100(25)
WLS2 ($\sigma = 2.0, 15 \times 15, m = 30$, without check)	3.39	6.55	100
WLS2 ($\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.99$)	1.50	2.22	59.1
WLS6 ($\sigma = 2.0, 15 \times 15, m = 30$, without check)	3.51	6.48	100
WLS6 ($\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.99$)	1.44	1.92	55.9
WTLS2 ($\sigma = 2.0, 15 \times 15, m = 30$, without check)	3.74	8.09	100
WTLS2 ($\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.99$)	1.61	2.60	71.2
WTLS6 ($\sigma = 2.0, 15 \times 15, m = 30$, without check)	3.67	7.37	100
WTLS6 ($\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.99$)	2.46	4.71	82.0
WTLS6 ($\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.999$)	1.55	2.34	51.6

Note: Same as in Table 1.

Figure 12. Snapshot taken from the Otte sequence.

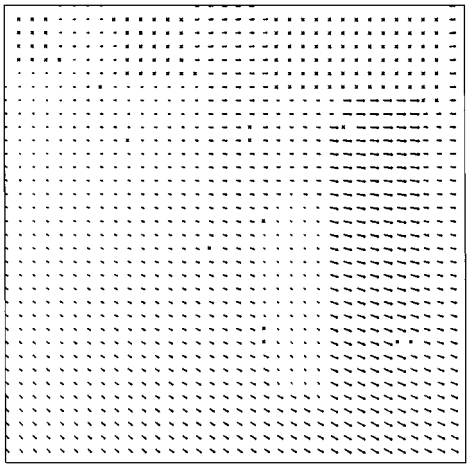

Figure 13. Otte Sequence—Correct Flow. Note that, unlike most test sequence with known velocity, there are actually patches of unknown velocity in this sequence (marked with $\times$ symbols).

details of the sequence see (Otte and Nagel, 1994). The scene contains many sharp discontinuities in both depth and motion. The results of our experiments are shown in Table 4. One can clearly see, from these results that our methods perform better than other published approaches. The results for the Fleet and Jepson's (1990) method is created using the software provided by Barron et al. (1994).

The above results show that both methods are very robust to the existing depth and motion discountiniuties (see Figs. 13–15). Although the results for both the LS and TLS based methods are very similar, the TLS based method seems to perform slightly better than the LS based method (after reliability check).

6. Conclusion

Two new robust methods for solving a system of over-determined linear equations have been developed for the purpose of calculating optic flow. The essence of

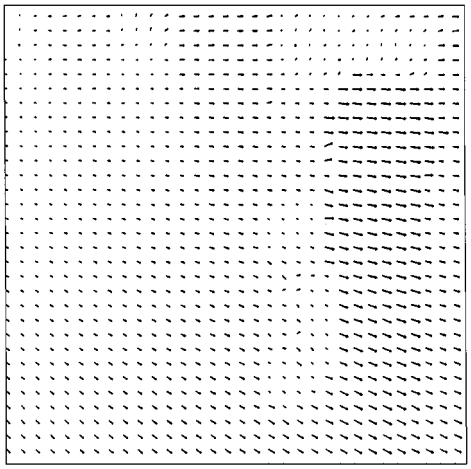

Figure 14. Otte Sequence—WLS Flow (no validation).

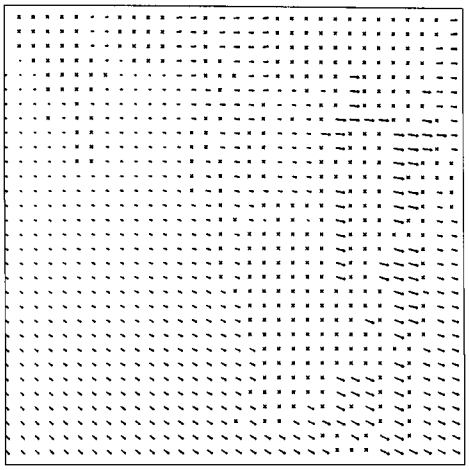

Figure 15. Otte Sequence—WLS Flow ($R^2 = 0.99$).

the first method is that we use an *approximate* Least median of Squares approach to identify outliers. This is particularly good at identifying and removing the effects of the breakdown of the motion consistency assumptions underlying all optic flow formulations (implicitly or explicitly, all formulations *must* use a form of this assumption to "beat the aperture problem"). Having detected and removed outliers, a simple Least Squares approach is used, followed by a validation step. The validation procedure, although probably useful in other situations, is very necessary to detect situations where the local area does not contain one dominant motion.

The second method focuses on solving a system of over-determined linear equations when all the parameters of the equations are contaminated with both noise and outliers. The proposed algorithm uses a new robust regression method named the least median of squares orthogonal distances in conjunction with the well-known total least squares method for dealing with the outliers and noise, respectively. A fast method for computing an approximate solution to the LMSOD is also proposed which makes the computation inexpensive.

An important ingredient of both approaches is that we develop an effective validation procedure by which the unreliable estimates are detected.

Although the presented algorithms are conceptually very straightforward, it was shown by a number of experiments over the synthetic and real image sequences that they outperform any other (often very sophisticated) optic flow technique appearing in computer vision literature.

Finally, it should be stressed that, since the essence of our approach is that it is a robust method to solve an over-determined linear system of equations, the approach should be applicable to a wide variety of problems (including and beyond other problems drawn from computer vision research).

Acknowledgments

During the course of this work, the first author is supported by scholarships from Ministry of Culture and Higher Education, Iran and Monash University, Australia.

Note

1. The Frobenius norm of a $m \times n$ matrix M, with entries m_{ij}, is defined as $\|M\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} m_{ij}^2}$.

References

Bab-Hadiashar, A. 1996. Least squares and closest point problem in optic flow estimation: The duality or the confusion. Technical Report MECSE 96–2, Monash University, Clayton, 3168, Australia, available at: http://batman.eng.monash.edu.au/ali/.

Barron, J.L., Fleet, D.J., and Beauchemin, S.S. 1994. Systems and experiment performance of optical flow techniques. *Intern. J. Comput. Vis.*, 12:43–77.

Bergen, J.R., Anandan, P., Hana, K.J., and Hingorani, R. 1992. Hierarchical model-based motion estimation. In *Proc. Secd. Europ. Conf. Comp. Vis., ECCV-92*, Springer-Verlag, pp. 237–252.

Bergen, J.R., Burt, P.J., Hana, K.J., Hingorani, R., Jeanne, P., and Peleg, S. 1991. Dynamic multiple motion estimation. *Artif. Inte. Comp. Vis. Proc. Israeli Conf.*, North Holland, pp. 147–156.

Black, M.J. 1994. Recursive non-linear estimation of discontinous flow field. In *Proc. of Third Europ. Conf. on Comp. Vis., ECCV-94*, Springer-Verlag, pp. 138–145.

Black, M.J. and Anandan, P. 1991. Robust dynamic motion estimation over time. In *Proc. of Computer Vision and Pattern Recognition, CVPR-91*, Maui, HI, pp. 296–302.

Black, M.J. and Anandan, P. 1993. A framework for the robust estimation of optical flow. In *Proc. Int. Conf. on Computer Vision, ICCV-93*, Berlin, pp. 231–236.

Black, M.J. and Anandan, P. 1996. The robust estimation of multiple motion: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104.

Black, M.J. and Jepson, A.D. 1994. Estimating multiple independent motions in segmented images using parametric models with local deformations. Workshop on Motion of Non-rigid and Articulated Objects, Austin, pp. 220–227.

Bober, M. and Kittler, J. 1994a. Estimation of complex multimodal motion: An approach based on robust statistics and hough transform. *Image and Vision Computing*, 12(10):661–668.

Bober, M. and Kittler, J. 1994b. Robust motion analysis. In *Proc. of Computer Vision and Pattern Recognition, CVPR-94*, Seattle, pp. 947–952.

Burt, P.J., Bergen, J.R., Hingorani, R., Kolczynski, R., Lee, W.A., Leung, A., Lubin, J., and Shvaytser, H. 1989. Object tracking with a moving camera: An application of dynamic motion analysis. In *Proc. of the Workshop on Visual Motion*, Irvin, CA, pp. 2–12.

Cafforio, C. and Rocca, F. 1976. Methods for measuring small displacement of television images. *IEEE Trans. Inform. Theory*, IT-22:573–579.

Chaudhuri, S. and Chatterjee, S. 1991. Performance analysis of total least squares method in three-dimensional motion estimation. *IEEE Trans. on Robotics and Automation*, 7(5):707–714.

Chu, C.H. and Delp, E.J. 1989. Estimating displacement vector form an image sequence. *J. Opt. Soc. Am. A*, 6(6):871–878.

Darrell, T. and Pentland, A. 1991. Robust estimation of a multilayered motion representation. *IEEE Proc. of Workshop on Visual Motion*, Princeton, NJ, pp. 173–178.

Edelsbrunner, H. and Souvaine, D.L. 1990. Computing least median of squares regression lines and guided topological sweep. *J. of the American Statistical Association*, 85(409):115–119.

Fennema, C. and Thompson, W. 1979. Velocity determination in scenes containing several moving objects. *Comput. Graph. Image Process*, 9:301–315.

Fleet, D.J. and Jepson, A.D. 1990. Computation of component image velocity from local phase information. *Intern. J. Comput. Vis.*, 5:77–104.

Giachetti, A. and Torre, V. 1996. Refinement of optical flow estimation and detection of motion edges. In *Proc. of Fourth Europ. Conf. on Comp. Vis., ECCV-96*, Springer-Verlag, pp. 151–160.

Hampel, F.R., and Ronchetti, E.M., Rousseeuw, P.J., and Stahel, W.A. 1986. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley: New York.

Heeger, D. 1997. Private correspondence.

Horn, B.K.P. 1986. *Robot Vision*. MIT Press: Cambridge.

Horn, B.K.P. and Schunck, B.G. 1981. Determining optical flow. *Artificial Intelligence*, 17:185–204.

Iu, S.L. 1995. Robust estimation of motion vector field with discontinuity and occlusion using local outliers rejection. *J. of Visual Communication and Image Representation*, 6(2):132–141.

Jähne, B. 1993. *Spatio-Temporal Image Processing, Theory and Scientific Applications*. Springer-Verlag: New York.

Jepson, A.D. and Black, M.J. 1993. Mixture models for optical flow computation. In *Proc. of Compu. Vision and Pattern Recognition, CVPR-93*, New York, pp. 760–761.

Jepson, A.D. and Black, M.J. 1995. Mixture models for optical flow computation. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 19, pp. 271–286.

Ju, S.X., Black, M.J., and Jepson, A.D. 1996. Skin and Bones: Multilayer, locally affine, optical flow and regularaization with transparency. In *Proc. of Computer Vision and Pattern Recognition, CVPR-96*, San Francisco, pp. 307–314.

Kvalseth, T.O. 1985. Cautionary note about R^2. *The American Statistician*, 39(4):279–285.

Limb, J.O. and Murphy, J.A. 1975. Measuring the speed of moving objects from television signals. *IEEE Trans. on Communications*, 23(4):474–478.

Liu, H., Hong, T., Herman, M., and Chellappa, R. 1996. Accuracy vs. efficiency trade-offs in optical flow algorithms. In *Proc. of Fourth Europ. Conf. on Comp. Vis., ECCV-96*, Springer-Verlag, pp. 174–183.

Meer, P., Mintz, D., Rosenfeld, A., and Kim, D.Y. 1991. Robust regression methods for computer vision: A review. *Intern. J. Comput. Vis.*, 6(1):59–70.

Mitiche, A. 1994. *Computational Analysis of Visual Motion*. Plenum: New York.

Nagel, H.H. 1987. On the estimation of optical flow. *Artificial Intelligence*, 33:299–324.

Nagel, H.H. 1995. Optical flow estimation and the interaction between measurement errors at adjacent pixel positions. *Intern. J. Comput. Vis.*, 15:271–288.

Nesi, P., Del Bimbo, A., and Ben-Tzvi, D. 1995. A robust algorithm for optical flow estimation. *Computer Vision and Image Understanding*, 62(1):59–68.

Odobez, J.M. and Bouthemy, P. 1995. Robust multiresolution estimation of parametric motion models. *J. of Visual Communication and Image Representation*, 6(4):348–365.

Otte, M. and Nagel, H.H. 1994. Optical flow estimation: Advances and comparisons. In *Proc. of Third Europ. Conf. on Comp. Vis., ECCV-94*, Springer-Verlag, pp. 51–60.

Rousseeuw, P.J. 1984. Least median of squares regression. *Journal of the American Statistical Association*, 79:871–880.

Rousseeuw, P.J. and Leroy, A.M. 1987. *Robust Regression and Outlier Detection*. John Wiley: New York.

Schunck, B.G. 1989. Image flow segmentation and estimation by constraint line clustering. *IEEE Trans. on Pattern Anal. and Mach. Intell., PAMI*, 11(10):1010–1027.

Shi, J. and Tomasi, C. 1994. Good features to track. In *Proc. of Computer Vision and Pattern Recognition, CVPR-94*, Seattle, pp. 593–600.

Steele, J.M. and Steiger, W.L. 1986. Algorithms and complexity for least median of squares regression. *Discrete. Appl. Math.*, 14:93–100.

Stewart, C. 1996. Bias in robust estimation caused by discontinuities and multiple structures. Technical Report, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York TR-12180–3590.

Szeliski, R. and Coughlan, J. 1994. Hierarchical spline-based image registration. In *Proc. of Computer Vision and Pattern Recognition, CVPR-94*, Seattle, pp. 194–201.

VanHuffel, S. and Vandewalle, J. 1991. *The Total Least Squares Problem: Computational Aspects and Analysis*. 1st edition, SIAM: Philadelphia.

Van Mieghlem, J.A., Avi-Itzhak, H.I., and Melen, R.D. 1995. Straight line extraction using iterative total least squares method. *J. of Visual Communication and Image Representation*, 6(1):59–68.

Wang, S., Markandey, V., and Reid, A. 1992. Total least squares fitting spatiotemporal derivatives to smooth optical flow field. In *Proc. of the SPIE: Signal and Data Processing of Small Targets*, vol. 1698, pp. 42–55.

Weber, J. and Malik, J. 1993. Robust computation of optical flow in a multi-scale differential framework. In *Proc. of Int. Conf. on Computer Vision, ICCV-93*, Berlin, pp. 12–20.

Weber, J. and Malik, J. 1995. Robust computation of optical flow in a multi-scale differential framework. *Intern. J. Comput. Vis.*, 14:67–81.