



**Revue d'article**

# **Improving 3D Lidar Point Cloud Registration Using Optimal Neighborhood Knowledge**

**Lucas Fourest**

Master IASD - Université PSL

March 21, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Contribution originale</b>	<b>1</b>
2.1	Voisinage optimal et descripteurs géométriques . . . . .	1
2.2	Optimisation de l'ICP . . . . .	2
2.2.1	Etape de sélection . . . . .	3
2.2.2	Etape de pondération . . . . .	3
2.2.3	Etape de rejet . . . . .	4
<b>3</b>	<b>Idées et expériences personnelles</b>	<b>4</b>
3.1	Propositions d'améliorations . . . . .	4
3.2	Implémentation et expériences . . . . .	5
3.2.1	Détermination des voisinages optimaux . . . . .	5
3.2.2	Datasets et expériences . . . . .	5
<b>4</b>	<b>Résultats et discussion</b>	<b>6</b>
<b>5</b>	<b>Analyse critique et idées futures</b>	<b>8</b>

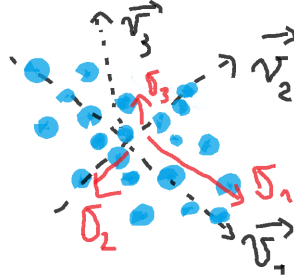
# 1 Introduction

Dans ce rapport nous présentons notre revue de l'article "Improving 3D Lidar Point Cloud Registration Using Optimal Neighborhood Knowledge" [1], abordant des potentielles améliorations de l'algorithme *Iterative Closest Point (ICP)* basés sur l'existence d'un *voisinage optimal* propre chaque point et une variété de descripteurs géométriques les caractérisant. Nous nous attacherons d'abord à résumer la contribution originale et à retranscrire de manière concise les résultats globalement observés, suite à quoi nous expliciterons nos idées et expériences personnelles ainsi que leur résultats afin de les comparer avec ceux de l'auteur, puis nous finirons par amener notre point de vue extérieur critique sur l'article. Avant toute chose, il est important de préciser que ce travail se place dans le contexte (revendiqué par l'auteur) de la *fine registration*, c'est-à-dire du recalage de paires bénéficiant d'un "bon" alignement *a priori*, par conséquent les écarts initiaux trop importants sont hors de notre cadre.

## 2 Contribution originale

### 2.1 Voisinage optimal et descripteurs géométriques

Pour un point  $P$  donné, notons  $\mathcal{V}_P^r$  son voisinage local sphérique de rayon  $r$ . La méthode de PCA appliquée à  $\mathcal{V}_P^r$  permet d'obtenir les valeurs propres  $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$  de la matrice de covariance du voisinage local (centré) en  $P$ , ainsi que les vecteurs propres associés  $v_1, v_2, v_3$ . Le voisinage local  $\mathcal{V}_P^r$  d'un point peut être (approximativement) vu comme une ellipsoïde, dont les  $v_i$  représentent les directions des 3 axes principaux et les  $\lambda_i$  la variance de la répartition des points selon chaque axe, i.e leur "intensité". De ce fait l'écart-type  $\sigma_1$  (resp.  $\sigma_2, \sigma_3$ ) représente la longueur du premier (resp. deuxième, troisième) axe principal de l'ellipsoïde  $\mathcal{V}_P^r$ , ou  $\sigma_i = \sqrt{\lambda_i}$ .



Dans ce contexte il est possible d'évaluer le caractère linéaire (1D), planaire (2D) et sphérique (3D) de la forme décrite par l'ellipsoïde du voisinage à l'aide des 3 grandeurs suivantes

$$a_{1D} = 1 - \frac{\sigma_2}{\sigma_1} \text{ (linearité) }, \quad a_{2D} = \frac{\sigma_2 - \sigma_3}{\sigma_1} \text{ (planarité) }, \quad a_{3D} = \frac{\sigma_3}{\sigma_1} \text{ (sphéricité) }$$

Ces 3 quantités (notons que  $0 \leq a_{kD} \leq 1$ ), normalisées de telle sorte à ce que  $a_{1D} + a_{2D} + a_{3D} = 1$ , représentent une distribution de probabilité  $p^r$  ( $p^r(d = k) = a_{kD}$ ) qui permet de quantifier, pour le rayon de voisinage choisi  $r$ , "à quel point"  $P$  appartient à chaque catégorie (1D, 2D ou 3D) et de lui attribuer un label de dimension. Le concept de voisinage optimal consiste à trouver  $r^* = r_P^*$  qui "maximisera le niveau de décision" et l'information apportée par  $p^r$ . Il faut pour cela trouver le paramètre  $r$  qui minimisera l'entropie de Shannon de cette distribution  $E_f(V_P^r) = -a_{1D}\ln(a_{1D}) - a_{2D}\ln(a_{2D}) - a_{3D}\ln(a_{3D})$

$$r_P^* = \arg \min_{r \in [r_{min}, r_{max}]} (E_f(V_P^r))$$

où l'intervalle de recherche  $[r_{min}, r_{max}]$  reste à déterminer. Le voisinage optimal  $\mathcal{V}^* = \mathcal{V}_P^{r^*}$  sera ensuite utilisé pour attribuer un label de dimension  $d^* = \arg \max_{k \in \{1,2,3\}} (a_{kD})$  à  $P$  avec un niveau de confiance "maximal": il s'agit du choix de voisinage local qui dégage le mieux l'une des caractéristiques et l'appartenance à une dimension donnée.

A partir de la connaissance de  $V^*$  et des  $\sigma_i$  associés, plusieurs descripteurs géométriques sont retenus pour chaque point, à savoir

$$\mathcal{G} = (n, r^*, d^*, E_f^*, V)$$

où  $n = \frac{v_3}{\|v_3\|}$  et  $V = \prod_{i=1}^3 \sigma_i$ . Intuitivement,  $n$  (direction unitaire du troisième axe principal) correspond à la normale du voisinage local lorsque celui-ci peut-être approché par un plan, d'où sa notation. Quant à  $V$ , il s'agit d'une grandeur nommée *l'omnivariance* qui est proportionnelle au volume de l'ellipsoïde décrite par  $\mathcal{V}^*$ , qui permet encore une fois de caractériser la forme du voisinage. Toutes ces caractéristiques/descripteurs géométriques, dérivés du voisinage optimal, sont à la base des améliorations de l'ICP qui vont suivre, proposées par l'auteur.

## 2.2 Optimisation de l'ICP

L'algorithme d'ICP vise à aligner un nuage de point mobile *data* à un nuage de référence *ref* (immobile et supposé en grande partie similaire) en lui appliquant une succession de transformations spatiales  $(R_i, t_i)$ , jusqu'à ce que les parties correspondantes se superposent. Son déroulement repose sur plusieurs étapes clés qui ouvrent chacune la possibilité à une multitude de variations.

La **sélection** est une étape préliminaire qui consiste à sous-échantillonner le nuage *data* (et possiblement *ref*) en vue de la suite de l'algorithme. Elle vise entre autres à réduire le temps de calcul global, mais peut également être utile pour sélectionner les "meilleurs candidats" et/ou éliminer le bruit du processus. Il s'agit d'une étape cruciale car c'est la que sont choisis les points "définitifs" pour l'appariement et le calcul des  $(R_i, t_i)$ .

L'**appariement** est la première étape effectuée à chaque itération, elle consiste à associer chaque point de *data* à un point de *ref*. Le but est que les paires soient correspondantes, mais aucun élément explicite ne permet de trouver ces correspondances, de plus certains points de *data* peuvent ne pas avoir de correspondant dans *ref* (et inversement). Sous l'hypothèse d'un bon alignement a priori (cf fine registration), une méthode commune et heuristique efficace consiste à choisir pour chaque point de *data* sont plus proche voisin dans *ref*: c'est l'ICP "point par point". L'auteur reste dans ce cadre pour cette étape, aucune variation ne sera proposée ici.

La **pondération** consiste à attribuer des poids à chaque paire de points, qui serviront ultérieurement dans le calcul de  $(R, t)$  via la minimisation de l'erreur de transformation (cf après). Par défaut, les poids sont identiques pour chaque paire (donnant l'erreur quadratique originale). Cette étape peut être suivie par celle **derejet** qui consiste à écarter les "pires" paires afin qu'elles ne soient pas prises en compte dans la minimisation de l'erreur, sur la base par exemple de comparaison avec des statistiques globales portant sur tout le nuage de points.

La dernière étape d'une itération est la **minimisation** de l'erreur de recalage, qui donne la transformation optimale  $(R, t)$  à appliquer au nuage mobile. En toute généralité, en prenant en compte la pondération, l'erreur à minimiser est

$$f_w(R, t) = \sum_i w_i \|p_i - (R \times p'_i + t)\|^2$$

La solution analytique à ce problème de minimisation de l'erreur quadratique pondérée est donnée par la décomposition SVD de  $C = \frac{\sum_i w_i p'_i (p_i)^T}{\sum_i w_i} - p'_m (p_m)^T$  [2] où  $p_m = \frac{\sum_i w_i p_i}{\sum_i w_i}$ ,  $p'_m = \frac{\sum_i w_i p'_i}{\sum_i w_i}$

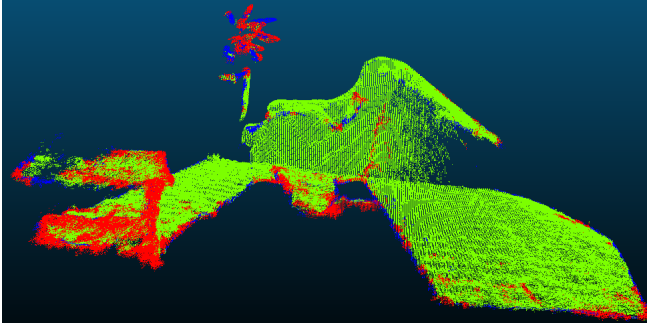
$$\begin{cases} R = VU^T \in \mathbb{R}^{3 \times 3} \\ t = p_m - Rp'_m \in \mathbb{R}^3 \end{cases}$$

On observe qu'avec des poids constants on retrouve bien  $C \propto \sum_i (p'_i - p'_m)(p_i - p_m)^T$  ce qui correspond bien à la matrice de covariance menant à la solution dans le cas classique non pondéré [3]

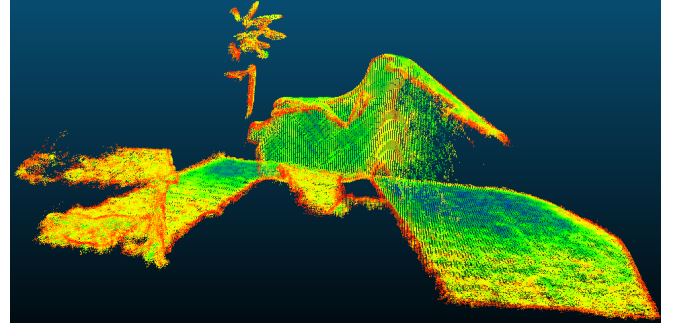
Mis à part la sélection qui est préliminaire, ces étapes sont répétées jusqu'à atteinte du critère d'arrêt (nombre d'itérations maximal et/ou seuil d'erreur). L'auteur préconise des méthodes spécifiques pour certaines d'entre elles, ayant recours au descripteurs géométriques déterminés au préalable avec les voisinages optimaux [1] (à la fois dans *data* et *ref*). Le but est de mettre en avant la "richesse" des voisinages optimaux en terme d'information apportée sur chaque point au profit de l'ICP.

### 2.2.1 Etape de sélection

- Une première idée est d'utiliser le critère de l'entropie  $E_f^*$ , qui décrit le niveau de certitude que l'on a sur un point quant à la forme de son voisinage, pour sélectionner les "meilleurs candidats" à garder. Intuitivement, plus  $E_f^*$  est petit plus la caractérisation est confiante avec une des trois dimensions qui ressort et plus la géométrie locale est suffisamment simple pour permettre une décision claire: on gardera ces points là car ils correspondent à des régions saillantes et/ou "clairement identifiables" de l'objet sous-jacent, qui ont plus de chances d'apparaître clairement à la fois dans *ref* et *data*. Ainsi  $E_f^* < E_0$  pourra être un premier critère de sélection



(a) Visualisation du label dimension sur un exemple 3.2.2 (bleu:  $d = 1$ , vert:  $d = 2$ , rouge:  $d = 3$ )



(b) Visualisation du champ scalaire de l'entropie sur un exemple 3.2.2 (bleue: basse, rouge: haute)

- L'auteur fait également le constat que les régions linéaires (1D) correspondent souvent à des extrémités, frontières entre surfaces et/ou à des objets fins 1a. Hors ces régions problématiques peuvent être à l'origine d'ambiguïtés/instabilités, notamment lors du processus d'appariement ou elles ont de grandes chances d'introduire de fausses correspondances. On peut tirer la même conclusion pour les régions (très) dispersés (3D) 1a, comme la végétation qui peut avoir un échantillonnage différent selon le point d'acquisition. Un autre critère de sélection pourrait donc être de ne garder que les points de dimension  $d^* = 2$ .

### 2.2.2 Etape de pondération

Plusieurs choix existent pour pondérer une paire de points  $(P_1, P_2)$ . Au delà des simples poids constants (absence de pondération), on peut favoriser les paires de points les plus proches ou bien s'appuyer sur les disparités en termes de caractéristiques géométriques pour en tirer plusieurs alternatives:

- Pondération euclidienne:

$$w(P_1, P_2) = 1 - \frac{d(P_1, P_2)}{d^{max}}$$

où  $d(P_1, P_2) = \|P_1 - P_2\|_2$  et  $d^{max}$  la distance maximale observée entre toutes les paires. Cette méthode repose sur l'idée sous-jacente qu'une paire de points "proches" a plus de chances d'être correspondante (cf fine registration) et ne fait pas appel au descripteurs géométriques.

- Pondération via l'omnivariance:

$$w_V(P_1, P_2) = 1 - \frac{d_V(P_1, P_2)}{d_V^{max}}$$

où  $d_V(P_1, P_2) = |V_{P_1} - V_{P_2}|$ . Cette méthode prend en compte l'écart de volume entre les ellipsoïdes locales associés aux deux points, supposées identiques si les points sont correspondants.

- Pondération via la compatibilité des normales:

$$w(P_1, P_2) = |\vec{n}_{P_1} \cdot \vec{n}_{P_2}|$$

### 2.2.3 Etape de rejet

Pour cette étape, l’auteur propose de manière similaire à la pondération une méthode basée sur des critères de distance et/ou compatibilité des caractéristiques géométriques, à savoir:

- Rejeter les  $n\%$  de paires ayant la plus grande distance euclidienne  $d(P_1, P_2)$
- Rejeter les  $n\%$  de paires ayant la plus grande incompatibilité d’omnivariance  $d_V(P_1, P_2)$

D’autres alternatives (seuil absolu,  $d > \bar{d} + 2.5\sigma$  ...) existent et sont mentionnées mais ne sont pas considérées ici.

## 3 Idées et expériences personnelles

### 3.1 Propositions d’améliorations

Nous allons désormais aborder quelques points pour lesquels nous voyons des modifications ou améliorations potentielles judicieuses à apporter.

- Recherche du **voisinage optimal**:

L’auteur se base sur un paramètre de rayon  $r \in [r_{min}, r_{max}]$  pour la recherche du voisinage (sphérique) optimale d’un point donné. Cependant, comme nous l’avons mentionné 2.1, les bornes restent à déterminer. En effet, l’intervalle de recherche doit être adapté à la dimension du nuage de points. Qui plus est, les nuages eux-mêmes peuvent présenter des disparités locales importantes en terme de densité de points: la distance caractéristiques entre voisins peut significativement varier d’un endroit à l’autre du nuage, il serait donc difficile de trouver des bornes statiques robustes sur tout le nuage.

Dans ce contexte, nous préférons directement raisonner en terme de nombre de voisins plutôt que de rayon. En effet, faire la requête des *k plus proches voisins (kNN)* (plutôt que des voisins contenus dans la sphère de rayon  $r$ ) nous assure directement de trouver un voisinage du point avec précisément  $k$  individus, il suffit alors que  $k$  soit "assez petit" par rapport au nombre total de points dans le nuage  $N^{total}$  pour assurer le caractère local du voisinage quel que soit le point de référence. La pertinence du paramètre  $k$  ne dépend alors plus que de la densité globale du nuage  $N^{total}$ : tester les petits pourcentages de cette valeur en guise de paramètre  $k$  peut être une idée. Nous choisissons donc arbitrairement l’intervalle de recherche:

$$k_{min} = \lfloor \alpha_{min} * N^{total} \rfloor, \quad k_{max} = \lfloor \alpha_{max} * N^{total} \rfloor$$

ou  $\alpha_{min} =$  et  $\alpha_{max} =$ . Nous testerons  $n = 5$  valeurs de  $k$  uniformément réparties dans cette intervalle.

- Etape d’**appariement**:

L’article original [1] reste dans le cadre de l’ICP point par point pour la recherche des paires, qui consiste à appairer chaque point de *data* avec son plus proche voisin dans *ref*. Cependant nous voyons encore ici l’opportunité d’utiliser les descripteurs géométriques dont on dispose, en particulier l’omnivariance  $V$ , comme élément d’information supplémentaire pour optimiser nos chances de trouver des (vraies) paires. En effet, nous pouvons à nouveau utiliser  $d_V(P_1, P_2)$  comme mesure de compatibilité: pour tout point de *data*, notre idée consiste à chercher ses  $k$  plus proches voisins dans *ref* et retenir celui ayant la plus grande compatibilité:

$$\forall P_1 \in data, P_2 = ref[i^*] \text{ ou } i^* = \arg \min_{i \in \mathcal{P}_k} (d_V(P_1, ref[i]) \text{ et } \mathcal{P}_k = kNN(P_1, ref)$$

Une telle méthode devrait selon notre intuition favoriser la correspondance des paires  $(P_1, P_2)$ , typiquement dans le scénario ou il y a un défaut d’alignement et que le (vrai) point correspondant dans *ref* se situe dans les  $k$  plus proches voisins de  $P_1$ , sans être le plus proche (auquel cas les descripteurs géométriques sont un critère de distinction).

## 3.2 Implémentation et expériences

Dans cette section nous allons clarifier certains points de notre implémentation concernant la détermination préalable des voisinages optimaux. Etant donné qu'il s'agit d'une étape cruciale dont tout le reste du travail découle, nous nous concentrerons sur cette partie et n'aborderons pas les éléments de code relatifs à l'algorithme de l'ICP en lui-même ici, car (moyennant l'ajout de quelques fonctions pour l'amélioration des étapes) il reste très similaire au code fourni dans le TP2 du cours, que nous avons repris pour notre implémentation, et ne semble donc pas nécessiter de clarifications particulières. Dans un deuxième temps, nous expliciterons nos choix de datasets et les expériences que nous avons conduit.

### 3.2.1 Détermination des voisinages optimaux

Afin de faire comprendre plus clairement la manière dont les voisinages optimaux sont au préalable déterminés dans notre implémentation (au sein du fichier `descripteurs.py`), le pseudo code "intuitif" de la méthode `OPTIMAL_NEIGHBORHOODS` est fourni 2. Celle-ci garde simplement un suivi pour chaque point de la plus petite entropie trouvée jusque ici (et son paramètre associé  $k^*$ ). On itère sur les différentes valeurs de  $K$  en les testant simultanément sur tout les points, ce qui permet de faire l'ensemble des opérations de manière vectorielle (important pour le temps d'exécution), néanmoins chaque point termine avec son propre paramètre optimal  $k^* \in K$  et les descripteurs géométriques associés.

Le reste du code et des fonctions de `descriptors.py` (dont la fonction `COMPUTE_FEATURES`) est principalement repris du TP3 portant sur les voisinages locaux (dont le calcul des normales et des descripteurs).

---

#### 1 OPTIMAL\_NEIGHBORHOODS

---

**Require:** cloud,  $K = \{k_1, \dots, k_n\}$

```

1:  $k^*, n, V, d^* \leftarrow \text{INIT\_ZEROS}()$ 
2:  $E^* \leftarrow \infty \times \text{INIT\_ONES}()$ 
3:  $\text{tree} \leftarrow \text{OCTREE}(\text{cloud})$ 
4: for  $k$  in  $K$  do
5:    $\text{normals}, a1d, a2d, a3d, \text{volume} \leftarrow \text{COMPUTE\_FEATURES}(\text{cloud}, \text{tree}, k)$ 
6:    $E = -a1d \cdot \log(a1d) - a2d \cdot \log(a2d) - a3d \cdot \log(a3d)$ 
7:    $\text{change\_mask} \leftarrow \text{np.where}(E < E^*)$   $\triangleright$  index pour lesquels  $k$  est le nouveau meilleur voisinage
8:    $E^*[\text{change\_mask}], k^*[\text{change\_mask}] \leftarrow E[\text{change\_mask}], k$ 
9:    $d^*[\text{change\_mask}] \leftarrow \text{np.argmax}(\text{np.hstack}(a1d, a2d, a3d)[\text{change\_mask}, :], \text{axis} = -1)$ 
10:   $V[\text{change\_mask}], n[\text{change\_mask}] \leftarrow \text{volume}[\text{change\_mask}], \text{normals}[\text{change\_mask}]$ 
11: end for
```

---



---

#### 2 COMPUTE\_FEATURES

---

**Require:** cloud, tree,  $k$

```

1:  $\text{eigenvalues}, \text{eigenvectors} \leftarrow \text{LOCAL\_PCA}(\text{cloud}, \text{tree}, \text{n\_neighbors}=k)$ 
2:  $\text{normals} \leftarrow \text{eigenvectors}[:, :, 0]$ 
3:  $\lambda_3, \lambda_2, \lambda_1 \leftarrow \text{eigenvalues}[:, 0], \text{eigenvalues}[:, 1], \text{eigenvalues}[:, 2]$ 
4:  $\theta_1, \theta_2, \theta_3 \leftarrow \sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3}$ 
5:  $V \leftarrow \theta_1 \cdot \theta_2 \cdot \theta_3$ 
6:  $a1d, a2d, a3d \leftarrow \text{FEATURES}(\theta_1, \theta_2, \theta_3)$ 
7: return normals,  $a1d$ ,  $a2d$ ,  $a3d$ ,  $V$ 
```

---

### 3.2.2 Datasets et expériences

A présent nous allons détailler nos idées d'expériences ainsi que nos choix de datasets. A l'instar du papier original, nous souhaitons analyser l'influence marginale de chacune des améliorations d'étapes précédemment décrites 3.1, y compris celle que nous proposons pour l'appariement 3.1. Nous allons donc, pour chaque étape, lancer les différentes variantes de l'ICP et comparer leur résultats en terme d'erreur quadratique.

$$RMSE(\text{aligned}, \text{ref}) = \sum_{(p, p') \in \mathcal{P}} \|p - p'\|^2$$

L'étude des combinaisons de plusieurs variantes sera laissée à des travaux futurs. Pour nos expériences, nous devons veiller à rester dans le cadre de la fine registration et choisir des paires de nuages de points bénéficiant d'un bon alignement a priori. Il est néanmoins intéressant de faire en sorte de varier les formes caractéristiques mises en avant par chaque nuage (linéaire, planaire, ovoïde...), directement connectées aux descripteurs géométriques, afin d'observer leur influence sur les méthodes. On pourra alors choisir des objets compacts mettant d'avantage en avant les rondeurs\*\*, d'autres au caractère plus complexe présentant beaucoup d'aspérités\*\*, ou bien alors des scans plus "réalistes" de scènes entières composées d'une multitude d'objets (plantes...)\*\*\*. Après avoir exploré les datasets de la littérature appropriés à la tâche de recalage, nous avons retenu les éléments suivants (stockés au format `.ply`):

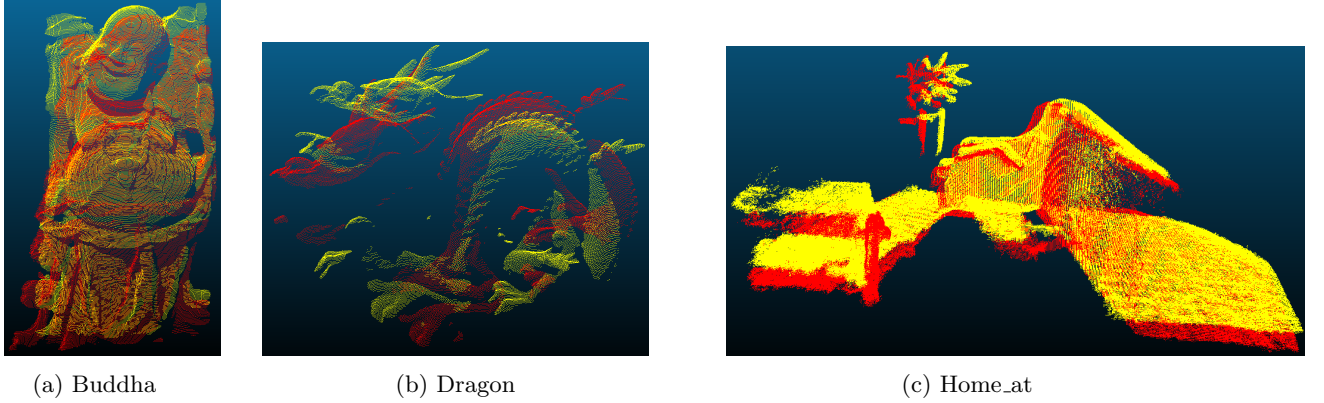


Figure 2: Visualisation des paires de nuages pour chaque dataset (jaune: *data*, rouge: *ref*)

- Buddha\*: paire (*data*, *ref*) identifiée (24,0) de `buddha/happy_side` (Stanford3D)
- Dragon\*\*: paire (*data*, *ref*) identifiée (192,168) de `dragon/dragon_side` (Stanford3D)
- Home\_at\*\*\*: paire (*data*, *ref*) identifiée (45,44) de `home_at` (3DMatch, Geometric Registration Benchmark)

	<i>data</i>	<i>ref</i>
Buddha	51266	59544
Dragon	23218	24557
Home_at	236914	260766

Table 1: Nombre de points des différents nuages retenus

## 4 Résultats et discussion

Ci-dessous nous fournissons par dataset les résultats des différentes variantes de chaque étape de l'ICP sous forme de courbes d'évolution temporelle de l'erreur d'alignement. Nous discuterons de l'influence marginale de chaque modification en terme de *RMSE* mais aussi de temps d'exécution, et nous pourrions comparer ces résultats avec ceux du papier original. Les critères d'arrêt de notre implémentation de l'ICP sont le seuil de *RMSE*  $T$  et le nombre d'itérations  $n_{iter}$ . Tout au long des expériences,  $T = 1e - 4$  et  $n_{iter}$  est précisé dans l'intitulé des sous-figures.

Nous constatons globalement que les étapes de matching, pondération et rejet n'apportent dans le cas de nos datasets aucun gain significatif que ce soit en terme d'erreur où de temps, voire une perte considérable. Pour ce dernier critère, les observations sont cohérentes car chacune de ces étapes implique des opérations supplémentaires (à chaque itération, calcul des  $k$  plus proches voisins et évaluation de leur compatibilité et/ou calcul des poids des paires et/ou détection des paires à rejeter), bien que le rejet soit censée alléger les calculs de l'étape de minimisation.

On notera tout de même que sur le dataset Dragon, dont les nuages comportent significativement moins de points que ceux des deux autres, certaines variantes de la pondération et du rejet sont telles que



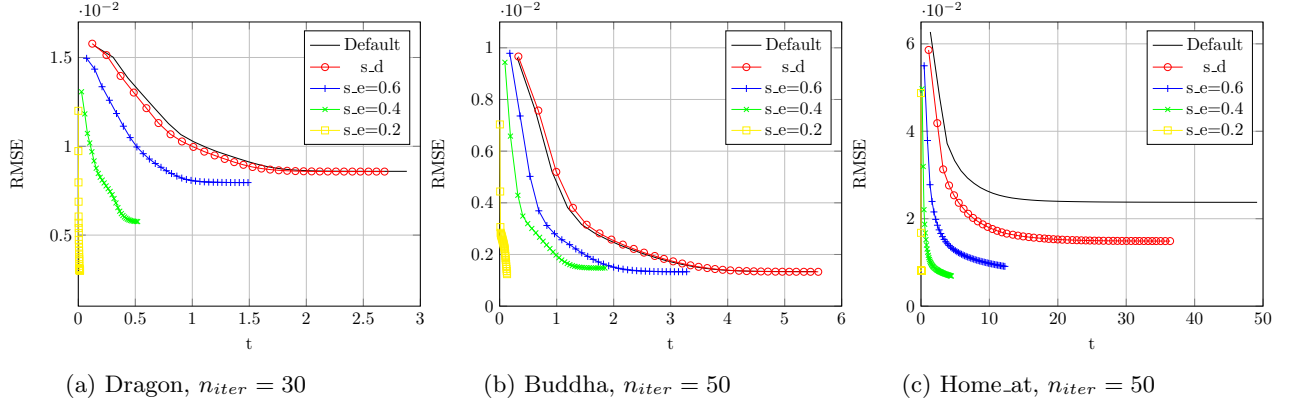


Figure 3: Selection step

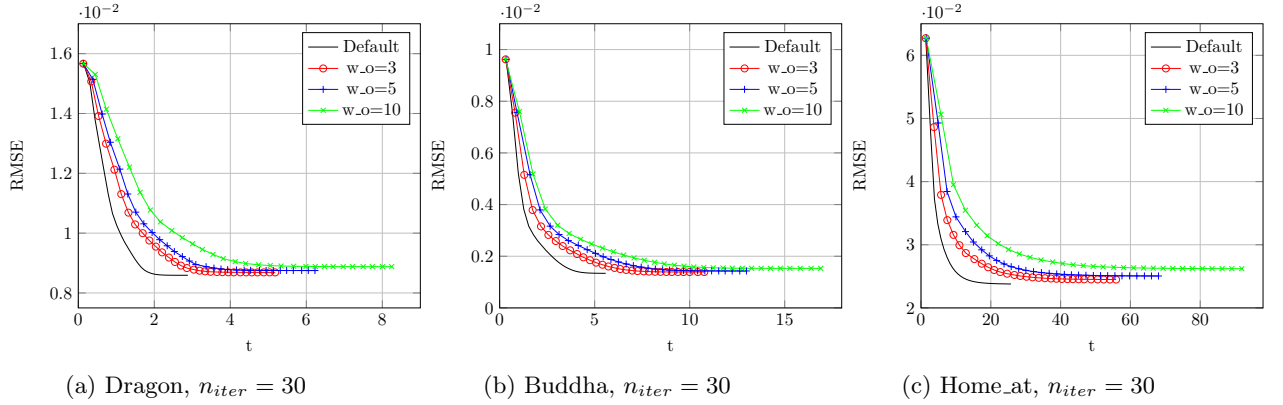


Figure 4: Matching step

l'erreur atteint son minimum sensiblement plus vite qu'avec l'ICP par défaut: le fait de mettre en avant ou rejeter des paires en amont de l'étape de minimisation semble légèrement accélérer la convergence avec des nuages à "peu" de points. En revanche, aucun avantage notable n'est à relever pour ces deux étapes sur Buddha et Home\_at, peut être que les "mauvaises paires" ont un impact relatif plus important avec peu de données. De manière globale, ces variantes entraînent au mieux une convergence un peu plus rapide, sinon dégradent les performances d'erreur et/ou de temps. Il y a en réalité un compromis entre le fait de dévaloriser et éliminer les pires paires et le fait de garder le plus de données possibles: plus de paires permettront intuitivement de mieux approcher la transformation mais d'un autre côté certaines peuvent introduire du bruit dans les calculs. Le dilemme semble pencher en faveur de la conservation équitable de l'intégralité des paires (ce qui est fait "par défaut") dans nos expériences. Enfin, la variante que nous avons imaginé pour l'étape de matching détériore grandement les performances de l'ICP sur tout les plans: notre contribution ne s'avère au final pas très pertinente.

Finalement, la seule étape qui apporte une amélioration significative à l'ICP dans nos expériences est celle de sélection, pour laquelle le bilan est très positif. La sélection des points sur le critère d'entropie minimale se démarque particulièrement. L'utilité de cette étape est à double tranchant: le sous-échantillonnage réduit drastiquement le temps de calcul et le fait de ne considérer que les zones de haute confiance assure la qualité des futures paires et correspondances à minimiser: une grande partie du bruit est éliminée d'emblée. Les performances et le gain lui-même semblent croître avec le seuil de sélection, jusqu'à des valeurs de filtrage radicalement grandes. Les bénéfices sont considérables à tout les niveaux, et qui plus est le temps ajouté par la sélection (effectuée une seule fois au début) est négligeable devant celui qu'elle nous fait gagner.

En comparaison, l'auteur ne fait pas les mêmes observations que nous, et ses résultats globaux mettent plutôt en avant le gain de temps apporté par la sélection (mais pas l'erreur minimisée) ainsi que l'étape de rejet (à tout les niveaux), tandis que celle de pondération n'est, comme dans notre cas, globalement pas concluante. Nous pensons que les décalages entre ses observations et les nôtres sont principalement dus à la nature des datasets utilisés: nos expériences ont recours à des datasets plutôt "simples" (scans

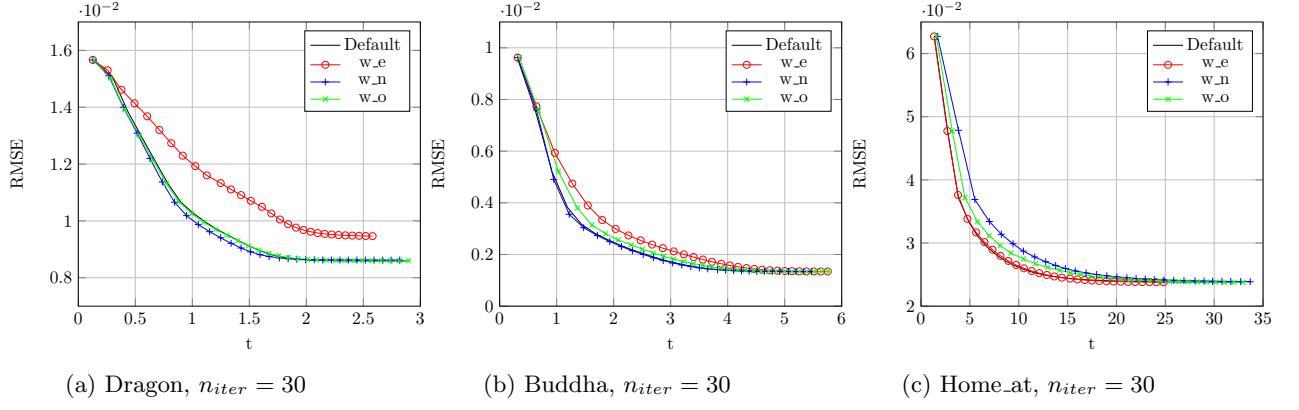


Figure 5: Weighting step

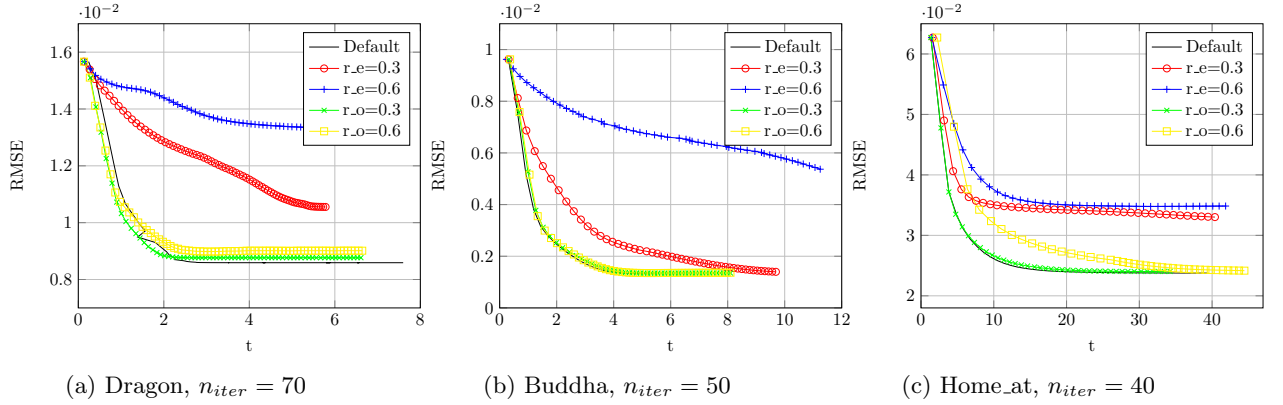


Figure 6: Rejecting step

d'objets de taille modérée et de l'intérieur d'une pièce) tandis que l'auteur opère sur des scans à l'échelle de bâtiments/zones urbaines (à l'exception de TLS) comportant plusieurs centaines de milliers de points. Les données sont donc plus lourdes mais aussi, de par la taille et la complexité des environnements de scan, plus propices au bruit et erreurs d'acquisition. Il est donc possible que cela favorise la présence de beaucoup de "points incertains" et donc la probabilité d'obtention de "mauvaises paires". En particulier, l'étape de rejet semble plus pertinente dans un cadre comme celui-ci, ce qui pourrait notamment expliquer les différences observées. Selon ce raisonnement, il en irait de pair pour la pondération, pourtant il semble que ce ne soit pas le cas. On remarque que les résultats sur TLS (scans d'intérieur, 20000 points) coïncident déjà plus avec les nôtres et notamment ceux de Home\_at, ce qui semble cohérent. Finalement, peut être que la contribution est plus pertinente dans le cadre de nuages de points "bruités" acquis dans un environnement extérieur complexe.

## 5 Analyse critique et idées futures

Après avoir discuté de nos résultats et les avoir mis en parallèle avec ceux de l'auteur, nous finirons par amener notre analyse critique des travaux de ses travaux, les limites des nôtres et nous terminerons par partager nos idées futures. Tout d'abord, le papier originale présente plusieurs ambiguïtés et/ou points à éclaircir que nous estimons nécessiter d'avantage d'explications. Premièrement, la méthode originale n'est pas transparente quant à la détermination ou le choix des bornes  $r_{min}$  et  $r_{max}$  pour la recherche du voisinage optimal, alors qu'il s'agit d'une étape cruciale de la méthode, notamment pour la reproductibilité des expériences. C'est pourquoi nous avons pris le soin de détailler cette étape plus en profondeur ici. Nous constatons également pour l'étape de sélection que l'auteur évoque un seuil minimal d'entropie (" $E_f^* > \dots$ ") alors même qu'il s'agit normalement d'un critère à minimiser selon la définition de celle-ci: nous ignorons si un élément de compréhension ou une subtilité nous échappe, mais ce processus de sélection n'est pas clair par rapport aux notions introduites plus haut. Enfin, nous nous

interrogeons quant à l'ordre présenté des étapes de pondération et de rejet, dans le sens où la méthode calcule à chaque itération le poids de paires qu'elle rejette (les étapes de pondération et de rejet ne sont pas incompatibles et peuvent en toute généralité être simultanément "actives"). Nous nous demandons alors si l'ordre inverse ne serait pas plus judicieux pour gagner du temps. Cependant, les poids obtenus ne seront pas les mêmes sans les paires rejetés, et il n'est pas évident de savoir si ils seront plus pertinent dans ce dernier cas (bien que nous en calculions moins). Encore une fois, nous pensons que cela aurait du être élucidé par l'auteur.

Nos travaux sont également discutables sur plusieurs points, entre autres par que nous utilisons notre propre erreur de minimisation  $\propto RMSE$ , qui est aussi notre métrique d'évaluation. Cependant l'auteur minimise la distance au plan et évalue ses performances avec la n-résolution  $R_n$ , qui s'apparente à une  $MAE$  lissée, utilisant la moyenne de la distance (à *ref*) des 5 plus proches voisins. Cette métrique est plus complète que la notre car elle tient compte du fait que certains points de *data* n'ont pas de correspondant dans *ref*. Ainsi tout cela limite la pertinence de nos comparaisons avec les résultats de l'auteur, car en réalité nous n'utilisons pas exactement les mêmes éléments (sans compter que nous procédons aussi différemment pour déterminer le voisinage optimal).

Dans de futurs travaux, nous pourrions aller plus loin que l'étude des influences marginales et explorer les combinaisons de variantes (de différentes étapes), afin de comprendre lesquelles sont compatibles voire synergiques. En particulier, il serait intéressant de comprendre leur lien avec les caractéristiques des nuages de points, et les éléments qui peuvent favoriser la méthode. Nous pourrions également, à l'image de ce qui a été dit précédemment, utiliser un panel de métriques plus élargi afin de compléter nos expériences.

## References

- [1] A. Gressin, C. Mallet, N. David., "Improving 3D Lidar Point Cloud Registration Using Optimal Neighborhood Knowledge", <https://isprs-annals.copernicus.org/articles/I-3/111/2012/isprsannals-I-3-111-2012.pdf>
- [2] K. Naus, L. Marche, "Use of a Weighted ICP Algorithm to Precisely Determine USV Movement Parameters", <https://www.mdpi.com/2076-3417/9/17/3530>
- [3] F. Goulette, "3DPoint Cloud and Modeling : Point Cloud Registration", <https://www.caor.minesparis.psl.eu/presentation/cours-npm3d/>