

TP 1 : Extraction des marquages et détection des obstacles avec la v-disparité

<http://perso.lcpc.fr/tarel.jean-philippe/ensta19/tp1/>
J.-P. Tarel

1 Introduction

Après quelques questions pour rappeler le fonctionnement de la toolbox *matlab* sur le traitement d'image, nous allons voir dans la première partie de ce TP comment la méthode d'extraction des marquages MLT (Median Local Threshold) peut être programmée et évaluée [1].

Grâce au calibrage d'une paire de caméras sur un véhicule, nous disposons de paires d'images stéréoscopiques. Le but de la deuxième partie de ce TP est de réaliser la détection des obstacles en utilisant la méthode v-disparité [2]. Pour cela, nous allons utiliser l'extracteur MLT sur l'image de gauche et de droite de la paire stéréoscopique, puis apparier ces extractions entre les deux images pour en déduire l'histogramme v-disparité. À partir de ce dernier, le profil de variation vertical de la disparité de la route est estimé afin, ensuite, de détecter les obstacles.

Remarque : Le code et les résultats doivent être commentés dans le compte rendu. Chaque étape devra être présentée graphiquement de manière à permettre l'évaluation visuelle des résultats. Les fonctions en *matlab* seront aussi jointes au compte rendu.

2 Lecture et affichage d'une image

La liste des fonctions *matlab* et de la *toolbox image processing* utiles pour les parties 2, 3 et 4 :

1. max, min, mean, sum,
2. imread, imwrite, double, rgb2gray
3. figure, plot, hold,
4. imshow, axis, colorbar, colormap,
5. fspecial, imfilter, medfilt2, imopen.

Une image de route nommée "01D.png" est fournie. Nous allons travailler dessus pour les parties 2, 3 et 4 afin de se familiariser avec la manipulation des images :

1. Lire l'image "01D.png" et la stocker dans la variable "ima". Utiliser la fonction "help imread" pour obtenir les propriétés de la fonction.
2. Observer le format de l'image lue dans le gestionnaire de variables. De quel type est chaque valeur de pixel ? Afficher l'image avec imshow.
3. Créer une deuxième variable "imad" contenant la même image en format double. Afficher l'image "imad" avec imshow. Que se passe t'il ? Diviser l'image "imad" par la valeur 255. Refaire l'affichage.
4. Convertir l'image "ima" en niveau de gris et la stocker dans "imagray". Convertir l'image "imad" en niveau de gris et la stocker dans "imadgray". Comparer numériquement les deux images (ne pas oublier de multiplier "imadgray" par 255).
5. Sauvegarder l'image résultant de la comparaison.

3 Propriétés d'une image

Sur l'image "imadgray" :

1. Afficher la valeur du pixel (232,212). Afficher la dimension de l'image. Stocker le nombre de ligne "nl", nombre de colonne "nc", et le nombre de bandes/canaux dans la variable "nb".
2. Créer un vecteur "maxlig" de taille "nl". Mettre la valeur max de chaque ligne dans le vecteur "maxlig" avec une boucle for. Faire de même avec le minimum et la moyenne sans boucle for. Faire un tracé de ces trois vecteurs sur le même graphique.
3. Calculer la moyenne, le max et le min et la somme de toute l'image.

4 Extraction des marquages

1. Filtrer l'image "imadgray" avec un filtre moyen et afficher le résultat.
2. Filtrer l'image "imadgray" avec un filtre médian, afficher le résultat et comparer avec le résultat du filtre moyen. Chercher un taille de filtre qui permet de faire disparaître les marquages.
3. Soustraire à l'image "imadgray", l'image obtenue avec le filtre médian.
4. Binariser l'image avec une valeur de seuil $s_{marquage}$ à choisir de façon à ce que les marquages soient bien détectés (à stocker dans "imask") et afficher le résultat. Est-ce que seulement des marquages sont extraits ?
5. Reprendre les trois questions précédentes en remplaçant le filtre médian par une ouverture morphologique. Est-ce que les résultats sont comparables ?

5 Évaluation de l'extraction

Différentes valeurs du seuil peuvent être utilisées lors de l'extraction des marquages. Un résultat de détection peut être évalué par sa courbe ROC (Receiver Operating Characteristic) définie comme le taux de bonnes détections en fonction du taux de fausses alarmes, lorsque l'on fait varier le seuil de détection, $s_{marquage}$ dans notre cas. Soit tp le nombre de vrai positif (pixels de marquage correctement extraits), fn le nombre de faux négatifs (pixels de marquage non extraits), tn le nombre de vrai négatif (pixels où aucun marquage n'est extrait), fp le nombre de faux positifs (pixels où un marquage est extrait alors qu'il est absent), le taux de bonnes détections est donné par $tpr = \frac{tp}{tp+fn}$ et le taux de fausses alarmes est donné par $fpr = \frac{fp}{fp+tn}$.

1. Sur l'image "01D.png", grâce à l'image de référence des marquages "01DR.png", calculer le taux de bonnes détections et le taux de fausses alarmes, pour la valeur précédemment choisie de $s_{marquage}$.
2. **Courbe ROC** : Pour plusieurs valeurs de $s_{marquage}$, calculer le taux de bonnes détections et de fausses alarmes. Construire la courbe ROC à partir de ces points.
3. **Comparaison** : Construire la courbe ROC de l'extracteur en changeant le filtre médian par une ouverture morphologique. Comparer les deux courbes ROC. Que pouvez-vous en déduire ?

6 Calcul des disparités

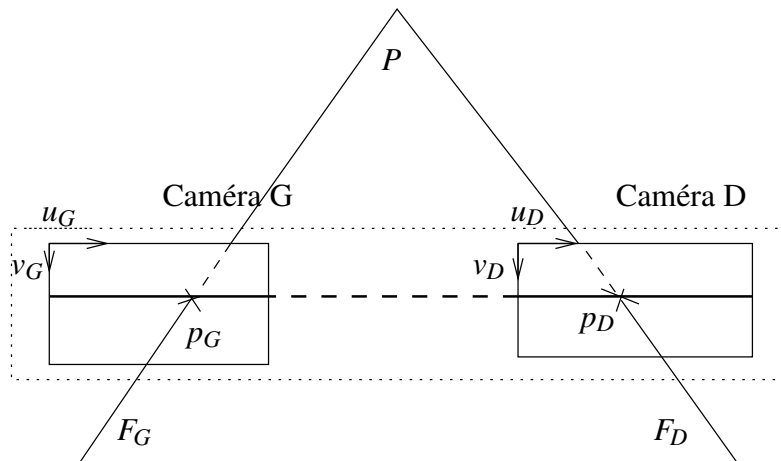


FIGURE 1 – La géométrie épipolaire pour le cas de caméras rectifiées.

Pour simplifier les calculs lors de l'appariement, on se place souvent en géométrie rectifiée où les deux plans images sont confondus dans un même plan, voir figure 1. En géométrie quelconque, il faut faire la reconstruction 3D de chaque paire de points appariés par triangulation. En géométrie rectifiée, un point 3D, P , a pour images (u_G, v_G) et (u_D, v_D) sur les caméras Gauche et Droite respectivement, avec $v_G = v_D$. De plus, la disparité définie par $u_G - u_D$ est inversement proportionnelle à la profondeur du point P .

Vous disposez d'une première paire stéréoscopique avec les images nommées 01G.png et 01D.png (G et D pour Gauche et Droite). Après application du détecteur MLT sur ces deux images et sélection des centres horizontaux, une paire stéréoscopique d'images binaires de centres est obtenue. Par appariement des centres de l'image gauche avec ceux de l'image droite, une carte de disparité partielle de l'image gauche sera produite.

1. Tracer sur les images 01G et 01D mises côte à côte, une ligne horizontale, pour différentes hauteurs. Que concluez-vous sur la géométrie épipolaire de la paire stéréoscopique ?
2. **Centres** : Appliquer l'extracteur MLT et obtenir les centres selon l'horizontale sur les images 01G et 01D avec la fonction *bwmorph*. Expliquer votre choix de la valeur du seuil d'extraction.
3. **Appariement** : Programmer la fonction d'appariement des pixels de l'image gauche avec ceux de l'image droite par un unique balayage par ligne sur l'image gauche. Construire pour chaque ligne les deux tableaux des positions des centres dans l'image de gauche et dans celle de droite. Faire tous les appariements possibles entre les positions à gauche et à droite, et calculer la disparité associée à chaque appariement. Afficher comme un nuage de points 3D, les points obtenus $(u_G, v_G, \text{disparité})$. Que peut-on dire sur le résultat obtenu ?
4. **Sélection** : Certains des appariements précédents doivent être éliminés. Ainsi, il n'est pas possible d'avoir une disparité négative car la scène est visible seulement devant les caméras. Il est aussi possible de mettre une borne sur la disparité maximale, en regardant l'objet le plus proche de la scène qui a la disparité maximale. Comparer les résultats en ajoutant ces différents critères de sélection.
5. **Carte de disparité** : Collecter les appariements dans une image vide de même taille que 01G pour construire la carte de disparité.

7 Détection des obstacles

Les résultats précédents permettent la construction de l'histogramme v -disparité. Cet histogramme est 2D avec pour abscisse v_G et la disparité. Il contient en ordonnée le nombre d'appariements pour chaque couple de valeurs $(v_G, \text{disparité})$. Son contenu est principalement composé du profil vertical de la route et des

profils verticaux des obstacles extraits. Ce contenu sera décomposé afin de faire la détection des obstacles.

1. **v-disparité** : À partir de l'ensemble de points $(u_G, v_G, \text{disparité})$, construire par accumulation l'histogramme 2D du nombre d'appariements pour chaque couple de valeurs $(v_G, \text{disparité})$. Afficher l'histogramme 2D comme une image à côté de l'image des contours à gauche. Que remarquez vous ?
2. **Profil de la route** : Faire la régression robuste d'une ligne sur le nuage de points $(v_G, \text{disparité})$ ayant une valeur supérieure à un seuil s_{vd} à choisir de façon adéquate. La fonction *robustfit* de la *statistic toolbox* sera utilisée. Commentez l'effet d'un changement de la valeur de s_{vd} .
3. **Obstacles** : Construire l'histogramme en disparité du nombre de pixels dans l'image v-disparité qui ont une valeur supérieure à s_{vd} mais qui ne sont pas trop proches du profil de la route obtenu précédemment. Le seuillage de cet histogramme avec un seuil de valeur $s_{obstacle}$ donne la position des obstacles selon la disparité. Comment utilisez la disparité associée à chaque détection pour localiser l'obstacle sur le plan de la route ? Afficher sur l'image gauche la ligne jusqu'à laquelle il n'y a pas d'obstacle.
4. Appliquer les traitements précédents sur la deuxième paire d'images nommées "02G.png" et "02D.png". Quelles modifications sont nécessaires pour détecter correctement la route ? Proposer des idées pour détecter correctement le véhicule ?

Références

- [1] T. VEIT, J.-P. TAREL, P. NICOLLE et P. CHARBONNIER : Evaluation of road marking feature extraction. *Dans les actes de 11th IEEE Conference on Intelligent Transportation Systems (ITSC'08)*, pages 174–181, Beijing, Chine, octobre 2008.
- [2] R. LABAYRADE, D. AUBERT et J.-P. TAREL : Real time obstacle detection in stereo vision on non-flat road geometry through v-disparity representation. *Dans les actes de IEEE Intelligent Vehicle Symposium (IV'2002)*, volume 2, pages 646–651, Versailles, France, juin 2002.