

# Cours - Introduction à la vision par ordinateur

## TP 2 : Détection de contours

ISIA - 2015-2016

### 1 Détection de contours

L'objectif de ce projet consiste à comparer différents algorithmes de détection de contours. Vous avez déjà trois algorithmes présents de détection de contours dans OpenCV : Sobel, Laplace et Canny (`cv::Canny`<sup>1</sup>, `cv::Sobel`<sup>2</sup>, `cv::Laplacian`), il vous faudra en programmer une autre de votre choix. N'hésitez pas à vous documenter sur d'autres méthodes de détection de contours que celles vues en cours, et expliquez-les brièvement dans votre rapport.

#### 1.1 Evaluation quantitative des détecteurs

Vous allez faire une comparaison de 4 détecteurs de contours (Sobel, Laplace, Canny + une autre de votre choix) avec 10 images différentes de votre choix. Vous pouvez utiliser les images de la base d'images de Berkeley<sup>3</sup> qui contient la vérité terrain (segmentation manuelle des contours par des humains). Le dataset *Berkeley Segmentation Data Set 300* est suffisant et vous pouvez récupérer vos 10 images ainsi que leur vérité terrain en parcourant le jeu de données en ligne. Vous pouvez utiliser le script sur Claroline.

Pour faire cette comparaison, vous avez besoin de programmer quelques fonctions simples de mesure de performances.

---

<sup>1</sup>[http://opencv-python-tutroals.readthedocs.org/en/latest/py\\_tutorials/py\\_imgproc/py\\_canny/py\\_canny.html#canny](http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html#canny)

<sup>2</sup>[http://opencv-python-tutroals.readthedocs.org/en/latest/py\\_tutorials/py\\_imgproc/py\\_gradients/py\\_gradients.html#gradients](http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_gradients/py_gradients.html#gradients)

<sup>3</sup><http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images.html>

Pour mener votre étude quantitative, pour chaque méthode, vous pourrez utiliser le plan suivant:

1. Calcul du gradient et seuillage. Il faudra essayer différentes valeurs de seuil pour une méthode afin de trouver la meilleure valeur possible, celle qui fonctionne le mieux pour toutes les images.
2. Comparaison des images de contours obtenue par un algorithme avec la référence (vérité terrain), en utilisant 5 mesures :
  - **contours-detectes** = nombre de pixels contours dans l'image calculée.
  - **contours-reference** = nombre de pixels contours dans l'image de référence.
  - **contours-correc** = nombre de pixels contours correctement détectés dans l'image calculée selon l'image référence. Pour cette grandeur, il faut tenir compte d'un décalage possible entre les images. Ce sont aussi les **vrai-positifs**.
  - **faux-positifs** = nombre de pixels détectés comme contours mais qui ne le sont pas dans l'image de référence.
  - **faux-negatifs** = nombre de pixels non détectés comme contours mais qui le sont dans l'image de référence.
  - **vrais-negatifs** = nombre de pixels non détectés comme contours et qui ne le sont pas dans l'image de référence.
3. En déduire les trois mesures suivantes pour évaluer les détecteurs :
  - $P = \text{contours-correc} / (\text{contours-correc} + \text{faux-positifs})$
  - $\text{TFP} = \text{faux-positifs} / (\text{faux-positifs} + \text{vrai-negatifs})$
  - $\text{TFN} = \text{faux-negatifs} / (\text{contours-correc} + \text{faux-negatifs})$
4. Analyser les résultats en discutant les trois critères d'évaluation suivant:
  - Bonne détection : est ce que tous les contours sont bien détectés ?

- Bonne localisation : est ce que les contours sont détectés au bon endroit ?
- Réponse unique : est ce qu'on obtient un seul résultat (contour) pour chaque contour réel ?

Pour l'ensemble des méthodes, faire un tableau de comparaison quantitative. Commentez les résultats des méthodes étudiées.

## 2 Fiabilité de la vérité terrain

Observez les vérités terrain des images proposées par différents utilisateurs pour une image donnée. Par exemple:

- <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images/gray/175032.html>
- <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images/gray/78004.html>
- <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images/gray/126007.html>
- <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images/gray/300091.html>
- <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images/gray/76053.html>

Décrivez qualitativement (mais synthétiquement!) les similitudes et différences entre annotateurs. Pouvez-vous qualifier perceptivement les traits que l'on retrouve systématiquement ? Quid des différences ?

Adaptez vos programmes d'évaluation pour estimer quantitativement l'accord inter-évaluateur, c'est-à-dire le nombre de pixels de contour sur lesquels ils s'accordent (ou pas). Commentez les chiffres obtenus.

## 3 Lecture conseillée

- Contour Detection and Hierarchical Image Segmentation P. Arbelaez, M. Maire, C. Fowlkes and J. Malik. IEEE TPAMI, Vol. 33, No. 5, pp. 898-916, May 2011; disponible sur Caroline