

INF6804 Vision par ordinateur

H2020 – Travail Pratique 2

Description et comparaison de régions d'intérêt

Objectifs:

- Permettre à l'étudiant de se familiariser avec la description d'images par histogramme de gradients orientés (HOG).
- Permettre à l'étudiant de se familiariser avec la description d'images par BRIEF.
- Apprendre les principes de l'estimation de disparités en stéréoscopie.

Remise:

- *Tout votre code source* (nous devrions pouvoir exécuter vos tests)
- Un rapport (*.pdf format* de 8 à 15 pages avec une taille de police de 10)
- Remise avant le 13 Mars 2020, 17h00, sur Moodle –*aucun retard accepté*
- *Vous devez aussi soumettre votre rapport sur TurnItIn!*
 - Enregistrez-vous sur www.turnitin.com en utilisant les informations sur Moodle!

Références:

- Voir les notes de cours sur Moodle (Chapitre 2 et chapitre 4)

Autres directives:

- Les TPs doivent obligatoirement être faits en équipe de deux, ne remettez qu'une seule version du rapport/code!
-

Présentation

Dans ce TP, vous devez caractériser deux méthodes de description de régions d'intérêt dans une image et déterminer quelle méthode est la meilleure et dans quelles circonstances. Une description de votre travail, de vos expériences, et les réponses aux questions de cet énoncé doivent se retrouver dans un rapport.

Dans ce travail, vous devez comparer deux méthodes de description de région d'intérêt dans des images, soit une technique basée sur la description par HOG, et une technique basée sur BRIEF. Celles-ci ont été vues brièvement en classe – vous pouvez utiliser vos notes de cours comme référence pour comprendre leur fonctionnement de base. Pour plus de détails, allez creuser sur internet! Votre but est de déterminer quelle approche fonctionne mieux pour la recherche de régions correspondantes entre deux images dans des conditions variées (e.g. en utilisant différentes tailles de régions, sous une variation d'illumination, sous une transformation affine, etc.). Pour comparer les deux méthodes, vous devrez résoudre une problématique d'estimation de disparité en stéréoscopie. Vous allez devoir décrire chaque image avec les descripteurs et ensuite utiliser un algorithme de correspondance stéréo, comme *semi-global matching* (SGM), pour faire l'estimation des disparités.

Dans votre rapport, vous devez inclure les éléments suivants (noté sur 20 pts):

1. Présentation des deux méthodes (4 pts):

Dans vos propres mots, donnez la description générale et les principes de vos deux méthodes.

2. Hypothèses de performance dans cas spécifiques (3 pts):

Identifier, selon votre compréhension théorique des deux techniques de segmentation étudiées uniquement, quelle approche devrait être meilleure que l'autre en fonction d'au moins TROIS cas d'utilisation. Par exemple, quelle est la meilleure approche à utiliser si les régions comparées sont très grandes (e.g. 200 pixels par 200 pixels)? Pourquoi? Et si leur contenu est relativement uniforme?

3. Description des expériences, bases de données et critères d'évaluation (2 pts):

Décrire en détail les expériences réalisées pour vérifier vos hypothèses du point précédent. Quelles bases de données avez-vous utilisé? Quelles sont les difficultés de celle-ci? Quels sont vos critères d'évaluation?

4. Description des deux implémentations utilisées (2 pts):

Décrire l'implémentation des deux approches étudiées ainsi que l'algorithme choisi pour réaliser l'estimation de disparités. Si vous n'avez pas tout écrit le code vous-même, d'où provient-il? A-t-il demandé des modifications? Sinon, de quel articles (ou site web) vous êtes-vous inspiré pour l'écrire?

Dans tous les cas, quels sont les paramètres principaux utilisés? Comment ont-ils été choisis?

5. Présentation des résultats de tests (3 pts):

Donner les résultats d'évaluation de performance tirés de vos expériences en lien avec les hypothèses du premier point. Utilisez un format approprié — tableaux, figures, ...

6. Discussion des résultats et retour sur hypothèses (3 pts):

Discuter des résultats du quatrième point en fonction des hypothèses du premier point. Quelles hypothèses sont supportées par les résultats? Lesquelles ne le sont pas? Quels tests sont restés sans conclusion? Quels tests pourriez-vous améliorer, et comment?

7. Lisibilité et complétude (3 pts):

À part le contenu, le format doit être soigné et complet.

Vous pouvez utiliser n'importe quelle base de données pour ce travail, mais si vous n'en connaissez pas, nous vous suggérons d'utiliser les bases de données mentionnées dans la section ressources.

Lors des séances de laboratoire, n'hésitez pas à poser des questions aux chargés — ils peuvent vous aider avec tout problème technique si vous travaillez sur Windows/Linux, ou bien avec votre code si vous travaillez avec du C/C++ ou avec Python/Matlab.

Vous serez pénalisés de 50% de la note totale si vous ne remettez pas votre code. De plus, si votre rapport n'est pas remis sur TurnItIn, celui-ci ne sera pas corrigé. L'ordre de la présentation n'est pas important, tant que tous les éléments ci-haut sont présents. N'oubliez pas de citer vos références!

Estimation de disparités en stéréoscopie

Vous trouverez la référence à un cours sur la correspondance stéréo dans la section ressources. Simplement dit, la tâche d'estimation de disparités consiste à, pour un pixel donné p à la position (x, y) dans une image de référence, de trouver le même pixel q à la position $(x + d, y)$ dans une autre image. La disparité d correspond donc à la distance entre le même pixel des deux images. Pour prédire d , il faut tout d'abord décrire les images gauches et droites avec les descripteurs. Par la suite, il faut former un volume de coût, soit un tenseur de dimensions $H \times W \times D$ (H = hauteur de l'image, W = largeur de l'image et D = nombre de disparités), où le coût d'un élément du volume correspond à la distance (L1, L2 ou autre) entre les éléments des descripteurs. La figure 1 montre un petit exemple montrant la formation du volume de coût, où les deux descripteurs sont des images 4×4 et la disparité maximale est aussi de 4. Ce processus peut être vu comme glisser une image par dessus l'autre, et à

chaque décalage possible, calculer la distance entre les descripteurs. Le processus est répété pour l'ensemble des valeurs de disparités possibles. Finalement, à la figure 2, on observe un exemple d'une paire d'image stéréo avec la carte de disparité correspondante pour l'image de gauche. Une couleur de pixel plus claire signifie une disparité plus grande.

Après avoir calculé le volume de coût, il faut appliquer un algorithme de

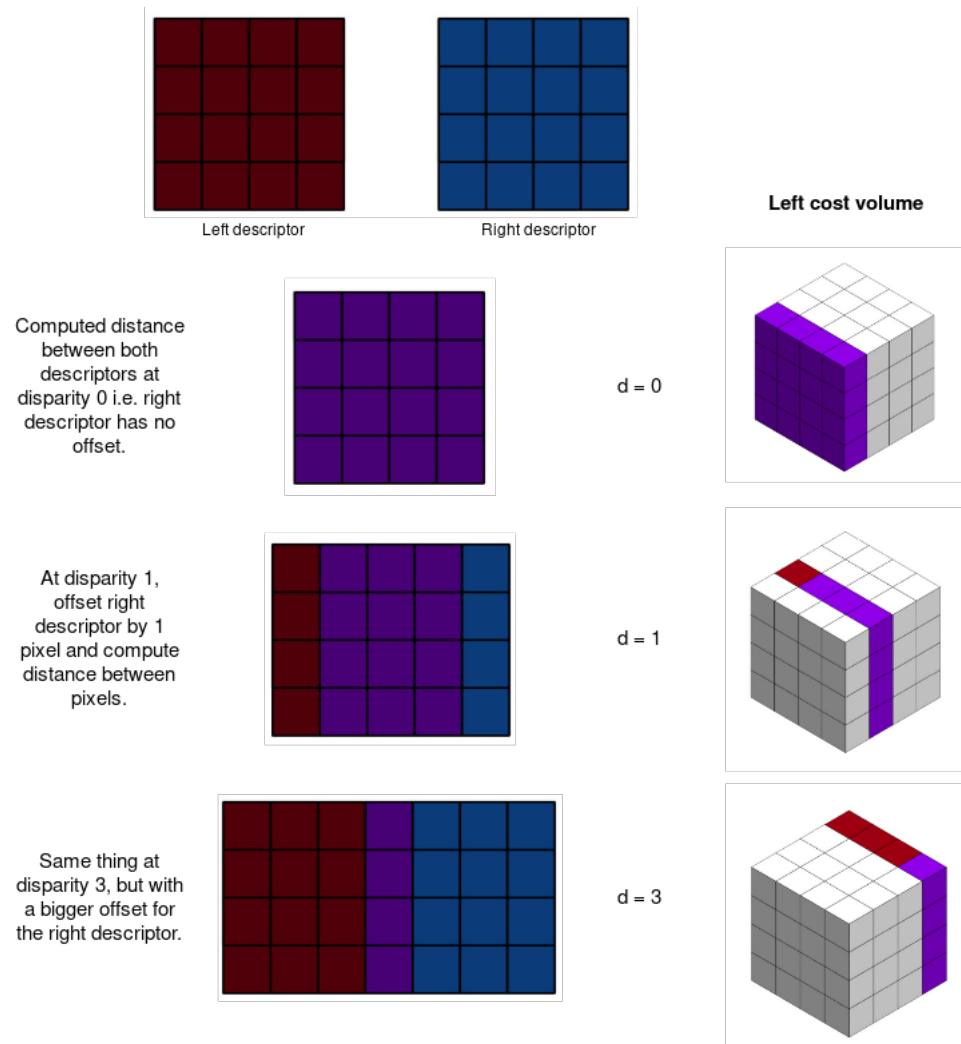


Figure 1: Diagramme montrant comment calculer le volume de coût à partir des descripteurs des images de gauche et droite. **Rouge**: descripteur gauche, **Bleu**: descripteur droit et **Mauve**: distance calculée entre les descripteurs.



Figure 2: Pair d'image stéréo avec la carte de disparité calculée.

correspondance stéréo pour raffiner le volume. Pour cette partie, vous êtes libre d'utiliser l'algorithme de votre choix. Ceci étant dit, nous vous suggérons de regarder l'algorithme SGM puisqu'il s'agit d'une des meilleures méthodes classiques et qu'il y a un grand nombre d'implémentations en ligne (Github ou OpenCV). La référence à l'article original peut être retrouvée dans la section ressources. Chaque algorithme de correspondance stéréo aura sa propre manière de produire la carte de disparité finale.

Ressources

Ensembles de données:

- KITTI dataset (http://www.cvlibs.net/datasets/kitti/eval_stereo.php)
- Middlebury dataset (<http://vision.middlebury.edu/stereo/data/>)
- CATS dataset (<http://bigdatavision.org/cats/data.html>)

Stéréo/correspondance semi-globale(SGM):

- Cours sur la stéréo de l'université de Toronto (http://www.cs.toronto.edu/~fidler/slides/2015/CSC420/lecture12_hres.pdf)
- Article SGM (<https://core.ac.uk/download/pdf/11134866.pdf>)

Librairies de vision par ordinateur:

- OpenCV (https://docs.opencv.org/4.0.0/d9/df8/tutorial_root.html)
- scikit-image (https://scikit-image.org/docs/stable/auto_examples/index.html)

Librairies d'apprentissage profond:

- PyTorch (<https://pytorch.org/tutorials/>)
- Tensorflow (<https://www.tensorflow.org/tutorials>)

Python:

- Guide (<https://wiki.python.org/moin/BeginnersGuide/Programmers>)
- NumPy (<https://docs.scipy.org/doc/numpy/user/quickstart.html>)

Matlab:

- Guide (http://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf)
- Aide-mémoire (<http://web.mit.edu/18.06/www/Spring09/matlab-cheatsheet.pdf>)