

Détection de cylindres dans les nuages de points de scènes industrielles basée sur l'utilisation de modèles 3D a priori

Aurélien BEY^{1,2}

¹Université Lyon 1, LIRIS, UMR5205, 69622, France

²Electricité De France Recherche & Développement, 92140, Clamart, France

Résumé

Nous nous intéressons dans cet article au problème de la reconstruction de modèles 3D CAO à partir de nuages de points 3D issus d'acquisition laser en environnement industriel. Nous proposons pour cela d'utiliser des modèles 3D existants approximant les nuages de points. Ces modèles peuvent en effet être considérés comme des estimations initiales des scènes traitées, et constituent à ce titre une source de connaissance a priori importante. Nous nous intéressons ici plus particulièrement au traitement des cylindres. La solution que nous proposons permet de générer aléatoirement des ensembles de cylindres candidats potentiellement présents dans le nuage de points, et s'appuie pour cela sur l'information a priori. Le recours à l'a priori permet de diminuer la quantité de candidats non pertinents par rapport aux approches aléatoires existantes (RANSAC), et de produire des cylindres plus fiables. Nous présentons notamment les résultats obtenus sur quelques jeux de données. Pour finir, nous discutons des possibilités d'intégration de cette solution dans une approche plus globale visant à produire un modèle 3D consistant à partir des cylindres candidats ainsi produits.

Mots-clés : Reconstruction 3D, reconnaissance de formes, nuages de points, CAO

1. Introduction

La reconstruction de nuages de points est le processus par lequel, à partir d'un ensemble de points issus d'acquisition laser par exemple, on obtient un modèle 3D correspondant à ces données échantillonnées. Dans le cadre des installations industrielles, les modèles 3D consistent en des assemblages cohérents de primitives géométriques (plans, sphères, cylindres, cônes, tores), proposant une représentation de haut niveau sémantique de la scène. Il existe à l'heure actuelle de nombreux travaux abordant ce problème de reconstruction. Mais à notre connaissance, aucune approche ne permet de garantir un résultat fiable en temps raisonnable sur des nuages de points bruités et incomplets tels que ceux issus d'acquisitions sur les sites industriels.

Partant du constat que certains environnements industriels sont relativement semblables les uns par rapport aux autres (typiquement, les paires de bâtiments nucléaires - cf. figure 1), et disposant d'un corpus de modèles 3D existants pour un certain nombre d'installations, nous proposons d'utiliser ces modèles 3D comme une information *a priori* permettant de guider la reconstruction des nouveaux nuages de points.

Chaque scène (nuage et modèle) est exprimée dans un repère propre et nous considérons que les repères sont mis en commun de sorte que les scènes sont initialement recalées. Les parties similaires telles que les murs doivent alors se superposer.

En matière de reconstruction, nous concentrons plus particulièrement nos recherches sur le traitement des cylindres. Il s'avère en effet que la détection simultanée de différents types de primitives n'est pas une tâche simple, et peut induire des erreurs dans les résultats. Les scènes industrielles étant en grande partie composées de cylindres (tuyauteries et gros composants), nous choisissons donc de privilégier la détection des cylindres dans un premier temps. Par la suite, il est envisageable de compléter cette première reconstruction en y intégrant les tores et cônes, qui servent principalement de connecteurs entre cylindres dans les composants de tuyauterie.

La méthode que nous proposons s'appuie sur l'approche stochastique RANSAC, et permet de guider l'exploration aléatoire du nuage sur la base des connaissances préalables apportées par le modèle 3D *a priori*. L'aspect aléatoire permet d'assurer une certaine tolérance aux changements pouvant survenir entre la scène traitée et l'*a priori*. Dans cet article, nous nous intéressons plus spécifiquement à la manière



Figure 1: Comparaison de deux scènes industrielles. Bien que très semblables, on constate que les deux environnements présentent quelques différences significatives (tuyaux surlignés).

dont on peut générer des cylindres fiables à partir des données *a priori*. L'utilisation de ces cylindres pour construire un modèle 3D consistant n'est pas abordée en détail, bien que des possibilités soient mentionnées en conclusion.

Notre contribution principale au domaine de la reconstruction de nuages de points consiste en un ensemble d'outils probabilistes permettant de tirer parti d'une estimation initiale de la scène pour améliorer la fiabilité des candidats générés aléatoirement. Notons que, dans le cas des approches de type RANSAC, cette amélioration de la pertinence des résultats produits induit une diminution des temps de calcul.

Cet article est structuré de la manière suivante : dans un premier temps (section 2), nous présentons quelques travaux traitant de la reconnaissance de formes dans les nuages de points. En second lieu (section 3), nous définissons le problème de la détection de cylindre étant donné un modèle 3D *a priori*, puis nous détaillons la solution probabiliste que nous proposons pour le résoudre (section 4). Par la suite (section 6), nous montrons l'efficacité de notre approche en présentant les résultats obtenus sur quelques jeux de données. Pour finir (section 7), nous concluons quant à la méthode proposée, et discutons des perspectives permettant d'intégrer cette approche dans une démarche plus globale aboutissant à la reconstruction de modèles 3D consistants à partir de nuages de points.

2. État de l'art

Le traitement des nuages de points en vue de leur reconstruction fait l'objet de très nombreux travaux.

Ceux de [Cha02] traitent de la détection de ligne de tuyauterie dans les nuages de points représentant des scènes industrielles. L'approche proposée repose sur une étude locale du nuage permettant la détection de portions de cylindres, qui sont ensuite étendues par propagation dans le nuage. Le

point servant d'origine à la propagation doit être spécifié par l'utilisateur. Dans [Bos10], l'auteur propose une approche en deux temps permettant de recalibrer finement un modèle 3D *a priori* sur un nuage de points qui lui correspond approximativement. La première phase consiste en un recalage grossier réalisé à l'aide de correspondances entre modèle et nuage spécifiés par l'utilisateur. Ensuite chacun des éléments du modèle est recalé individuellement par le biais d'un algorithme Iterative Closest Points (ICP). Cette approche ne peut fonctionner que dans les cas où les différences entre le modèle et le nuage sont insignifiantes, puisque l'ICP ne peut converger que lorsque les objets à recalibrer sont initialement très proches les uns des autres.

Dans certains autres travaux, le problème de reconstruction est décomposé en deux sous-tâches : la segmentation, visant à identifier dans le nuage de points des sous-ensembles homogènes vis-à-vis d'un critère donné (habituellement, l'absence d'arêtes vives sur les surfaces) et la modélisation permettant de calculer la surface correspondant à un sous-ensemble de points. Le problème de modélisation est aujourd'hui relativement maîtrisé, via les approches de minimisation des distances au sens des moindres carrés [Ahn08, LMM97, BKV*02, Cha02]. La segmentation en revanche reste un problème complexe dans les nuages de points 3D désorganisés et bruités. Les travaux de [MLM01, GG04, WKWL02, YY08] proposent diverses méthodes intéressantes de segmentation de nuages de points ou d'images de profondeur.

Un certain nombre d'auteurs s'intéressent à la reconstruction de surfaces à partir de nuages de points sur la base d'informations *a priori*. Dans les travaux de [PMG*05, JWB*06, GSH*07], différentes approches visant à reconstruire une surface maillée à partir de nuages de points sont présentées. Chacune de ces méthodes tire parti d'informations disponibles quant aux données : base de modèles surfaciques

existants, supposition sur la distribution des points du nuage, patchs surfaciques *a priori*.

Outre les différentes approches présentées ci-avant, citons deux méthodes de détection de formes dans les nuages de points connues pour leur robustesse vis-à-vis du bruit d'acquisition et des points aberrants du nuage.

Les transformées de Hough généralisées fonctionnent sur un principe de vote : l'espace paramétrique des formes recherchées est discrétisé, produisant ainsi une grille dont chaque cellule correspond à une instance de forme. Chacune des cellules stocke ensuite le nombre de points du nuage correspondant à la forme qu'elle représente. De cette manière, les cellules associées à un pic de votes indiquent les formes qui sont effectivement présentes dans le nuage de points. Ce processus, bien qu'étant fiable, est très coûteux puisqu'il requiert un parcours de l'ensemble de la grille paramétrique, dont la taille peut être considérable. Dans le cas du cylindre par exemple, l'espace paramétrique possède 5 dimensions, et ne permet pas un échantillonnage fin de la grille. Dans [Rv05] les auteurs proposent une approche permettant de diminuer cette complexité dans le cas des cylindres. Pour cela, le processus est décomposé en deux phases : la première permet de détecter l'axe du cylindre (problème bidimensionnel), alors que la seconde gère le calcul de la position et du rayon (problème tridimensionnel). Bien que les résultats obtenus soient très convaincants, la seconde partie reste tout de même complexe à gérer, puisqu'elle fait intervenir les positions et rayons qui nécessitent une grille pouvant être très large.

L'algorithme RANdom SAMple Consensus (RANSAC) quant à lui repose sur une démarche stochastique. On tire aléatoirement un nombre minimal de points définissant une instance de la forme recherchée. La forme candidate construite à partir de ce quorum détermine ensuite son support comme étant le nombre de points du nuage qui lui correspondent. Ces tirages sont répétés un certain nombre de fois, après quoi le candidat dont le support est le plus grand est validé, et les points qui lui correspondent sont retirés du nuage. Ce processus est finalement réitéré jusqu'à ce que les points restant dans le nuage ne permettent plus la détection de formes. Le problème de cette approche, lorsqu'elle est appliquée naïvement, réside dans le fait que le nombre de tirages nécessaire à la détection effective de formes est prohibitif : il s'exprime comme une fonction inverse de la probabilité de tirer les points du quorum sur une surface commune. Pour diminuer cette complexité, [SWK07] propose de tirer les points de chaque quorum dans une même cellule d'octree, dont la profondeur est déterminée dynamiquement en fonction des résultats obtenus jusqu'alors. De cette manière, la probabilité que ces points appartiennent à une même surface augmente, et le nombre de tirages nécessaire diminue en conséquence.

Les résultats obtenus par l'algorithme de [SWK07] sont

intéressants, y compris sur les données industrielles. Nous avons toutefois constaté quelques problèmes. Tout d'abord, lorsqu'il gère la détection simultanée de différents types de formes (cylindres, cônes et tores typiquement), l'algorithme ne parvient pas vraiment à fournir une modélisation correcte du nuage, et tend à mélanger les différentes primitives (tores à la place de cylindres par exemple). De plus, les candidats générés par cette méthode sont imprécis, et il n'est pas rare de voir plusieurs formes se superposer. Cette imprécision est notamment due au fait que la quantité d'information utilisée pour la création des candidats est minimale (les auteurs préconisent l'utilisation de 3 points associés à leurs normales), de sorte que la moindre perturbation sur les points influence considérablement le résultat.

Notre approche s'inspire des principes présentés dans [SWK07]. Nous proposons en effet une méthode stochastique permettant de construire des candidats à partir de points sélectionnés aléatoirement. Toutefois, le modèle *a priori* dont on dispose nous permet d'intégrer de la connaissance dans ce processus de sélection, de sorte que l'on puisse diminuer le nombre de candidats non pertinents générés, et que ces candidats soient plus précis.

3. Définition du problème

Nous nous posons le problème de la mise en correspondance des cylindres issus d'un modèle 3D *a priori* avec les parties d'un nuage à traiter.

Soient $\mathcal{P} = \{(\mathbf{p}_1, \mathbf{n}_1), \dots, (\mathbf{p}_n, \mathbf{n}_n)\}$ un nuage de points et M un modèle 3D pouvant être considéré comme une approximation initiale de ce nuage. Les paires $(\mathbf{p}_j, \mathbf{n}_j) \in \mathcal{P}$ se composent d'un point 3D \mathbf{p}_j et d'un vecteur normal unitaire \mathbf{n}_j associé à \mathbf{p}_j . Les normales peuvent être fournies avec les données issues de l'acquisition laser, ou calculées par ajustement de plans locaux (cf. [MN04]). Nous supposons par défaut que l'orientation correcte des normales n'est pas connue.

Étant donné un cylindre *a priori* $C \in M$, on cherche à trouver les cylindres $\{C_0, \dots, C_{r-1}\}$ présents dans \mathcal{P} , de sorte que chaque candidat C_m puisse être considéré comme étant une réplique de C soumis à une légère transformation. En effet, dans la scène représentée par \mathcal{P} , un cylindre correspondant à l'*a priori* C peut être :

- décalé : la position \mathbf{c}_C est modifiée. Les déplacements de cylindres sont relativement courants entre deux scènes industrielles (cf. figure 1). Ils peuvent être d'une amplitude considérable, mais restent toutefois restreints à une zone limitée de la scène (typiquement, un équipement reste dans une même salle).
- réorienté : la direction \mathbf{d}_C de l'axe est modifiée. Dans les installations industrielles, les cylindres sont majoritairement orientés de manière verticale ou horizontale. Lorsqu'ils sont horizontaux, les cylindres sont souvent contraints par la présence des murs. Dans la plupart

des cas, les cylindres ne changent pas significativement d'orientation entre deux scènes, et on admettra donc que cette modification est peu probable (bien qu'elle ne soit en réalité pas impossible).

- redimensionné : le rayon r_C est modifié. Le rayon d'un cylindre est plus ou moins lié à la fonction de l'équipement qu'il représente. À ce titre, deux cylindres (issus de deux scènes semblables) qui se correspondent possèdent la plupart du temps un rayon similaire. On admet donc que les modifications de rayon sont majoritairement non significatives.

Ces hypothèses de stabilité des paramètres de cylindres permettent la mise en correspondance : un cylindre C_m correspond d'autant plus probablement au cylindre C lorsque les paramètres des ces deux formes sont semblables.

Le nombre r de cylindres dans \mathcal{P} correspondant effectivement à C ne peut être connu par avance. Dans la plupart des cas, il devrait être égal à un. Toutefois, il peut être nul, indiquant ainsi que C est absent du nuage de points. À l'inverse, il peut être supérieur à 1, dans le cas où un équipement correspondant à C se retrouverait dupliqué dans la scène représentée par \mathcal{P} . Le fait qu'on ne contraigne pas le nombre de correspondances r devrait ainsi permettre de tenir compte des suppressions d'équipements, et de modéliser les parties présentes dans \mathcal{P} sans qu'elles ne figurent forcément dans le modèle 3D *a priori* M .

Par défaut, la longueur des cylindres candidats $\{C_0, \dots, C_{r-1}\}$ étant indéterminée, nous considérons dans un premier temps que les candidats sont de longueur infinie. Pour permettre la modélisation, il est toutefois nécessaire de déterminer la portion valide de ces candidats infinis.

4. Principe de la solution

Étant donné un cylindre *a priori* C , notre méthode génère des candidats $\{C_0, \dots, C_{r-1}\}$ qui sont pertinents vis-à-vis de \mathcal{P} ainsi que de C . De la même manière que les approches RANSAC, la création de chaque candidat s'effectue à partir de points $\{\mathbf{q}_0, \dots, \mathbf{q}_{k-1}\}$ tirés aléatoirement dans le nuage \mathcal{P} . Toutefois, pour diminuer l'imprécision provoquée par le bruit du nuage sur les candidats, nous proposons d'augmenter la quantité k de points servant à leur création : plus les points utilisés sont nombreux, plus la forme engendrée est stable par rapport au bruit.

Pour assurer la pertinence de chaque candidat, il est nécessaire que les k points ayant permis sa création soient issus d'une surface cylindrique commune, qui soit aussi semblable que possible à la surface de C (conformément à l'hypothèse de stabilité entre les cylindres correspondants). Nous proposons une exploration probabiliste du nuage qui permet de faire en sorte que l'on puisse tendre vers ces conditions. Cette approche fonctionne en 3 phases :

précalcul : avant tout traitement, des sous-ensembles sont

Distribution des directions de cylindres

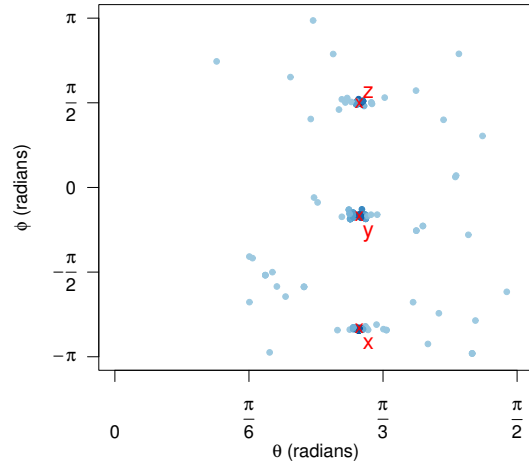


Figure 2: Distribution des axes de cylindres dans une scène industrielle. Chaque point de ce graphique représente les coordonnées sphériques (θ, ϕ) d'une direction de cylindre. Les points sont colorés selon leur densité (plus ils sont sombres, plus il y a de points dans leur voisinage). Les axes canoniques \mathbf{X} , \mathbf{Y} et \mathbf{Z} sont indiqués par des croix. On constate que la plupart des directions sont regroupées autour des axes canoniques.

identifiés dans le nuage de points. Chaque sous-ensemble ne contient que les points susceptibles d'être sur un cylindre de direction spécifiée. Étant donné le cylindre *a priori* C , on identifie alors le sous-ensemble $\mathcal{P}_C \subset \mathcal{P}$ correspondant à sa direction.

sélection d'un premier point : un premier point \mathbf{q}_0 est sélectionné dans \mathcal{P}_C . Ce point est choisi de manière à ce qu'il ne soit pas trop éloigné de C .

sélection des $k - 1$ points restants : nous utilisons ensuite l'idée formulée dans [SWK07] selon laquelle les points sont d'autant plus probablement issus de la même surface qu'ils sont proches les uns des autres. Les points $\{\mathbf{q}_1, \dots, \mathbf{q}_{k-1}\}$ sont donc choisis à proximité de \mathbf{q}_0 .

5. Génération de cylindres candidats

5.1. Précalcul : rassemblement des cylindres

On constate que dans les scènes fortement structurées comme le sont les scènes industrielles, la plupart des cylindres sont orientés selon quelques directions privilégiées (cf. figure 2). On peut alors, à l'aide d'un algorithme de clustering, rassembler les cylindres de M en quelques groupes $\{G_0, \dots, G_a\}$ de cylindres ayant une direction commune.

Considérons $D = \{\mathbf{d}_0, \dots, \mathbf{d}_b\}$ l'ensemble des directions des cylindres de M , où chaque direction \mathbf{d}_j s'exprime sous forme d'un vecteur 3D unitaire. Toute direction \mathbf{d}_j peut donc être vue comme un point sur la sphère unitaire centrée sur l'origine (S^2), d'où $D \subset S^2$. Les algorithmes habituels de clustering, tels que le Mean-Shift, sont pour la plupart difficilement utilisables dans les espaces non Euclidiens (S^2 n'étant pas Euclidien). Pour passer outre ce problème, on pourrait envisager d'utiliser les coordonnées sphériques associées à ces données. Mais ce changement provoquerait des distorsions aux pôles qui rendraient le clustering impraticable. Les algorithmes Medoid-Shift quant à eux peuvent fonctionner dans des espaces métriques non-Euclidiens, et sont donc adaptés à la résolution du problème de clustering tel que nous l'avons posé. [VS08] propose une implémentation efficace de ce type d'algorithme, dont la complexité est en $O(b^2)$ (rappelons que b est ici la quantité de directions à traiter).

La distance que nous utilisons dans cet algorithme est la distance orthodromique entre les points d'une sphère :

$$d(\mathbf{d}_i, \mathbf{d}_j) = \arccos(\mathbf{d}_i \cdot \mathbf{d}_j)$$

"." étant le produit scalaire de \mathbb{R}^3 . On peut toutefois omettre la fonction arccos afin d'accélérer les calculs, sans que cela ne nuise vraiment aux résultats. Comme les directions ne sont pas orientées, le clustering est en fait effectué sur l'ensemble $D = D^+ \cup D^-$, avec $D^\pm = \{\pm \mathbf{d}_0, \dots, \pm \mathbf{d}_b\}$. Il faut cependant filtrer les groupes issus du clustering. En effet, chaque groupe résultant possède un groupe symétrique par rapport au centre de S^2 : on ne garde alors qu'un élément de chaque paire.

Une fois les groupes $\{G_0, \dots, G_a\}$ calculés, on identifie pour chaque groupe G_i l'ensemble des points dans \mathcal{P} dont la normale est perpendiculaire (à une tolérance α près) à une des directions contenues dans G_i :

$$\mathcal{P}_i = \left\{ (\mathbf{p}, \mathbf{n}) \in \mathcal{P} \mid \exists C_m \in G_i \text{ t.q. } \left| \frac{\mathbf{d}_{C_m} \cdot \mathbf{n}}{\|\mathbf{d}_{C_m}\| \cdot \|\mathbf{n}\|} \right| \leq \sin(\alpha) \right\}$$

où \mathbf{d}_{C_m} est la direction du cylindre C_m et α est la tolérance angulaire spécifiée par l'utilisateur.

Par la suite, on appellera \mathcal{P}_C le sous-ensemble calculé à partir du groupe G_C auquel C est rattaché. Sous l'hypothèse de stabilité des directions de cylindres, seuls les points de \mathcal{P}_C sont réellement susceptibles de se trouver sur des cylindres correspondant à C .

5.2. Tirage des points

Pour générer un candidat, nous proposons de nous appuyer sur k points sélectionnés dans le nuage de points \mathcal{P} . Or tous les points échantillonnés à partir de cylindres correspondant à C se trouvent dans le sous-ensemble \mathcal{P}_C (sous l'hypothèse de stabilité des directions de cylindres). Il est donc plus probable que l'on génère un candidat qui corres-

ponde effectivement à C lorsque les points sont sélectionnés dans \mathcal{P}_C .

5.2.1. Tirage d'un premier point

Par une méthode d'échantillonnage reposant sur un test de rejet, il est possible de contrôler la distance du premier point tiré (\mathbf{q}_0) par rapport au cylindre *a priori* C recherché.

Considérons un point \mathbf{q} tiré aléatoirement dans \mathcal{P}_C . On évalue alors une probabilité que ce point soit effectivement sur un cylindre correspondant à C en étudiant sa distance au cylindre C :

$$P(\mathbf{q}|C, \sigma_c) = \frac{1}{\sigma_c \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{d(C, \mathbf{q})}{\sigma_c} \right)^2}$$

où $d(C, \mathbf{q})$ est la distance séparant \mathbf{q} de C . On utilise ici une loi normale centrée dont l'écart type σ_c est spécifié par l'utilisateur, et permet de moduler la tolérance vis-à-vis des déplacements de cylindres (σ_c correspond à une distance).

Il s'agit ensuite de décider si le point \mathbf{q} peut être utilisé en vue de créer un candidat correspondant à C . Pour se faire, on tire aléatoirement une valeur $\beta \in]0, 1]$ selon une loi uniforme. Le point \mathbf{q} est accepté si $\beta \leq P(\mathbf{q}|C, \sigma_c)$. Dans le cas contraire il est rejeté, un nouveau point est sélectionné, et le test d'acceptation/rejet est réitéré.

Pour permettre la rapidité des calculs, la distribution des distances des points de \mathcal{P} par rapport à C n'est pas prise en compte dans ce processus d'échantillonnage. Ce qui implique que la distribution résultante des distances $d(C, \mathbf{q}_0)$ pour les points \mathbf{q}_0 ainsi générés ne peut pas être déterminée précisément. Entre autres, on ne peut pas vraiment garantir une convergence rapide de ce processus d'échantillonnage par rejet. Pour cette raison, on se fixe un nombre maximal de tirages au delà duquel on retient $\mathbf{q}_0 = \operatorname{argmax}_{\mathbf{q}} P(\mathbf{q}|C, \sigma_c)$.

Quoi qu'il en soit, le point \mathbf{q}_0 retenu se trouve sur une surface qui, de par sa distance au cylindre C (et sous l'hypothèse de stabilité des positions de cylindres), lui correspond probablement.

5.2.2. Tirage des points restants

Une fois \mathbf{q}_0 sélectionné, nous faisons l'hypothèse que ce point se trouve effectivement sur un cylindre. Si ce n'est pas le cas, la génération d'un candidat à partir des points $\{\mathbf{q}_0, \dots, \mathbf{q}_{k-1}\}$ ne pourra de toute manière pas produire de candidat pertinent, quels que soient les points $\{\mathbf{q}_1, \dots, \mathbf{q}_{k-1}\}$ retenus par la suite : l'échec de la sélection de \mathbf{q}_0 entraîne l'échec du tirage.

Pour maximiser la probabilité que les points soient issus d'une surface commune (et produisent donc un candidat fiable), on se propose d'adopter une approche en deux étapes. La première consiste en la sélection d'un voisinage : plus les points sont proches les uns des autres, plus il est

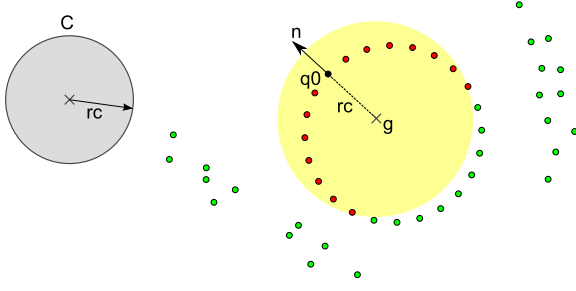


Figure 3: Principe du tirage dans un voisinage : exemple en dimension 2. Étant donnée une forme a priori C , on sélectionne un premier point \mathbf{q}_0 dans le nuage de sorte que ce point ne soit "pas trop loin" de C . Puis on détermine un point temporaire \mathbf{g} par translation de \mathbf{q}_0 le long de sa normale \mathbf{n} . Dans le cas où cette translation se fait dans le bon sens, \mathbf{g} se trouve à l'intérieur de la forme échantillonnée et le voisinage centré sur \mathbf{g} contient majoritairement ses échantillons (peu d'outliers).

probable qu'ils soient sur une même surface. Considérons le point issu du décalage de \mathbf{q}_0 le long de sa propre normale \mathbf{n} :

$$\mathbf{g} = \mathbf{q}_0 + (-1)^\eta \cdot r_C \cdot \mathbf{n}$$

r_C étant le rayon du cylindre recherché C . La valeur de η est choisie aléatoirement dans l'ensemble $\{1, 2\}$ (les deux éléments étant équiprobables), de manière à déterminer dynamiquement le sens de ce décalage parmi les deux possibles. Rappelons en effet que l'on ne présuppose pas que les normales sont correctement orientées.

Le point ainsi décalé se trouve donc à l'intérieur du cylindre sur lequel se trouve \mathbf{q}_0 (par hypothèse) avec une probabilité de $1/2$. Définissons le voisinage sphérique centré sur \mathbf{g} :

$$\mathcal{S}_\rho = \{\mathbf{p} \in \mathcal{P}_C \text{ s.t. } \|\mathbf{p} - \mathbf{g}\| \leq \rho \cdot r_C\}$$

$\rho \geq 1$ étant un facteur dont on proposera un calcul par la suite. Notons que le calcul de voisinages peut être géré efficacement à l'aide de structures d'accélération telles que les KD-Arbres pour lesquels chaque requête peut être gérée en $O(r \cdot \log(n))$ (n étant le nombre de points du nuage, r le nombre de points du voisinage résultant). Lorsque \mathbf{g} est effectivement "à l'intérieur de son cylindre", et si la valeur ρ est adaptée, \mathcal{S}_ρ contient majoritairement des points se trouvant sur la surface de ce même cylindre (cf. figure 3). Sous ces conditions, il est donc judicieux de tirer les points restants ($\mathbf{q}_{0 \leq j < k}$) dans \mathcal{S}_ρ .

Afin d'assurer la robustesse du processus de création du candidat, il est toutefois nécessaire que les points ne soient pas trop proches les uns des autres : plus les points sont proches, plus l'information apportée par chacun est insignifiante. Il est donc souhaitable que le rayon du voisinage sphérique soit suffisamment grand pour permettre d'explorer une partie du nuage qui contienne une quantité convenable d'information, tout étant suffisamment petit pour mi-

nimiser la quantité d'outliers (données n'appartenant pas à la surface recherchée et portant donc une information fausse). Pour ce faire, nous proposons d'utiliser $\rho = 1 + |\rho'|$ où ρ' est tiré aléatoirement suivant une distribution normale centrée d'écart type σ_r (spécifié par l'utilisateur). Notons au passage que σ_r permet aussi de paramétrer la tolérance aux grossissements de rayons de cylindres.

Dans le cas où le facteur ρ retenu serait trop grand, le voisinage \mathcal{S}_ρ contient potentiellement des outliers (qu'il ne faut pas tirer). La deuxième étape du processus consiste en une estimation probabiliste de la propension de chaque point du voisinage à se trouver effectivement sur la surface cylindrique recherchée. Considérons les normales associées aux points de \mathcal{S}_ρ puis, par analyse en composantes principales, le plan passant par l'origine et qui approxime au mieux ce nuage de normales. Cette approche est souvent utilisée pour estimer l'axe d'un cylindre échantillonné [Cha02, YY08, Rv05] : si la quantité d'outliers est raisonnable, le vecteur normal au plan ainsi calculé constitue une bonne estimation de la direction du cylindre échantillonné par \mathcal{S}_ρ . Considérons ensuite la projection de \mathcal{S}_ρ dans ce plan. Les projetés des points qui échantillonnent effectivement le cylindre décrivent alors un cercle. Par un algorithme MSAC (décrit dans [TZ00] par exemple), on peut détecter le cercle le plus vraisemblable passant par le projeté de \mathbf{q}_0 (rappelons que, par hypothèse, \mathbf{q}_0 appartient au cylindre recherché). Chaque point se voit ensuite affecter une probabilité en fonction de son éloignement à ce cercle de référence R :

$$P(\mathbf{q}_i | R) = \frac{1}{\varepsilon \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{d(R, \mathbf{q}_i)}{\varepsilon} \right)^2}$$

où $d(R, \mathbf{q}_i)$ est la distance séparant R de \mathbf{q}_i , le projeté de \mathbf{q}_i dans le plan, et ε est une estimation de l'amplitude du bruit dans le nuage (fournie par l'utilisateur).

Les $k - 1$ points restants sont alors tirés aléatoirement conformément à la distribution de probabilités ainsi estimées.

5.3. Création du candidat à partir des points

Une fois les points $\{\mathbf{q}_0, \dots, \mathbf{q}_{k-1}\}$ sélectionnés, le cylindre candidat correspondant peut être construit comme étant la forme approximant au mieux ces points.

Dans les approches de type RANSAC existantes, les candidats sont générés à partir d'une quantité minimale de points. Comme nous l'avons remarqué, cette approche ne permet pas de générer des candidats fiables (forte exposition au bruit), et le problème n'est pas simple à résoudre lorsque le nombre de points est vraiment faible (cf. [BF06]). Toutefois, c'est un mal nécessaire dans ces approches puisque le tirage d'un trop grand nombre de points augmente considérablement la probabilité d'intégrer un point \mathbf{q}_j qui sabote la création du candidat.

Dans notre cas, la méthode proposée permet en théorie de fournir des points pertinents si \mathbf{q}_0 appartient bien à une surface cylindrique et si la valeur de η retenue est appropriée (ce qui arrive une fois sur deux). La probabilité d'échec dans notre approche ne dépend que faiblement du nombre k de points et nous pouvons donc nous permettre d'utiliser une quantité confortable ($k > 10$) de points pour générer les cylindres candidats.

Dans ce contexte de redondance des données, le cylindre candidat C_m peut être calculé comme la forme minimisant l'erreur au sens des moindres carrés :

$$C_m = \operatorname{argmin}_{C_\omega} \sum_{j=0}^{k-1} d(C_\omega, \mathbf{q}_j)^2$$

Des expressions analytiques de la distance $d(C_\omega, \mathbf{q}_j)$ pourront être trouvées dans [Cha02, MLM01].

Quant à la résolution de ce problème d'optimisation non linéaire, elle peut être réalisée à l'aide d'algorithmes itératifs tels que l'algorithme de Levenberg-Marquardt. Cependant, la plupart de ces algorithmes nécessitent une estimation initiale qui ne soit pas trop éloignée de la solution recherchée pour garantir la convergence vers le minimum global de la fonction traitée. De par le processus ayant permis la sélection des points, on dispose de paramètres fournissant une bonne approximation initiale de la solution. En effet, le plan de projection utilisé (section 5.2.2) fournit, via son vecteur normal, une estimation de l'axe du cylindre optimal. Quant à la position et au rayon, ils sont fournis par le cercle optimal R calculé par MSAC.

Du fait du nombre raisonnable de points utilisés (nous préconisons $k \leq 50$), et de la bonne qualité de l'estimation initiale, la convergence devrait être relativement rapide. Si l'on se fixe un nombre maximal d'itérations τ , l'algorithme d'optimisation réalise au plus τ calculs de la fonction à optimiser et de ses dérivées partielles. Dans la mesure où le calcul de la fonction nécessite un parcours des k points, chaque candidat est donc créé en $O(k \cdot \tau)$ (avec un grand coefficient dépendant de la dimension du problème traité - ici 5). Ce processus est toutefois considérablement plus lent que l'approche proposée dans [SWK07], dans laquelle les cylindres sont créés instantanément.

Pour déterminer la portion valide du cylindre (qui lors de sa création, est considéré comme étant infini), nous utilisons les points du nuage. Nous rassemblons dans un premier temps les points dont la distance au candidat C_m est inférieure à un seuil ε (le même que celui présenté en section 5.2.2). Ces points sont ensuite projetés sur l'axe de C_m , puis nous calculons la composante connexe dans l'ensemble des projetés qui contient la projection de \mathbf{q}_0 . Les valeurs extrêmes de cette composante connexe déterminent les limites du candidat.

6. Résultats

6.1. Illustration sur un cas simple

Nous avons testé notre approche sur des données synthétiques bruitées, afin de les comparer avec les approches RANSAC existantes. La figure 4 montre les résultats obtenus après génération de 1000 candidats dans une scène constituée de 6 cylindres relativement proches les uns des autres. Tous les cylindres sont de rayon 1 et sont échantillonnés par le même nombre de points (7000 par cylindre). Les points du nuage sont entachés d'une erreur Gaussienne dont l'écart type vaut 3% des rayons des cylindres. L'*a priori* est construit à partir d'un des cylindres du nuage que nous avons légèrement décalé, désorienté et redimensionné. Sur cet exemple, nous n'utilisons pas l'*a priori* de position ($P(\mathbf{q}|C, \sigma_c) = 1 \forall \mathbf{q} \in \mathcal{P}_C$). La scène disposant de normales correctement orientées, nous avons testé notre approche avec (en fixant $\eta = 2$) et sans ($\eta \in \{1, 2\}$) l'utilisation de cette information. Nous avons ensuite testé, sur le même nombre de candidats, un RANSAC (que nous qualifions de "localisé") semblable à celui proposé par [SWK07], dans lequel les points sont tirés dans une même cellule d'octree. Pour favoriser la méthode, la profondeur la plus appropriée est calculée conformément au rayon des cylindres recherchés. En pratique, les auteurs proposent de générer les cylindres à partir de 2 points et de leurs normales, par un calcul relativement simple. Pour finir, nous avons testé une approche que nous qualifions de naïve, dans laquelle les 2 points sont tirés aléatoirement et indépendamment l'un de l'autre.

Nous présentons ici la distribution des rayons et supports des cylindres générés. Le support étant la proportion des points du nuage dont la distance à la forme étudiée est inférieure à un seuil ε (ici, $\varepsilon = 0.25$ - notons qu'il s'agit du même paramètre que celui présenté en section 5.2.2).

Les résultats montrent clairement sur cet exemple que notre approche permet, en tenant compte de l'*a priori*, de créer majoritairement des cylindres candidats conformes aux attentes. On note notamment la précision des cylindres générés. En effet, la plupart possèdent un rayon proche de celui attendu : 90% des candidats ont un rayon de 1 ± 0.05 . De même, les candidats créés sont pertinents vis-à-vis du nuage puisque 92% possèdent un support supérieur ou égal à 0.1666 (le support attendu étant de $1/6 \approx 0.16666$).

Les approches existantes ne génèrent qu'une petite quantité de candidats pertinents : la méthode naïve crée massivement des cylindres incohérents, tandis que l'approche localisée n'est pas suffisamment précise (on note toutefois qu'elle améliore globalement la qualité des candidats créés par rapport à l'approche naïve).

Il est intéressant de constater que le choix d'ignorer les orientations de normales n'est pas forcément handicapant. En effet, les divers traitements effectués (notamment l'estimation du cercle de référence proposé en section 5.2.2) com-

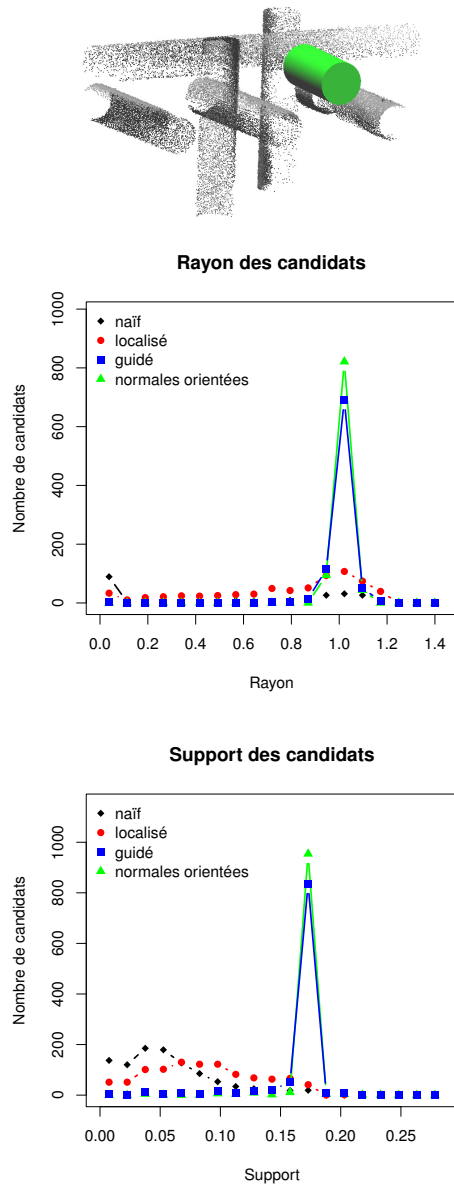


Figure 4: Résultats des méthodes de génération sur une scène synthétique simple bruitée. De haut en bas : la première vignette présente le nuage de points et le cylindre *a priori* utilisé. La seconde montre la répartition des rayons des candidats générés, pour chacune des méthodes testées, tandis que la dernière présente la distribution des supports des candidats. On constate aisément que l'approche que nous proposons produit une quantité importante de candidats pertinents (pics au dessus des valeurs attendues qui sont de 1 pour les rayons et 0.1666 pour les supports) contrairement aux approches existantes.

binés à un paramètre σ_r relativement permissif permettent de passer outre ce problème pour la plupart des candidats.

Comme nous l'avons précédemment mentionné, notre méthode est significativement plus lente dans la génération d'un candidat que ne le sont les autres approches. On constate ainsi, sur processeur dual core 1.73Ghz, que le RANSAC naïf produit instantanément 1000 candidats (sans calcul du support), tandis que notre approche requiert 20 secondes (avec $k = 20$ et un nuage de 42000 points). Cependant, au vu de la qualité des résultats fournis, on peut estimer qu'un faible nombre d'essais suffit à assurer la détection des cylindres, contrairement aux approches rapides.

6.2. Illustration sur une scène industrielle complexe

Afin d'évaluer la capacité de notre méthode à reconnaître les formes qui lui correspondent, nous présentons dans cette section les résultats obtenus sur un nuage de points issu d'acquisition laser sur site industriel. Cette fois encore, le cylindre servant d'*a priori* est extrait du nuage de points, puis décalé, désorienté et redimensionné. La figure 5 montre les distributions des paramètres (rayon, position et direction) des candidats générés par notre méthode. Le cylindre *a priori* ainsi que la forme à partir de laquelle il a été créé sont marqués dans chacun des espaces de paramètres.

Globalement, on constate que les candidats produits sont à la fois semblables les uns par rapport aux autres (les cylindres sont groupés dans les espace de paramètres représentés), et semblables à l'*a priori*. La plupart des cylindres générés vérifient donc, vis-à-vis de l'*a priori*, l'hypothèse de stabilité des paramètres formulée en section 3. Ils sont donc potentiellement en correspondance avec cet *a priori*. Le fait qu'ils soient relativement semblables les uns par rapport aux autres confirme que la méthode tend à générer des candidats cohérents.

Finalement, on peut constater sur la visualisation simultanée du nuage et des meilleurs candidats générés que la méthode trouve la plupart des cylindres qu'on aurait pu attendre en résultat. La méthode est donc apte, sur le cas présenté en tout cas, à traiter convenablement des données industrielles.

7. Conclusion et perspectives

Étant donné un cylindre *a priori*, l'approche que nous proposons permet la détection dans un nuage de points des cylindres qui lui sont semblables à une légère transformation près. Cette méthode améliore les approches aléatoires existantes en y intégrant la connaissance préalable disponible. Nous avons montré, sur des exemples expérimentaux, que la majorité des candidats générés par notre méthode sont cohérents vis-à-vis du modèle *a priori* et du nuage de points. En effet, la démarche probabiliste mise en œuvre permet l'utilisation d'une information à la fois pertinente et abondante issue du nuage de points, qui permet d'augmenter considérablement la fiabilité et la précision des formes créées. Le fait

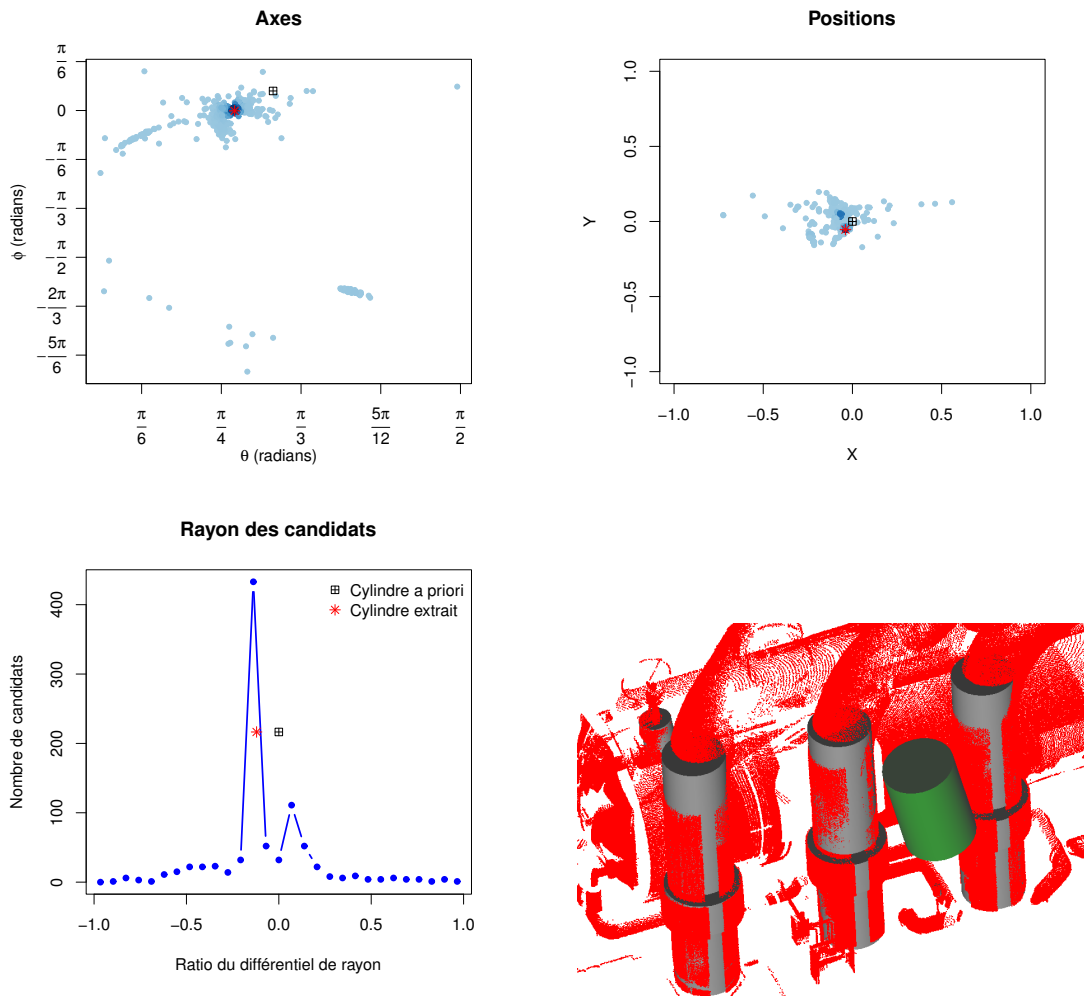


Figure 5: Distribution des paramètres de 1000 candidats générés sur une scène industrielle. Chacun des trois graphiques montre la répartition des points dans l'espace de paramètre qu'il représente. Les rayons sont représentés sous forme de ratio (moins 1) par rapport au rayon du cylindre a priori. Dans le cas des positions et des directions (axes), la couleur des points figure leur densité (plus ils sont sombres, plus la zone dans laquelle ils se trouvent est dense). L'étoile représente le cylindre à partir duquel l'a priori a été créé ("cylindre extrait"), le carré représentant l'a priori. La dernière vignette présente la superposition du nuage et des meilleurs candidats générés (déterminés par un algorithme de gestion des conflits entre les candidats). Le cylindre a priori utilisé y figure au premier plan.

que la méthode puisse être indépendante de l'orientation des normales constitue un atout non négligeable.

Malgré la lenteur de la génération des candidats, cette approche reste tout de même efficace dans la mesure où elle peut fournir des résultats satisfaisants avec un nombre minimal d'essais. Quoiqu'il en soit, en vue d'une exploitation industrielle, la priorité doit être accordée à la fiabilité du résultat plutôt qu'à la rapidité d'exécution des algorithmes mis en œuvre. En effet, la vérification des résultats effectuée par

l'opérateur humain est autrement plus coûteuse que l'exécution du code par la machine.

Concernant l'utilisation de cette méthode, elle doit s'inscrire dans le cadre d'une approche élaborée qui permette la gestion intelligente des candidats générés. Il s'agit en effet de proposer une solution permettant de régir les interactions entre les différents candidats ainsi que leur assemblage en un modèle 3D consistant. Nous envisageons pour cela d'utiliser les contraintes présentes dans le modèle 3D *a priori*

(typiquement le chaînage des cylindres en lignes de tuyauterie).

Globalement, le "générateur de cylindre" présenté dans cet article peut être perçu comme un échantillonneur pertinent de l'espace des configurations de modèles 3D, et peut donc s'intégrer dans le cadre d'une approche de Monte Carlo. Le problème de la recherche du modèle 3D s'ajustant au mieux sur les observations (nuage de points \mathcal{P}) pourrait se formuler dans le paradigme Bayésien. Étant donné un modèle 3D *a priori* et une incertitude sur ce modèle (en l'occurrence, les paramètres α , σ_c et σ_r que nous avons présentés fournissent une telle description), il s'agit de trouver la configuration X qui maximise la probabilité *a posteriori* :

$$\pi(X|\mathcal{P}) = \frac{P(\mathcal{P}|X).P(X)}{\int P(\mathcal{P}|X).P(X)dX}$$

où $P(\mathcal{P}|X)$ quantifie l'adéquation entre les observations et la configuration retenue, et $P(X)$ est la distribution *a priori* associée à l'espace des configurations (elle est définie par la donnée d'une instance de modèle *a priori* et des incertitudes qui s'y rattachent). Dans une telle situation, l'échantillonneur que nous proposons permet de fournir des configurations pertinentes qui ménagent simultanément les quantités $P(\mathcal{P}|X)$ (les cylindres générés sont conformes au nuage) et $P(X)$ (les cylindres générés sont conformes à l'*a priori*), et devrait donc permettre une convergence "rapide" vers l'optimum recherché.

Cette démarche doit être approfondie et fera l'objet de notre attention dans nos prochains travaux.

8. Remerciements

Nous remercions l'atelier CAO de l'Unité Technique Opérationnelle d'EDF, qui nous a fourni les données industrielles présentées dans cet article et autorisé à les diffuser, ainsi que Ruwen SCHNABEL et Liu YU pour leurs réponses à nos questions.

References

- [Ahn08] AHN S. J. : Geometric Fitting of Parametric Curves and Surfaces. *Journal of Information Processing Systems* (2008). 2
- [BF06] BEDER C., FORSTNER W. : Direct solutions for computing cylinders from minimal sets of 3D points. *Lecture Notes in Computer Science 3951* (2006), 135–6
- [BKV*02] BENKO P., KÓS G., VÁRADY T., ANDOR L., MARTIN R. : Constrained fitting in reverse engineering. *Computer Aided Geometric Design 19*, 3 (2002), 173–205. 2
- [Bos10] BOSCHÉ F. : Automated Recognition of 3D CAD Model Objects and Calculation of As-built Dimensions for Dimensional Compliance Control in Construction. *Elsevier Journal of Advanced Engineering Informatics* (2010). 2
- [Cha02] CHAPERON T. : *Segmentation de nuages de points pour la modélisation automatique d'environnements industriels numérisés*. PhD thesis, Ecole des mines de Paris, 2002. 2, 6, 7
- [GG04] GELFAND N., GUIBAS L. J. : Shape Segmentation Using Local Slippage Analysis. *Eurographics Symposium on Geometry Processing* (2004). 2
- [GSH*07] GAL R., SHAMIR A., HASSNER T., PAULY M., COHEN-OR D. : Surface Reconstruction using Local Shape Priors. *Proceedings of the fifth Eurographics symposium on Geometry processing* (2007), 253–262. 2
- [JWB*06] JENKE P., WAND M., BOKELOH M., SCHILLING A., STRASSER W. : Bayesian Point cloud Reconstruction. *Computer Graphics Forum* (2006). 2
- [LMM97] LUKACS G., MARSHALL D., MARTIN R. : *Geometric least-squares fitting of spheres, cylinders, cones and tori*. Tech. rep., University of Wales, 1997. 2
- [MLM01] MARSHALL D., LUKACS G., MARTIN R. : Robust Segmentation of Primitives from Range Data in the Presence of Geometric Degeneracy. *IEEE PAMI 23* (2001), 304–314. 2, 7
- [MN04] MITRA N. J., NGUYEN A. : Estimating Surface Normals in Noisy Point Cloud Data. *Special Issue of International Journal of Computational Geometry and Applications 14* (2004), 261–276. 3
- [PMG*05] PAULY M., MITRA N. J., GIESEN J., GROSS M., GUIBAS L. J. : Example-Based 3D Scan Completion. *Proceedings of the third Eurographics symposium on Geometry processing* (2005). 2
- [Rv05] RABBANI T., VAN DEN HEUVEL F. : Efficient Hough Transform for Automatic Detection of Cylinders in Point Clouds. *Proceedings of the 11th Annual Conference of the Advanced School for Computing and Imaging (ASCI '05)* (2005). 3, 6
- [SWK07] SCHNABEL R., WAHL R., KLEIN R. : Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum* (2007). 3, 4, 7
- [TZ00] TORR P. H. S., ZISSERMAN A. : MLESAC : A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding* (2000). 6
- [VS08] VEDALDI A., SOATTO S. : Quick shift and kernel methods for mode seeking. *Computer Vision-ECCV 2008* (2008), 705–718. 5
- [WKWL02] WOO H., KANG E., WANG S., LEE K. H. : A new Segmentation Method for Points Cloud Data. *International Journal of Machine Tools & Manufacture 42* (2002), 167–178. 2
- [YY08] YU L., YOULUN X. : Automatic segmentation of unorganized noisy point clouds based on the Gaussian map. *Computer Aided Design 40*, 5 (2008), 576–594. 2, 6