



# Améliorations d'une approche de programmation par contraintes pour l'estimation de paramètres

Bertrand Neveu, Martin De La Gorce, Gilles Trombettoni

## ► To cite this version:

Bertrand Neveu, Martin De La Gorce, Gilles Trombettoni. Améliorations d'une approche de programmation par contraintes pour l'estimation de paramètres. Journées Francophones de Programmation par Contraintes 2016, Jun 2016, Montpellier, France. <hal-01376625>

**HAL Id: hal-01376625**

**<https://hal-enpc.archives-ouvertes.fr/hal-01376625>**

Submitted on 5 Oct 2016

**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.

# Améliorations d'une approche de programmation par contraintes pour l'estimation de paramètres

Bertrand Neveu<sup>1</sup>Martin de la Gorce<sup>1</sup>Gilles Trombettoni<sup>2</sup><sup>1</sup> LIGM, Ecole des Ponts, UPEM, ESIEE Paris, CNRS, UPE, Champs-sur-Marne, France<sup>2</sup> LIRMM, Université de Montpellier, France

Bertrand.Neveu@enpc.fr martin.delagorce@gmail.com Gilles.Trombettoni@lirmm.fr

## Résumé

Cet article présente diverses améliorations d'une méthode combinatoire complète proposée par Jaulin, Walter et Didrit [2] pour trouver, dans le contexte de l'estimation de paramètres, toutes les instances d'un modèle paramétré compatibles avec au moins  $q$  observations dans les limites d'une tolérance donnée. Cette méthode a été appliquée à la détection de formes (cercles en 2D, plans en 3D). Cet article est un résumé de l'article publié dans les actes d'ICTAI 2015 [3].

## 1 Introduction

L'estimation de paramètres est un problème difficile qui apparaît souvent dans les sciences de l'ingénieur. Cela consiste à déterminer les  $n$  paramètres numériques d'un modèle à partir de  $m$  observations. L'étalonnage et la géolocalisation en sont des exemples. Un modèle paramétré est défini par une équation implicite  $f(\mathbf{x}, \mathbf{p}) = 0$ ,  $\mathbf{p}$  étant le vecteur de paramètres que l'on cherche à déterminer. Dans cet article, nous cherchons toutes les instances de modèle paramétré compatibles, dans les limites d'une tolérance donnée  $\tau$ , avec au moins  $q$  observations parmi les  $m$  (plus exactement, une instance par ensemble maximal d'observations compatibles de cardinal supérieur ou égal à  $q$ ). Nous appliquons cette méthode à la détection de formes géométriques comme des cercles en 2D et des plans en 3D, dans un nuage de points.

## 2 Etat de l'art

La principale méthode utilisée en vision par ordinateur et traitement d'image pour l'estimation de paramètres en présence d'observations aberrantes

(*outliers*) est RANSAC (RANdom SAMple Consensus), méthode stochastique, procédant par tirages aléatoires successifs dans les observations pour déterminer les paramètres du modèle ( $n$  observations pour  $n$  paramètres), en cherchant à maximiser le nombre d'observations compatibles. Une version présentée dans [4] est adaptée à la détection de *plusieurs* objets, mais sans garantie de trouver toutes les solutions. Une première version d'un algorithme déterministe complet effectuant une recherche arborescente dans l'espace des paramètres a été proposée dans [2]. Le problème peut être formulé comme la recherche de solutions d'un CSP numérique, comprenant  $n$  variables réelles  $p_1, \dots, p_n$  ayant chacune pour domaine un intervalle et liées par la contrainte globale indiquant qu'au moins  $q$  observations sont compatibles, l'ensemble  $C$  représentant les observations  $o_i$  compatibles.

$$C(p_1, \dots, p_n) = \{\mathbf{o}_i | \text{abs}(f(\mathbf{o}_i, p_1, \dots, p_n)) \leq \tau\} \quad (1)$$

$$|C((p_1, \dots, p_n))| \geq q \quad (2)$$

Chaque nœud de l'arbre de recherche correspond à une partie de l'espace exploré, c.-à-d. une boîte  $[p_1] \dots [p_n]$ . On utilise un opérateur de contraction (filtrage) basé sur la  $q$ -intersection des observations pour réduire les intervalles sur chacune des dimensions de la boîte courante. Dans l'espace des paramètres, pour chaque observation, l'ensemble des points compatibles dans la boîte courante est tout d'abord approximé par une boîte. La  $q$ -intersection de ces boîtes est la boîte enveloppe de l'union de toutes les intersections de  $q$  de ces boîtes. Trouver la  $q$ -intersection exacte est un problème DP-complet [1], aussi nous avons utilisé l'algorithme non optimal classique de projection sur les différentes dimensions. Cet algorithme, de complexité totale en  $O(nm \log(m))$ , permet de détecter

des intervalles ne pouvant pas être compatibles avec  $q$  observations et donc de réduire l'intervalle de la boîte courante dans cette dimension. Nous sommes partis de la méthode de [2] implantée dans Ibex et l'avons améliorée sur les points suivants.

### 3 Améliorations de la méthode

Nous avons apporté à l'algorithme complet de recherche de solutions des améliorations génériques (indépendantes de l'application de détection de plans ou de cercles) et des améliorations spécifiques pour ces applications.

#### 3.1 Améliorations génériques

**Validation des solutions.** Dans l'arbre de recherche, nous maintenons l'ensemble des observations possibles sur la boîte courante et des observations valides en un point de cette boîte (sous ensemble des possibles). Une condition d'arrêt de l'algorithme dans la branche courante est atteinte quand ces ensembles sont les mêmes.

L'ensemble des observations possibles est mis à jour directement par l'algorithme de q-intersection.

**q-intersection supplémentaire.** L'algorithme de contraction par q-intersection procède par projections des observations sur les différentes dimensions de la boîte courante. Nous réalisons une projection sur une direction supplémentaire dans laquelle nous espérons obtenir des petits intervalles et aboutir à une q-intersection vide. Pour cela, nous linéarisons les équations de chaque observation en les relaxant et projetons les parallélogrammes obtenus sur la direction correspondant à la moyenne de leurs normales (Figure 1).

#### 3.2 Améliorations spécifiques

**Contraintes dédiées.** Au lieu d'utiliser l'algorithme général de contraction effectuant des calculs sur la représentation des fonctions mathématiques en Ibex, nous avons réécrit ces calculs pour les contraintes simples (linéaires ou quadratiques) utilisées dans notre modélisation.

**Heuristique de bisection.** On bissecte en premier les domaines des variables de direction pour les plans et de coordonnées du centre pour les cercles, et on termine par la distance à l'origine pour les plans et le rayon pour les cercles.

**Modélisation linéaire.** Pour la détection de plans, nous proposons une modélisation linéaire des contraintes liant les paramètres, ce qui permet des calculs plus efficaces.

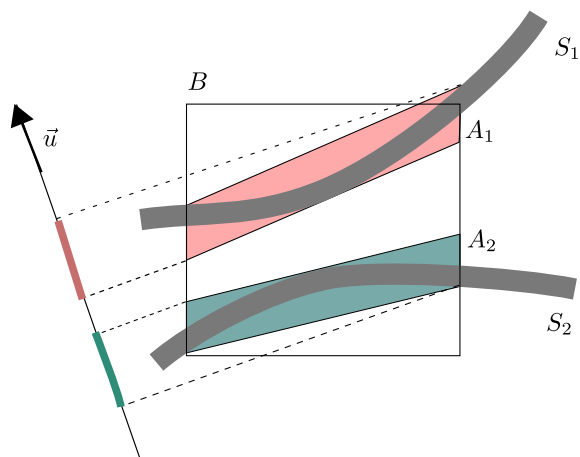


Figure 1: Projection des parallélogrammes  $A_i$ , relaxations linéaires des zones de compatibilité  $S_i$ , sur la direction de la moyenne de leurs normales.

### 4 Validation expérimentale

Nous avons testé notre algorithme sur des problèmes de détection de plans sur des exemples artificiels et sur une scène réelle ainsi que sur la détection de cercle en 2D pour trouver une bouée dans une image sous-marine. Les expérimentations ont montré que chacune des améliorations apportait un gain en temps de calcul significatif. Sur chaque instance, 2 à 3 ordres de grandeur ont été gagnés en tout par rapport à l'algorithme originel.

### References

- [1] C. Carbonnel, G. Trombettoni, P. Vismara, and G. Chabert. Q-intersection algorithms for constrained-based robust parameter estimation. In *Proc. of AAAI 2014*, pages 2630–2636, 2014.
- [2] L. Jaulin, E. Walter, and O. Didrit. Guaranteed robust nonlinear parameter bounding. In *CESA '96 IMACS Multiconference (Symposium on Modelling, Analysis and Simulation)*, pages 1156–1161, 1996.
- [3] B. Neveu, M. de la Gorce, and G. Trombettoni. Improving a constraint programming approach for parameter estimation. In *27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015, Vietri sul Mare, Italy, November 9–11, 2015*, pages 852–859, 2015.
- [4] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226, 2007.