

Algorithmie Géométrique : Détection de composantes principales

Arnaud TANGUY
Jean-Dominique Favreau
University of Polytech'Nice-Sophia

2 mars 2014

Résumé

Durant ce projet, nous avons exploré des méthodes de détection de composantes principales. La méthode que nous avons sélectionnée est basée sur un accumulateur : la *sphère de Gauss*. Nous avons ensuite exploité les directions principales afin de chercher les plans principaux présent dans divers nuages de points.

Table des matières

1	Sphère de Gauss	2
1.1	Projection Stéréographique	2
1.2	Détection de plans	5
1.3	Détection de cylindres	6
2	Reconstruction de plans	7
3	Résultats	8
4	Détails d'implémentation	10
4.1	Sphère de Gauss sur GPU	10
5	Conclusion	11

1 Sphère de Gauss

Afin de détecter les directions principales, nous avons choisi de nous intéresser en particulier à un accumulateur : la *sphère de Gauss*. Le principe est simple : on détermine la normale à chaque point d'un nuage de point, et on accumule la direction de cette normale sur une sphère.

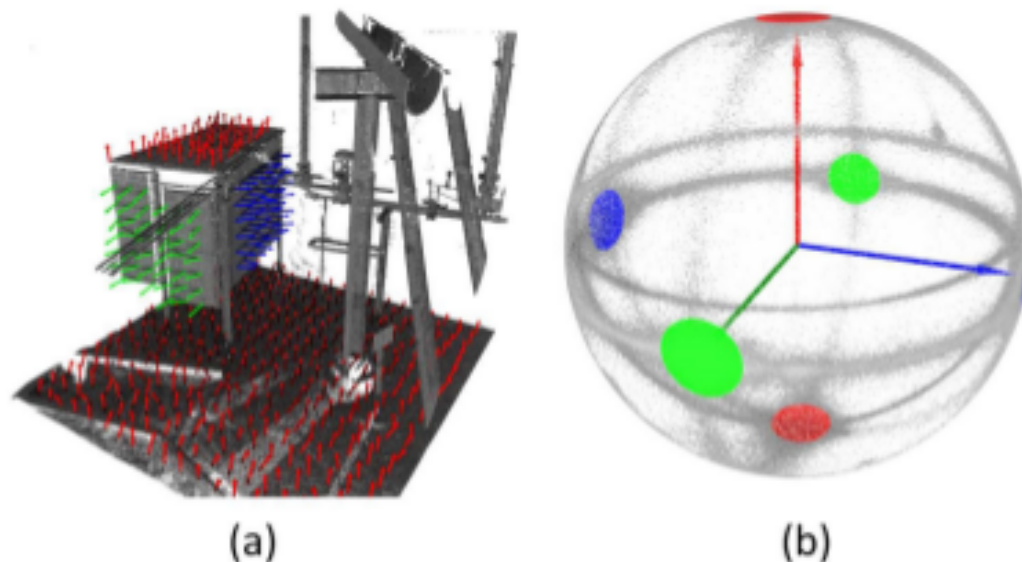


FIGURE 1 – Sphère de Gauss telle que présentée dans (Gau). (a) Visualisation des normales, (b) Projection des normales sur la sphère de Gauss.

1.1 Projection Stéréographique

Nous avons choisi d'accumuler les normales selon une projection stéréographique. Cette projection permet de représenter une sphère sur un plan en définissant la position des points projetés comme l'intersection de droites passant par les points de la sphère avec un plan. Dans la suite, on définit les coordonnées par rapport à un repère cartésien (x, y, z) centré en O : centre de la sphère de Gauss, définie comme une boule unité centrée en O .

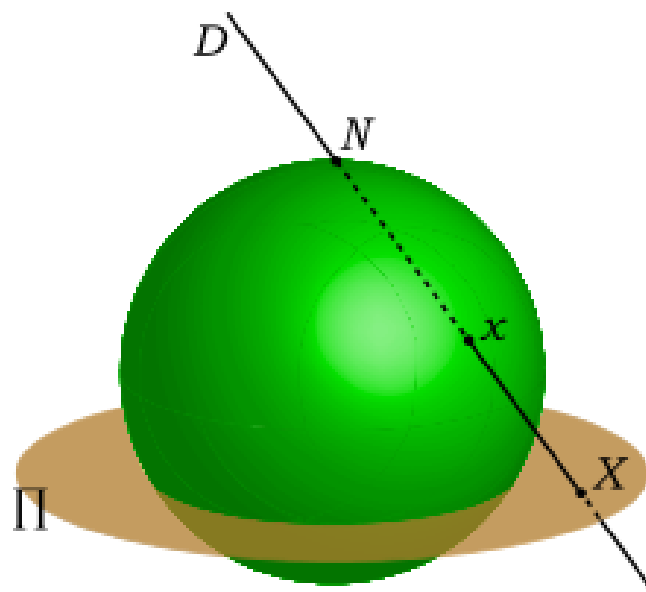


FIGURE 2 – Schéma de la projection stéréographique depuis le point N sur le plan Π

- On définit un point $N = \begin{pmatrix} 0 \\ 0 \\ \alpha \end{pmatrix}$ comme origine de la projection.
- Le point $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ coïncide avec les normales, pour peu que celles-ci aient été normalisées au préalable.
- Le point X représente le point de projection stéréographique sur le plan $\Pi : z = -\beta$
On a : $X = \begin{pmatrix} 0 \\ 0 \\ \alpha \end{pmatrix} + \lambda \begin{pmatrix} x_1 \\ x_2 \\ x_3 - \alpha \end{pmatrix}$ avec $\lambda = \frac{\alpha + \beta}{\alpha - x_3}$

En utilisant cette méthode, nous avons généré une projection de la sphère de Gauss en 2 hémisphère. En voici un exemple sur un nuage de point représentant un appartement.

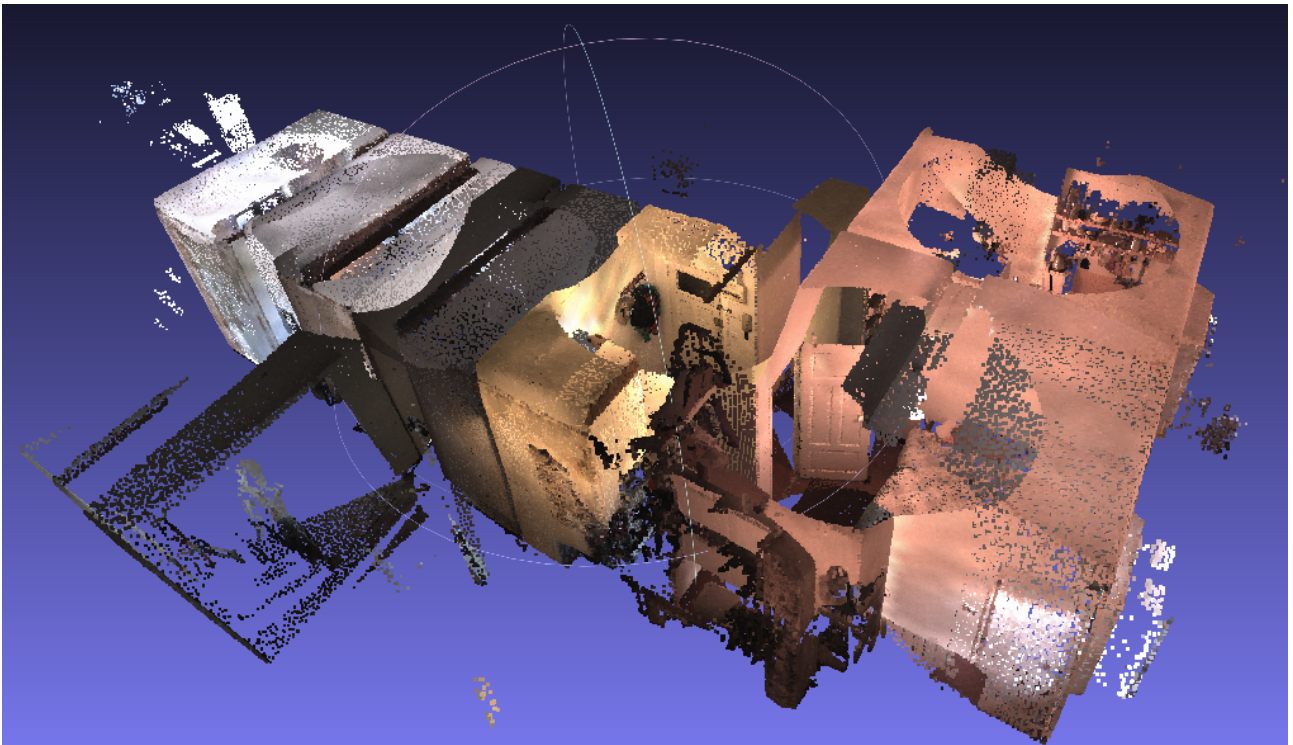


FIGURE 3 – Appartement

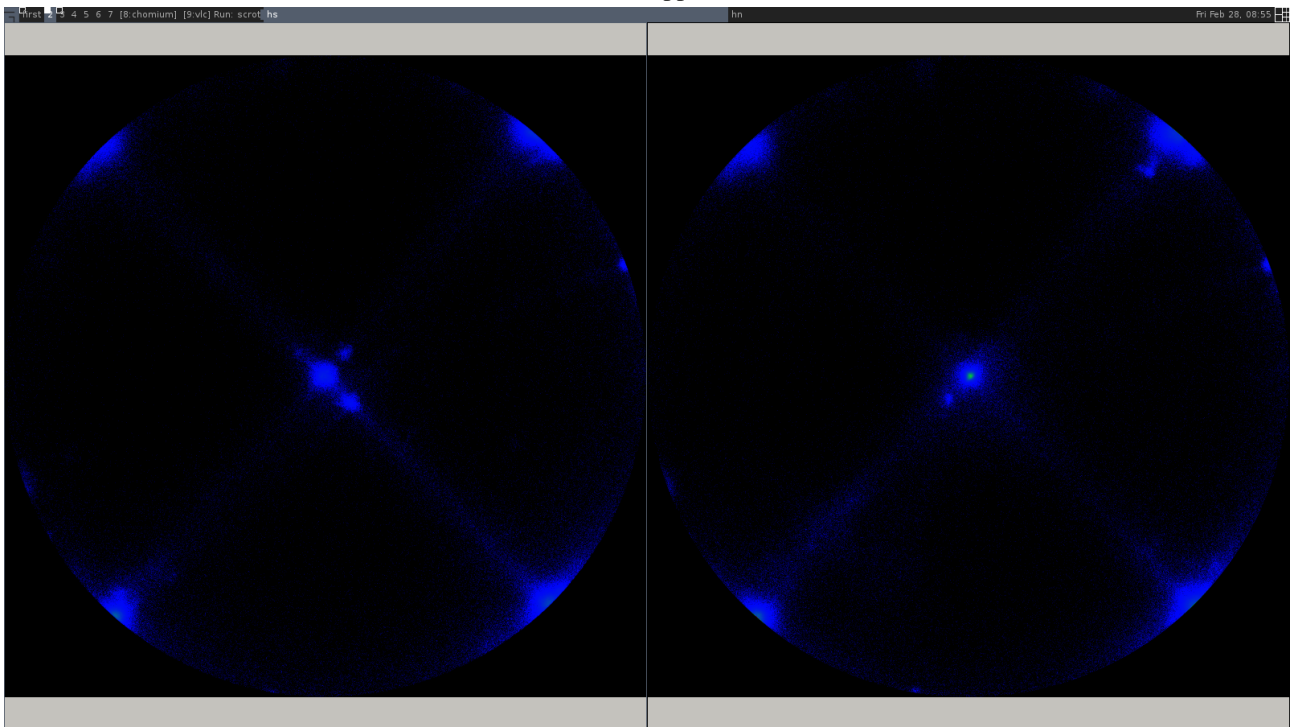


FIGURE 4 – Sphère de Gauss normalisée, avec $\alpha = 1$, $\beta = 1000$. La partie gauche représente l'hémisphère sud, la droite l'hémisphère nord.

Les zones bleues représentent la position à laquelle les normales ont majoritairement été projetées. On constate nettement que la majorité des normales est concentrée en 6 clusters indépendants : les 4 clusters présent sur le bord de l'image représentent les directions principales des murs verticaux, tandis que la zone centrale représente respectivement le sol (gauche) et le plafond (droite).

Ce type de projection présente de nombreux avantages :

- La sphère de Gauss est représentée en 2D, permettant ainsi l'utilisation d'algorithme de traitement d'images classiques, tels que le mean-shift.
- En modifiant les paramètres α et β de la projection, on obtient diverses résolution de représentation des normales.

On peut alors utiliser ces différents niveaux de résolution afin de rechercher simplement et efficacement de façon hiérarchique les directions principales.

Elle présente cependant un avantage majeur : les aires ne sont pas respectées. Il est cependant possible de calculer l'expansion d'aire en fonction de l'angle de projection, et ainsi d'appliquer une distorsion restaurant la notion d'aire.

1.2 Détection de plans

La projection stéréographique présente de nombreux avantages, notamment pour la détection de plans. Un plan est idéalement représentable comme un point sur la sphère de Gauss, et donc sur sa projection. En pratique, les normales sont bruitées, et forment un blob par direction principale (voir 4).

Afin d'obtenir la normale idéale représentant la direction principale de chaque cluster, il nous faut :

- Extraire les différents blobs de l'image
- Déterminer la normale optimale dans le cluster. On peut utiliser un algorithme tel que *mean shift* afin de déterminer le centre du cluster.

Pour cela, on peut commencer à travailler sur une sphère de Gauss à faible résolution (α élevé et β faible).

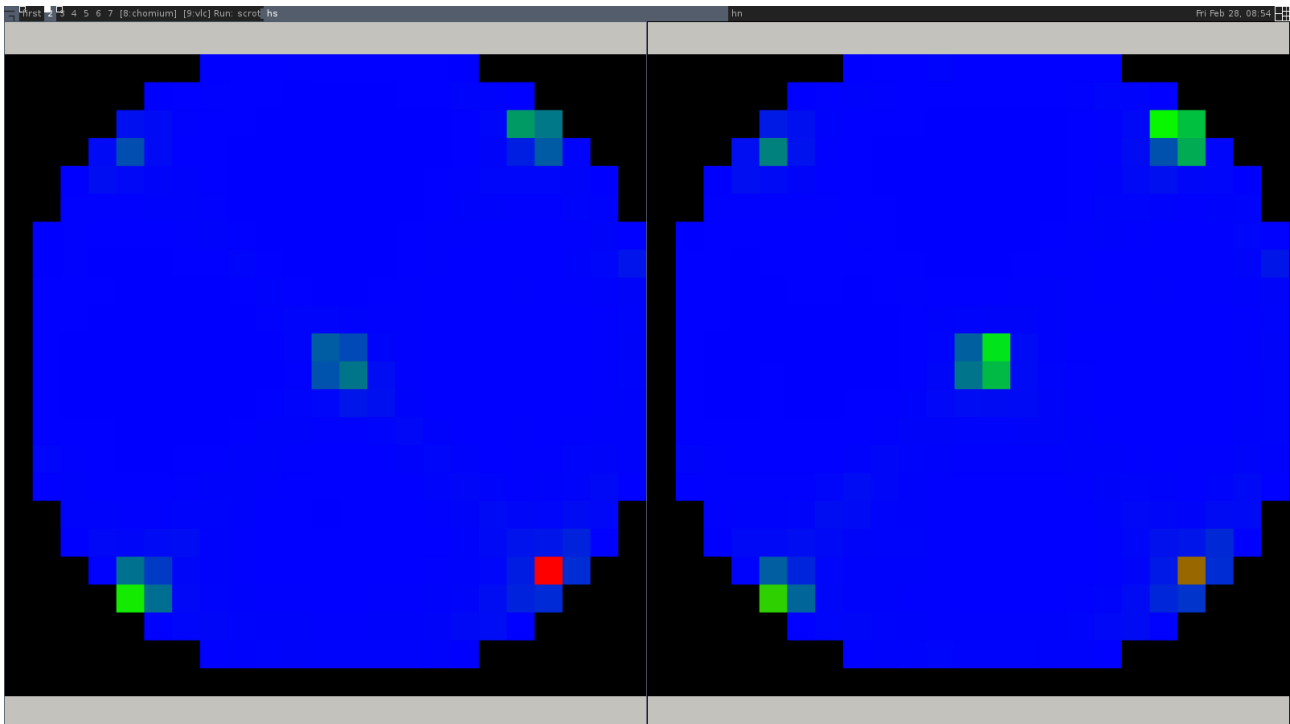


FIGURE 5 – Sphère de Gauss de l'appartement avec $\alpha = 1$ et $\beta = 10$

Comme on peut le voir en comparant la figure 4 et 5, il est aisé de trouver les clusters de façon hiérarchique. On peut par exemple choisir de chercher le pixel ayant l'intensité maximale sur l'image basse résolution (figure 5). Ce pixel correspondra à un cluster sur une image à résolution (figure 4) plus élevée. En effectuant un mean-shift sur le cluster sélectionné dans l'image haute résolution, il est alors possible de trouver la normale représentant au mieux le cluster.

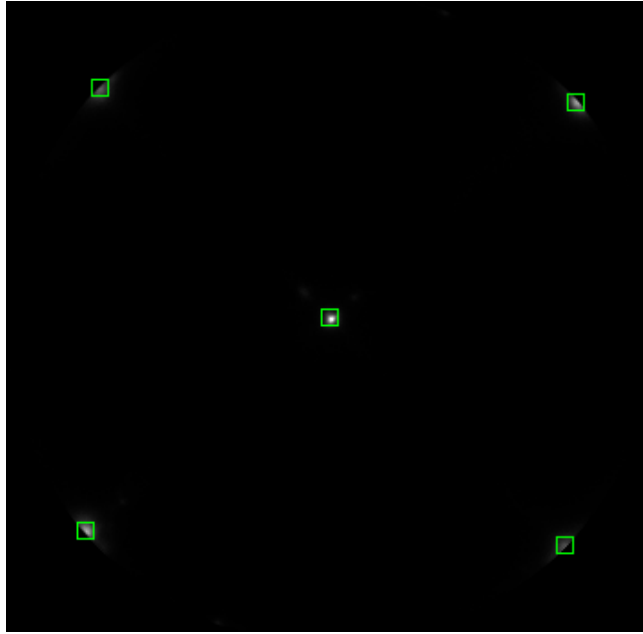


FIGURE 6 – Détection des clusters principaux dans la sphère de Gauss

1.3 Détection de cylindres

Les formes cylindriques accumulent leur normales selon un cercle sur la sphère de Gauss. La direction principale du cylindre est orthogonale au plan de support de ce cercle.

Dans notre espace de représentation projeté, un cylindre résulte en

Un cercle Si la direction principale est orientée selon $\pm \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$

Une ellipse dans tous les autres cas.

Voici un exemple de résultat obtenu sur un nuage de point de l'Église St. Jean, ayant une forme cylindrique.

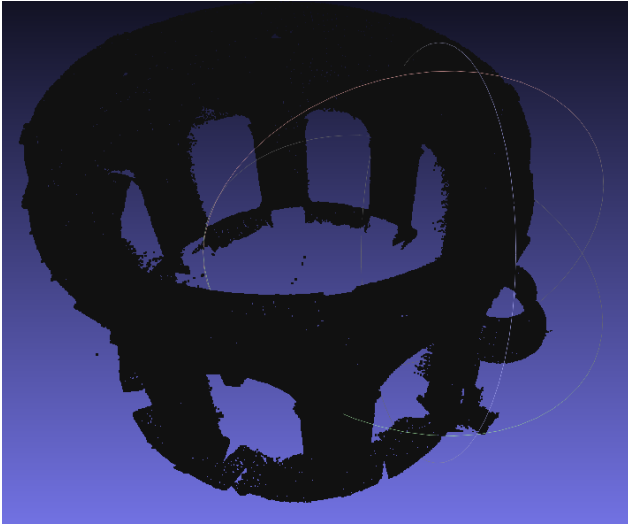


FIGURE 7 – église S^t Jean

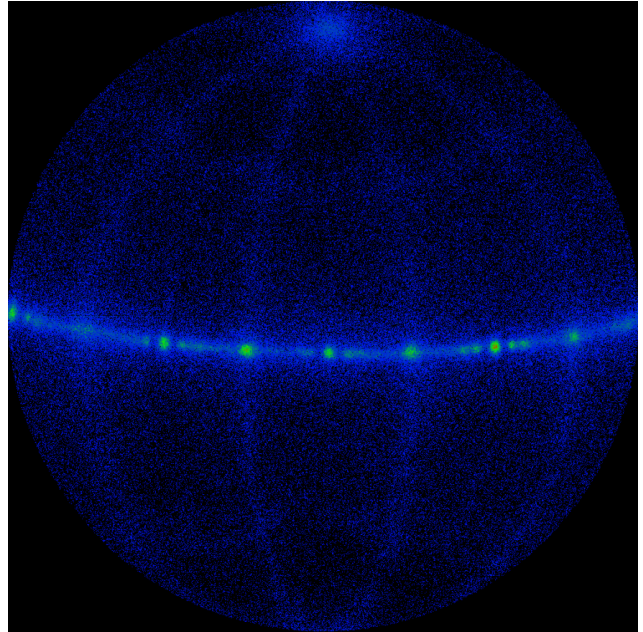


FIGURE 8 – Sphère de Gauss de l'église, avec $\alpha = 1$ et $\beta = 1000$

L'arc bleu clair horizontal sur la sphère de gauss représente la direction des normales horizontales de l'église, c'est à dire toute la façade circulaire. Chaque point sur cet arc correspond à une colonne. Les arcs de cercle verticaux représentent la courbure des arches.

Cet espace de représentation est donc loin d'être adapté à la détection des cylindres.

2 Reconstruction de plans

Une fois les directions principales des plans obtenues, il s'agit de trouver les plans dans le nuage de point. Il peut y avoir plusieurs plans à des profondeurs différentes selon une direction de normale. Pour cela, on a étudié la distribution des points selon la normale. L'idée est que les points représentant un même plan seront tous plus ou moins à la même distance de l'origine dans la direction de la normale.

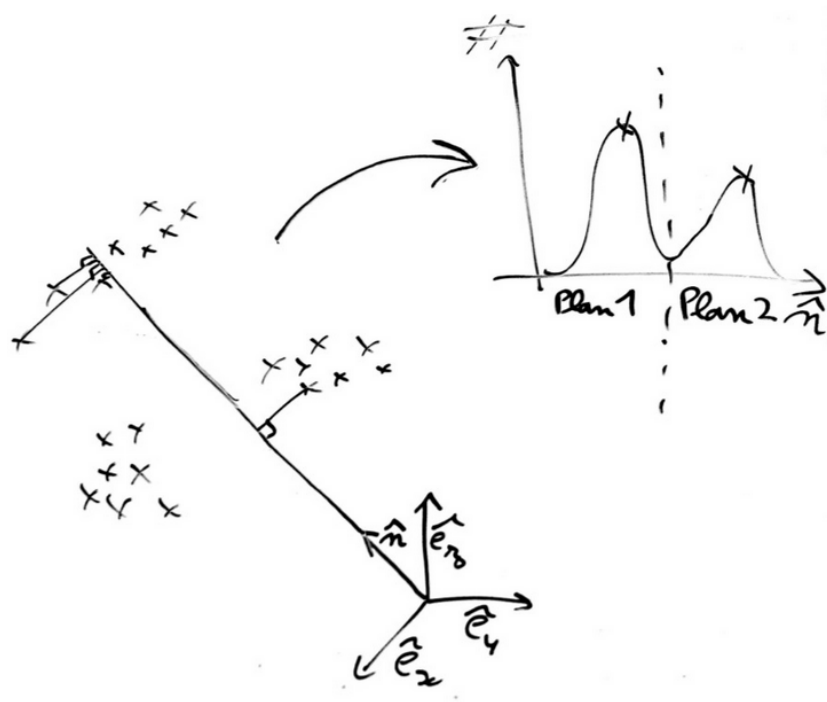


FIGURE 9 – Distribution des plans selon la normale. On représente le nombre de points en fonction de la distance par rapport à l'origine dans la direction de la normale.

Sur la figure ci-dessus, il est clairement visible que les points sont répartis selon 2 plans. Afin de les détecter, on seuille l'histogramme et cherche le maximum de chacun des pics. Sur des nuages de points réels, l'histogramme est bien sur bien moins régulier.

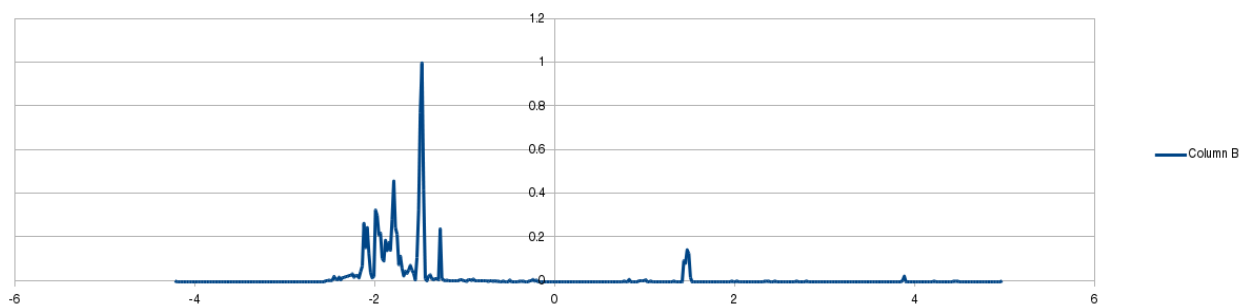


FIGURE 10 – Histogramme dans une direction principale de l'appartement (normale horizontale correspondant aux murs).

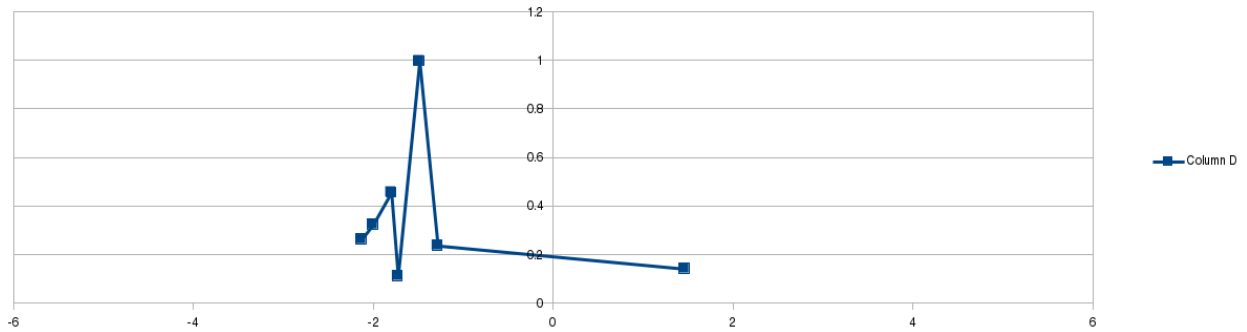


FIGURE 11 – Histogramme seuillé : les points représentent plans détecté (leur abscisse correspond à la position en mètres depuis l’origine du plan).

Afin d’améliorer la détection de la position des plans, il serait intéressant d’imposer une distance minimale entre 2 plans. Considérant que le bruit dans les données est assimilable à du bruit Gaussien, on pourrait également chercher les distributions gaussiennes représentant au mieux l’histogramme, et déterminer la position des plans en fonction de la moyenne et variance de ces distributions.

Les plans sont ensuite reconstruit de façon très approximative, en considérant seulement le plus petit rectangle englobant toutes les données classifiées comme appartenant à une profondeur de plans. Cette méthode n’est bien sur par robuste aux outliers, et ne permet pas de distinguer différents plans à une même profondeur. Afin d’obtenir un meilleur résultat, il serait intéressant d’utiliser une combinaison de RANSAC et K-Means afin de clusteriser les plans et reconstruire autant de plans qu’il y a de clusters trouvés sans tenir compte des outliers.

3 Résultats



FIGURE 12 – Appartement : nuage de points de référence

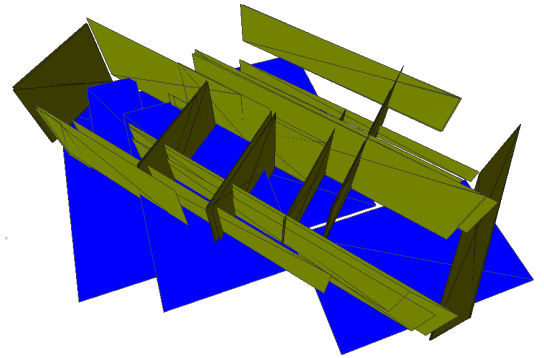


FIGURE 13 – Plans reconstruits avec β élevé.

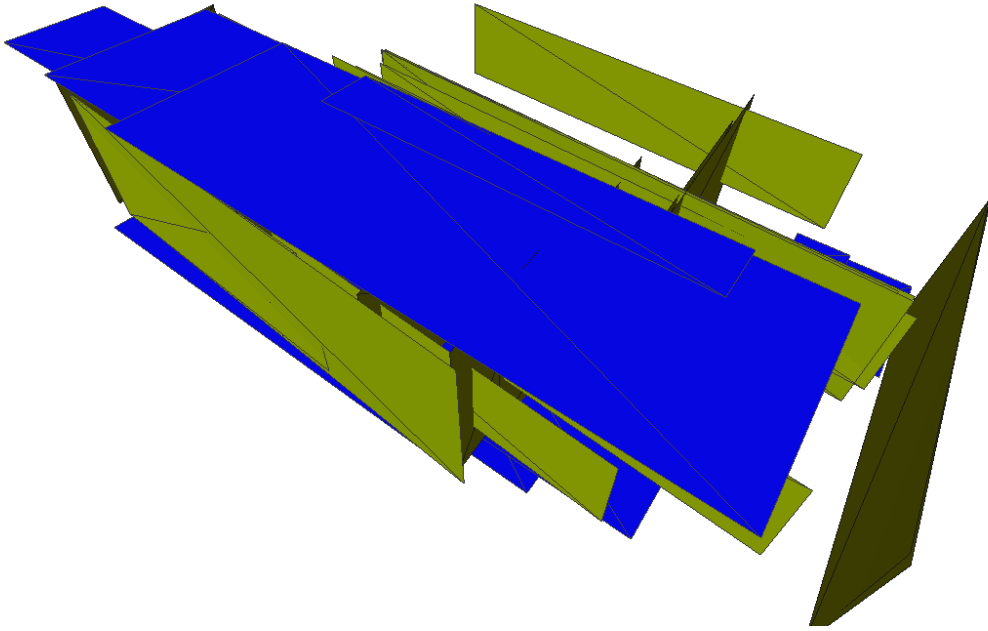


FIGURE 14 – Plans reconstruits, avec β plus faible. On remarque que cette fois-ci le toit a été détecté. Cela est dû au fait que les normales représentant le toit sont relativement bruitées, et diminuer la résolution de la projection induit un phénomène de clustering concentrant toutes les normales sur une plus petite zone.

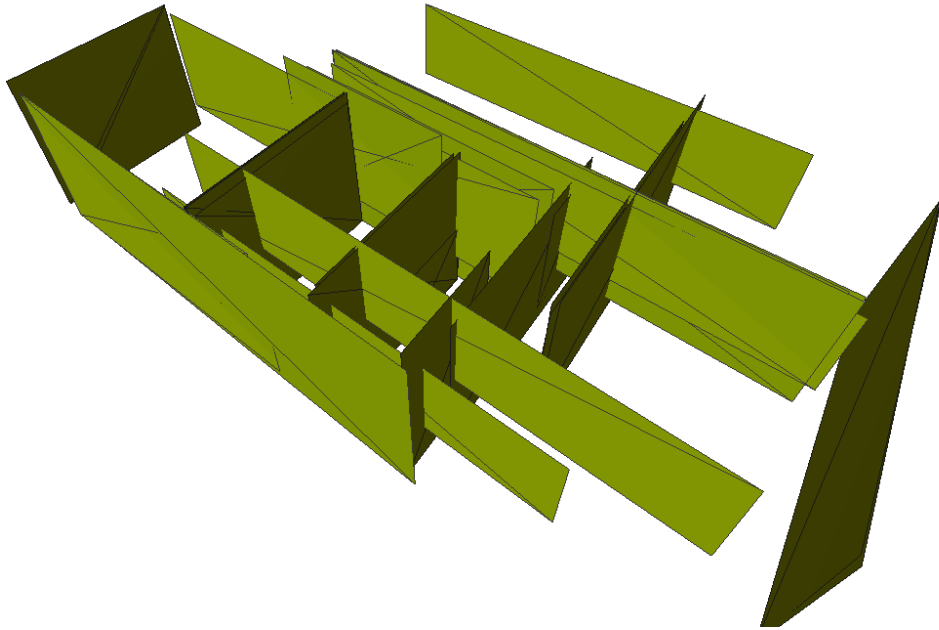


FIGURE 15 – Si l'on enlève le toit de la figure précédente, on peut voir les plans reconstruits à l'intérieur de l'appartement. On constate que diminuer la résolution de la sphère de Gauss a introduit des erreurs d'orientation de normales. Par exemple, le plan le plus à droite est très mal orienté.

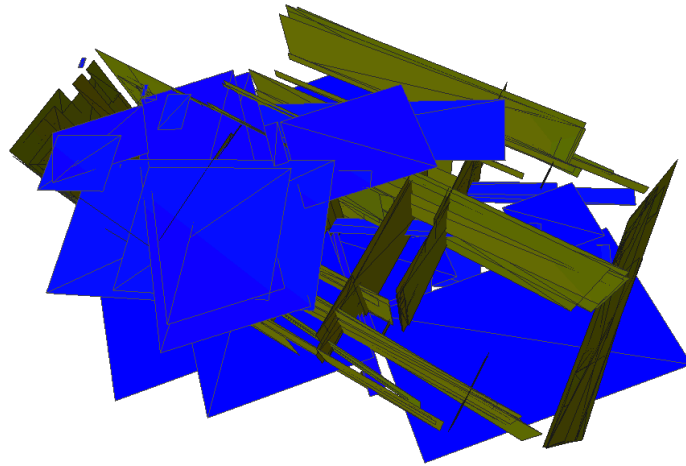


FIGURE 16 – Les erreurs d’orientations sont généralement compensée en prenant le centre d’un cluster comme orientation de normale. Cette figure représente le résultat obtenu lorsque le clustering n’est pas effectué et que la direction des normales est directement prise comme étant le maximum de chaque cluster.

4 Détails d’implémentation

4.1 Sphère de Gauss sur GPU

La projection stéréographique des normales sur la sphère de Gauss est hautement parallélisable. En effet, la sphère de gauss étant un simple accumulateur, la projection de toutes les normales peut être implémentée en parallèle.

Cependant, beaucoup de normales doivent être accumulées au même endroit, entraînant des délais d’écriture du fait de l’atomicité de l’accumulation (il n’est pas possible d’accumuler plusieurs normales au même endroit en même temps, cette opération doit être effectuée séquentiellement). Ce problème peut être fortement allégé en prenant les normales dans un ordre aléatoire, réduisant ainsi le nombre de normales devant être simultanément accumulées au même endroit.

Voici le *kernel* OpenCL responsable de la projection

```
void kernel_gauss_sphere(volatile global float4* normal, global int*
north_hemisphere, global int* south_hemisphere, const float alpha_g, const
float beta_g)
{
    private const float beta = beta_g;
    private const float alpha = alpha_g;
    private const int rows = 2*(1+(int)ceil(beta/alpha)) + 1;

    private int index = get_global_id(0);
    private float tmp;
    private int2 coord;
    private float4 n = normal[index];
    // Compute projection for the south hemisphere
    if(n.z<0) {
        tmp = (beta+alpha)/(alpha-n.z);
        coord = convert_int2((float2)(floor(tmp*n.xy)))+(int2)(rows/2, rows/2);
        private int tex = coord.y * rows + coord.x;
        // Atomic operation to increment the value for the current normal.
        // This ensures that no other normal is currently being accumulated at
        // this position.
        atomic_inc(&south_hemisphere[tex]);
    }
    // North hemisphere
    else {
```

```

    tmp = (beta+alpha)/(alpha+n.z);
    coord = convert_int2((float2)(floor(tmp*n.xy))) + (int2)(rows/2, rows/2)
    ;
    private int tex = coord.y * rows + coord.x;
    atomic_inc(&north_hemisphere[tex]);
}
}

```

5 Conclusion

La méthode de détection de composante principales basée sur la sphère de Gauss est très prometteuse. Elle est très efficace et performante dans le cadre de plans car ceux-ci projettent énormément de normales au même endroit. Elle n'est cependant pas limitée aux plans, et peut prendre en compte bien d'autres composantes, telle que cylindres, cônes, sphères...

Afin d'améliorer le travail présenté ici, les articles suivants fournissent des techniques utiles : (Gau), (3DR) et (Li et al., 2011)

Références

3d reconstruction and lod generation for urban scenes. Unpublished as of now.

Robust 3d reconstruction of complicated scenes with industrial structures. Unpublished as of now.

Yangyan Li, Xiaokun Wu, Yiorgos Chrysanthou, Andrei Sharf, Daniel Cohen-Or, and Niloy J. Mitra. Globfit : Consistently fitting primitives by discovering global relations. *ACM Transactions on Graphics*, 30(4) :52 :1–52 :12, 2011.