

Méthodes robustes d'estimation pour la vision robotique

Ezio MALIS
Projet ICARE
INRIA Sophia-Antipolis
Ezio.Malis@inria.fr

Éric MARCHAND
Projet Lagadic
IRISA-INRIA Rennes
Eric.Marchand@irisa.fr

Résumé

L'objectif de cet article est de présenter un état de l'art des méthodes d'estimation robustes utilisées en vision par ordinateur, avec une attention particulière aux applications robotiques. Dans ce contexte particulier, les contraintes dues au temps de calcul doivent être prises en compte pour le choix des algorithmes d'estimation. Parmi les nombreuses techniques qui ont été proposées dans la littérature pour obtenir des estimations robustes, on peut citer, dans être exhaustif, la transformée de Hough, RANSAC (Random Sample Consensus), les LMS (Least Median of Squares), les M-estimateurs, etc. Dans cet article nous décrirons les différentes méthodologies en prenant comme exemple des applications robotiques.

Mots Clef

Robustesse, optimisation, estimation de paramètres, vision robotique

1 Introduction

L'objectif de cet article est de présenter un état de l'art des méthodes robustes d'estimation utilisées en vision par ordinateur, avec une attention particulière aux applications robotiques. Dans ce contexte particulier, les contraintes dues au temps de calcul doivent être prises en compte pour le choix des algorithmes d'estimation. Nous adoptons la définition de robustesse suivante: un algorithme d'estimation est dit "robuste" s'il garde ses propriétés malgré les incertitudes sur le modèle, les erreurs de mesure et les changements de l'environnement.

Le fait que l'information visuelle issue de l'image doive être calculée avec une précision suffisante est une hypothèse importante. Si les caméras fournissent une informations de bas niveau extrêmement riche sur l'environnement, il est extrêmement difficile d'extraire l'information, paramètres de haut niveau, pertinente pour résoudre tel ou tel problème de vision robotique. À partir de ces informations et un modèle du processus d'acquisition, il est possible d'estimer les paramètres recherchés. Ces informations de haut niveau peuvent prendre différentes formes en fonction du problème considérée : information de profondeur, position de la caméra par rapport à la scène, déplacement de la caméra ou du robot, paramètres intrinsèques, etc. Pour de nombreuses applications, des modèles pho-

tométriques et géométriques simples peuvent suffire. La géométrie projective est, par exemple, un outils mathématique très bien adaptée à la modélisation de la géométrie de l'environnement et du processus d'acquisition des caméras. Toutefois, des lors que l'on se confronte à des images réelles (et particulièrement dans les applications robotiques en environnement extérieur) la modélisation devient inexacte et des algorithmes robustes sont nécessaires. De plus, des mesures aberrantes peuvent être provoquées par un changement de l'éclairage ou par une occultation. Deux groupes principaux d'algorithmes robustes permettent de prendre en compte ces données aberrantes (*outliers*). La première approche consiste à détecter les outliers avant de procéder à l'estimation des paramètres. Les approches de ce type les plus classiquement utilisées sont la transformée de Hough et l'algorithme Ransac (Random Sample Consensus) [13]. Ce dernier algorithme consiste à estimer les paramètres recherchés avec le minimum de mesures nécessaires puis à vérifier si d'autres mesures confirment cette première estimation. Si un consensus est obtenu, l'estimation est retenue. La seconde approche permet de résoudre simultanément le problème de la détection des outliers et de l'estimation (e.g., LMS et LTS [35], M-estimateurs, L-Estimateurs ou R-Estimateurs [21]). Ces techniques visent à redéfinir la fonction d'objectif à minimiser afin que le minimum global de la fonctionnelle ne soit pas affecté par les données aberrantes. Ces approches cherchent par ailleurs à estimer de façon robuste l'écart type des "bonnes" mesures ou des mesures non aberrantes. La caractéristique principale d'une méthode robuste est son "point de rupture" ("breakdown point" en anglais). Le point de rupture est le pourcentage de mesure aberrante qui mettent en défaut l'algorithme. À titre d'exemple, l'algorithme des moindres carrés a un point de rupture de 0% car il suffit d'une seule mesure aberrante pour obtenir une estimation fautive des paramètres. D'autres caractéristiques très importantes dans les applications robotiques sont la complexité algorithmique et la vitesse de convergence des algorithmes. Ces deux facteurs différents se traduisent en une diminution de la fréquence des boucles de commande. Nous verrons qu'il y a un compromis entre la vitesse de convergence et le point de rupture des algorithmes. Pour illustrer ces techniques d'estimation nous considérerons par la suite trois problèmes classiques en vision robotique: le calcul de pose, le suivi dans une séquence

d'images et l'asservissement visuel.

2 Estimation de paramètres

Le problème que nous cherchons à résoudre est celui de l'estimation d'un certain nombre de paramètres à partir d'une ou plusieurs images. Nous considérons ici des applications robotiques. Pour cette raison nous tiendrons compte des performances en temps de calcul des algorithmes considérés.

2.1 Les moindres carrés

Supposons que nous disposions d'un modèle des signaux $s_k(\mathbf{x})$, qui changent en fonction d'un certain nombre de paramètres inconnus \mathbf{x} , et d'une mesure $s_k^* = s_k(\mathbf{x}^*)$ de ces signaux. Nous définissons le résidu $r_k = s_k(\mathbf{x}) - s_k^*$ et $\mathbf{r}(\mathbf{x})$ le vecteur ($n \times 1$) contenant tous les résidus ordonnés par ordre croissant, de telle manière que $r_1^2(\mathbf{x}) \leq r_2^2(\mathbf{x}) \leq \dots \leq r_n^2(\mathbf{x})$. Notre problème est de retrouver la valeur \mathbf{x}^* des paramètres à partir des signaux mesurés. La méthode la plus utilisée, qui remonte à Gauss et Legendre [35], est la méthode des moindres carrés LS (*Least Squares*). On cherche à minimiser la fonction de coût suivante:

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

La solution de ce problème d'optimisation peut être obtenue de manière itérative en partant d'une estimation initiale \mathbf{x}_0 des paramètres.

2.2 Une exemple très simple

Nous considérons dans cet article un exemple très simple qui nous servira de "cas d'école" afin de comparer les différentes méthodes d'estimation. Supposons $n = 100$ points d'intérêts vus dans deux images différentes.

Supposons que la caméra a effectué une translation de 10 mm suivant l'axe \vec{x} . Notre objectif est d'estimer le déplacement de la caméra à partir des coordonnées des points dans les deux images et la connaissance des profondeurs dans le repère d'une des deux images. En effet, l'équation qui lie les coordonnées normalisées des points dans l'image est la suivante:

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

Nous avons alors un seul paramètre à estimer ($\mathbf{x} = t_x$) et nous pouvons visualiser plus facilement les fonctions de coût des différentes méthodes.

Dans ce cas d'école la fonction de coût est quadratique et son minimum est $x^* = 10$ mm. Le problème LS est donc facilement résolu pour n'importe quelle condition initiale. Toutefois, nous considérons ici dans toutes les exemples que la condition initiale est $\mathbf{x}_0 = 0$.

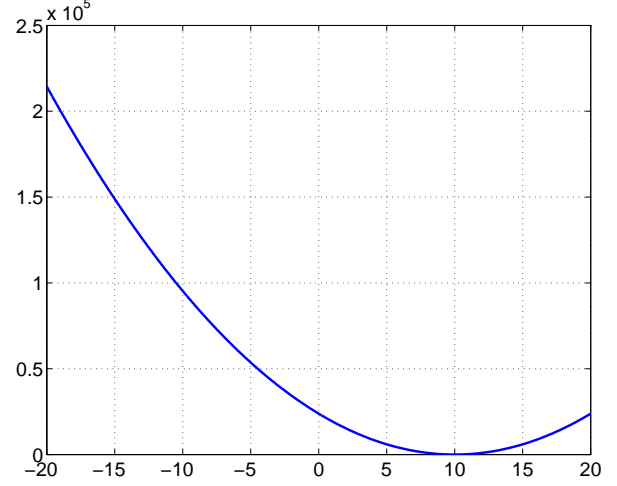


FIG. 1 – Fonction de coût des moindres carrés (LS).

2.3 Problèmes avec des mesures aberrantes

En présence d'un certain nombre de mesures aberrantes (erreur de mise en correspondance, occultations, changement d'éclairage,...) la fonction de coût du LS est modifiée de telle manière que le minimum ne correspond plus à la valeur réelle des paramètres à estimer. Par exemple, sur la figure 2a on peut voir la fonction de coût du LS quand l'ensemble des mesures est corrompu de manière aléatoire avec 20 % d'"outliers". Le minimum de la fonction de coût est $x = 6$ mm au lieu de $x = 10$ mm. De manière similaire dans la figure 2b on peut voir la fonction de coût du LS quand l'ensemble des mesures est corrompu de manière aléatoire avec 40 % d'"outliers". Le minimum de la fonction de coût est $x = 4$ mm au lieu de $x = 10$ mm. Dans ces graphiques, les points verts représentent la valeur initiale de x et les points rouges représentent la valeur trouvée après minimisation. La méthode LS est très peu robuste aux mesures aberrantes. Son "breakdown point" est de 0% car il suffit d'une seule mesure aberrante pour fausser le minimum. Dans certains cas, le minimum peut changer de manière considérable si l'"outliers" a une influence très importante (on parle alors d'effet de levier [35]). Les méthodes d'estimation robustes que nous allons décrire à présent sont capables d'estimer la valeur correcte des paramètres malgré la présence de ces mesures aberrantes.

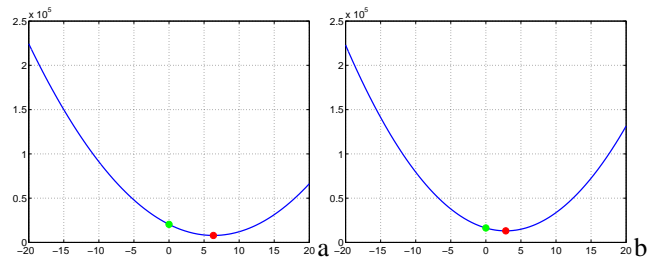


FIG. 2 – Fonction de coût du LS en présence de (a) 20% et (b) 40% de mesures aberrantes.

3 Méthodes robustes

Les moindres carrés ne sont pas robustes car la fonction de coût peut grandir indéfiniment et des mesures aberrantes peuvent devenir prépondérants par rapport aux mesures vérifiant le modèle correcte (*inliers*). Les méthodes robustes d'estimation modifient la fonction de coût de manière à limiter l'influence des résidus les plus importants. La conséquence principale est un ralentissement de la vitesse de convergence des algorithmes d'optimisation. En effet, il est très difficile (voir impossible) de distinguer dans un premier temps entre les mesures aberrantes et les mesures correctes. Donc, des résidus correctes importants sont aussi initialement filtrés ce qui comporte une réduction de la vitesse de convergence. Le cas le plus difficile à résoudre se présente quand des résidus faibles mais aberrants déplacent le minimum de la fonction de coût.

3.1 LMS (Least Median of Squares)

La méthode robuste LMS [34, 35] minimise la fonction de coût suivante:

$$C(\mathbf{x}) = \text{median}((r_1^2(\mathbf{x}), r_2^2(\mathbf{x}), \dots, r_n^2(\mathbf{x}))) \quad (2)$$

La médiane est vu comme un estimateur robuste car elle ne considère pas les 50% des résidus les plus importants. Elle a cependant deux inconvénients majeurs:

- la fonction de coût n'est généralement pas différentiable. Des méthodes de minimisation basées sur le gradient de la fonction de coût sont donc très délicates à mettre en œuvre;
- la vitesse de convergence des algorithmes de minimisation peut être extrêmement lente si les résidus sont distribués de telle manière que la médiane a un gradient très faible.

Dans le cas d'école considéré le minimum est correctement localisé même en présence de 20% ou 40% d'*outliers* sur la totalité des mesures. La figure 3 montre l'allure de la fonction de coût dans les deux cas. En partant de $\mathbf{x}_0 = 0$ (le point vert dans la figure) on trouve correctement le minimum (le point rouge dans la figure). Cette méthode a un "breakdown point" de 50% (le plus haut possible) mais aussi une convergence extrêmement lente. Afin d'augmenter la vitesse de convergence plusieurs techniques sont possibles. L'une d'elle consiste à effectuer des tirages aléatoires d'un sous ensemble des mesures de manière similaire à l'algorithme RANSAC. Une autre possibilité est d'utiliser la méthode LTS.

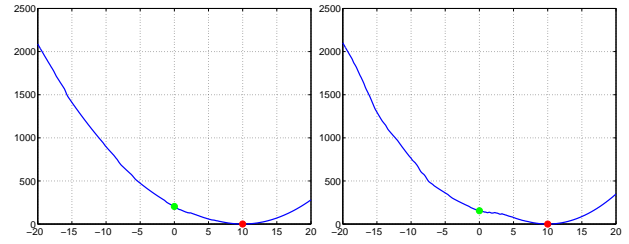


FIG. 3 – Fonction de coût du LMS en présence de (a) 20% et (b) 40% de mesures aberrantes.

3.2 LTS (Least Trimmed Squares)

Afin d'améliorer la vitesse de convergence de la méthode LMS, Rousseeuw [35] a proposé la méthode dite LTS. Elle consiste à minimiser la somme des carrés des q premiers résidus:

$$C(\mathbf{x}) = \sum_{k=1}^q r_k^2(\mathbf{x}) \quad (3)$$

En général, on choisit $q = n/2$ mais une connaissance *a priori* du taux d'*outliers* permet d'optimiser le choix de la valeur de q . La fonction de coût a une allure similaire à celle de la méthode LMS mais le gradient est généralement plus élevé. La contribution des q premiers résidus rends la fonction de coût un peu plus lisse. Toutefois cette fonction reste généralement non différentiable. La figure 4 montre l'allure de la fonction de coût dans les cas précédemment considérés. Le minimum est trouve correctement en présence de 20 % d'*outliers*. Quand le pourcentage d'*outliers* est de 40 % il est toujours possible de trouver le vrai minimum mais un minimum local apparaît dans la fonction de coût. Ce problème, qui peut arriver aussi pour d'autre méthodes robustes (y compris pour le LMS), dépends de la distribution des *outliers* et il est très difficile à analyser. De manière générale, il est *a priori* impossible de prévoir quelle méthode robuste permettra d'éviter la formation de ces minima locaux.

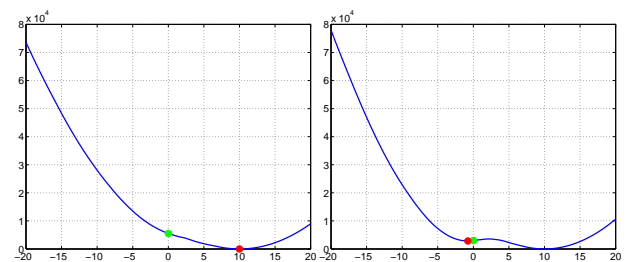


FIG. 4 – Fonction de coût du LTS en présence de (a) 20% et (b) 40% de mesures aberrantes.

3.3 M-estimateurs

Le principe des M-estimateurs est de modifier la fonction de coût du LS en pénalisant les résidus les plus grands. La

fonction de coût s'écrit:

$$C(\mathbf{x}) = \sum_{k=1}^n \rho(r_k(\mathbf{x})) \quad (4)$$

où la fonction ρ est au moins C^0 . Différentes fonction ρ ont été proposées dans la littérature. Le plus utilisées sont celles proposées par Tukey [1] et Huber [21]. Le *break-down point* des M-estimateurs est théoriquement de 0% comme pour le LS. Toutefois, il s'agit du pire des cas et on verra qu'il faut des mesures aberrantes importantes avant de mettre en défaut les M-estimateurs.

Huber. La fonction ρ proposée par Huber est la suivante:

$$\rho(r_k(\mathbf{x})) = \begin{cases} \frac{1}{2}r_k^2(\mathbf{x}) & \text{if } r_k^2(\mathbf{x}) \leq c \\ c(|r_k(\mathbf{x})| - \frac{c}{2}) & \text{if } r_k^2(\mathbf{x}) > c \end{cases} \quad (5)$$

où $c = 1.345\hat{\sigma}$ et $\hat{\sigma}$ est une estimation robuste de l'écart type. En générale on utilise le Mad (*Median Absolute Deviation*):

$$\hat{\sigma} = 1.48\text{median}(|\mathbf{r} - \text{median}(\mathbf{r})|). \quad (6)$$

Dans la fonction ρ proposée par Huber, les résidus les plus faibles sont considérés comme dans un LS alors que les résidus les plus importants sont rapidement limités (mais non annulés). La figure 5 montre l'allure de la fonction de coût dans les cas considérés. Même si théoriquement le *break-down point* est de 0%, la fonction de coût des M-estimateur à son minimum global correctement localisé. Toutefois, quand le pourcentage d'*outliers* est élevé des minima locaux peuvent, là encore, apparaître.

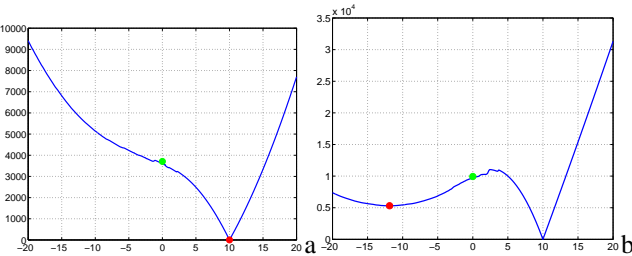


FIG. 5 – Fonction de coût du M-estimateur de Huber en présence de (a) 20% et (b) 40% de mesures aberrantes.

Beaton-Tukey. La fonction ρ proposée par Beaton-Tukey [1] est la suivante:

$$\rho(r_k(\mathbf{x})) = \begin{cases} \frac{c^2}{6} \left[1 - \left(1 - \left(\frac{r}{c} \right)^2 \right)^3 \right] & \text{if } r_k^2(\mathbf{x}) \leq c \\ \frac{c^2}{6} & \text{if } r_k^2(\mathbf{x}) > c \end{cases} \quad (7)$$

où $c = 4.6851\hat{\sigma}$ et où $\hat{\sigma}$ est une estimation de l'écart type du bruit sur les bonnes mesures (voir paragraphe 3.3). Comme dans le cas précédent, dans la fonction ρ proposée par Beaton-Tukey, les résidus le plus faibles sont considérés comme dans un LS alors que les résidus le plus importants sont rapidement annulés (totalement cette fois). La

figure 6 montre des résultats très similaires aux résultats obtenus par le M-estimateur de Huber.

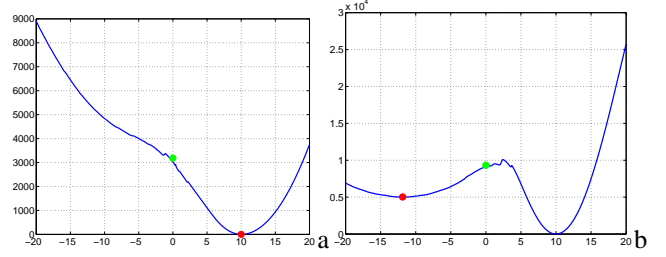


FIG. 6 – Fonction de coût du M-estimateur de Tukey en présence de (a) 20% et (b) 40% de mesures aberrantes.

Algorithme IRLS. En pratique, les M-estimateurs sont implémentés avec l'algorithme des moindres carrés pondérés itérés IRLS (Iteratively Reweighted Least Squares). L'algorithme vise à résoudre le système suivant $\mathbf{D}\mathbf{A}\mathbf{x} = \mathbf{D}\mathbf{b}$ où $\mathbf{D} = \text{diag}(w_1, \dots, w_k)$ est une matrice diagonale. Le calcul du poids w_i associé à chaque mesure représente la confiance que l'on a dans chacune d'elles.

Cet algorithme opère de la façon suivante : estimation des poids en utilisant l'un des multiples critères robustes présentés dans la littérature (Huber, Tukey), estimation de la valeur de \mathbf{x} en résolvant le système précédent, et réitérer jusqu'à convergence.

Les poids w_i , éléments de la matrice \mathbf{D} , reflètent la confiance en chaque primitive et sont définis par [21] :

$$w_i = \frac{\psi(ds)}{ds} \quad (8)$$

où $\psi(ds) = \partial\rho(\delta_i)/\partial\delta_i$, δ_i est le résidu normal donné par $\delta_i = \Delta_i - \text{median}\Delta$ ($\text{median}\Delta$ correspond à la valeur médiane des résidus). Le paramètre σ , qui représente la valeur de l'écart type du bruit sur les "bonnes" mesures, peut varier énormément au cours du processus de minimisation. σ est souvent traitée comme une variable d'ajustement qui est choisie manuellement en fonction d'une application particulière. Il est aussi possible, afin d'améliorer la précision de détection des données aberrantes, d'estimer la valeur de σ parallèlement à la minimisation de l'erreur en utilisant une statistique robuste. Le Mad pour *Median Absolute Deviation* est l'une de ces statistiques :

$$\hat{\sigma} = 1.48\text{Med}_i(|\delta_i - \text{Med}_j(\delta_j)|). \quad (9)$$

4 Méthodes robustes de vote

L'estimation des paramètres avec les méthodes de vote repose sur l'utilisation du minimum de données nécessaires à l'estimation. Chaque estimation, avec un jeu de données particulier, correspond à un "vote" pour les paramètres obtenus. Le jeu de paramètres élu, *i.e.* le plus "voté", est retenu comme résultat de l'estimation.

4.1 Transformée de Hough

La transformée de Hough [20] est une méthode de vote très robuste. La version originale de la méthode proposée par Hough a été modifiée par [11]. Depuis plusieurs variantes ont été proposées [23]. Cette approche repose sur une discrétisation de l'espace des paramètres. On obtient alors des hypercubes dans l'espace d'état auquel sont associés des accumulateurs. Pour un jeu de données de taille minimale, les paramètres recherchés sont estimés et l'accumulateur correspondant de l'hypercube est incrémenté. Ce processus est itéré jusqu'à considérer toutes les combinaisons possibles des données à disposition. L'accumulateur ayant la valeur la plus importante correspond alors à la meilleure estimation des paramètres.

La transformée de Hough est bien adaptée aux problèmes ayant un nombre important de données par rapport au nombre des paramètres à estimer. En effet, si les données et les inconnues sont de taille équivalente il est difficile de trouver un accumulateur prépondérant par rapport aux autres. En plus, dû à la discrétisation et au bruit il est possible que l'optimum soit delocalisé. La transformée de Hough est très robuste car elle effectue une recherche globale et exhaustive. Finalement, cette technique est capable de segmenter les données en plusieurs populations qui vérifient le modèle de référence. Toutefois, la transformée de Hough est très rarement utilisée en vision robotique car pour des problèmes qui nécessitent l'estimation de plus de trois ou quatre paramètres les temps de calculs deviennent prohibitifs.

4.2 RANSAC

La méthode RANSAC [13] (en anglais Random Sample Consensus) est une méthode de vote probabiliste qui a été proposée afin de réduire le temps de calcul des méthodes de votes classiques (comme par exemple la transformée de Hough). À partir d'un sous ensemble minimal de s signaux mesurés il est possible de calculer les paramètres dans une situation non dégénérée. Ensuite, on calcule une fonction la fonction de coût suivante:

$$C(\mathbf{x}) = \sum_{k=1}^n \rho(r_k(\mathbf{x})) \quad (10)$$

ou:

$$\rho(r_k(\mathbf{x})) = \begin{cases} 0 & \text{if } r_k^2(\mathbf{x}) \leq c \\ 1 & \text{if } r_k^2(\mathbf{x}) > c \end{cases} \quad (11)$$

et $c = 2.5 \hat{\sigma}$.

Soit p la probabilité de trouver la bonne solution, s le nombre minimum de signaux nécessaire pour l'estimation des paramètres et r le pourcentage d'inliers. Le nombre m de tirages aléatoires nécessaire pour avoir une probabilité p de retrouver les bons paramètres est :

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

Dans la figure 7 est représentée la fonction de coût du RANSAC. Les barres rouges représentent les résultats d'estimation des paramètres à partir des tirages aléatoires (pour certains tirages on obtient le même jeu de paramètres). On peut voir que pour 20% d'*outliers* 5 tirages aléatoires seulement sont suffisants pour avoir une probabilité de 95% de trouver la bonne solution. Quand le pourcentage d'*outliers* est de 40% 13 tirages sont nécessaires.

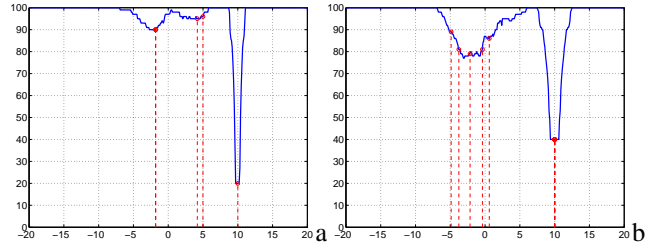


FIG. 7 – Fonction de coût du RANSAC en présence de (a) 20% et (b) 40% de mesures aberrantes.

5 Application à la vision robotique

Un certain nombre de problèmes en vision par ordinateur sont intéressants pour la robotique. Nous considérons ici les exemples suivants:

- suivi d'un objet dans une séquence d'images ;
- localisation de la caméra ;
- asservissement visuel.

5.1 Suivi d'un objet dans l'image.

Les algorithmes d'estimation robuste que nous avons décrit précédemment peuvent être appliqués à un problème classique de la vision robotique : le suivi d'un objet dans une séquence d'images. Si l'on suppose que l'objet est plan le problème de suivi peut être résolu en estimant la matrice d'homographie qui lie deux points homologues dans deux images successives. La méthode utilisée est présentée dans [2]. La matrice d'homographie est de dimension (3×3) et elle est définie à un facteur d'échelle. Seulement 8 paramètres sont donc à estimer. L'estimation de ces paramètres par une technique de moindres carrés donne généralement de très bons résultats même en présence de certaines perturbations. Cependant, dès que des erreurs aberrantes se produisent (comme par exemple une réflexion spéculaire sur la surface de l'objet), l'algorithme de suivi n'est plus capable d'estimer correctement les paramètres. Il est donc souhaitable d'utiliser une méthode d'estimation robuste. La figure 8 montre un exemple où une estimation reposant sur l'utilisation des M-estimateurs. L'objet à suivre dans la séquence d'images est sélectionné dans première image et il est entouré par un rectangle rouge. Dans les images suivantes, l'objet est suivi malgré des très grands changements d'éclairage et de grands déplacements. En utilisant un moindres carrés classique il n'est pas possible de suivre l'objet pendant toute la séquence. À partir de l'homographie estimée il est ensuite possible de commander un



FIG. 8 – Suivi d’un objet dans l’image par estimation robuste d’une transformation homographique. L’objet suivi est entouré en rouge.

robot en utilisant des techniques d’asservissement visuel.

5.2 Localisation d’une caméra.

Le calcul de pose vise à déterminer la position d’une caméra en effectuant un recalage 2D-3D. C’est un problème très anciens en vision par ordinateur (citons les travaux sur le P4P [13] – *Perspective from 3 points* –) et en photogrammétrie [3] mais qui ont suscité et continuent de susciter de nombreuses études. Considérons, pour illustrer ce point, le problème de la localisation 3D à partir de la projection de points. Ce problème de recalage 2D-3D revient à déterminer les paramètres extrinsèques de la caméra, définis par la matrice homogène de changement de repère cM_o , qui minimise l’erreur de reprojction suivante :

$$\Delta = \sum_{i=1}^N (\mathbf{p}_i - \mathbf{K}^c M_o \circ \mathbf{P}_i)^2 \quad (12)$$

où $\circ \mathbf{P}$ représente la position des N points considérés dans un repère lié à la scène (modèle de l’objet), \mathbf{p}_i leur projection dans le plan image et \mathbf{K} est la matrice de projection perspective.

Dans le cas où un faible nombre de primitives est disponible, il existe des solutions purement analytiques à ce problème consistant à résoudre directement le système d’équations non-linéaires issu de l’équation 12 [13, 19, 9]. Par nature, ces problèmes sont non linéaires par rapport aux paramètres de pose mais il existe des solutions linéaires (e.g., [12, 25]) reposant sur la résolution de systèmes linéaires aux moindres carrés pour estimer la pose et éventuellement les paramètres intrinsèques de la caméra. Dans ce cas, l’efficacité de ces approches repose principalement sur la représentation choisie pour la matrice de rotation et des contraintes retenues pour assurer l’orthonormalité de cette matrice. Elles sont cependant extrêmement sensibles aux bruits de mesure. Ces méthodes ne fournissant généralement pas un résultat de très bonne qualité,

d’autres approches comme les techniques de minimisation non-linéaire (e.g., [26, 27, 10, 29, 8, 6]) peuvent alors être considérées.

Nonobstant la méthode de minimisation retenue, le critère 12 peut être optimiser de façon robuste. Historiquement, l’algorithme Ransac [13] développé par Fischler et Bolles au SRI a été initialement décrit dans ce contexte (problème du P3P). Les M-estimateurs ont aussi été largement mis à contribution [17, 10, 6, 37].

La figure 9 montre le résultat d’un suivi 3D en utilisant un algorithme de localisation 3D robuste (M-estimation Tukey) dans chaque image [6]. Cette localisation est utilisé pour réaliser une expérience d’asservissement visuel 2 1/2 D.

Des problèmes similaires sont aussi souvent résolue en utilisant des techniques robuste. Citons par exemple l’estimation d’homographie à partir de point 2D mis en correspondance [18, 14, 33, 38], l’estimation d’un mouvement 2D [31, 16], l’estimation d’une transformation rigide entre deux ensemble de points 3D [14], l’estimation de la matrice essentielle ou fondamentale (RANSAC [13], LMedS [41])

5.3 Asservissement visuel robuste aux mesures aberrantes

Si l’asservissement visuel est très efficace pour réaliser des tâches de positionnement, il apparaît cependant que la précision de positionnement est très sensible aux erreurs inhérentes au processus d’extraction des données. L’efficacité de l’asservissement visuel dépend en effet de la précision de localisation de cette information visuelle mais aussi de la précision de l’appariement entre les valeurs courante et désirée de cette information. Si la mise en correspondance entre les informations visuelles est entachée d’erreur ou si l’estimation de la valeur de s est imprécise, la précision de la tâche de positionnement sera imprécise, voire même, dans certains cas, l’asservissement sera un échec.

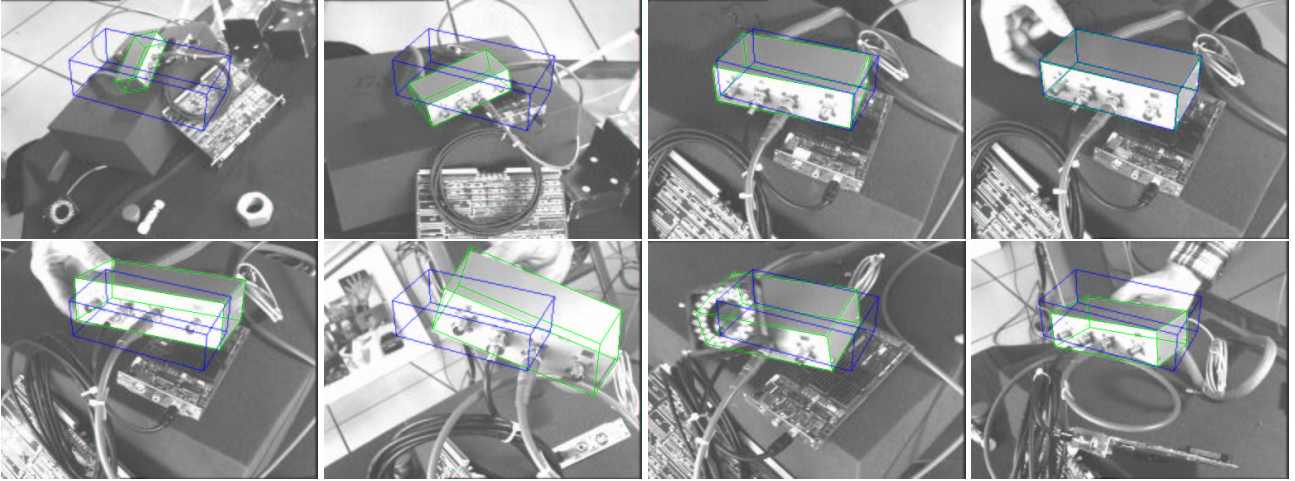


FIG. 9 – Suivi d’un objet pendant une expérience d’asservissement visuel 2D 1/2. L’objet suivi est en vert et sa position désirée dans l’image est en bleu. Les images de la première ligne correspondent à l’étape initiale de positionnement. Dans la suivante, à la fois l’objet et le robot sont en mouvement et subi de multiples occultations correctement traitées par les estimateurs robustes.

Traditionnellement, la robustesse d’une loi de commande est définie par : “*stability results which remain true in the presence of modeling errors or certain classes of disturbance*” [36]. Deux solutions peuvent donc être exhibées pour assurer la robustesse de la loi de commande : la première est de créer un modèle le plus précis possible du système considéré (perturbations potentielles comprises) et la seconde est de traiter (limiter) au mieux les perturbations en travaillant directement sur la commande. Dans le premier cas, il est raisonnable de penser qu’une modélisation et une estimation correcte de l’ensemble des paramètres intrinsèques du système permettent d’améliorer les résultats. En asservissement visuel, ce type d’approche a conduit à modéliser la caméra par un modèle de projection perspective, à disposer d’une formulation analytique de la matrice d’interaction [4] et à estimer en ligne l’information de profondeur présente dans cette matrice [30, 15, 5, 39], etc. D’autres sources d’erreurs proviennent du bruit dans l’extraction des indices visuels, ou d’erreurs de suivi voire d’importantes erreurs de mise en correspondance entre primitives courantes et désirées. La prise en compte de ces erreurs se fait le plus souvent en aval de la loi de commande, c’est-à-dire au niveau de l’extraction des indices visuels (voir figure 10a) : amélioration de la qualité des algorithmes de suivis [40] ou sélection de primitives particulières [32], fusion d’informations redondantes (par des approches de vote ou de consensus [24]).

Les solutions mentionnées dans le paragraphe précédent sont des solutions partielles pouvant prendre en compte certains types d’erreurs bien définis. Une séquence d’images acquises à la cadence vidéo est cependant une source quasi infinie d’erreurs qu’il est impossible de caractériser et de traiter de manière exhaustive. Ceci inclut les problèmes dus au mouvement plus ou moins rapide des objets, aux occultations éventuellement multiples, aux chan-

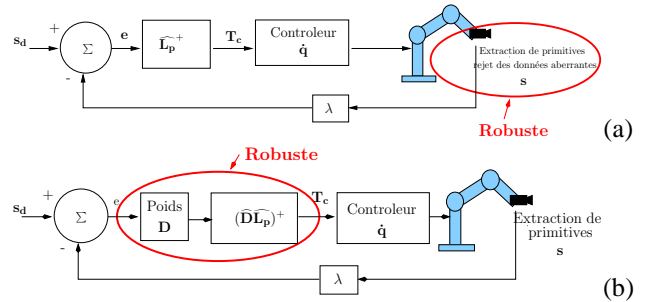


FIG. 10 – (a) Asservissement visuel robuste “classique” : le rejet des données aberrantes se fait dans l’extraction des données, (b) Nouvelle loi de commande : le rejet des données aberrantes se fait dans la loi de commande.

gements d’illumination, etc. Il semble évident qu’établir un catalogue analytique de toutes les sources de perturbations possibles et de proposer une solution pour traiter chacune d’entre elles est un travail complexe voire impossible à réaliser. Nous avons donc décidé d’essayer de limiter l’effet des perturbations potentielles en modifiant la loi de commande. Nous avons en effet considéré le problème de l’asservissement visuel robuste en introduisant directement dans la loi de commande des estimateurs robustes permettant de quantifier la confiance dans chacune des informations visuelles et, si nécessaire, de les rejeter (voir figure 10b). L’incertitude sur chaque primitive est donc modélisée statistiquement, ce qui permet de prendre en compte tout type de perturbations dans l’extraction des données.

Nous considérons la tâche générique qui consiste à déplacer une caméra pour observer un objet à une position donnée dans l’image. Ceci est accompli en minimisant l’erreur Δ entre un état désiré des primitives dans l’image s^* et leur

état courant \mathbf{s} .

En asservissement visuel, la loi de commande qui réalise la minimisation de Δ est traitée habituellement par une approche aux moindres carrés [4, 22]. Cependant, s'il y a des données aberrantes, la réalisation de la tâche sera en échec et une prise en compte explicite de ce problème est nécessaire. La fonction à minimiser est donc modifiée afin de réduire la sensibilité aux données aberrantes. L'erreur à minimiser est alors donnée par :

$$\Delta_{\mathcal{R}} = \sum_{i=1}^N \rho(s_i(\mathbf{r}) - s_i^*)^2, \quad (13)$$

où $\rho(u)$ est une fonction robuste [21]. De façon similaire au problème des moindres carrés pondérés itérés, nous introduisons dans la loi de commande une matrice de pondération, où les poids reflètent la confiance dans chaque primitive visuelle. Nous avons donc une nouvelle loi de commande qui assure une minimisation robuste de Δ définie par l'équation 13. L'erreur à minimiser est donc :

$$\mathbf{e} = \mathbf{D}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*), \quad (14)$$

où \mathbf{D} est une matrice diagonale. Le calcul du poids w_i associé à chaque information visuelle représente la confiance que l'on a dans chacune des informations visuelles. Sans entrer dans les détails (voir [7]), on obtient une loi de commande donnée par :

$$\mathbf{v} = -\lambda(\widehat{\mathbf{D}}\widehat{\mathbf{L}}_{\mathbf{s}})^+ \mathbf{D}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*). \quad (15)$$

où un modèle ou une approximation $\widehat{\mathbf{L}}_{\mathbf{s}}$ de $\mathbf{L}_{\mathbf{s}}$ sont utilisés (un modèle $\widehat{\mathbf{D}}$ de \mathbf{D} peut aussi être considéré).

Les résultats obtenus ont montré l'efficacité d'une telle approche (comme le montrent les résultats de la figure 11. Des résultats plus complets sont données dans [7, 28]). Il reste que l'utilisation d'une telle loi de commande robuste n'est pas incompatible, loin de là, avec un processus efficace d'extraction des données. Une fusion des deux schémas de la figure 10 est non seulement possible mais souhaitable.

6 Conclusion

Dans cet article nous avons passé en revue les méthodes d'estimation robuste les plus utilisées en vision robotique. L'utilisation de ces méthodes est nécessaire afin de réaliser des tâches en environnement réel. Le prix à payer est un temps de calcul un peu plus élevé et une vitesse de convergence réduite. Si les techniques de vote (Hough, Ransac) sont très efficaces, le temps de calcul est souvent trop élevé pour assurer une utilisation des algorithmes de vision à une cadence compatible avec la commande d'un robot. Comme on a pu le voir dans les exemples donnés dans la section précédente l'utilisation de M-estimateurs représente un bon compromis entre robustesse et efficacité algorithmique.

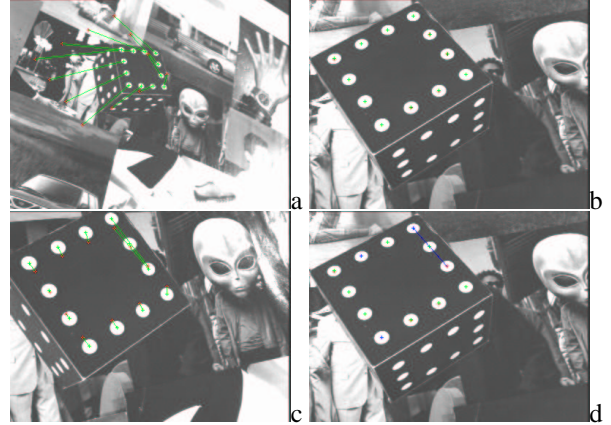


FIG. 11 – Tâche classique de positionnement en considérant une loi de commande classique et une loi de commande robuste. L'image (a) montre l'image initiale acquise avant le début de la tâche. Les trois autres images correspondent aux images acquises à l'issue de la tâche de positionnement : (b) correspond à un positionnement reposant sur une loi de commande classique mais sans données aberrantes (expérience de référence), (c) reprend la même loi de commande classique mais la mise en correspondance entre points courants et désirés est faussée introduisant des données aberrantes (c'est un des cas d'erreurs possibles, mais d'autres cas sont envisageables [7, 28]). Comme on peut s'y attendre la commande converge vers un minimum local, (d) considère la même expérience mais avec une loi de commande robuste. Malgré les données aberrantes, la tâche de positionnement se déroule correctement.

Références

- [1] A.E. Beaton and J.W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16:147–185, 1974.
- [2] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE/RSJ Int. Conf. on Intelligent Robots Systems*, Sendai, Japan, October 2004.
- [3] D.C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 4(2):127–140, March 1971.
- [4] F. Chaumette. Asservissement visuel. In W. Khalil, editor, *La commande des robots manipulateurs*, Traité IC2, chapter 3, pages 105–150. Hermès, 2002.
- [5] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin. Structure from controlled motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(5):492–504, May 1996.
- [6] A.I. Comport, E. Marchand, and F. Chaumette. Efficient model-based tracking for robot vision. *Advanced Robotics*, 2005.
- [7] A.I. 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.
- [8] N. Daucher, M. Dhome, J.T. Lapreste, and G. Rives. Modelled object pose estimation and tracking by monocular vision. In *British Machine Vision Conf., BMVC'93*, pages 249–258, Guildford, UK, September 1993.
- [9] M. Dhome, M. Richetin, J.-T. Lapresté, and G. Rives. Determination of the attitude of 3D objects from a single perspective view. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, December 1989.
- [10] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):932–946, July 2002.
- [11] R.O. Duda and P.E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communication of the ACM*, 15:11–15, 1972.
- [12] O. Faugeras, F. Lustman, and G. Toscani. Motion and structure from motion from point and line matches. In *iccv*, pages 25–34, London, UK, June 1987.
- [13] N. Fischler and 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.
- [14] A.W. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(12-13):1145–1153, December 2003.
- [15] E. Grosso, G. Metta, A. Oddera, and G. Sandini. Robust visual servoing in 3D reaching tasks. *IEEE Trans. on Robotics and Automation*, 12(5):732–742, October 1996.
- [16] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, October 1998.
- [17] R. Haralick, H. Joo, C. Lee, X. Zhuang, V Vaidya, and M. Kim. Pose estimation from corresponding point data. *IEEE Trans on Systems, Man and Cybernetics*, 19(6):1426–1445, November 1989.
- [18] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2001.
- [19] R. Horaud, B. Conio, O. Le Boulleux, and B. Lacolle. An analytic solution for the perspective 4-points problem. *Computer Vision, Graphics and Image Processing*, 47(1):33–44, July 1989.
- [20] P.V.C. Hough. Machine analysis of bubble chamber pictures. In *Int. Conf. on High Energy Accelerators and Instrumentation*, pages 554–556, CERN, 1959.
- [21] P.-J. Huber. *Robust Statistics*. Wiley, New York, 1981.
- [22] 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.
- [23] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision, Graphics, and Image Processing*, 44(1):87–116, 1988.
- [24] D. Kragic and H. Christensen. Cue integration for visual servoing. *IEEE Trans. on Robotics and Automation*, 17(1):19–26, February 2001.
- [25] Y. Liu, T.S. Huang, and O.D. Faugeras. Determination of camera location from 2D to 3D line and point correspondences. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):28–37, January 1990.
- [26] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–394, March 1987.
- [27] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.
- [28] E. Marchand, A.I. Comport, and F. Chaumette. Improvements in robust 2D visual servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, volume 1, pages 745–750, New Orleans, Louisiana, April 2004.
- [29] F. Martin and R. Horaud. Multiple camera tracking of rigid objects. *Int. Journal of Robotics Research*, 21(2):97–113, February 2002. (Rapport INRIA RR-4268, septembre 2001).
- [30] S.-J. Maybank and O. Faugeras. A theory of self calibration of a moving camera. *Int. Journal of Computer Vision, IJCV*, 8(1):123–152, 1992.
- [31] J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, December 1995.
- [32] N. P. Papanikolopoulos and P. K. Khosla. Selection of features and evaluation of visual measurements for 3D robotic visual tracking. *Int. Symp. on Intelligent Control.*, pages 320–325, August 1993.
- [33] M. Pressigout and E. Marchand. Model-free augmented reality by virtual visual servoing. In *IAPR Int. Conf. on Pattern Recognition, ICPR'04*, volume 2, pages 887–891, Cambridge, UK, August 2004.
- [34] P.J. Rousseeuw. Least median of squares regression. *Journal American Statistic Association*, 79:871–880, 1984.
- [35] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, New York, 1987.
- [36] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991.

- [37] G. Simon and M.-O. Berger. A two-stage robust statistical method for temporal registration from features of various type. In *Int. Conf. on Computer Vision, ICCV'98*, pages 261–266, Bombay, India, Janvier 1998.
- [38] G. Simon and M.-O. Berger. Pose estimation for planar structures. *IEEE Computer Graphics and Applications*, 22(6):46–53, November 2002.
- [39] C.-J. Taylor, J.-P. Ostrowski, and S.-H. Jung. Robust visual servoing based on relative orientation. *Int Conf on Computer Vision and Pattern Recognition*, pages 574–580, June 1999.
- [40] 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.
- [41] Z. Zhang, R. Deriche, O. Faugeras, and 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.