

Compte rendu de la réunion numéro 7

19 Avril 2023

1 Membres présents :

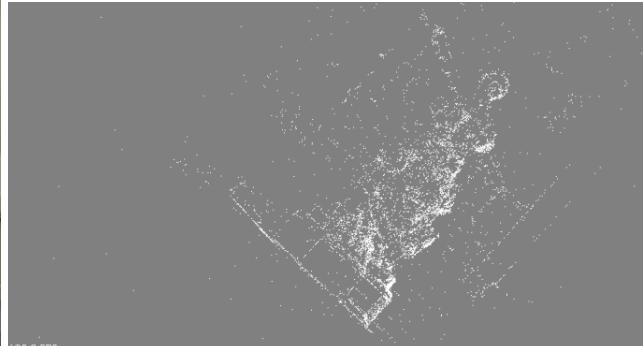
- William Puech : Oui;
- Marie Bocquelet : Oui;
- Alexandre Fleury : Oui;
- Fabien Caballero : Oui;
- Thibault Gasc : Oui;

2 Ordre du jour et objectifs :

-Montrer les nuages de points 3D que l'on a réussis à obtenir à partir d'images 2D. La difficulté là dedans était d'avoir les bons paramètres caméra et d'appliquer les bonnes transformations aux points afin d'avoir quelque chose de cohérent. Nous avons d'abord tenté de la faire sur des images dont nous avions le nuage de points 3D auquel se référer pour voir si notre résultat était correct. Ci-dessous, on peut voir l'image de la statue avec son nuage de points correspondant :



(a) Photo de la statue



(b) Nuage de points obtenus à partir des points de référence

Comme on peut le voir, on distingue très bien la statue dans le nuage de points. Ensuite, nous avons essayé de reconstruire les points 3D nous mêmes à partir d'un algorithme, car nous allons en avoir besoin pour visualiser le nuage de points 3D obtenu par match entre les images.

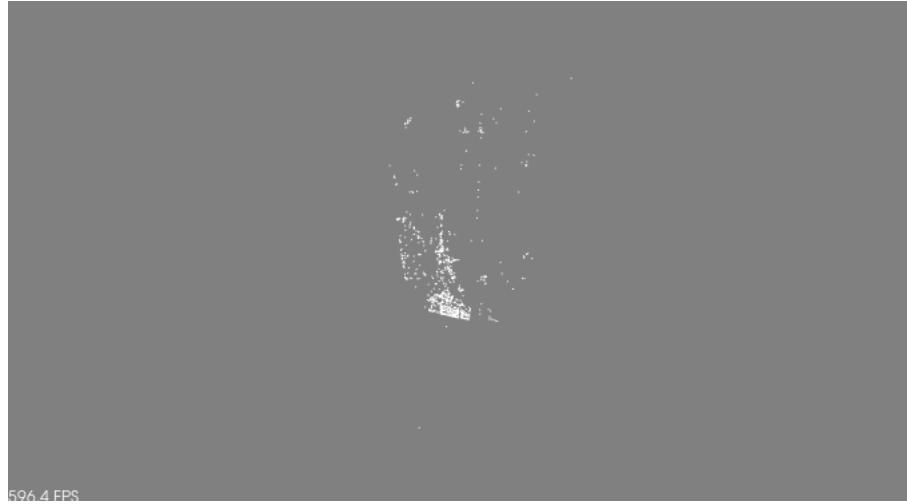
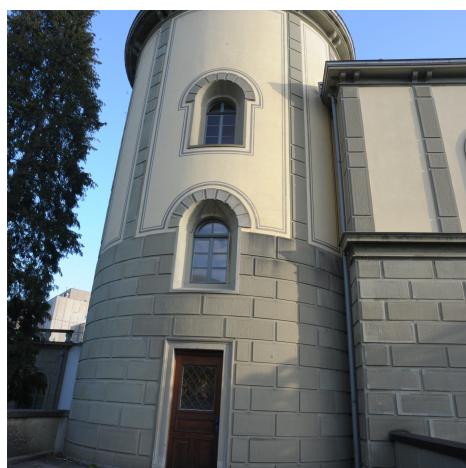


Figure 2: Nuage de points obtenu pour 2 images de la statue

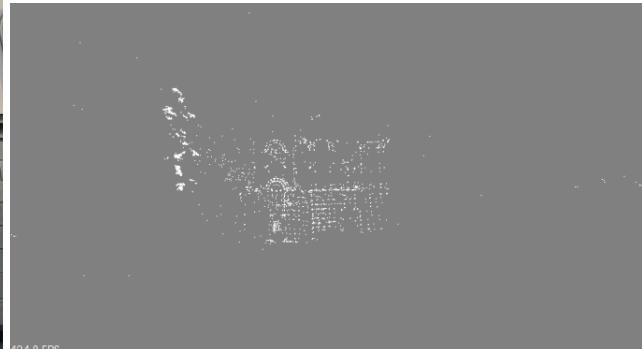
Et voici le résultat obtenu pour seulement deux images :

Comme on peut le voir, le nuage de points est beaucoup moins précis puisque l'on a seulement deux images. Cependant on peut tout de même distinguer la forme de la statue et surtout son socle.

Pour confirmer que notre algorithme fonctionne bien, nous avons testé sur deux autres images qui sont complètement différentes de la statue :



(a) Photo d'un observatoire



(b) Nuage de points obtenu à partir des deux images

On peut voir que seulement avec deux images, on distingue bien les arcs des fenêtres.

-Montrer la reconstruction d'un maillage à partir d'un nuage de points. Pour ceci nous avons pris un nuage de points bien connu représentant un lapin :

Tous les nuages de points sont affichés grâce à un programme codé par nous même utilisant la librairie PCL. La reconstruction d'un maillage à partir de ce nuage de point se fait également grâce

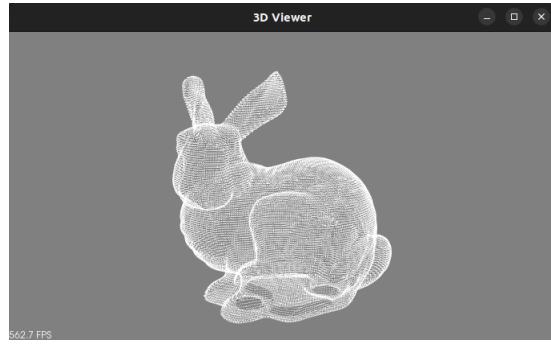


Figure 4: Nuage de point bunny.ply

à la librairie PCL. Donc nous avons testé la reconstruction du maillage du lapin et voici le résultat obtenu :

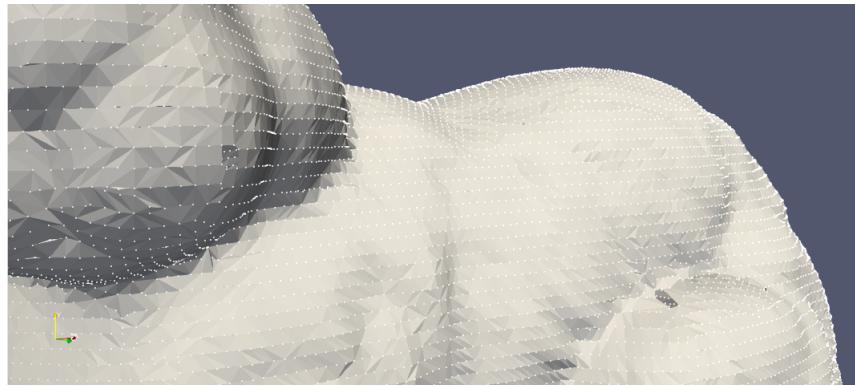
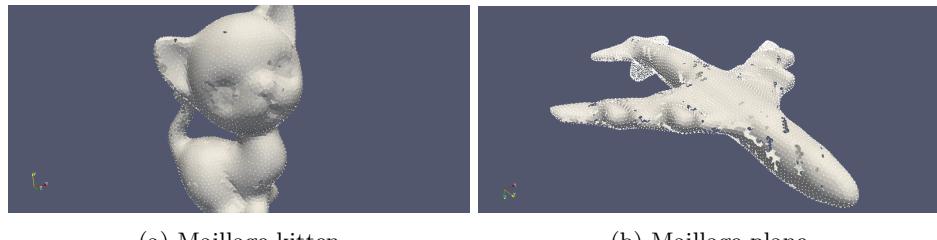


Figure 5: Maillage bunny

Ici, on peut voir que le maillage a très bien été reconstruit. En effet on distingue bien les points et les triangles les reliant.

Cependant, nous avons remarqué que cela ne fonctionne pas pour tous les maillages :



Comme on peut le voir sur les photos ci-dessus, la reconstruction des maillages ne fonctionne pas tout le temps. On remarque des trous dans le maillage, ce qui va poser problème au moment de l'impression 3D.

3 Résultats de la réunion :

-Maintenant que nous avons une triangulation des points correcte, nous devons commencer à appliquer ça pour toutes les images, et non plus seulement sur deux images. Pour ceci nous avons deux possibilités : soit faire en version 3D, c'est à dire effectuer un nuage de point pour chaque paire d'images (1 et 2, 2 et 3...). Une fois les nuages de points obtenus, effectuer un recalage des points dans les différents nuages, pour ne garder que les points qui diffèrent entre les nuages car nous ne souhaitons pas afficher plusieurs fois les même points 3D. L'autre possibilité serait en version traitement d'image, pour chaque nouvelle image, on vérifie si on retrouve les mêmes points qui matchent. Si c'est le cas, on ne garde ces points que dans le premier nuage, et si ce n'est pas le cas, on ajoute dans le nuage les nouveaux points. Nous avons choisi la version traitement d'images;

-Nous devons trouver un moyen de boucher les trous du maillage, nous allons tenter d'appliquer l'algorithme du Marching Cube pour ça;

4 A faire pour la prochaine fois :

-Avoir fait le multi-view avec recalage des points 3D;

-Avoir réussis à boucher les trous des maillages;

5 Date et heure de la prochaine réunion :

Mercredi 10 Mai à 15h30 à la Faculté des Sciences, Bâtiment 16.