Experiments with robust estimation techniques in real-time robot vision

Ezio Malis, Eric Marchand

▶ **To cite this version:**

Ezio Malis, Eric Marchand. Experiments with robust estimation techniques in real-time robot vision. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'06, 2006, Beijing, China. pp.223-228. inria-00350297

HAL Id: inria-00350297
https://inria.hal.science/inria-00350297

Submitted on 6 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'06,
Beijing, China, Octobre 2006

Experiments with robust estimation techniques in real-time robot vision

Ezio Malis
INRIA Sophia-Antipolis, Icare project ;
Ezio.Malis@inria.fr

Eric Marchand
INRIA, IRISA, Lagadic project ;
Eric.Marchand@irisa.fr

Abstract— **The goal of this paper is to present an overview of robust estimation techniques with a special focus on robotic vision applications. In this particular context, constraints due computation time have to be considered in the choice of the estimation algorithm. Among the numerous techniques proposed in the literature to obtained robust estimation we have, not being exhaustive, Hough transform, RANSAC (Random Sample Consensus), the LMedS (Least Median of Squares), the M-estimators, etc. In this overview, we describe these various approaches in the light of a simple example. Finally, we illustrate the use of robust estimation techniques by various examples in real-time robot vision.**

I. Introduction

The goal of this paper is to present an overview of robust estimation techniques with a special focus on robotic vision applications. In this particular context, constraints due computation time have to be considered in the choice of the estimation algorithm. We adopt the following definition for robustness: an estimation algorithm is "robust" if its properties (i.e. the ability of estimating correctly some parameters) remain valid despite the uncertainty in the model, measurements errors and modifications in the environment.

An image sequence represents an almost infinite source of information which is extremely difficult to model exhaustively by analytical development. This includes dynamic problems such as moving objects, multiple occlusions, changes in illumination due to varying light sources, different visual medium such as water and air, etc... It is clear that handling all the potential sources of error by analytical classification is a very complex and difficult task. If a camera provides very rich low-level information about the environment, extracting various high level information or visual features necessary to solve a specific problem is a non trivial issue. Furthermore, the fact that these visual features have to be extracted with a sufficient precision is an important and usually non realistic hypotheses.

The considered high level information may be very different according to the considered problem: depth information, relative camera/scene position, intrinsic camera parameters, robot parameters, etc. Considering most of applications, simple photometric or geometric models may be sufficient for estimating the information needed in most of robotic applications. Projective geometry is, for example, a powerful mathematical tool to model the geometry of the environment and the image acquisition process. Nevertheless, when highly complex images are considered (e.g., for robotics navigation in outdoor environment), this modeling process is no longer realistic nor valid and robust algorithm are required to overcome uncertainty in data extraction and important outliers.

In related literature, many different approaches exist to treat external sources of error. To handle the outlier issue, it is possible to classify robust estimation approaches in two main classes. The former approach consider an outlier detection prior to the parameters estimation process. Among this approaches, the most classical are voting methods such as the Hough transform or RANSAC (Random Sample Consensus) [9]. This latter algorithm performs an estimation of the considered parameters from a minimal set of data and other data that confirms this first estimation are then integrated in the estimation process. The best consensus is retained. Although these approach are very efficient from the robustness point of view, the are usually quite time consuming. The second class of approaches allows to solve simultaneously the outliers detection and parameters estimation process. Among this method we have Least Median of Squares (LMedS) [21], LTS, and M-estimators, L-estimators, R-estimators [15]. The objective function to be minimized is modified to reduce the sensitivity to outliers. To be efficient, this approaches also estimate the standard deviation of the inliers (ie, the "good" data). These approach are a good choice when algorithmic effectiveness is of interest.

One of the main characteristic of a robust method is its breakdown point. Breakdown point is the percentage of outliers leading to a wrong estimation. For example, considering the classical least squares algorithm, this breakdown point is of 0% since only one outlier is sufficient to obtain a wrong estimation of the parameters. Other characteristics, fundamental in the robotic context, are the complexity (from a computation time point of view) and the convergence rate. If complexity it to high or convergence too slow, in a robotics context the control rate decreases which may prove to be a problem. We will see that there is usually a trade-off between convergence rate and breakdown point.

II. Parameters estimation

The considered problem is the estimation of a set of parameters using data extracted from one or more images. The most classical algorithm to handle this problem is the well known least squares approach.

A. Least squares minimization

Assume that a model of a set of signals $s_k(\mathbf{x})$, that rely on a set of unknown parameters $\mathbf{x}$, and on measurements $s_k^* = s_k(\mathbf{x}^*)$ of these signals are available. We defined the residual r_k by $r_k = s_k(\mathbf{x}) - s_k^*$ and $\mathbf{r}(\mathbf{x})$ the (n$\times$1) vector that contains all the residual ordered by increasing order such

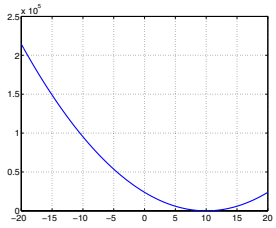

Fig. 1. Least squares objective function (LS).

that $r_1^2(\mathbf{x}) \leq r_2^2(\mathbf{x}) \leq ... \leq r_n^2(\mathbf{x})$. Our problem is to find the value $\mathbf{x}^*$ of the unknown parameters from measured signals s_k^*. The most classical approach which come back to Gauss and Legendre [21], is the least squares approach (LS). The goal is to minimize the following objective function:

$$C(\mathbf{x}) = \mathbf{r}^\top \mathbf{r} = \sum_{k=1}^{n} r_k^2(\mathbf{x}) \tag{1}$$

If $s_k(.)$ is linear in $\mathbf{x}$ the solution to this problem can be obtained by a classical linear least squares approach. Nevertheless in the general case, when $s_k(.)$ is a non linear function of $\mathbf{x}$ and the solution of this optimization problem can be obtained using an iterative approach (Newton-based minimization algorithms) considering an initial estimation $\mathbf{x}_0$ of the parameters.

B. An academic example

In the reminder of this paper we will consider the various estimation method in the light of an academic example. Let us consider that $n = 100$ points of interest are extracted and matched in two images. Let us assume that the camera has made a 10 mm translation along the $\vec{x}$ axes. The goal is to estimate this camera displacement using point coordinates and the knowledge of each point depth. The equation that links the normalized coordinates of these points in the two image is given by:

$$u_{2k} = u_{1k} + \frac{t_x}{Z_{1k}} \tag{2}$$

In this simple problem, we have only one parameter to estimate ($\mathbf{x} = t_x$) and it is easy to visualize the objective function related to the various methods. The objective function is quadratic and its minimum is $x^* = 10$ mm (see figure 1). The least square problem LS is easily solved for any initial condition. Nevertheless, we will always consider the same initial condition that is $\mathbf{x}_0 = 0$.

C. Outliers issue

The measurements process is never perfect, and it seems reasonable to assume that there are spurious data (outliers). In a robotic vision context, these outliers may due to matching errors, occlusions, lighting variation, etc.

In this case, the least squares objective function is modified and the minimum does not correspond to the actual value of the parameters to be estimated. For example, on Figure 2a the least squares objective function is displayed for a set of data where measures are randomly contaminated with 20% of outliers. The minimum of the objective function is $x = 6$ mm when the actual value is $x = 10$ mm. Similarly, figure 2b the least squares objective function is displayed for a set of data where measures are randomly corrupted with 40% of outliers. The minimum of the objective function is now $x = 4$ mm. In these two figures the green points represents the initial value of x and the red points the results of the minimization. As can be expected, it shows that the LS approach is not robust to outliers. It breakdown point is 0% which that only one outlier is enough to corrupt the estimation process. In some case, the minimum can be modified in a tremendous way is the outlier have a high influence (leverage data [21]).

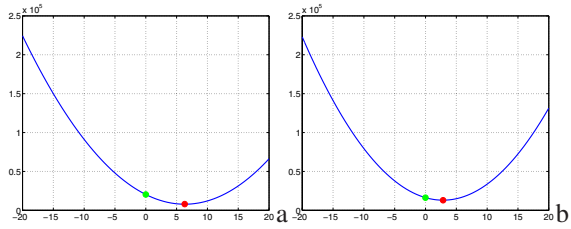

Fig. 2. Least squares objective function with (a) 20% and (b) 40% of outliers.

The robust estimation methods that are described in the next sections allows to estimate the actual value of the considered parameters despite the presence of these outliers.

III. Robust iterative methods

Least squares are not robust because the objective function can grow indefinitely and outliers may become dominating w.r.t. measurements that actually verify the actual model. Alternatively, the robust methods modify the objective function so as to limit the influence of the most significant residues. The principal consequence is usually, a slower convergence speed of the optimization algorithms. Indeed, it is very difficult to distinguish, in a first time, outliers and inliers. Therefore, some inliers may be initially filtered which leads to a reduction of the convergence speed. Nevertheless, the most difficult case to solve arises when small but aberrant residues move the minimum of the objective function.

A. LMedS (Least Median of Squares)

The robust approach LMedS [21] minimize the following objective function:

$$C(\mathbf{x}) = \text{median}\big(r_1^2(\mathbf{x}), r_2^2(\mathbf{x}), ..., r_n^2(\mathbf{x})\big) \tag{3}$$

The median is seen as a robust estimator since it does not consider 50% of the most significant residues. Nevertheless, it has two major disadvantages:

- the objective function index is not differentiable. Methods of minimization based on the gradient of the objective function are thus very delicate to implement;
- the convergence speed of the minimization algorithms can be extremely slow if the residues are distributed in such a way that the median has a very weak gradient.

In the considered academic case, the minimum is correctly localized even in the presence of 20% or 40% outliers in the measurements. Figure 3 shows the objective functions in both cases. Beginning from $\mathbf{x}_0 = 0$ (the green points in the

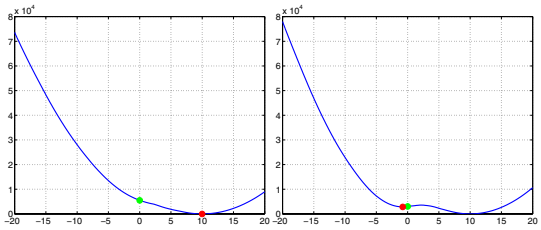

Fig. 4. LTS objective function with (a) 20% and (b) 40% of outliers.

figure) one finds the minimum correctly (the red point in the figure). This method has a breakdown point of 50% (highest possible) but also an extremely slow convergence. In order to increase the convergence speed several techniques are possible. One consists in carrying out random samples of measurement subsets in a way similar to RANSAC algorithm (see next section). Another possibility is to use method LTS.

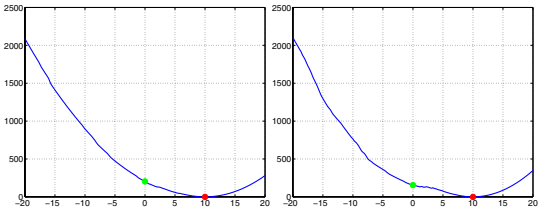

Fig. 3. LMedS objective function with (a) 20% and (b) 40% of outliers.

B. LTS (Least Trimmed Squares)

In order to improve the convergence speed of the LMedS method, Rousseeuw [21] proposed a method known as LTS. It consists in minimizing the sum of the squares of the q first residues:

$$C(\mathbf{x}) = \sum_{k=1}^{q} r_k^2(\mathbf{x}) \tag{4}$$

In general, $q = n/2$ is chosen but it is possible to optimize the choice of q with an a priori knowledge of the percentage of outliers. The objective function is similar to the LMedS' one but the gradient is usually higher. The contribution of the q first residues makes the objective function smoother. However, this function remains non differentiable. Figure 4 shows the objective function in the previously considered cases. The minimum is correctly found in the presence of 20% outliers.

When the percentage of outliers increases to 40%, it is always possible to find the true minimum but a local minimum appears in the objective function (although the global minimum is correctly located as can be seen on Figure 4b). This problem, which can also appear for other robust methods (including for the LMedS), depend on the distribution of the outliers and it is very difficult to analyze. In a general way, it is a priori impossible to figure out which robust method will allow to avoid the formation these local minima.

C. M-estimation

The principle of the M-estimators [15] is to modify the objective function of the least squares by penalizing the largest residues. The objective function is then defined by:

$$C(\mathbf{x}) = \sum_{k=1}^{n} \rho(r_k(\mathbf{x})) \tag{5}$$

where the function ρ is at least C^0. Various function ρ were proposed in the literature. The breakdown point of the M-estimators is theoretically 0% as for the LS. However, this is the worst case M-estimators failed only with a significant number of spurious measurements.

Among all the possible robust estimator ρ, the most popular are those proposed by Tukey [1] and Huber [15]. Function ρ proposed by Huber is defined by:

$$\rho(r_k(\mathbf{x})) = \begin{cases} \frac{1}{2} r_k^2(\mathbf{x}) & \text{if } r_k^2(\mathbf{x}) \leq c \\ c\left(|r_k(\mathbf{x})| - \frac{c}{2}\right) & \text{if } r_k^2(\mathbf{x}) > c \end{cases} \tag{6}$$

where $c = 1.345\widehat{\sigma}$ and $\widehat{\sigma}$ is a robust estimation of the standard deviation of the inliers. It is usually defined by the *Median Absolute Deviation* (MAD) given by:

$$\widehat{\sigma} = 1.48\text{median}(|\mathbf{r} - \text{median}(\mathbf{r})|). \tag{7}$$

In the function ρ proposed by Huber, the weakest residues are regarded as in a LS whereas the most significant residues are quickly limited (but not canceled). The figure 5 shows the objective function in the considered cases. Even if theoretically the breakdown point is 0%, the performance index of the M-estimators has its global minimum correctly localized. However, when the percentage of *outliers* is high, local minima can, here again, appear (although the global minimum is still correctly located).

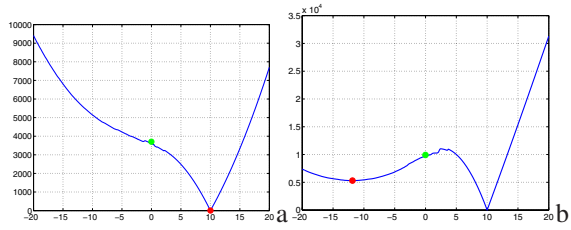

Fig. 5. M-estimator objective function with the Huber estimator with (a) 20% and (b) 40% of outlier.

Function ρ proposed by Beaton-Tukey [1] is defined by:

$$\rho(r_k(\mathbf{x})) = \begin{cases} \frac{c^2}{6}\left[1 - \left(1 - \left(\frac{r}{k}\right)^2\right)^3\right] & \text{if } r_k^2(\mathbf{x}) \leq c \\ \frac{c^2}{6} & \text{if } r_k^2(\mathbf{x}) > c \end{cases} \tag{8}$$

where $c = 4.6851\widehat{\sigma}$ and where $\widehat{\sigma}$ is a robust estimation of the standard deviation of the inliers . As in the previous case, in the function ρ proposed by Beaton-Tukey, the weakest residues are regarded as in a LS whereas the most significant residues are quickly canceled (completely this time). The figure 6 shows results very similar to the results obtained by the Huber's M-estimator.

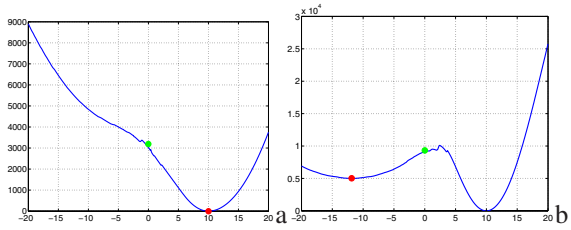

Fig. 6. M-estimator objective function with the Tukey estimator with (a) 20% and (b) 40% of outlier.

An advantage of the M-estimation is that it can be implemented using a simple iterative re-weighted least squares algorithm. Considering a linear problem, the actual least squares problems aims at solving for $\mathbf{x}$ the following linear system $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{x}$ and $\mathbf{b}$ are vectors and $\mathbf{A}$ is a matrix. Iterative re-weighted least squares algorithm aims at solving the following system $\mathbf{DAx} = \mathbf{Db}$ where $\mathbf{D} = diag(w_1, \ldots, w_n)$ is a diagonal matrix where w_i reflect the confidence of each feature. The algorithm acts as follows: estimate weights using one of the many robust criterion suggested in the literature (see next paragraph for example), estimate the value of $\mathbf{x}$ by solving the weighted system, and reiterate until convergence. These methods act like automatic outlier rejectors since large residual values lead to very small weights.

The weights w_i, which reflects the confidence of each feature, are usually given by [15]: $w_i = \frac{\psi(r)}{r}$, where the influence function $\psi(r) = \frac{\partial \rho(r)}{\partial r}$. When Tukey's function is considered, this influence function is given by:

$$\psi(r) = \begin{cases} r(C^2 - r^2)^2 & , \text{ if } |r| \leq c \\ 0 & , \text{else,} \end{cases} \tag{9}$$

Let us note that is all the cases, the choice of the optimization algorithm (Gradient, Newton, Gauss Newton, Levenberg Marquardt, Simplex) is then of great importance. Gradient approach may fail when the cost function is not differentiable but on the other side other minimization approaches such as simplex are usually prohibitive from a computational point of view[1].

IV. Robust voting methods

The estimation of the parameters with the voting methods rely on the use of the minimum of set of measurements. Each estimation, with a particular measurement sample, corresponds to a "vote" for the computed parameters. The set of chosen parameters, *i.e.* the one that gets the highest number of vote, is retained as the correct one.

A. Hough transform

The Hough transform [14] is a very robust voting method. The original version of the method suggested by Hough was modified by [8]. Several alternatives have been proposed [17]. This approach relies on a discretization of the parameters space. One then obtains hypercubes in the space state. Accumulators are associated with these hypercubes. For a specific sample, the required parameters are estimated and the corresponding accumulator in the hypercube is incremented. This process is reiterated until all the possible combinations have been considered. The accumulator having the most significant value corresponds to the best estimate of the parameters.

The Hough transform is well adapted to problems with a significant number of data compared to the number of parameters to be estimated. However, if the number data is similar with the number of unknowns, it is difficult to find a dominating accumulator. Moreover, due to the discretization issue and to the noise it is possible to obtained a wrong optimum. Nevertheless, the Hough transform is very robust since it carries out a total and exhaustive research. Finally, this technique is able to segment the data in several clusters which verify the reference model. However, the transform of Hough is very seldom used in robotic vision but for problems which require the estimate of less than three or four parameters. Indeed the computing times become rapidly prohibitive.

B. RANSAC

RANSAC [9] (Random Sample Consensus) is a probabilistic voting method which have been proposed to reduce the prohibitive computation time of classical voting approaches (such as the Hough transform). Considering a minimal set of measurements s, it is usually possible to compute the parameters in a non degenerate configuration. The following objective function is then computed:

$$C(\mathbf{x}) = \sum_{k=1}^{n} \rho(r_k(\mathbf{x})) \text{ with } \rho(r_k(\mathbf{x})) = \begin{cases} 0 & \text{if } r_k^2(\mathbf{x}) \leq c \\ 1 & \text{if } r_k^2(\mathbf{x}) > c \end{cases} \tag{10}$$

and $c = 2.5\ \widehat{\sigma}$. Let s the number of necessary measurements to compute a solution and r the percentage of inliers. The number m of random samples necessary to have a probability p to find the actual parameter is given by:

$$m = \frac{\log(1-p)}{\log(1-(1-r)^s)}$$

Figure 7 shows the objective function associated with RANSAC. Red lines are the results of the estimation considering a random sample. For 20% of outliers, only 5 random sample are sufficient to have a 95 % probability to find the actual solution. When the number of outliers increases to 40%, 13 random samples are necessary.

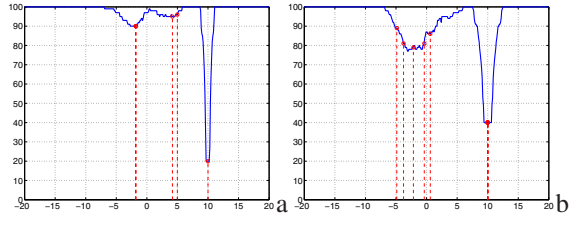

Fig. 7. RANSAC objective function with (a) 20% and (b) 40% of outliers.

The main advantage of Ransac w.r.t. the M-estimators is that it does not required an initial estimate of the solution.

V. Application to robots vision

Many computer vision problems are of interest for robot control or navigation. For example the following have been

[1]In all these experiment, a Gauss-Newton method has been considered (matlab `fminunc` function)

considered: homography estimation from point correspondence [13], [10], [6], [22], 2D motion estimation [19], [11], 3D rigid transformation from a set of 3D points [10], pose estimation [9], [12], [7], [6], essential or fundamental matrix estimation [24], camera calibration, eye-to-hand calibration, slam, visual servoing, etc. We now illustrates the previous section with some robotics problem where robust estimation have successfully considered: 2D object tracking in an image sequence, 3D camera localization, visual servoing. Let us note that, in each case, time is a critical issue and a trade-off between highly robust and fast approach has to be considered.

A. Visual tracking of a target in an image sequence.

Fig. 8. Robust visual tracking of a planar target with the ESM algorithm. The output of the algorithm is a homography matrix.

The robust estimation algorithms described in the previous sections can be applied to a standard robotic vision problem: the real-time visual tracking of a target in an image sequence. If we suppose that the target is planar, the visual tracking problem can be solved by estimating the homography matrix which links the coordinates of corresponding points in two images. We use the ESM visual tracking algorithm proposed in [2]. The (3×3) homography matrix is defined up to a scale factor. Thus, only 8 parameters have to be estimated. These parameters can be estimated by solving a standard least squares problem. However, as soon as outliers measures are used (as for example a specular reflection on the target surface), the visual tracking algorithm is not able to correctly estimate the parameters. It is thus necessary to use a robust estimation algorithm. Figure 8 show an example where M-estimator are used to reject the outlier measures. The target is selected manually in the first image and it is marked with a red rectangle. In the following images, the target it tracked despite the big illumination changes and big displacement of the object in the image. By solving a standard least squares problem it is not possible to track the object in all the sequence. On the other hand, robust methods need more computation time. It is thus important to use optimization techniques that compensate the time spent at each iteration with a faster convergence. It is the case of the ESM algorithm which use a second-order optimization technique (ESM stands for "Efficient Second-order Minimization") allowing for a faster convergence. The computation time is crucial in robotics applications. For example, from the estimated homography matrix it is possible to control a robot in real-time using a visual servoing technique [3].

B. 3D camera localization.

When dealing with 3D camera localization or pose computation, most of the approaches proposed in the literature rely on a 3D registration issue. Although it is a very old computer vision problem, e.g. [9], [4], it is still an important research issue. To illustrate the problem, let us consider the 3D camera localization from a set of 2D point. The pose estimation consists in estimating the frame transformation ${}^c\mathbf{M}_o$ that minimize the forward projection error:

$$ {}^c\widehat{\mathbf{M}}_o = \underset{{}^c\mathbf{M}_o}{\operatorname{argmin}} \sum_{i=1}^{N} \left(\mathbf{p}_i - pr_\xi({}^c\mathbf{M}_o, {}^o\mathbf{P}_i)\right)^2 \qquad (11) $$

where ${}^o\mathbf{P}$ are the 3D coordinates of object points in the object frame (object model), $\mathbf{p}_i$ are their projection in the image plane and $pr_\xi(.)$ is the projection function according to camera intrinsic parameters ξ.

To obtain the solution of this minimization problem, robust estimation can be considered: Ransac [9] was originally developed to solve this problem, but the M-estimator have been widely and successfully considered [12], [7], [6].

Figure 9 shows the result of a 3D tracking process using a robust model-based pose estimation process (Tukey M-estimator) in each image [6]. This localization is used here within a 2 1/2 D visual servoing experiment.

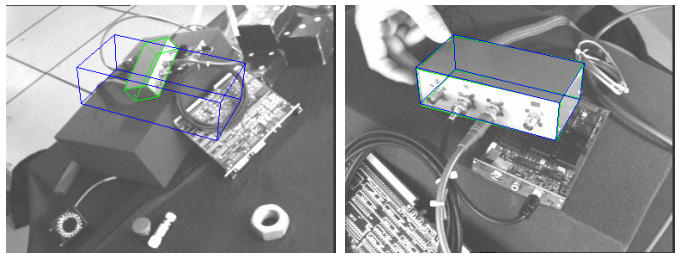

Fig. 9. Tracking within a 2D 1/2 visual servoing experiments. The four first images have been acquired in initial visual servoing step. In the reminder images object is moving along with the robot (a-b) camera velocity in rotation and translation (c) error vector $\mathbf{s} - \mathbf{s}^*$. Images are acquired and processed at frame rate (50Hz)

C. Robust visual servoing

Visual servoing is known to be a very efficient method for positioning and target tracking tasks [16]. However, its efficiency relies on correspondences between the position of tracked visual features in the current image and their position in the desired image which define a set of errors to be minimized. If these correspondences contain errors then visual servoing usually fails or converges upon a wrong position.

Dealing with robust visual servoing, the main (although it is not the only one) source of errors include those introduced by local detection and matching of features between the current and desired images. Overcoming this class of error is often achieved by improving the quality of tracking algorithms [23], [18], [6] and feature selection methods [20]. These approaches provide a robust input to the control law and as such treats outlier rejection, in an image processing step, prior to the visual servoing task (see Figure 10a). Nevertheless, it is clear that handling all the potential sources of error by analytical

classification is a very complex and difficult task. It has been shown that [5] the problem of *statistically robust* visual servoing can be implemented directly at the control law level (see Figure 10b).

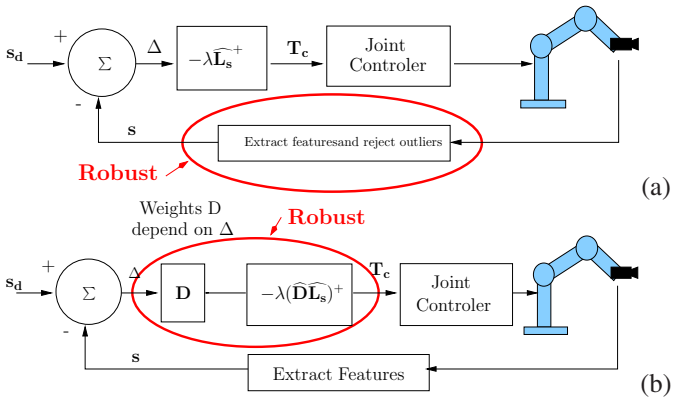

Fig. 10. Visual servoing with (a) traditional outliers rejection process at the image processing level, (b) Outliers rejection within the control law

This approach features three main advantages. With respect to more classical robust visual servoing techniques that rely on a robust extraction of the visual features, it bypasses intermediary decision steps which usually require thresholds to be tuned for each specific application. Second, the confidence in each visual feature relies on the value of all the other features. Finally, the computed uncertainty values do not act as a binary weight which completely rejects or accepts the feature. Each feature may either gain or loose certainty over time and during the execution of the control law.

Let us consider the generic positioning task. The goal of visual servoing is essentially to minimize the error $\Delta = \mathbf{s}(\mathbf{r}) - \mathbf{s}^*$ between a set of visual features $\mathbf{s}(\mathbf{r})$, that depends of the camera pose $\mathbf{r}$, and a set of desired visual features $\mathbf{s}^*$. The control law that performs Δ minimization is usually handled using a least square approach [16]. However when the data contains outliers, such a classical approach is no longer efficient.

To embed a robust minimization in visual servoing, a modification of the control law is required to allow outlier rejection. For that, a weight is associated to each feature to specify a confidence in its location. This leads to the following error to be minimized (see Fig 10b and [5]):

$$\mathbf{e} = \mathbf{D}\left(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\right), \tag{12}$$

where $\mathbf{D}$ is a diagonal weighting matrix: $\mathbf{D} = diag(w_1, \ldots, w_k)$. The weights w_i reflects the confidence we have in each visual feature and are computed using M-estimators. The new control law is given by [5]:

$$\mathbf{v} = -\lambda(\mathbf{DL})^{+}\mathbf{D}\big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big). \tag{13}$$

The obtained results showing the efficiency of this approach are given in [5]). It remains that the use of such a robust control law is far from being incompatible with an effective and robust tracking process. A fusion of the two diagrams of the figure 10 is not only possible but desirable.

VI. Conclusion

In this paper, we have reviewed the main robust estimation methods used in robotic vision. Considering such methods is necessary in order to achieved efficiently vision-based robotic tasks in real environment. The counterpart for the robustness is a higher computing time and a reduced convergence speed. If the voting techniques (Hough, Ransac) or LMedS are very efficient, the computing time is often too high to be used within a robotic control law. Another advantage of Ransac is that it provides an initial estimate of the parameters that is required by the M-estimators. Once this initial estimate is computed M-estimators represents a very good trade-off between robustness and algorithmic effectiveness.

References

[1] A.E. Beaton, J.W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16:147–185, 1974.
[2] S. Benhimane, E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. *IEEE/RSJ IROS'04*, pp. 943–948, Sendai, Japan, October 2004.
[3] S. Benhimane, E. Malis. Homography-based 2d visual servoing. *IEEE ICRA'06*, Orlando, Fl, May 2006.
[4] D.C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 4(2):127–140, March 1971.
[5] A.I. Comport, E. Marchand, F. Chaumette. Statistically robust 2d visual servoing. *IEEE Trans. on Robotics*, 22(2):415–421, apr 2006.
[6] A.I. Comport, E. Marchand, M. Pressigout, F. Chaumette. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Trans. on Visualization and Computer Graphics*, 12(4):615–628, July 2006.
[7] T. Drummond, R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. on PAMI*, 24(7):932–946, July 2002.
[8] R.O. Duda, P.E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Comm. of the ACM*, 15:11–15, 1972.
[9] N. Fischler, R.C. Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communication of the ACM*, 24(6):381–395, June 1981.
[10] A.W. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(12-13):1145–1153, December 2003.
[11] G. Hager, P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on PAMI*, 20(10):1025–1039, October 1998.
[12] R. Haralick, H. Joo, C. Lee, X. Zhuang, V Vaidya, M. Kim. Pose estimation from corresponding point data. *IEEE Trans on SMC*, 19(6):1426–1445, November 1989.
[13] R. Hartley, A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2001.
[14] P.V.C. Hough. Machine analysis of bubble chamber pictures. *Int. Conf. on High Energy Accelerators and Instrumentation*, pp. 554–556, 1959.
[15] P.-J. Huber. *Robust Statistics*. Wiler, New York, 1981.
[16] S. Hutchinson, G. Hager, P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.
[17] J. Illingworth, J. Kittler. A survey of the hough transform. *CVGIP*, 44(1):87–116, 1988.
[18] D. Kragic, H. Christensen. Cue integration for visual servoing. *IEEE Trans. on Robotics and Automation*, 17(1):19–26, February 2001.
[19] J.-M. Odobez, P. Bouthemy. Robust multiresolution estimation of parametric motion models. *J. of Visual Communication and Image Representation*, 6(4):348–365, December 1995.
[20] N. P. Papanikolopoulos, P. K Khosla. Selection of features and evaluation of visual measurements for 3D robotic visual tracking. *Int. Symp. on Intelligent Control.*, pp. 320–325, August 1993.
[21] P.J. Rousseeuw, A.M. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, New York, 1987.
[22] G. Simon, M.-O. Berger. Pose estimation for planar structures. *IEEE Computer Graphics and Applications*, 22(6):46–53, November 2002.
[23] T. Tommasini, A. Fusiello, E. Trucco, V. Roberto. Making good features track better. *CVPR*, pp. 178–183, Santa Barbara, June 1998.
[24] Z. Zhang, R. Deriche, O. Faugeras, Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, October 1995.