

SY32 – Vision et apprentissage

Projet : Mise en correspondance stéréo

11 mai 2023

1 Consignes et métriques d'évaluation

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 TD-V04, vous pouvez repartir de cette base ou bien opter pour un tout autre algorithme.

Les métriques suivantes seront utilisées pour comparer l'efficacité de vos algorithmes (pour vous permettre de quantifier vous-même vos améliorations, le code utilisé est donné dans l'archive *projetstereo.zip* fournie avec le sujet) :

- **qualité** => pourcentage de pixels avec une erreur de disparité $> s$ pour $s \in \{1, 2\}$ (incluent les erreurs de détection d'occultations, cf § cartes d'occultations)
- **rapidité** => temps d'exécution en ms
- **graphe de ratio qualité/rapidité** => à partir des deux métriques précédentes

N'incluez pas d'évaluation dans le code final que vous rendez !

Dans vos cartes de disparités, codez les pixels pour lesquels vous jugez la disparité invalide par la valeur particulière 0 (avec les scènes étudiées, vous ne devriez jamais obtenir de disparités valides à 0 car elles signifient un point 3D à l'infini). Les disparités estimées indiquées comme invalides devraient correspondre à la détection d'occultations et sont prises en compte lors de l'évaluation de la qualité globale des cartes de disparités.

2 Date de rendu

A rendre pour le **11 juin** maximum, minuit (la date ne sera pas repoussée, anticipez autant que se peut).

Compactez votre travail dans une archive zip, puis déposez-le dans la section « Projets > Rendu projet Stéréo » du Moodle SY32.

3 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 450×375 . 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 correspondant 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, la gestion des occultations est prise en compte pour éviter de favoriser les algorithmes qui densifient coûte que coûte les cartes de disparités estimées, ou à l'inverse, ceux qui se contenteraient de ne faire correspondre qu'une faible portion des points de la scène les plus sûrs.

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

Les données à utiliser, ainsi qu'un script d'évaluation, sont également regroupés dans le fichier *projetstereo.zip* disponible sur le Moodle.

4 Travail à rendre

Éléments à rendre :

- Rapport en PDF, de 5 à 10 pages
- Code source « `stereomatch.py` » (+ éventuelles dépendances si décomposé en plusieurs fichiers)
- **Rien d'autre, et ne mettez pas `stereomatch.py` dans un sous-dossier**

Arborescence standard à respecter, pour un bon fonctionnement du script d'évaluation de l'ensemble des projets :

```

/NOM_GROUPE/
├── :
├── /rapport_GROUPE.pdf .2 :
├── stereomatch.py
└── :

```

Le code doit être compatible python 3.8 et se limiter aux bibliothèques listées à la fin de ce document (bibliothèques de base ou vues en TD).

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 des disparités 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 $[[0;255]]$, **dans une image à un seul canal de niveaux de gris** (de manière similaire au codage des disparités vérité, ce qui permet un affichage bien visible avec tout logiciel de visualisation d'images).

Votre code ne doit rien faire d'autre que de calculer la carte des disparités et l'enregistrer dans une image !

Afin de pouvoir évaluer vos algorithmes avec un script standard, le code doit pouvoir être appelé en ligne de commande avec exactement le nom et les options suivantes (seul le nom du groupe doit être adapté) :

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

Note : N'adaptez pas des paramètres différents pour chaque scène étudiée ! Optimisez vos paramètres de manière globale pour l'ensemble des scènes.

4.1 Comment proposer des algorithmes alternatifs ?

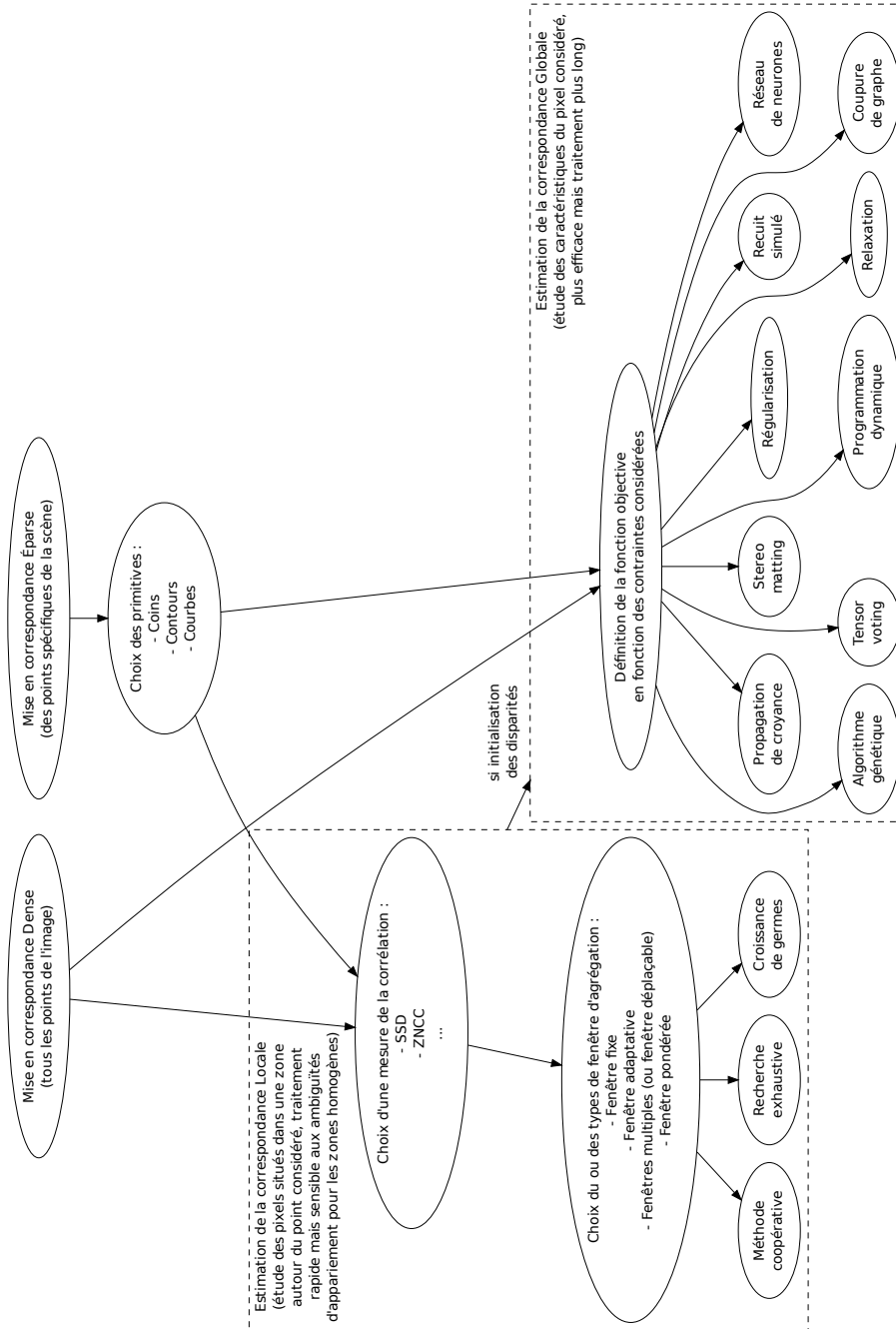
Au cours de vos essais, vous pouvez avoir remarqué plusieurs méthodes ou jeux de paramètres intéressants (méthode plus précise, méthode plus rapide, etc). Si vous souhaitez proposer plusieurs codes, organisez alors chaque algorithme dans un dossier séparé dont le nom commence par « algo » **en lettres minuscules**, sous la forme :

```

/NOM_GROUPE/
├── :
├── /algoXXX/
│   ├── :
│   ├── stereomatch.py
│   └── :
├── /algoYYY/
│   ├── :
│   └── stereomatch.py
└── :

```


7 Taxinomie de méthodes existantes (pour info)



8 Liste des bibliothèques python permises

- argparse
- csv
- getopt
- io
- json
- matplotlib
- numba
- numpy
- opencv (excepté les fonctions toutes faites de calcul des disparités)
- os
- pandas
- pillow (PIL)
- scikit-image
- scikit-learn
- scipy
- sys
- time
- timeit
- tqdm