

Exercice 0.1 Mise en correspondance de points

1 - Que pensez vous du résultat ? Les correspondances ne sont pas toutes correctes. Par exemple, on a une grande droite en diagonale qui met en correspondance un point du sol, et un point de la coupole sur la figure [1]. On peut corriger certains de ces points en imposant la contrainte épipolaire évoquée en cours.



Figure 1 – Mise en correspondance des points dans l'image - sans contrainte épipolaire (à gauche) - avec contrainte épipolaire (à droite).

Exercice 0.2 Reconstruction d'un nuage de points par triangulation

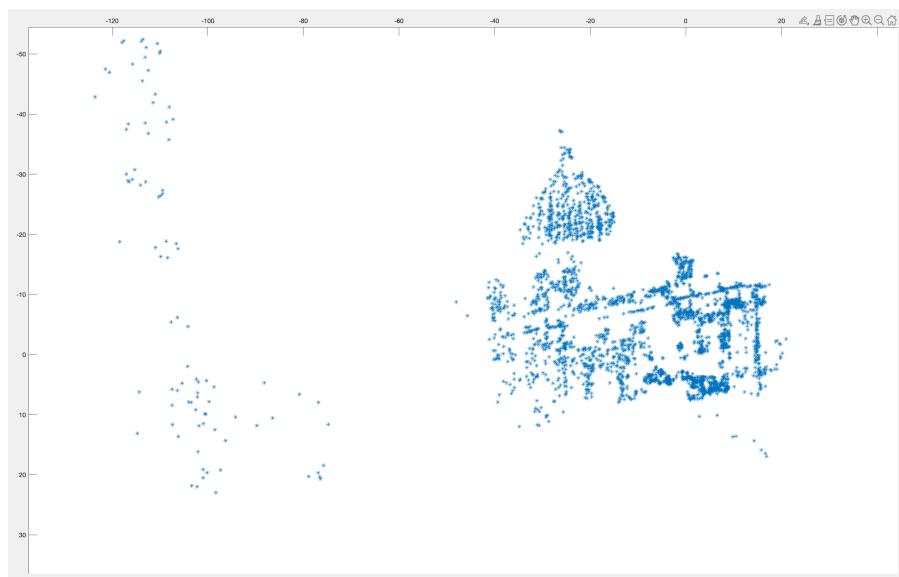


Figure 2 – Reconstruction de la cathédrale par triangulation (reconstruction NON dense (?)).

1 - Que pensez vous du résultat ? Le résultat est assez satisfaisant : on devine la forme de la cathédrale ; mais un peu moins la statue. On aimerait avoir un peu plus de points, en particulier au niveau de la statue pour avoir une meilleure reconstruction.

Exercice 0.3 Reconstruction dense

1 - Pourquoi la carte correspond à $[-64, 0]$ est-elle erronée ? Pour commencer, il ne faut pas oublier qu'il faut que les images soient rectifiées avant de calculer disparité ! Ensuite, il n'est pas impossible de

manière générale que la disparité soit négative. Cependant cela dépend du mouvement de la caméra entre les deux prises. Il se trouve qu'ici, la caméra bouge de telle sorte que la carte de disparité calculée entre les deux images est forcément positive : il n'est donc pas pertinent de prendre $[-64, 64]$ comme plage de valeurs pour afficher la disparité.

2 - Regarder les valeurs de disparités sur la statue, la grande porte en bois et l'extrémité gauche du bâtiment. Ces valeurs sont-elles cohérentes ? pourquoi ?

La figure 3 présente la carte de disparité calculée avec Matlab. On lit alors les disparités :

- Sur la statue : environ 33
- sur la grande porte en bois (ou plutôt la grande fenêtre en verre, étant donné que je ne vois pas de porte en bois) : environ 12
- à l'extrémité gauche du bâtiment : environ 5 (bleu un peu plus foncé qu'au niveau de la "porte-fenêtre")

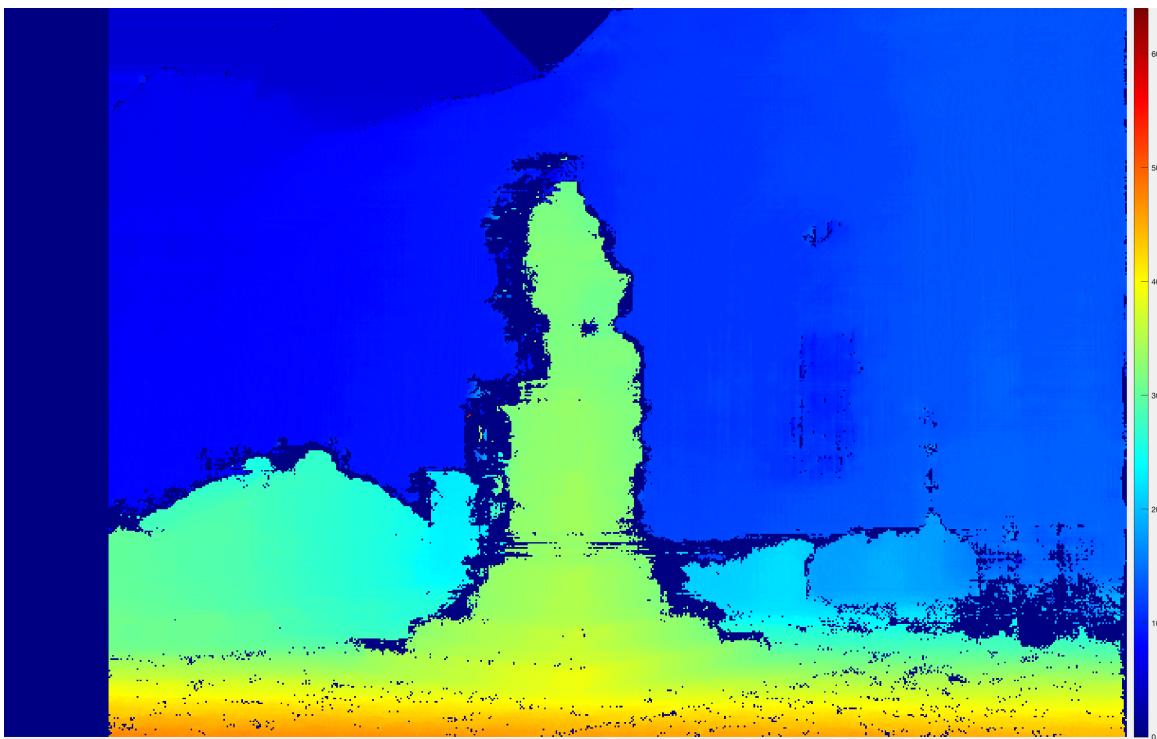


Figure 3 – Carte de disparité pour la cathédrale, obtenue avec le code Matlab.

Ces disparités sont cohérentes avec les informations que l'on déduit visuellement de l'image : du plus proche au plus loin, on a la statue, puis la partie droite du bâtiment ("porte-fenêtre"), et enfin la partie gauche du bâtiment, ce qui est transcrit dans une disparité de plus en plus petite au fur et à mesure que l'on considère des objets éloignés.

3 - Regarder maintenant les valeurs dans le ciel. Qu'en pensez-vous ?

Les valeurs dans le ciel sont estimées un peu (mais pas beaucoup) plus loin de la caméra que le bâtiment. Cela est bien évidemment assez peu cohérent avec la réalité ; on s'attend à une différence significativement plus grande comme par exemple le triangle de bleu foncé que l'on peut deviner juste au-dessus de la statue. Cette sur-estimation de la disparité peut être dû à la résolution de la caméra, ou encore au côté uniforme du ciel qui rend difficile la mise en correspondance des points et donc fausse le calcul de disparité.

4 - A quoi correspondent les valeurs marquées 'NaN' autour de la statue ?

Les valeurs marqués "NaN" autour de la statue correspondent à des points qui n'ont pas pu être mis en correspondance car ils sont présents dans une image, et pas dans l'autre (occultation). En effet la caméra change de point de vue entre les deux images, et une partie de la statue visible dans une des deux images ne l'est pas dans l'autre.

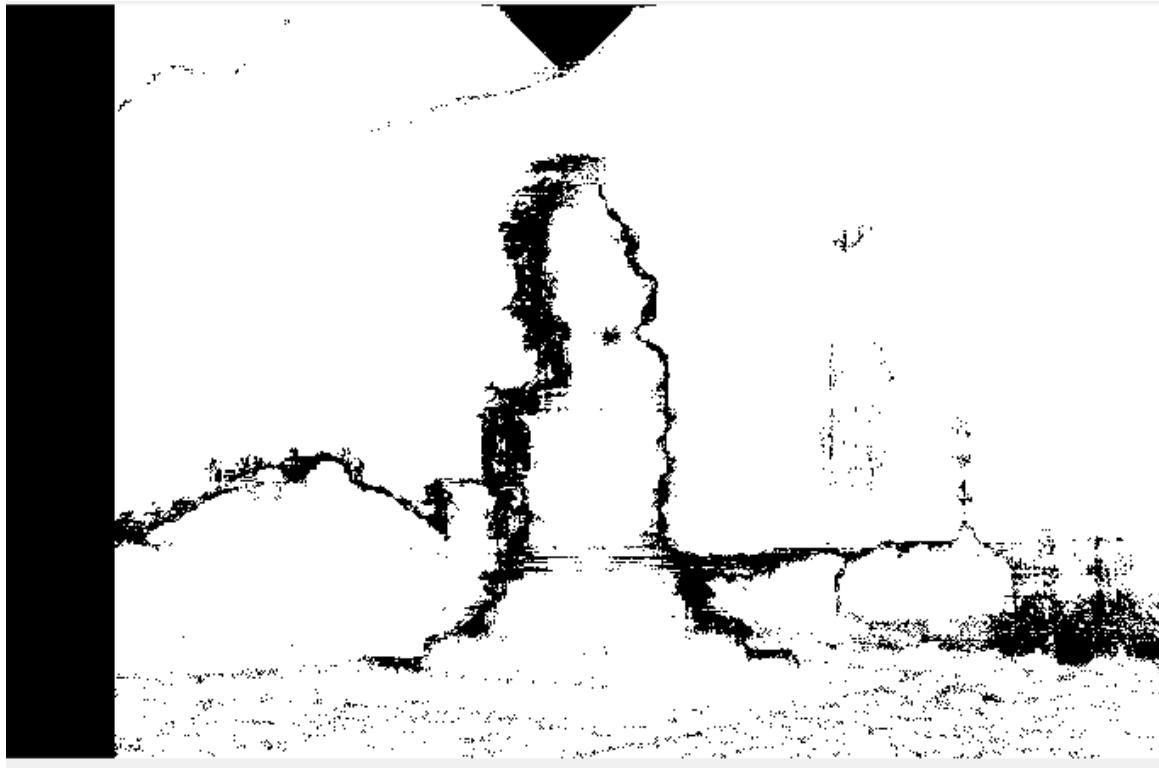


Figure 4 – Visualisation de la carte de disparité avec imtool.

5 - Où sont situés en majorité les points pour lesquels aucune disparité n'est calculée ? Pourquoi ces éléments causent-ils des problèmes pour établir la mise en correspondance ?

De manière générale, comme on l'a vu en cours, un point peut ne pas avoir de correspondant d'un point de vue à un autre, en particulier au sein des zones où il y a une occultation entre les 2 vues. Ainsi, autour de la statue, sur la partie gauche de l'image (tout le bleu foncé dans la figure 3), au niveau du "balcon" derrière la statue ou encore des points camouflés par les passants à droite d'une des deux images, on trouve des valeurs "NaN" dans la carte de disparité.

6 - Quelles sont les difficultés rencontrées pour obtenir une visualisation correcte ? Quelles est l'origine de ces difficultés ? Pourquoi la reconstruction du dôme de verre donne-t-elle n'importe quoi ?

Plusieurs difficultés se manifestent lorsque l'on veut utiliser la carte de disparité pour reconstruire la scène en 3D (comme on l'a fait figure 2).

- Pour commencer, il faut choisir la bonne fonction pour visualiser les points ; en l'occurrence on utilise ici *pcshow* comme recommandé par mail dans Matlab qui permet d'afficher des "worldPoints" en *mappant* les points d'une image d'origine, afin de récupérer les couleurs. Ca n'est pas en soit un gros problème mais c'est déjà une première différence par rapport à la partie précédente.
- Il faut aussi choisir la bonne échelle d'affichage afin que les points soient visibles. En l'occurrence, *pcshow* le fait pour nous.
- Ensuite, une fois que les points sont affichés à l'aide de la commande *view(player, xyzPoints)*, on se retrouve avec une sorte de "d'amas de points" qui ressemble (avec un peu d'imagination) à la scène d'origine. Il faut alors tourner cela un peu dans tous les sens pour "retrouver" la cathédrale et ses environs. Finalement, on arrive à la figure 5 où l'on devine la cathédrale, avec quelques trous et la coupole qui se délite un peu (beaucoup).

On remarque que le dôme se "délite" un peu, en plusieurs points de plus en plus éparses. En effet, le dôme se fond dans le ciel niveau disparité (voir figure 3) du fait de sa transparence, et le fait qu'il réfléchisse par endroits la lumière peut aider à expliquer pourquoi sa reconstruction n'est pas être satisfaisante.

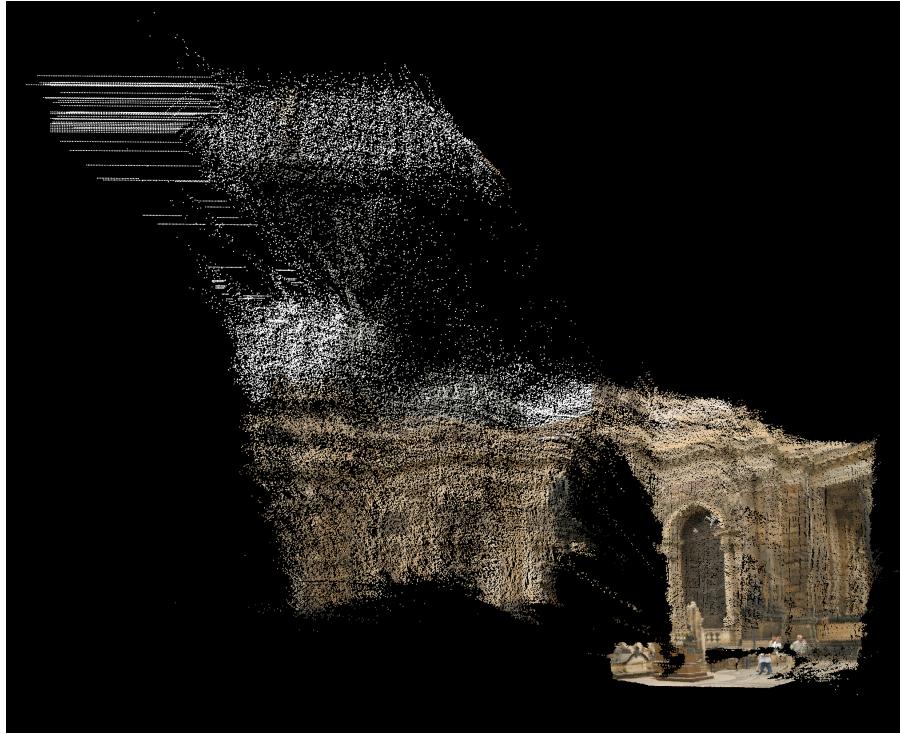


Figure 5 – Affichage des points 3D xyzPoints obtenus avec `reconstructScene`.

Une autre manière de visualiser la reconstruction et proposée par la documentation. Elle est moins impressionnante que ce que j'avais en tête et ce qui est présenté figure 5, mais on peut tout de même faire des commentaires et des observations avec cette base.

On visualise ainsi les points compris entre 4 et 20 mètres de la caméra figure 6. On déduit de cette image que la statue est située à moins de 4 mètres de la caméra (en pratique entre 2 et 4 mètres, car on arrive à dévoiler quasi-totalement la statue en imposant $Z > 2$ dans le code Matlab associé). C'est en revanche apparemment aussi le cas de la coupole, ce qui est bien sûr faux. Difficile d'expliquer exactement pourquoi. Peut-être est-ce que la contrainte d'ordre (évoquée en cours) n'est pas respectée (car on a une discontinuité de profondeur). En effet ce comportement est tout de même étonnant, car on voit figure 3 qu'à priori, la coupole est détectée tout de même plus loin que la statue. La reconstruction de la coupole est probablement rendue difficile par son aspect transparent qui réfléchit la lumière, et sa couleur qui est proche du ciel, difficile à reconstituer.

Figure 7. On voit également qu'une partie des points à droite de la coupole est détectée à moins de 2m. Ce sont les mêmes points détectés très loin sur la carte de disparité. En réalité, en visualisant avec `imtool` (figure 4), il semblerait que ces points n'ont pas été mis en correspondance, d'où leur affichage incohérent.

7 - citer des exemples de tâches pour lesquelles une reconstruction dense est indispensable et des exemples de tâches où une reconstruction éparsse est suffisante.

Pour des robots autonomes (rovers sur Mars, voitures sans chauffeur), la détection d'objets 3D est importante et une reconstruction dense et préférable. Pour des technologies comme le MoCap (Motion Capture) utilisées dans le cinéma pour réaliser des caractères CGI (*Computer Generated Imagery*), une reconstruction éparsse est suffisante.

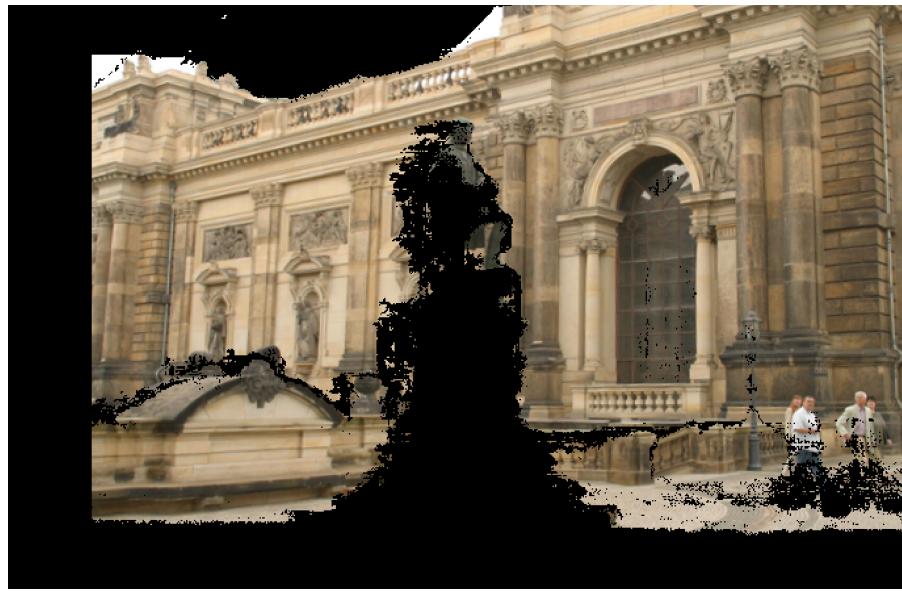


Figure 6 – Points de l'image entre 4 et 20 mètres de la caméra.

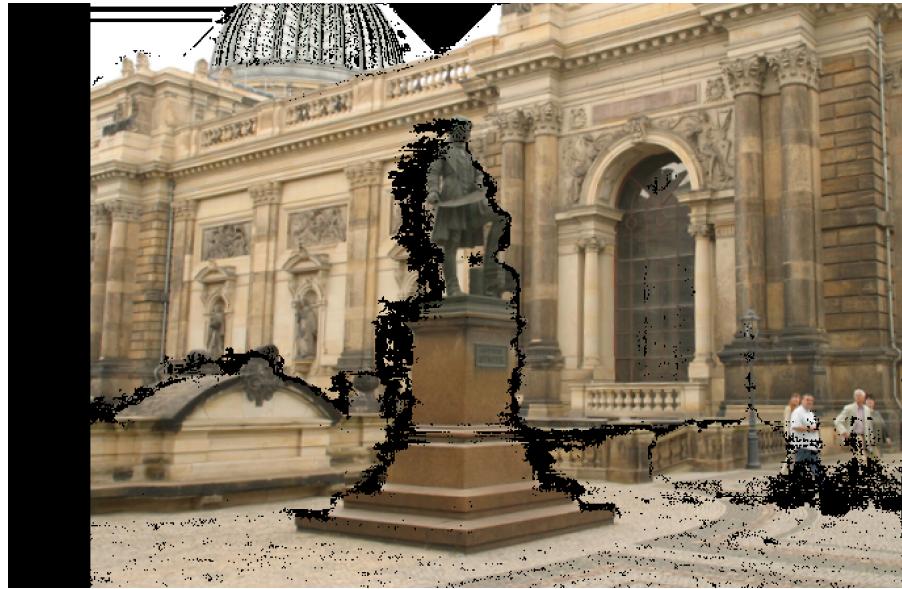


Figure 7 – Points de l'image entre 2 et 20 mètres de la caméra.