

TP : Détection de véhicule

A. Cord, J.-P. Tarel, T. Ridene

1 Introduction

Nous allons implanter un détecteur de véhicule à partir d'une caméra embarquée. Les images utilisées sont issues d'un simulateur de conduite et sont extraites de la base d'évaluation d'algorithmes de restauration de la visibilité nommée FRIDA (<http://perso.lcpc.fr/tarel.jean-philippe/bdd/frida.html>).

Dans votre compte rendu de TP, vos réponses devront être commentées et illustrées graphiquement afin de faciliter une validation visuelle des résultats. Les codes Matlab seront joints.

2 Exploration des données fournies

1. Lire et afficher les 5 images sans brouillard de nom `00?.bmp` et les 5 images avec brouillard de 80 mètres de distance de visibilité de nom `U00?.bmp`. Chaque image sans brouillard sera affichée dans une figure avec à côté sa version avec brouillard. La détection se fera sur les images tests à plusieurs échelles pour tenir compte de la variation de taille des véhicules suivant la distance.
2. Lire le fichier de données `BDD_vehicule.mat` qui contient trois variables :
 - `BDD_vehicule` : ensemble de 51 imagerie de taille 30x36x3 (valeur entière entre 0 et 255) format la base d'apprentissage
 - `BDD_vehiculesmall` : base d'apprentissage précédente construite à partir d'images en codage double (donc normalisées entre 0 et 1), avec centrage par mise à zéro de l'intensité moyenne de chaque imagerie, et réduction à la taille 5x6x3 par interpolation bilinéaire.
 - `Seach_Pattern` : structure contenant les différentes échelles pour la détection dans les images, et les différents paramètres pour chaque échelle, comme les régions d'intérêt (ROI), les pas d'extraction et les seuils de détection.

3. Afficher les 7x7 premières imagerie de la base d'apprentissage (BDD_vehicule) et les 7x7 imagerie de la base après réduction (BDD_vehiculesmall).

3 Détection à plusieurs échelles dans une image

Pour réaliser la détection sur une image, des imagerie sont extraites de l'image et chaque imagerie extraite va être comparée à celles contenues dans la base d'apprentissage. La comparaison entre deux imagerie se fera simplement par le calcul de la distance Euclidienne. Afin de pouvoir les comparer à celles de la base d'apprentissage réduite BDD_vehiculesmall, elles seront donc centrées en intensité et réduites à la taille 5x6x3. La base d'apprentissage contenant 51 imagerie de référence, le score de dissemblance entre une imagerie test et la base d'apprentissage sera calculé comme la moyenne des 3 distances les plus petites à ces 51 imagerie de référence. Ce score de dissemblance sera donc d'autant plus petit que l'imagerie considérée est ressemblante avec quelques imagerie de la base d'apprentissage.

Pour ne pas rater de détections, les imagerie seront extraites à intervalle régulier sur l'image à tester. Ce parcours de l'image sera réalisé à quatre échelles différentes, afin de tenir compte du changement de taille des véhicules en fonction de la distance. Pour ne pas chercher à détecter des véhicules dans des zones improbables de l'image comme le ciel, une région d'intérêt (ROI) est définie pour chaque échelle. On impose à ce que le coin supérieur gauche de chaque imagerie extraite soit dans cette région d'intérêt. Le choix des échelles et le réglage de tous ces paramètres, qui peuvent varier avec l'échelle, étant assez laborieux, un réglage a été sauvegardé dans la variable `Search_Pattern`. Cette variable `Search_Pattern` est un tableau de structures, chacune contenant les éléments suivants :

- `Search_Pattern.scale` est le facteur d'échelle des imagerie à extraire (4, 6, 8 et 12). Les imagerie extraites auront donc pour taille $[5*scale, 6*scale, 3]$.
- `Search_Pattern.ROI` donne la région d'intérêt dans l'image. La première valeur est le nombre de colonnes pour le début de la ROI, la deuxième le nombre en lignes du début de la ROI, la troisième la largeur en nombre de colonnes de la ROI et la quatrième la hauteur en lignes de la ROI.
- `Search_Pattern.StepX` et `StepY` est le pas horizontal et vertical en pixels entre deux imagerie extraites voisines.
- `Search_Pattern.Thresh` est le seuil de détection sur le score de dissemblance en dessous duquel une imagerie est considérée comme contenant

un véhicule.

- `Search_Pattern.color` est une couleur pour afficher les boîtes englobantes des détections avec différentes couleurs selon l'échelle.

A programmer en Matlab :

1. Pour chacune des quatre échelles `scale` de détection, afficher sur l'image test les contours des imagerie sélectionnées dont le coin supérieur gauche est dans la ROI, à l'échelle `[5*scale, 6*scale, 3]` et avec un pas correspondant à `StepX` et `StepY`. La fonction Matlab `rectangle` sera utilisée.
2. Ajouter au parcours construit à la question précédente, le calcul du score de dissemblance. Pour chaque imagerie extraite de l'image en double, il faut la centrer en annulant son intensité moyenne, la réduire à la taille `[5, 6, 3]`, calculer les distances Euclidiennes avec toutes les images réduites de `BDD_vehiculesmall`. Score est obtenu comme moyenne des trois distances les plus faibles. Si le score est inférieur au seuil de détection, il y a détection et il faut afficher le contour de l'imagerie extraite sur l'image avec la fonction `rectangle` et de préférence avec la couleur indiquée dans `Search_Pattern.color`. Il est avantageux d'utiliser les fonctions Matlab `mean`, `imresize`, et `sort`, entre autres.
3. Faire la détection sur les cinq images sans brouillard. Est-ce qu'il y a des fausses alarmes ? Est-ce qu'il y a des non détections ?
4. Introduire un facteur constant, indépendant de l'échelle sur le seuil de décision `Search_Pattern.Thresh` afin de diminuer les erreurs de détection. Quelle est la valeur qui vous semble optimale ? Est-ce possible de trouver un facteur qui permet d'avoir une solution sans fausse alarme et sans non détection ?
5. Appliquez la détection sur les cinq images avec du brouillard. Que constatez-vous ? Expliquez ?

4 Question Bonus

1. Remplacer le seuillage sur le score de dissemblance par une autre méthode de classification de votre choix. Régler les paramètres, comparez et discutez les résultats obtenus.