Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Improvements in robust 2D visual servoing

Éric Marchand, Andrew Comport, François Chaumette
IRISA - INRIA Rennes
Campus de Beaulieu, 35042 Rennes Cedex, France
E-Mail: Firstname.Name@irisa.fr

***Abstract*— A fundamental step towards broadening the use of real world image-based visual servoing is to deal with the important issues of reliability and robustness. In order to address this issue, a closed loop control law is proposed that simultaneously accomplishes a visual servoing task and is robust to a general class of image processing errors. This is achieved with the application of widely accepted statistical techniques of robust M-estimation. Furthermore improvement have been added in the weight computation process: memory, initialization. Indeed, when the error between current visual features and desired ones are large, which occurs when large robot displacement are required, M-estimator may not detect outliers. To address this point, the method we propose to initialize the confidence in each feature is based on the LMedS estimators. Experimental results are presented which demonstrate visual servoing tasks which resist severe outlier contamination.**

I. Introduction

Visual servoing is targeted at controlling the movement of robotic systems by exploiting image sensor information. A general task being to move an end-effector to a certain pose with respect to particular objects or features in the image. This is known to be a very efficient method for positioning tasks [6]. However, its efficiency is subject to varying degrees of error. The efficiency of visual servoing relies on correspondences between the position of tracked visual features in the current image and their desired positions in the desired image. These correspondences are typically exploited in the form of a image error to be minimized. If these correspondences contain errors then visual servoing usually fails or convergences upon an imprecise position.

Other sources 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 [11], [7], [1] and feature selection methods [9]. These approaches provide a robust input estimate to a control loop and as such treats outlier rejection, in an image processing step, prior to the visual servoing task (see Figure 1a). It is clear that handling all the potential sources of error by analytical classification is a very complex and difficult task. In this paper the problem of *statistically robust* visual servoing is implemented directly at a transient control law level (see Figure 1b). In order to represent all the possible external sources of error the correspondences may contain, a statistical measure of position uncertainty is sought.

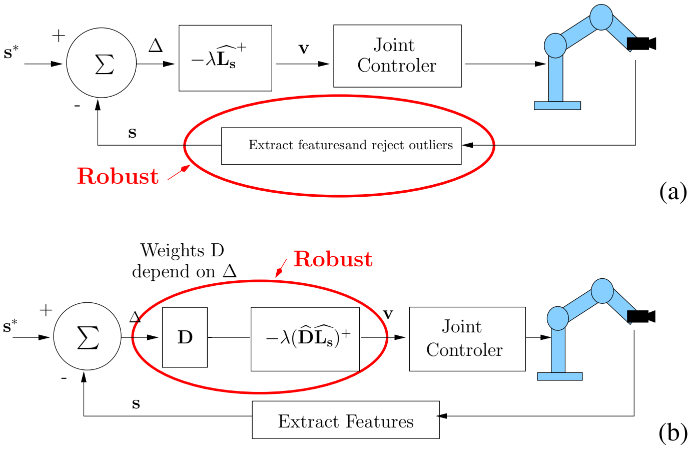

s*
+
Σ
-
Δ
$-\lambda\widehat{\mathbf{L_s}}^+$
v
Joint Controler
s
Extract featuresand reject outliers
Robust
(a)
Weights D depend on Δ
Robust
s*
+
Σ
-
Δ
D
$-\lambda(\mathbf{D}\widehat{\mathbf{L_s}})^+$
v
Joint Controler
s
Extract Features
(b)

Fig. 1. (a) Traditional outlier rejection (b) New proposed control law (see Section II-B for details)

Let us note here that this paper is a sequel of a recent [2]. In this initial paper robust M-estimators [5] are employed because they give a solid statistical basis for detailed analysis and have even been considered to be a unifying banner for these estimation techniques [8]. The estimation of the standard deviation of the inlier data (scale) using the Median Absolute Deviation (MAD) means that no tuning is required. Furthermore, formulation in terms of an Iteratively Re-weighted Least Square (IRLS) allows simple integration directly into the visual control law. We had presented the details of the control law (including stability results) and compared various M-estimators (Huber, Cauchy, Tukey). Experimental results were presented on both simple images to show the validity of the control law and real images where points (detected using a Harris detector) were tracked using SSD trackers. However, though results were satisfactory a problem remains at the beginning of the positioning task when the error between current and desired position of the visual feature are large. In that case, outliers may not be detected and the robot trajectory may not be perfectly adequate. We address this problem in this paper using a LMedS-like method [10]. In addition we also consider a memory process on the weights in order to smooth the trajectory. New results will be presented.

Following an introduction to the method, a robust control scheme is recalled in Section II-B. This is achieved by introducing a weight matrix in the error minimization. We present, in Section II-C how to use M-estimators to compute the weights which reflect a confidence in each feature in the image. In Section II-D, a new method to initialized weights

0-7803-8232-3/04/$17.00 ©2004 IEEE

based on the LMedS approach is presented and finally in Section II-E we show how to combine both techniques to obtain satisfactory camera trajectory. Experimental results are presented in Section III.

II. Robust Visual Servoing

A. Overview and motivations

The goal of classical visual servoing [3], [6] is essentially to minimize the error Δ between a set of image features $\mathbf{s}(\mathbf{r})$, that depends of the camera pose $\mathbf{r}$, and a set of desired image features $\mathbf{s}^*$:

$$\Delta = \mathbf{s}(\mathbf{r}) - \mathbf{s}^*. \tag{1}$$

The camera then has to reach the pose $\mathbf{r_d}$ that minimizes this error.

However, as stated in the previous section, considering that $\mathbf{s}(\mathbf{r})$ is computed (from the image) with a sufficient accuracy is an important assumption. The control law that performs Δ minimization is usually handled using a least square approach [3], [6]. However when the data contains outliers, such a classical approach is no longer efficient.

A solution to handle this problem is to perform a robust minimization. M-estimators can be considered as a more general form of Maximum Likelihood Estimators (MLE) [5] because they permit the use of different minimization functions not necessarily corresponding to normally distributed data. Many functions have been presented in the literature which allow uncertain measures to be less likely considered and in some cases completely rejected. In the following subsections ρ is the objective function considered. The metric function to be minimized is modified to reduce the sensitivity to outliers. The robust optimization problem is then given by:

$$\Delta_{\mathcal{R}} = \rho\Big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\Big) \tag{2}$$

where $\rho(u)$ is a robust function [5] that grows subquadratically and is monotonically nondecreasing with increasing $|u|$.

Considering a robust minimization will allow to compute the confidence in each feature. However this confidence is a statistic that depend only on the error $\mathbf{s}(\mathbf{r}) - \mathbf{s}^*$. When this error is large it may be difficult to detect outliers. Indeed, the error $\Delta_i = s_i + \epsilon_i - s_i^*$ (where ϵ_i is an "aberration" due to imprecision in data extraction) is not statistically significant wrt. to the other errors Δ_j. Since such important errors occurs mainly at the beginning of the positioning task is necessary at this point to provide a good initialization of the confidence in each feature.

To embed a robust minimization in visual servoing, a modification of the control law is required to allow outlier rejection. The new control law is given in the next subsection while the weight computation method is presented in Section II-C (M-estimators), II-D (LMedS-based initialization) and II-E (weights fusion).

B. Robust Control Law

The objective of the control scheme is to minimize the objective function given in equation (2). This new objective is incorporated into the control law in the form of a weight, which is given to specify a confidence in each feature location. Thus, the error to be minimized is given by:

$$\mathbf{e} = \mathbf{D}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*), \tag{3}$$

where $\mathbf{D}$ is a diagonal weighting matrix given by

$$\mathbf{D} = diag(w_1, \dots, w_k)$$

The computation of weights w_i are described in the next subsections.

A simple control law can be designed to try to ensure an exponential decoupled decrease of $\mathbf{e}$ around the desired position $\mathbf{s}^*$ (see [2] for more details). We obtain the following control law given by (see Figure 1b):

$$\mathbf{v} = -\lambda(\widehat{\mathbf{D}}\widehat{\mathbf{L_s}})^{+}\mathbf{D}\big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big), \tag{4}$$

where $\mathbf{v}$ is the camera velocity, $\widehat{\mathbf{D}}$ a chosen model for $\mathbf{D}$ and $\widehat{\mathbf{L_s}}$ is a model or an approximation of the real interaction matrix $\mathbf{L_s}$ related to $\mathbf{s}$ ($\mathbf{L_s}$ links the camera motion $\mathbf{v}$ to the velocity of $\mathbf{s}$ in the image: $\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v}$ [6], [3]). This matrix depends on the value of the image features $\mathbf{s}$ and their corresponding depth $\mathbf{Z_d}$ in the scene.

It is classic in visual servoing to consider a constant Jacobian using the desired depth $\mathbf{Z_d}$ and the desired value of the features $\mathbf{s}^*$. In our case we can similarly define $(\widehat{\mathbf{D}}\widehat{\mathbf{L_s}})^{+}$ as:

$$(\widehat{\mathbf{D}}\widehat{\mathbf{L_s}})^{+} = \mathbf{L_s^+}(\mathbf{s}^*, \mathbf{Z_d}), \tag{5}$$

Let us note that in [2], we have demonstrated the local stability of such system around $\mathbf{s}^*$. Of course it is necessary to ensure that a sufficient number of features will not be rejected so that $\mathbf{DL_s}$ is always of full rank (6 to control the 6 dof of the robot). In fact, redundant features have to be used in $\mathbf{s}$ and our approach can thus not be applied for position-based or 2D 1/2 visual servoing since only six features are used in these methods.

C. Computing the weights

The weights w_i, which represent the different elements of the $\mathbf{D}$ matrix and reflect the confidence of each feature, are usually given by [5]:

$$w_i = \frac{\psi(\delta_i/\sigma)}{\delta_i/\sigma} \tag{6}$$

where $\psi\Big(\delta_i/\sigma\Big) = \frac{\partial\rho\big(\delta_i/\sigma\big)}{\partial\mathbf{r}}$ (ψ is the M-estimate and is also called the influence function) and δ_i is the normalized residue given by $\delta_i = \Delta_i - Med(\Delta)$ (where $Med(\Delta)$ corresponds to the median value taken across all the residues).

In [2] we considered two influence functions that exist in the literature: Huber's monotone function and Tukey's hard re-descending function [5]. We showed that, since Tukey's function allows to completely reject outliers and gives them a zero weight, it is more interesting in visual servoing so that detected outliers have no effect on the robot motion. The corresponding influence function is given by:

$$\psi(u) = \begin{cases} u(b^2 - u^2)^2 & , \text{ if } |u| \leq b \\ 0 & , \text{else} \end{cases} \quad (7)$$

where the proportionality factor for Tukey's function is $b = 4.6851$ which represents 95% efficiency in the case of Gaussian Noise [4].

The standard deviation of the inlier data or scale σ which appears in (6) is a robust estimate of the standard deviation of the good data and is at the heart of the robustness of the function. In visual servoing this estimate of the scale can vary dramatically during convergence. To avoid defining the scale as a tuning variable we use a robust statistic: the Median Absolute Deviation (MAD), given by:

$$\widehat{\sigma} = \frac{1}{\Phi^{-1}(0.75)} Med_i(|\delta_i - Med_j(\delta_j)|). \quad (8)$$

where $\Phi(.)$ is the cumulative normal distribution function and $\frac{1}{\Phi^{-1}(0.75)} = 1.48$ represents one standard deviation of the normal distribution.

D. Weights initialization using the LMedS approach

When error $\mathbf{s} - \mathbf{s}^*$ is large, typically at the beginning of the positioning task, the error due to a misstracking or to a matching problem may not be statistically significant wrt. the expected error. In that case, if some outliers are not detected as such, corresponding weights are not equal to zero and the robot trajectory can be strongly perturbed. Therefore it is important to detect the points that are likely to be outliers prior the beginning of the servo process and to initialize adequately the weights.

The Least-Median-of-Squares (LMedS) [10] method estimates the parameters by solving the non-linear minimization problem:

$$\min \underset{i=1,\dots n}{\text{med}} \; r_i^2$$

where r_i is the residual for each available measure and n is the number of measure.

In our case the camera velocity $\mathbf{v}$ is obtained by solving the following linear system: $\mathbf{L_s v} = -(\mathbf{s} - \mathbf{s}^*)$, the residual r_i is given by:

$$r_i^2 = \left(\mathbf{L}_{s_i}\mathbf{v} + (s_i - s_i^*)\right)^2$$

where $\mathbf{L}_{s_i}$, s_i and s_i^* are, respectively, the i-th line of the interaction matrix $\mathbf{L_s}$ and of the vector $\mathbf{s}$ and $\mathbf{s}^*$.

Unlike M-estimators, the LMedS method cannot be reduced to a weighted least squares problem (and this is why we do not use it inside the control law but only for the initialization) and cannot be solved analytically. It must be solved by a search in the space of possible estimates generated from the data. This space is usually very large. In our case if we have n independent visual features its size is given by $\binom{n}{k}$ where k is the minimal number of features that allows to perform a positionning task ($6 \leq k < n$).

The algorithm described below enables a robust detection of the outliers within the whole set of features. Given n features $s_i, i = 1 \dots n$:

1) draw m subsamples $\mathbf{s}_J, J = 1 \dots m$ of k independent visual features. The maximum number of subsamples is $m_{\max} = \binom{n}{k}$, therefore if n is large $m_{\max}$ may be very large and a Monte Carlo technique may be used to draw the m subsamples (see note below).
2) for each subsample J, we compute the camera velocity $\mathbf{v}_J$ according to:

$$\mathbf{v}_J = -\mathbf{L}_{\mathbf{s}_J}^{+}(\mathbf{s}_J - \mathbf{s}_J^*)$$

3) For each $\mathbf{v}_J$, we determine the median of square residuals, denoted M_J, with respect to the whole set of features, that is:

$$M_J = \underset{i=1\dots n}{\text{med}} \left(\mathbf{L}_{s_i}\mathbf{v}_J + (s_i - s_i^*)\right)^2$$

4) We retain the value M^* that is minimal among all m M_J's. The corresponding $\mathbf{v}_J$ could be also of interest to control the robot, but we prefer to use a weighted control as described in Section II-B for such purpose[1]. M^* will now be used to detect outliers.

LMedS must be carefully designed to detect and remove outliers. Rousseeuw [10] proposes to assign a binary weight to each feature according to:

$$w_i = \begin{cases} 1 & \text{if } |r_i| \leq 2.5\widehat{\sigma} \\ 0 & \text{else} \end{cases} \quad (9)$$

where $\widehat{\sigma}$ is (as in eq (8)) a scale estimate defined by the robust statistic given by (see [10] for details):

$$\widehat{\sigma} = 1.4826(1 + 5/(n-k))\sqrt{M^*} \quad (10)$$

Note dealing with an important number of data: When n increases the number of possible subsamples increases drastically. Assuming that the whole set of features contains up to ε outliers (e.g., $\varepsilon = 50\%$), the probability that at least one of the m subsamples contains only inliers is given by [10]:

$$P = 1 - [1 - (1 - \varepsilon)^k]^m$$

If we require a good probability of detection ($P \simeq 1$), m can be computed given P, m and ε:

$$m = \frac{\log(1 - P)}{\log[1 - (1 - \varepsilon)^k]}$$

For example, when $n = 20$ and $k = 6$ an exhaustive search leads to $m_{max} = \binom{20}{6} = 38760$ subsamples where as, with $P = 0.99$ and $\varepsilon = 50\%$, only 293 subsamples are required. These m sample can be drawned using a Monte Carlo technique.

[1] This is due to two reasons. The former is related to the complexity of this algorithm in term of computation. The latter is due to the fact that, if $\mathbf{v_j}$ is considered to control the manipulator, only k features are then considered regardless of the number of inliers. Data redundancy that usually improves the robustness wrt. image noise would not be considered

E. Full weights computation

Though not very complex, the LMedS-based outliers rejection algorithm requires some processing time that is not yet compatible with a 25Hz loop. That is why we consider it only for initialization. If we denote w^{tukey} the weights computed from (6) and w^{LMedS} the weights obtained from (9), a global weight can be defined as:

$$w_i = (1 - \alpha_t)w_i^{tukey} + \alpha_t w_i^{LMedS} \qquad (11)$$

where $\alpha = 1 - exp(-\beta_1 \|\mathbf{s} - \mathbf{s}^*\|)$. When $\mathbf{s} - \mathbf{s}^*$ is large, α is close to 1 and only w^{LMedS} are mainly considered. When the error $\mathbf{s} - \mathbf{s}^*$ decreases, it is easier and faster to detect outliers thanks to the M-estimator and weights w^{tukey} are considered.

Furthermore to smooth the weight evolution (and then the camera trajectory) it is also possible to introduce a memory process:

$$w_i'(t) = \beta_2 w_i(t) + (1 - \beta_2)w_i(t-1) \qquad (12)$$

with $\beta_2 \in [0:1]$ has to be tuned (1 if only the last computed weight $w_i(t)$ has to be considered).

III. Experimental results

The complete implementation of the robust visual servoing task, including tracking and control, was carried out on an experimental test-bed involving a CCD camera mounted on the end effector of a six d.o.f robot. We have considered positioning tasks. From an initial position, the robot has to reach a desired position expressed as a desired position of the object in the image.

a) Visual features and weights computation: Some simple visual servoing experiments are considered that are based on the tracking of a pattern made with twelve white dots. Tracking such a simple pattern allows to validate the efficiency of the new control law. Indeed due to this simplicity, if no noise is artificially introduced in the point matching or in the tracking, a "control-case" is then available. Let us recall that our approach requires some redundancy in image data therefore multiple points (12 in the reported experiments) are extracted in the images.

If n points are considered, $\mathbf{s}$ is a vector defined as $\mathbf{s} = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$ where (x_i, y_i) are the coordinates of the $i - th$ point. Interaction matrix $\mathbf{L_s}$ is a $2n\times$ 6 matrix given by $\mathbf{L_s} = (\mathbf{L_{s}}_1, \ldots, \mathbf{L_{s}}_n)$ with:

$$\mathbf{L_s}_i = \begin{pmatrix} -\frac{1}{Z_i} & 0 & \frac{x_i}{Z_i} & x_i y_i & -(1 + x_i^2) & y_i \\ 0 & -\frac{1}{Z_i} & \frac{y_i}{Z_i} & 1 + y_i^2 & -x_i y_i & -x_i \end{pmatrix}$$

Weights are computed using equation (12) where

$$\forall k = 1 \ldots n, \quad \left\{ \begin{array}{l} \Delta_{2k} = x_k - x_{k_d} \\ \Delta_{2k+1} = y_k - y_{k_d} \end{array} \right.$$

Since weights w_{2k} and w_{2k+1} reflect the confidence we have in the same point, we define elements of the weights matrix $\mathbf{D}$ as

$$\mathbf{D}_{2k,2k} = \mathbf{D}_{2k+1,2k+1} = \min(w_{2k}, w_{2k+1}).$$

b) Experiments: In these experiments, we considered our new visual servoing control law when large errors are introduced (other experiments are reported in [2]). Four experiments have been carried out:

- **[Exp 1]** a reference experiment: a classical visual servoing task using the control law given in (4) with $\mathbf{D} = \mathbf{I}$;
- **[Exp 2]** an experiment with the same control law (still $\mathbf{D} = \mathbf{I}$, that is no robust estimation is considered) but artificial noise has been added in data extraction: an important error was introduced into the extracted coordinates of two points (points 0 and 2 on Figure 2 were voluntarily inverted) ;
- **[Exp 3]** this experiment is similar to [Exp 2] but weights are computed as described by equation (6) using only Tukey M-estimator ;
- **[Exp 4]** in this experiment we extend [Exp 3] by initializing weights using the new approach presented in section II-D ;

As expected whereas the classical 2D visual servoing control law converges successfully toward the desired minimum when no error is introduced [Exp 1], a large error on two points [Exp 2] implies the convergence of the control law toward a local minimum. An error on the final coordinates of each point can be observed (see Figure 2c and plots on Figure 3 [Exp 2]), raw 2). Let us note that in some cases we can observe a complete divergence of the control law.

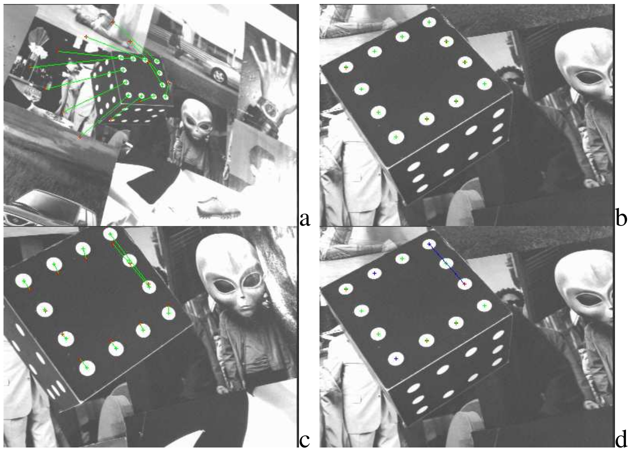
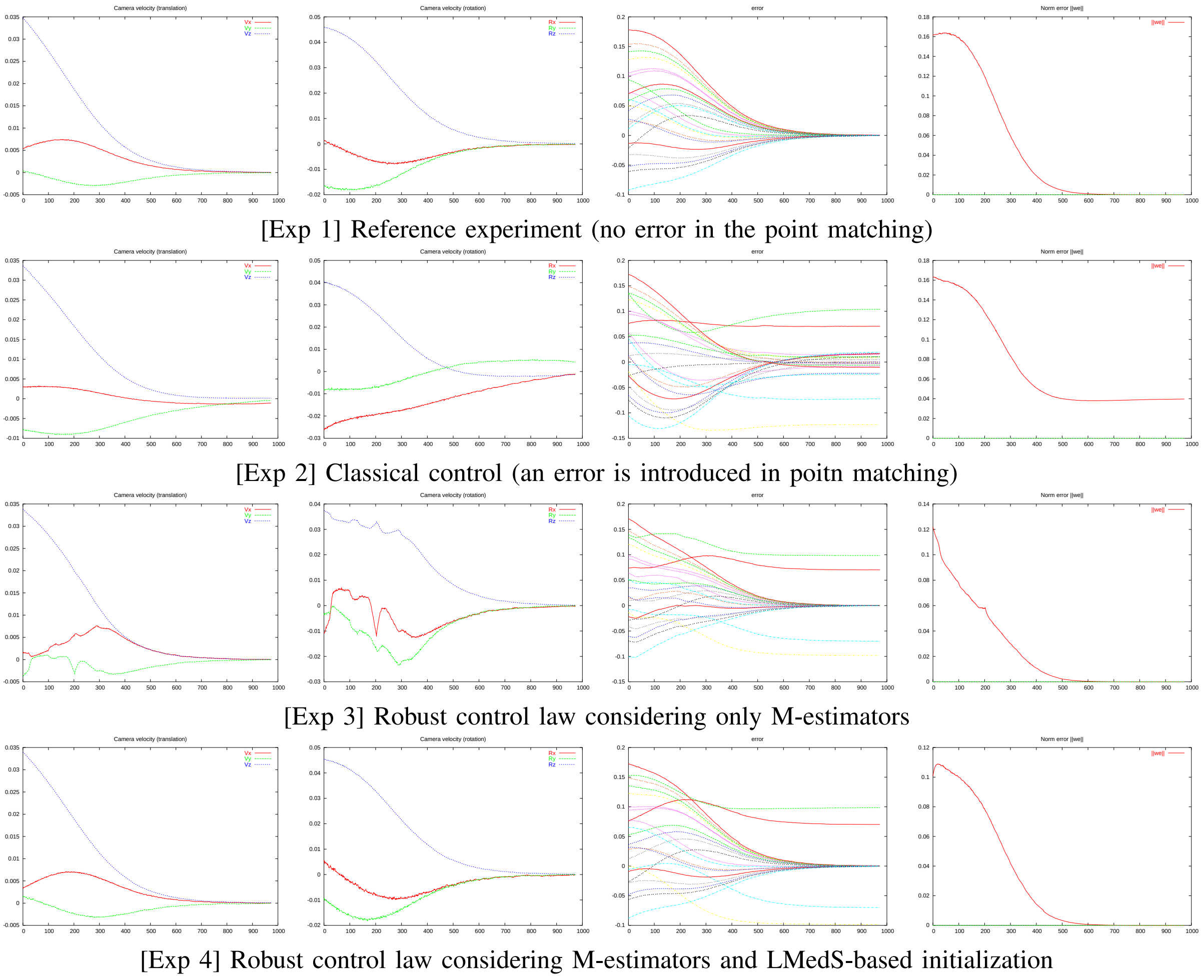


Fig. 2. Result of a positionning task using classical and robuste control law. On the image raw, first image show the initial camera position, the second shows the final camera position for the reference image, the third one shows the final camera view when no robust control law is considered (an error may be observed) and the later image shows the final camera view when the robust control law is considered).

Our new method improves the behavior of the positioning task. Indeed in the two other experiments where a weighting matrix is introduced in the control law [Exp 3-4] the camera reaches its final desired position with very good accuracy despite the errors introduced in the data. In [Exp 3] only M-estimator (Tukey) were considered in the weighting and the positioning precision is similar to the reference experiment (see Table I [Exp 3] vs [Exp 1]). We can note that at the



[Exp 1] Reference experiment (no error in the point matching)

[Exp 2] Classical control (an error is introduced in poitn matching)

[Exp 3] Robust control law considering only M-estimators

[Exp 4] Robust control law considering M-estimators and LMedS-based initialization

Fig. 3. Result of a positioning task using classical and robust control laws. On each raw, plots depict from left to right the translational camera and rotational camera velocities, the errors $\mathbf{s}_i - \mathbf{s}_i^*$ for all the features and finally the weighted error $\|\mathbf{D}(\mathbf{s} - \mathbf{s}^*)\|$

Position			Tx	Ty	Tz	Rx	Ry	Rz	$\|\|T\|\|$	$\|\|R\|\|$
no error	reference	[Exp 1]	-0,110	-0,092	0,026	-0,007	-0,015	0,075	0,146	0,077
large errors	no robust	[Exp 2]	-127,337	-102,909	29,873	-3,140	-15,629	22,696	166,426	27,735
	Mest tukey	[Exp 3]	-0,140	-0,078	-0,122	-0,022	-0,028	0,120	0,202	0,126
	LMedS		-0,133	-0,211	-0,185	-0,018	-0,052	0,129	0,310	0,140
	LMedS+Mest	[Exp 4]	-0,155	0,072	-0,185	-0,017	-0,069	0,070	0,252	0,100

TABLE I

POSITIONING ACCURACY SUMMARY: EACH LINE DISPLAYS THE ERROR (IN CM AND DG) BETWEEN THE DESIRED AND FINAL CAMERA POSITION.

beginning of the experiment, since all the error are large, as expected, the errors introduced in the point extraction $s_i + \delta - s_i^*$ is not statistically significant wrt. the other errors $s_i - s_i^*$ and the values of the related weights are close to 1 at the beginning of the task (as can be seen on Figure 4a that shows the weights evolution). This leads to a discontinuous and noisy control law (see Figure 3 [Exp 3]). Furthermore the camera trajectory is very different from the "reference" trajectory (see Figure 5).

When the new weight initialization process is considered [Exp 4], it is possible to detect outliers before the beginning of the positioning task since it provides a better detection than the M-estimator for large error. However, if this approach is more efficient for large errors, when the errors decrease M-estimator should be considered. Indeed LMedS are very conservative and some inliers point may be considered as outliers (in this experiment two other points are initially considered as outliers). When the error decreases, M-estimators become more efficient and weight of outliers remains null while weight of inliers

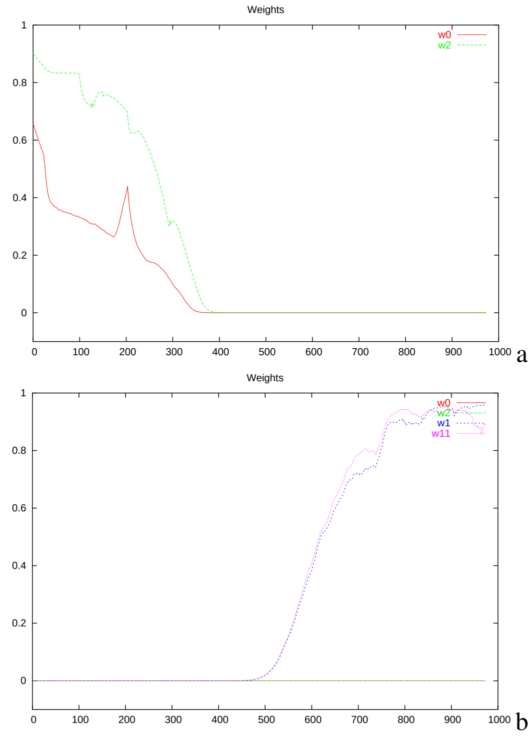


Fig. 4. Weights evolution for point detected as outliers (a) [Exp 3] only M-estimators (b) [Exp 5] LMedS and M-estimators.

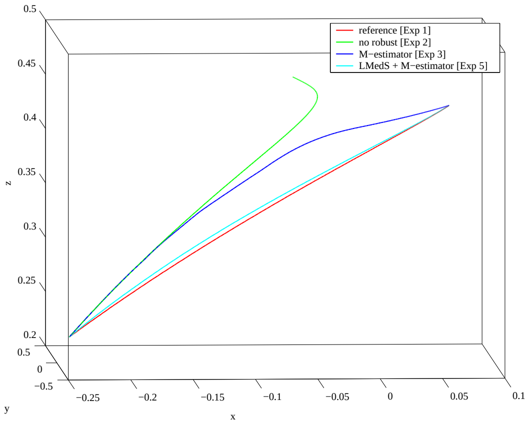


Fig. 5. Camera trajectory for important matching errors.

tends toward 1 (see Figure 4b) since LMedS initialization is "forgotten" according to equation (11). As can be seen on Figure 4b, final weights of outliers are always close to zero which implies that the evolution of the camera velocity (see Figure 3 [Exp 1-4]) as well as its trajectory (on Figure 5) may favorably be compared with the reference experiment.

In the last experiment (see Figure 6) we show that our approach does not depend on the target geometry and is also suitable for non-planar scene (here 18 non coplanar points have been used). In addition we also consider more outliers: two sets of points where inverted (see the blue lines on figure 6 and the white patch in the corner simulates an error in the extraction of the cog of another point. Despite these multiples errors and thanks to the robust control law, the expected minimum is reached.

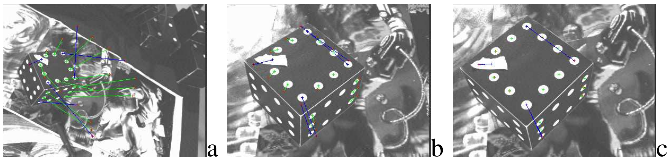


Fig. 6. Similar experiment but a 3D pattern is considered

IV. Conclusion

A novel visual servoing method has been proposed that rejects errors in feature extraction, tracking and matching at the transient control law level. To achieve this goal, a robust M-estimation was integrated directly via an iteratively re-weighted least squares method. Previous visual servoing methods have only considered outlier rejection in the image processing step. Extending the method proposed in [2], we considered the problem of weights initialization in order to detect earlier aberrant data. The resulting control is now more continuous and the robot trajectory is now similar to the reference trajectory (with no aberrant data). Experimental results show the efficiency of the approach for a positioning task on case-study examples for planar and non-planar objects. In both cases a great improvement in the positioning accuracy has been observed wrt. a non robust control law and a better trajectory is obtained wrt. the use of a robust control law considering only M-estimator as proposed in our earlier work.

References

[1] A. Comport, E. Marchand, and F. Chaumette. A real-time tracker for markerless augmented reality. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03*, pages 36–45, Tokyo, Japan, October 2003.

[2] A. Comport, M. Pressigout, E. Marchand, and F. Chaumette. A visual servoing control law that is robust to image outliers. In *IEEE Int. Conf. on Intelligent Robots and Systems, IROS'03*, volume 1, pages 492–497, Las Vegas, Nevada, October 2003.

[3] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[4] P.-W. Holland and R.-E. Welsch. Robust regression using iteratively reweighted least-squares. *Comm. Statist. Theory Methods*, A6:813–827, 1977.

[5] P.-J. Huber. *Robust Statistics*. Wiler, New York, 1981.

[6] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.

[7] D. Kragic and H. Christensen. Cue integration for visual servoing. *IEEE Trans. on Robotics and Automation*, 17(1):19–26, February 2001.

[8] P. Meer, C. V. Stewart, and D. E. Tyler. Robust computer vision: An interdisciplinary challenge. *Computer Vision and Image Understanding: CVIU*, 78(1):1–7, 2000.

[9] N. P. Papanikolopoulos and P. K Khosla. Selection of features and evaluation of visual measurements for 3-d robotic visual tracking. *Int. Symp. on Intelligent Control.*, pages 320–325, August 1993.

[10] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, New York, 1987.

[11] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto. Making good features track better. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 178–183, Santa Barbara, USA, June 1998.