M3DA - TP 3: stéréovision dense

François Lepan

13 octobre 2013

Introduction

Dans le TP précédent nous avons vu comment faire de la stéréovision via l'utilisation de "coins" et droites épipolaires . Cette méthode permettait de faire de la reconstruction 3D basé sur la géométrie des scènes. Dans ce TP nous allons voir comment faire de la stéréovision avec non plus quelques pixels mais tous les pixels de l'image ainsi que les niveaux de gris de la scène.

Afin de simplifier le problème on utilisera une configuration canonique. (*cf.* Fig.1). En effet dans cette configuration on va pouvoir directement calculer la disparité qui est la différence entre les abscisses de deux points homologues.

La dernière étape sera de calculer la disparité en chaque point et la stocker dans une image nous donnant la carte des disparités.

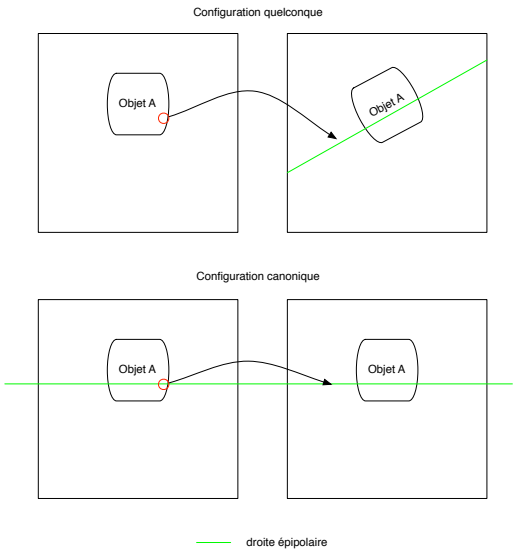

Figure 1 – En bas configuration canonique entre deux images d'une scène et en haut une configuration quelconque.

1 Similarité par SSD

La première étape est de maximiser la similarité et donc minimiser la dissimilarité entre deux voisinages.

Pour cela on va (*cf.* Fig.2) :
- définir un voisinage centré sur chaque pixel,
- utiliser une fonction de dissimilarité sur les deux voisinages courant,
- et enfin on va opérer un décalage sur l'une des images afin de repérer des voisinages similaires.

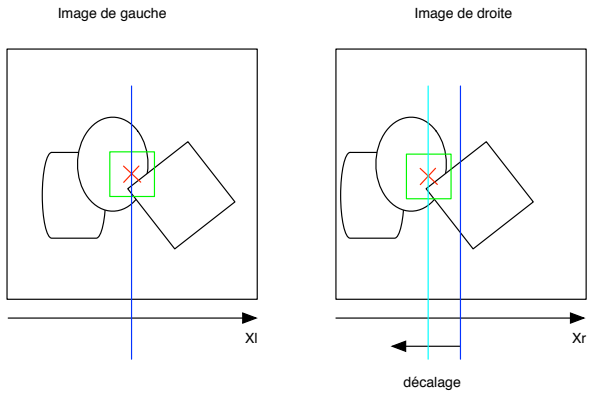

FIGURE 2 – Schéma du calcul de la minimisation de la dissimilarité

La fonction de dissimilarité utilisée sera la somme des différences au carré (SSD [1]).

Dans le TP on effectue cette opération de calcule du SSD sur toute l'image dans la fonction *iviLeftDisparityMap* avec l'image de gauche comme référence et *iviRightDisparityMap* avec l'image de droite comme référence.

On va calculer pour chaque décalage le SSD que l'on va stocker dans *mSSD* et on ne gardera que les valeurs pour lesquels le SSD est le minimum que l'on stockera dans *mMinSSD*. On remarque aussi que dans la fonction *iviLeftDisparityMap* les pointeurs permettant d'accéder aux pixels des images subissent un décalage égale à la moitié de la taille du voisinage. Ceci permettant de ne pas faire de dépassement de tableau lors du calcule du SSD.

Afin d'avoir la carte des disparités, pour chaque décalage on calcule le SSD avec la méthode *iviComputeLeftSSDCost* et on comparera les valeurs obtenu avec les autre décalages afin de récupérer les coûts (en décalage) minimum.

Après calcule on obtient la Fig.3.

1. Sum of Squared Differences

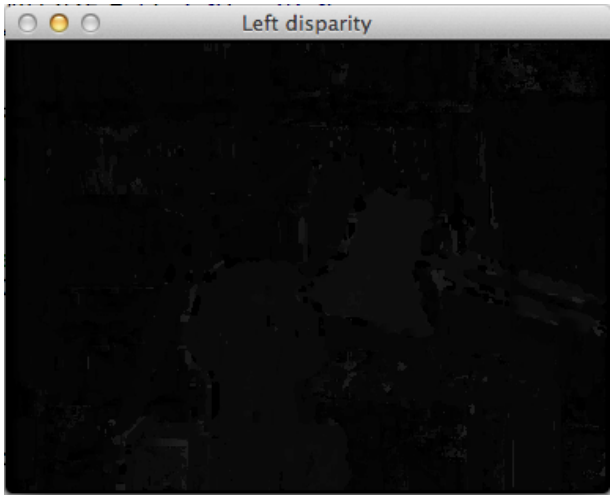

FIGURE 3 – Carte des disparités avec l'image de gauche comme référence

Cette image est rendu visible via les deux fonctions *minMaxLoc* et *normalize*. La fonction *minMaxLoc* permet de récupérer la valeur minimum (*min*) et la valeur maximum (*max*) de l'image. Ensuite on effectue une normalisation sur ces deux valeurs *min* et *max* c'est a dire une égalisation de l'histogramme (*cf.* Fig.4).

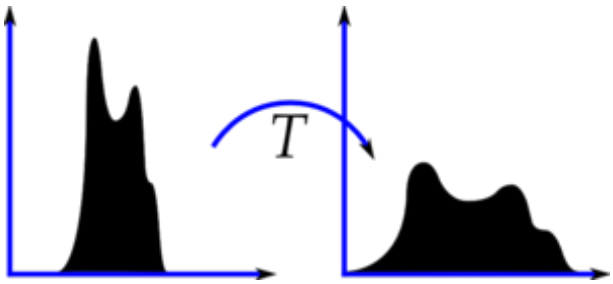

FIGURE 4 – Schéma de l'égalisation d'un histogramme (source : `http://en.wikipedia.org/wiki/Histogram_equalization`)

2 Vérification gauche-droite

La prochaine étape consiste à faire une carte de disparité qui est un combiné des deux cartes (gauche et droite). Dans cette carte on ne va garder que les disparités de la carte gauche (ayant l'image gauche comme référence) qui possède un homologue dans la carte de disparité droite.

Pour le moment on a que la carte de disparité avec l'image gauche comme référence. On va maintenant calculer la carte de disparité pour l'image droite en utilisant la fonction *iviRightDisparityMap*. La Fig.5 est le résultat de cette fonction.

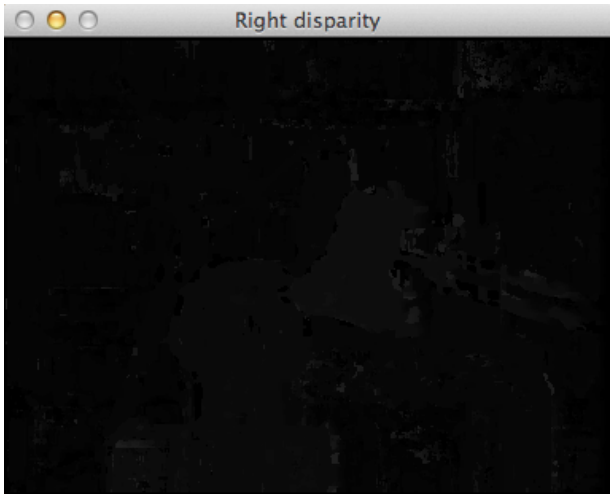

FIGURE 5 – Carte des disparités avec l'image de droite comme référence

Ensuite avec ces deux cartes on va rechercher les disparités homologues. Pour ce faire on va compléter la fonction *iviLeftRightConsistency* qui rempli deux matrices *mValidityMask* et *mDisparity*.

mValidityMask correspond à un masque de validité remplit de la façon suivante : si pour un pixel la disparité à bien été calculé alors on met 0 sinon on met 255. De ce fait les pixels marqué comme incorrecte sont bien visible (*cf.* Fig.6).

FIGURE 6 – Masque de validité

Enfin grâce à ce masque on peut récupérer les disparités correcte, il suffit de ne prendre que les disparités à la position ou le masque à comme valeur 0 (*cf.* Fig.7).

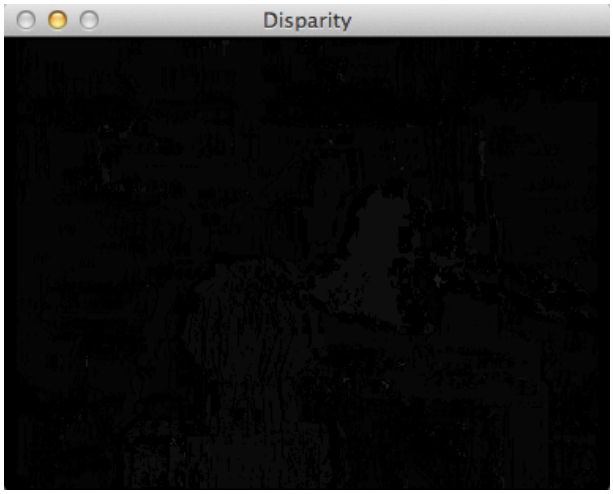

FIGURE 7 – Carte des disparités cohérente

Conclusion

Cette méthode de stéréovision est plus précise que la précédente car elle se base sur tout les pixels de l'image ainsi que la photométrie des deux images. Mais elle s'exécute plus lentement que la précédente et elle possède des contraintes lié à la photométrie de chaque image.