Mise en correspondance stéréoscopique

Yann DUBOIS

October 12, 2017

1 Introduction

La dernière fois nous avons des techniques permettant la reconstruction 3D d'image par stéréoscopie éparse. Dans ce rapport nous allons nous concentrer sur la stéréoscopie dense, c'est à dire que nous allons prendre comme hypothèse que tout les points de l'image de référence sont des points d'intérêt. Nous allons dans un premier temps calculer la disparité dans l'image de gauche, qui sera alors notre image de référence. Puis nous allons faire la même chose pour l'image de droite. On va ensuite vérifié à l'aide des deux résultats les points qui peuvent être en correspondance et enfin nous allons analyser une technique de calcul plus efficace pour le SSD (Sum of Squared Difference).

2 Similarité par SSD

Notre premier objectif est d'obtenir une carte des disparités de l'image de gauche. C'est à dire qu'on cherche à quelle niveau les pixels sont en corrélation dans l'image de gauche. Pour cela on va utiliser une fonction de dissimilarité qui se nomme SSD (Sum of Squared Difference). Cette technique permet de retrouver la différence de position d'un pixel dans une image en utilisant son voisinage. On calcul la SSD de cette façon :

$$SSD(x_l, y, s) = \sum_{i=-w_x}^{w_x} \sum_{j=w_y}^{w_y} (I_l(x_l + i, y + j) - I_r(x_l + i - s, y + j))^2$$

Avec x_l, la coordonnée x de l'image de gauche, y, la coordonnées y (étant donné qu'on est dans le cas de la stéréoscopie dense, les droite épipolaires sont horizontale et se confonde dans les deux images), s, le décalage en pixel, w_x la taille de la région de voisinage en x, et w_y la taille de la région de voisinage en y. On notera que cette formule est vrai uniquement si l'image de référence est l'image de gauche.

Pour trouver la carte des disparités, on va chercher la disparité minimale pour tout les décallage, c'est ce qui correspond dans notre code à la matrice mMinSSD. La variable mSSD par contre change pour chaque décallage et est comparé mMinSSD pour en sortir les disparités minimales. En informatique, on doit faire attention aux effet de bord, on va donc éviter de calculer la disparités des pixels ayant une fenêtre de corrélation dépassant de l'image, c'est pour ça que dans le code, on

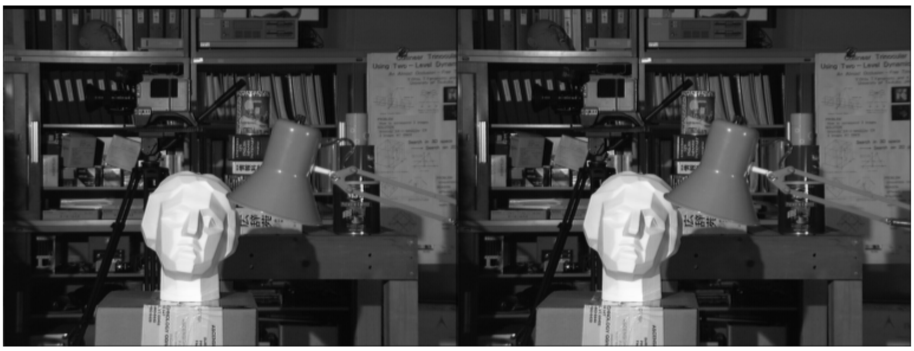

Figure 1: Images de références

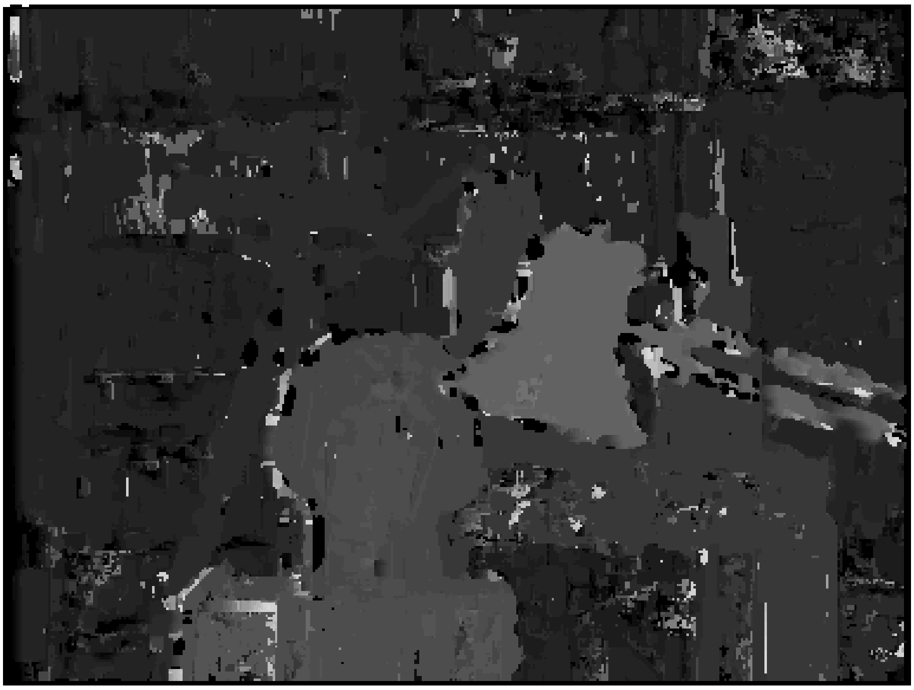

Figure 2: Carte de disparités de l'image de gauche

commence a un pixel se trouvant à une position correspondant au centre de la fenêtre. De même on va arrêter le calcul lorsqu'on se trouve à la limite des bords en abscisse et en ordonnées.

La carte des disparités de l'image de gauche est représenté par la Figure 2. Plus le pixel est sombre, plus la correspondance avec l'image de droite est forte, car celà veux dire que nous n'avons pas eu besoin de beaucoup de décalage pour trouver la point correspondant dans l'image de droite. On remarque que sur cette image, les points correspondant ne sont pas nombreux, on voit par exemple que les objets du premier plan comme le buste ou la lampe ont une disparité assez forte. En effet, plus un objet est proche du champ de vision, plus la différence de position sera grande. On peut faire l'expérience avec nos propre yeux en mettant un doigt assez proche puis en fermant l'oeil droit suivi de l'oeil gauche. On remarque alors que la perception de la position du doigt n'est pas la même pour les deux yeux.

Maintenant que nous avons pu calculer la carte de disparités de l'image de gauche, nous allons essayé de vérifier nos points pour éviter de prendre en compte des points qui ne serait pas présent sur les deux images.

3 Vérification gauche-droite

Pour faire la vérification gauche-droite, on va aussi calculer les disparités sur l'image de droite, avec la même méthode (SSD). On obtient alors pour l'image de droite la Figure 3. On remarque que les images sont très proche, mais il y a des différences essentiellement sur les bords des objets. En effet, certain points sont cachés dans l'image de gauche par rapport à l'image de droite, ce sont les points occultés. Dans cette méthode, on doit donc aussi inversé le iShift de "sens" et donc de le déplacer de la droite vers la gauche.

Lorsque l'on a les deux cartes de disparités, on peut calculer une image de vérification des disparités comme la Figure 4. On a coder cette image de manière à avoir du blanc pour les points occultés. On remarque alors bien ici que les bords sont des points occultés (la lampe par exemple). Grace à cette carte de validation, on peut générer la carte de disparité final (voir Figure 5). Le niveau de gris de chaque pixel correspond à la distance entre les pixels de l'image de gauche par rapport à l'image de droite (et inversement).

D'autres tests ont été fait pour une taille de fenêtre de corrélation de 5. On remarque qu'il y a un peu moins de points occultés.

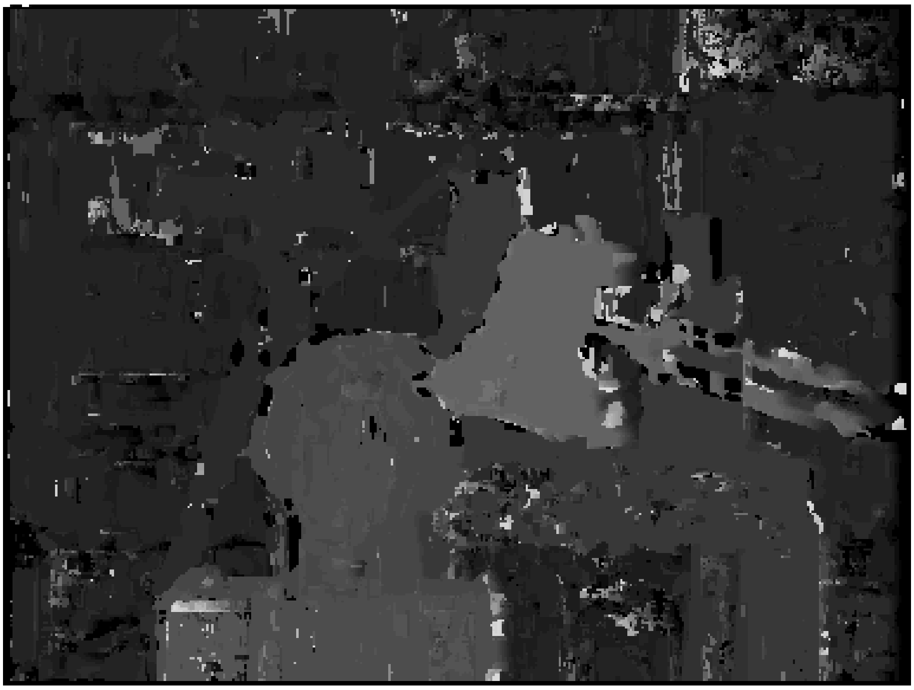

Figure 3:   Carte de disparités de l'image de droite

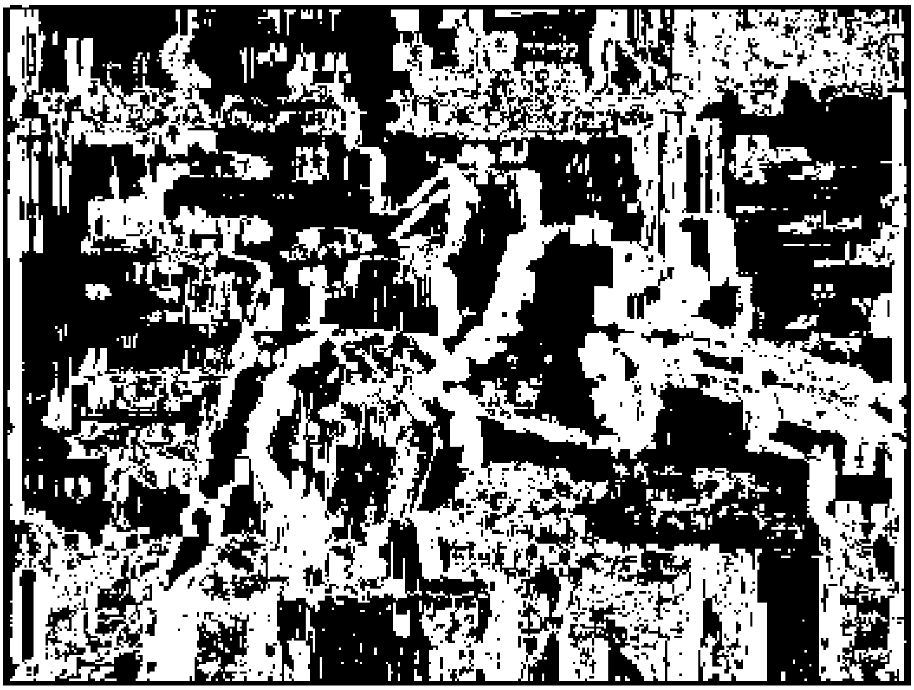

Figure 4:   Carte de validité des disparités gauche-droite

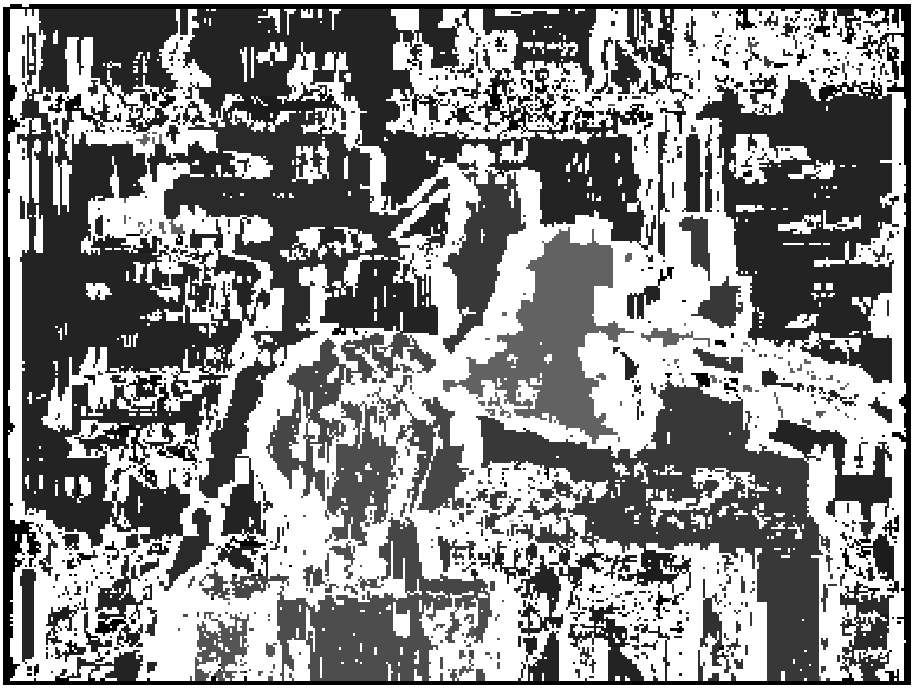

Figure 5: Carte de validité des disparités gauche-droite

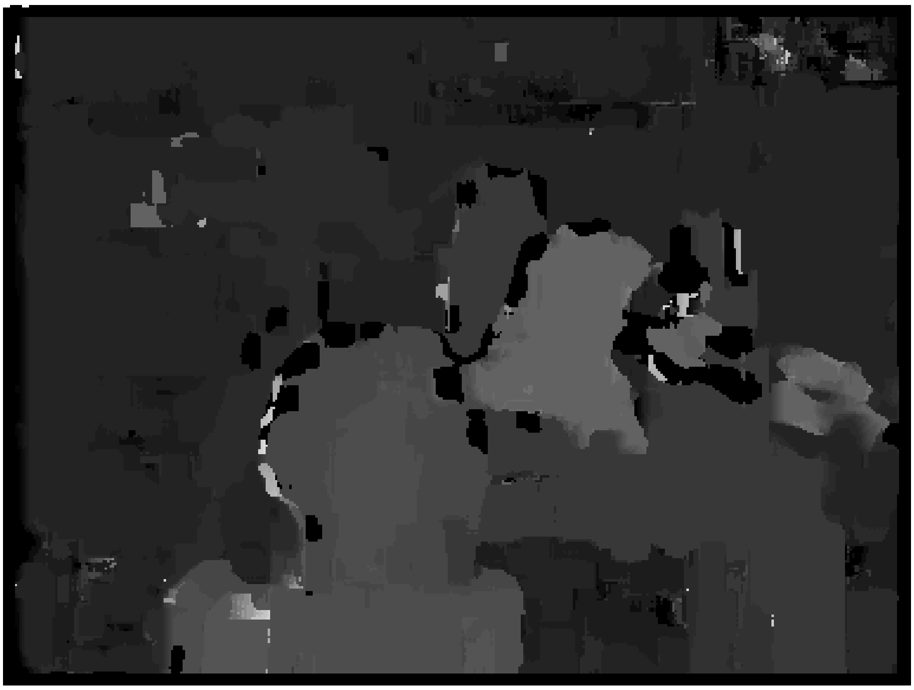

Figure 6: Carte de disparité de l'image de gauche pour une fenêtre de corrélation de 5

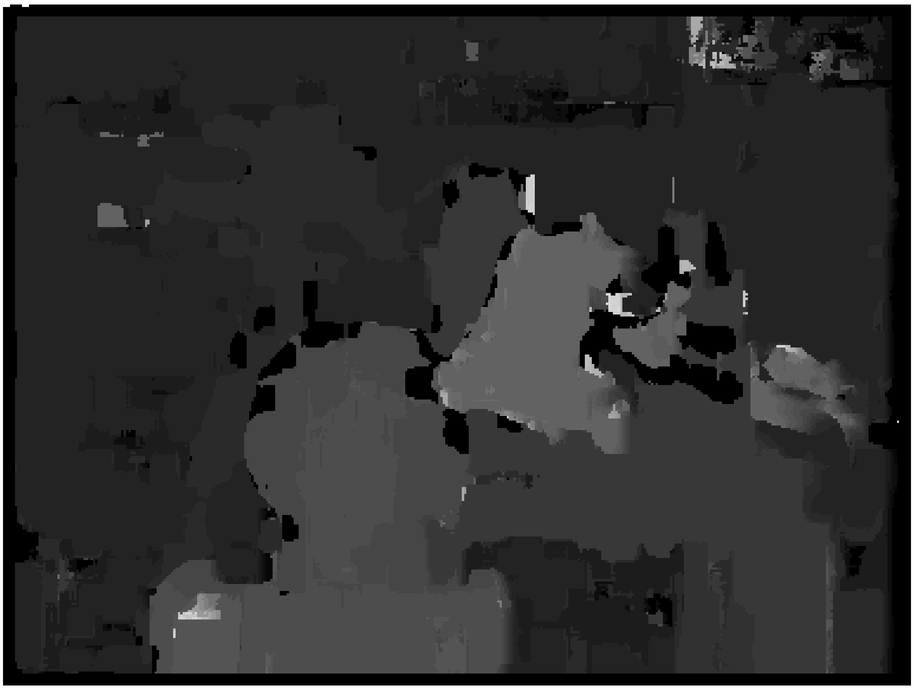

Figure 7: Carte de disparité de l'image de droite pour une fenêtre de corrélation de 5

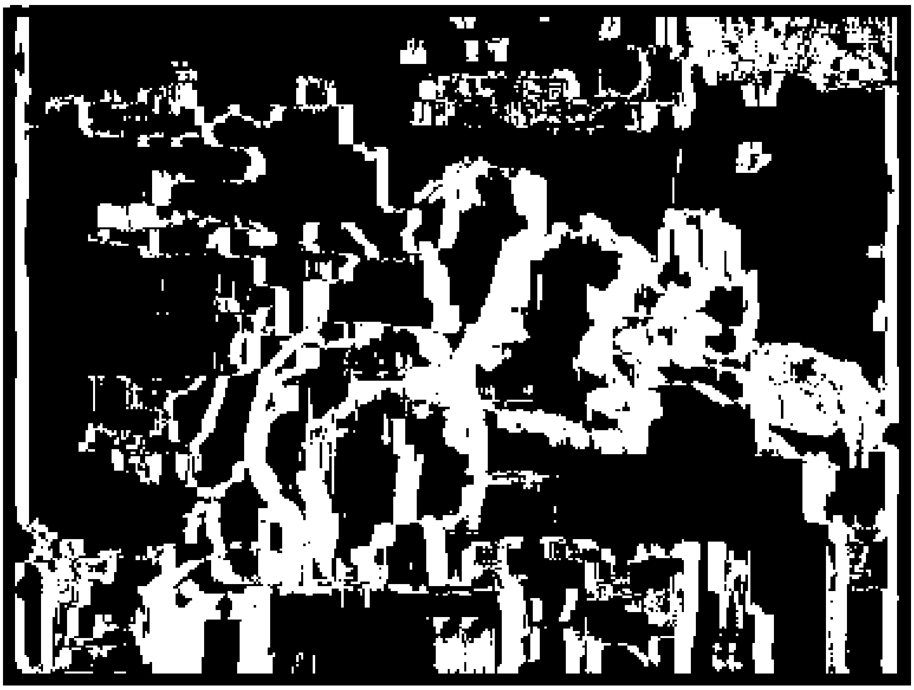

Figure 8: Carte de validité pour une fenêtre de corrélation de 5

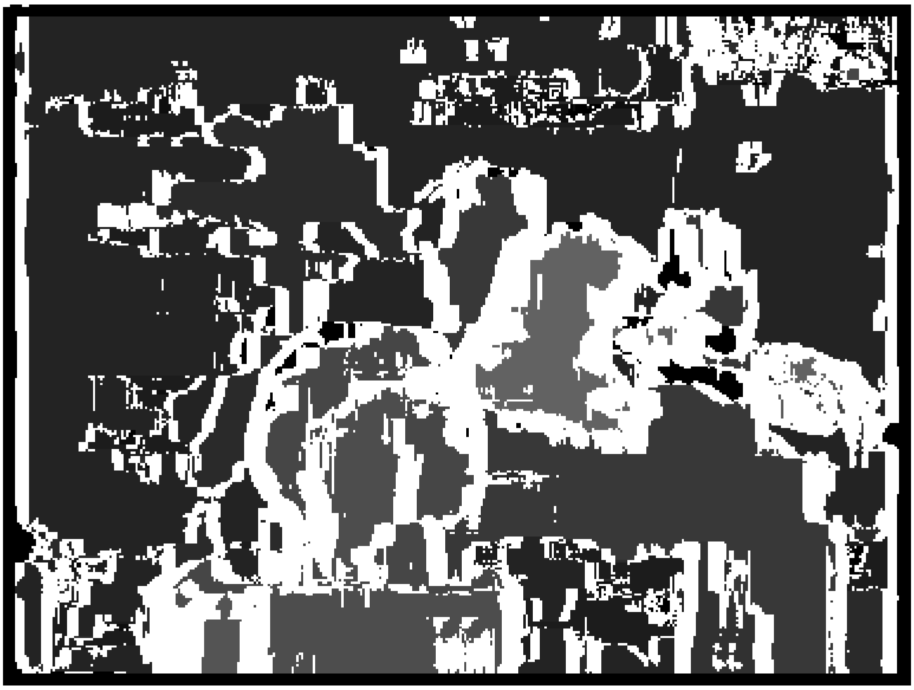

Figure 9: Carte de disparité finale pour une fenêtre de corrélation de 5

4 Calcul efficace de SSD

Pour rappel, la formule de la SSD est :

$$SSD(x_l, y, s) = \sum_{i=-w_x}^{w_x} \sum_{j=w_y}^{w_y} (I_l(x_l + i, y + j) - I_r(x_l + i - s, y + j))^2$$

On remarque alors que l'un des premiers éléments d'optimisation est de calcul une fois la différence et de multiplié le résultat par lui même. On remarque aussi des redondances dans le calcul et qu'on peut appliquer un algorithme récursif plutôt que deux boucles for. On va essayé alors de remplacer les paramètres i et j par deux équations de récurrence :

$$P(x, y, s) = (I_l(x_l, y) - I_r(x_l - s, y))^2$$
$$Q(x, 0, s) = \sum_j P(x, j, s)$$
$$Q(x, y + 1, s) = Q(x, y, s) + P(x, y + 2m + 1, s) - P(x, y, s)$$
$$N(0, y, s) = \sum_i Q(0, y, s)$$
$$N(x + 1, y, s) = N(x, y, s) + Q(x + 2n + 1, y, s) - Q(x, y, s)$$

Les deux équations de récurrences sont Q et N ici, on note n et m, la longueur et la hauteur de la fenêtre de corrélation. On obtient alors une équation qui nous calcul pour tout i, la disparité en chaque ligne j. Chaque appelle récursif fait un seul appelle récursif, on voit que N ne fait pas de multiplication et Q non plus. Seul P calcul une multiplication. Pour savoir le nombre de multiplication que l'on a, on peut donc calcul le nombre de fois où P sera appelée. Chaque Q appelle au plus deux fois P, chaque N appelle au plus deux fois Q. Pour un seul appelle à N, on calcul alors 2 * 2 = 4 P. Ici on a pas encore pris en compte les différents appels récursifs. Si on les prend en compte, on peut imaginer construire un arbre de récurrence, si on fait se travail, on remarque qu'il y aurait en moyenne n log(n) feuille dans l'arbre, soit n log n calcul de P pour l'équation Q. Si on ajoute N, on remarque alors qu'il y aura n log n calcul de Q. On a donc en moyenne n log n * n log n calcul de P, on est donc en $\theta(nlog(n)^2)$ qui est plus petit que n^2.

Le nombre d'addition quand a lui est identique à celui des multiplication (à une constante près), il y a donc en complexité le même nombre de multiplication que d'addition.

5 Conclusion

Dans ce rapport nous avons vu comment calculer la carte de disparités d'une image, pour cela nous avons utiliser la SSD qui consiste à calculer la différence au carré entre les positions des points de l'image de gauche et droite. Puis nous avons chercher à vérifier les points de disparités en les comparant avec les cartes de disparités des deux images, ce qui nous a donné une carte de disparité avec les points occultés positionnés. Et enfin nous avons analyser un moyen de calculer plus efficacement la SSD par rapport à ce que nous avions fait précédemment.