

# SY32 – Vision et apprentissage

## Projet : Mise en correspondance stéréo

### 1 Consignes

Implémentez vous-même un algorithme de mise en correspondance stéréo. Utilisez une technique de votre choix, qui ne soit pas basée sur de l'apprentissage automatique. Vous êtes libres d'apporter des pré- et post-traitements à votre implémentation.

*NB : L'algorithme de block-matching par recherche exhaustive avec une disparité limite est décrit dans le TD06, vous pouvez repartir de cette base ou bien opter pour un tout autre algorithme.*

Les métriques suivantes seront utilisées pour classer vos algorithmes :

- qualité => pourcentage de pixels avec une erreur de disparité  $> s$  pour  $s \in \{2, 3\}$  (en omettant les régions occultées, cf § cartes d'occultation)
- rapidité => temps d'exécution en ms
- ratio qualité/rapidité => à partir des deux métriques précédentes

### 2 Données

L'algorithme sera testé sur les images du jeu de données <https://vision.middlebury.edu/stereo/data/scenes2003/> qui est simple à prendre en mains. Les images sont déjà rectifiées de telle sorte à ce que les lignes de même hauteur soient les lignes épipolaires conjuguées.

Utilisez les images standard de ce dataset, c'est-à-dire *quarter-size*, de résolution  $450 \times 375$ . Choisissez la version au format png, donnée dans les fichiers *cones-png-2.zip* et *teddy-png-2.zip*.

Les images à traiter s'appellent *im2* et *im6*, respectivement pour les vues gauche et droite. Les disparités vérité correspondants à ces images comme référence sont données respectivement dans les fichiers *disp2* et *disp6* (images à 1 seul canal pour coder la disparité du pixel). Elles sont codées en valeur absolue, c'est-à-dire que la vérité est toujours donnée en disparités positives. Elles sont codées par des entiers dans  $[[0; 256[[$  pour des disparités réelles 4 fois plus faibles dans  $[0; 64[$ . Lors de la comparaison de vos disparités avec la vérité, il faut donc veiller à mettre à l'échelle  $\div 4$  les disparités vérité.

NOTE : Le codage des disparités vérité étant limité à un maximum de 63,75, il serait inconvenu de permettre le calcul de disparités supérieures à cette valeur

dans vos implémentations de mise en correspondance stéréo.

#### **Cartes d'occultations :**

Des cartes d'occultations sont disponibles sur la page du dataset, sous le nom *occl.png*. Elles servent pour l'évaluation des résultats. Des portions des scènes peuvent être non visibles à certains points de vue, ce qui empêche de trouver une bonne disparité; c'est ce que l'on appelle les occultations. Les cartes d'occultations sont en fait des masques binaires, où le blanc encode les régions non occultées et le noir les régions occultées. Elles sont données par rapport aux images *im2* (gauche) uniquement. Lors de l'évaluation, nous évaluons les disparités données aux positions de l'image gauche, et nous excluons les zones occultées données par ces cartes.

Il ne faut en aucun cas utiliser les cartes d'occultation dans votre algorithme d'estimation des disparités.

Les données à utiliser sont également regroupées dans le fichier *projetstereo.zip* disponible sur le Moodle.

### **3 Travail à rendre**

A rendre pour le 8 mai maximum, minuit (la date ne sera pas repoussée, anticipez autant que se peut).

Le code doit être compatible python 3.8 et se limiter aux librairies utilisées jusqu'au TD06.

Il doit pouvoir traiter une paire d'images (gauche et droite) données en argument. Il doit être en mesure d'enregistrer les disparités calculées dans une image, via un troisième argument. L'image de disparité en sortie doit correspondre aux disparités avec comme référence les points de l'image gauche. Les données de cette sortie doivent être multipliées par 4 et codées de manière standard par des entiers dans  $[[0; 255]]$  (de manière similaire au codage des disparités vérité, ce qui permet un affichage bien visible avec tout logiciel de visualisation d'images).

Le code doit pouvoir être appelé en ligne de commande avec exactement le nom et les options suivantes :

```
/NOM_GROUPE/stereomatch.py im_gche.png im_dte.png disp_sortie.png
```

ou

```
/NOM_GROUPE/stereomatch.py im_gche.png im_dte.png sans-sortie
```

de sorte à pouvoir utiliser un script pour mesurer le temps d'exécution, n'incluant pas le temps de la conversion/enregistrement des disparités dans un fichier.

### **4 Barème**

- 6pts : Rapport expliquant le fonctionnement de l'algorithme choisi et les raisons de ce choix, ainsi que les choix pour les paramètres, variantes et autres traitements éventuels.

- 3pts : Code.
- 1pt : Performances – les meilleurs algorithmes ou les algorithmes les plus spécifiques seront invités à être présentés en 5min par leurs auteurs lors du dernier cours.

**Attention :** 0 pour la partie code si le code ne marche pas selon les conditions demandées dans l'énoncé, après 1 relance et 1 seule.

## 5 Références à lire, sources d'inspiration

- Chambon, Sylvie. Mise en correspondance stéréoscopique d'images couleur en présence d'occultations. Thèse de doctorat.  
Lien : <http://thesesups.ups-tlse.fr/5/>
- Références indiquées à la page <https://vision.middlebury.edu/stereo/data/> (Scharstein, Hirschmüller et co-auteurs)

## 6 Taxinomie de méthodes existantes (pour info)

