

Flot optique

Compte rendu TP2 Vision Robotique



WU Zijian
(zijian-wu@outlook.com)

Répertoire

1	Introduction	2
2	Codage.....	2
2.1	Moyenne et écart type	2
2.2	Sigma-delta.....	2
2.3	Flot optique.....	3
3	Conclusion.....	5
4	Référence	5

1 Introduction

La méthode du flux optique est une méthode importante pour l'analyse d'images en mouvement. Le flux optique est la vitesse de déplacement du mode dans des images variant dans le temps. Le flux exprime le changement dans l'image et, puisqu'il contient des informations sur le mouvement de la cible, l'observateur peut l'utiliser pour déterminer le mouvement de la cible. On considère une séquence de 600 images prise par une caméra installée au-dessus d'une autoroute. L'objectif de ce TP est de détecter les voitures en mouvement.

Dans cet article, je présenterai d'abord deux méthodes d'extraction d'objets en mouvement : la moyenne, l'écart type et le sigma delta. Après avoir comparé les deux méthodes, je vais utiliser la méthode mentionnée dans le document de référence pour générer un flux optique.

2 Codage

2.1 Moyenne et écart type

J'ai d'abord essayé la méthode en utilisant la moyenne et la variance. J'ai d'abord lu toutes les images à tour de rôle et les ai stockées dans une variable appelée "recu". Ensuite, j'ai calculé la moyenne et la variance de toutes les cinq images,



Figure 1 Image Originale



Figure 2 Image de Moyenne

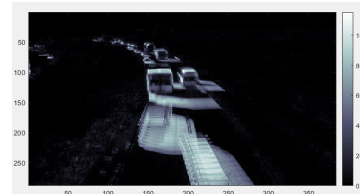


Figure 3 Image de l'Écart type

J'ai trouvé que cette méthode permet de bien distinguer les voitures, mais je ne peux pas voir la forme de la voiture, car le résultat est basé sur cinq images consécutives, et la forme de la voiture est également allongée. En raison de la complexité informatique de ce programme, je produis une image au format gif environ 50 fois plus vite. [Cliquez ici pour voir .gif.](#)

2.2 Sigma-delta

Sujet a mentionné que la méthode sigma-delta était meilleure que la méthode de la moyenne et l'écart type. J'ai suivi l'algorithme 1 mentionné dans la référence pour calculer M, O, V respectivement. La valeur finale estimée de E est le résultat final. Ici, de nombreux paramètres doivent être ajustés par eux-mêmes, et il n'y a pas de bonne référence dans le document. J'ai essayé de nombreuses combinaisons différentes et j'ai finalement trouvé que les résultats étaient meilleurs lorsque $V_{max} = 60$ et $V_{min} = 40$. J'ai comparé les résultats de différentes itérations comme indiqué sur la figure 4,5,6.

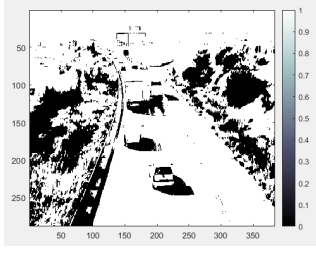


Figure 4 10 itérations

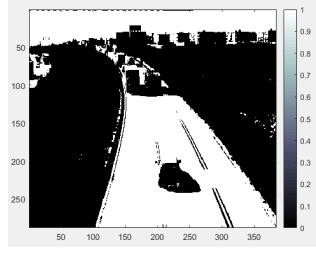


Figure 5 100 itérations

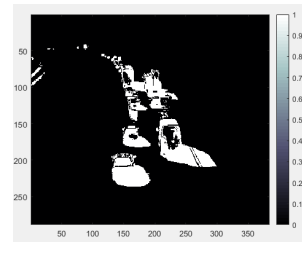


Figure 6 600 itérations

La figure montre que plus il y a d'itérations, meilleur est le résultat. Lorsque le nombre d'itérations est de 10, nous ne pouvons pas distinguer clairement les voitures des arbres en bordure de route. Lorsque le nombre d'itérations est de 100, la route et la voiture peuvent être clairement distinguées. Lorsque le nombre d'itérations est de 600 (presque toutes les images sont utilisées), seules les parties de la voiture sont blanches et les autres sont noires. J'ai donc utilisé 600 itérations pour calculer l'énergie et afficher les résultats de l'itération. J'ai constaté que si je n'ajoute pas l'intervalle, le programme ne générera pas toutes les images, mais le résultat de la dernière image une fois toutes les images traitées. Comme la quantité de calcul n'est pas grande ici, j'ai ajouté une pause (0.01) pour contrôler l'intervalle entre deux images. [La sortie du programme est ici.](#)

2.3 Flot optique

L'exercice 2 consiste à produire un flux optique. L'algorithme spécifique est mentionné dans la section 7 de la référence.

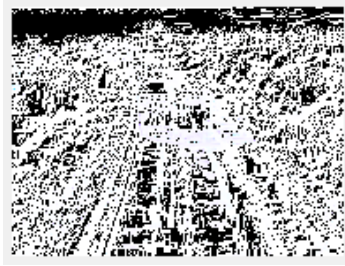
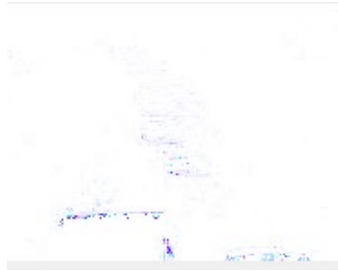
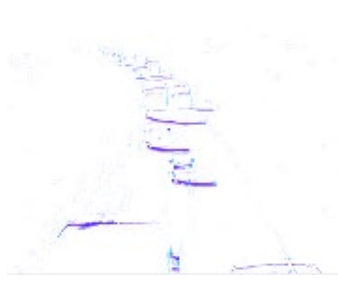
$$\begin{aligned}
 E_x &\approx \frac{1}{4} \{ E_{i,j+1,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i+1,j,k} \\
 &\quad + E_{i,j+1,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i+1,j,k+1} \}, \\
 E_y &\approx \frac{1}{4} \{ E_{i+1,j,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i,j+1,k} \\
 &\quad + E_{i+1,j,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i,j+1,k+1} \}, \\
 E_t &\approx \frac{1}{4} \{ E_{i,j,k+1} - E_{i,j,k} + E_{i+1,j,k+1} - E_{i+1,j,k} \\
 &\quad + E_{i,j+1,k+1} - E_{i,j+1,k} + E_{i+1,j+1,k+1} - E_{i+1,j+1,k} \}, \\
 \bar{u}_{i,j,k} &= \frac{1}{6} \{ u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k} \} \\
 &\quad + \frac{1}{12} \{ u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j-1,k} + u_{i+1,j+1,k} \}, \\
 \bar{v}_{i,j,k} &= \frac{1}{6} \{ v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k} \} \\
 &\quad + \frac{1}{12} \{ v_{i-1,j-1,k} + v_{i-1,j+1,k} + v_{i+1,j-1,k} + v_{i+1,j+1,k} \}.
 \end{aligned}$$

La formule ci-dessus peut être utilisée pour calculer la moyenne de E_x , E_y , E_t et u , v . Le principe est le suivant : un cube tridimensionnel est obtenu en huit mesures, à savoir la longueur, la largeur et la durée de l'image (c'est-à-dire différentes photos). La moyenne de premier ordre des trois directions est utilisée dans la formule au lieu des dérivées partielles des trois directions. Afin d'estimer la vitesse de flow, nous devons également approximer u et v pour calculer l'opérateur de Laplace. Une méthode d'approximation simple mentionnée dans la formule est mentionnée ci-dessus.

Afin de minimiser l'erreur, l'article mentionne l'utilisation de la formule suivante pour itérer. L'approximation précédente de Laplacien peut être utilisée pour obtenir une minimisation en trouvant une valeur appropriée de la vitesse du flux lumineux (u , v).

$$\begin{aligned}
 (\alpha^2 + E_x^2 + E_y^2)(u - \bar{u}) &= -E_x[E_x\bar{u} + E_y\bar{v} + E_t], \\
 (\alpha^2 + E_x^2 + E_y^2)(v - \bar{v}) &= -E_y[E_x\bar{u} + E_y\bar{v} + E_t].
 \end{aligned}$$

La mesure de la luminosité de l'image sera affectée par les erreurs de quantification et le bruit. Il est donc nécessaire de multiplier le coefficient α avant le laplacien. Ensuite, discutons de l'effet de la taille d'un sur le résultat.

Figure 7 $a=0$ Figure 8 $a=1$ Figure 9 $a=10$ Figure 10 $a=100$ Figure 11 $a=1000$ Figure 12 $a=100000$

J'ai sélectionné des valeurs entre 0 et 100 000 pour les tests et les résultats sont présentés dans la figure ci-dessus. J'ai trouvé qu'à $a = 0$, le bruit dépassait clairement le signal utile et nous pouvions à peine discerner l'existence d'un flux optique. Dans le cas où $a > 0$, le bruit peut évidemment être éliminé, et le flux optique est clairement vu en bleu. Dans l'intervalle de $a = 10$ à 100, les résultats ne changent pas beaucoup. Finalement, j'ai choisi $a = 100$ comme résultat final.

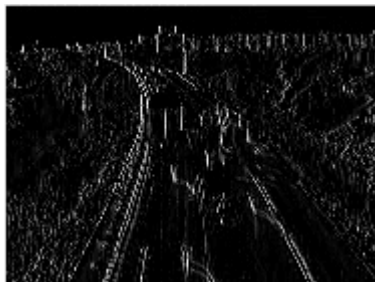
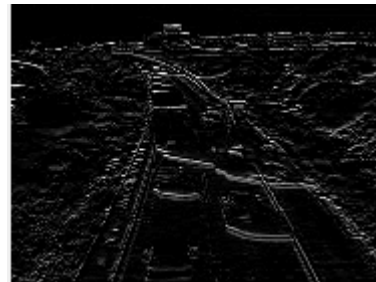
Figure 13 Estimation de E_x Figure 14 Estimation de E_x Figure 15 Estimation de E_t 

Figure 16 Flux optique

En comparaison, je pense que mon programme peut extraire correctement le flux optique et peut distinguer la direction du mouvement. Afin de présenter la sortie de mon programme, j'ai enregistré un fichier au format gif. [Cliquez ici pour voir.](#)

J'ai aussi essayé de tester le mouvement horizontal. J'ai téléchargé une autre vidéo sur le web. Dans mon programme, deux voitures se déplaçant horizontalement dans la vidéo ont été détectées, et les autres objets fixes sur la route ne sont pas apparus sur la photo. Comme indiqué dans la figure ci-dessous.

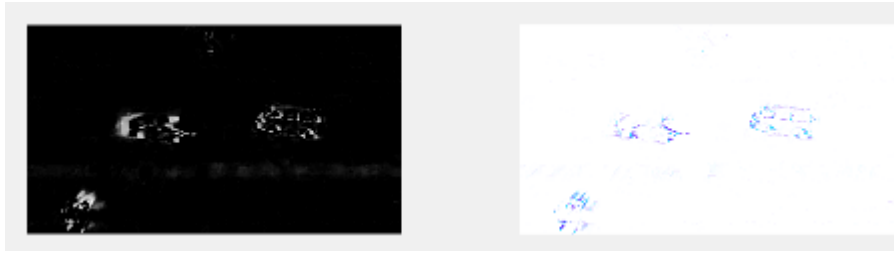


Figure 17 déplacement horizontal

3 Conclusion

Le but de TP cette fois-ci est de détecter des objets en mouvement et d'utiliser différentes couleurs pour distinguer les mouvements dans différentes directions. Pour le premier exercice j'ai comparé les avantages et les inconvénients des deux méthodes et j'ai réussi à générer le flux vidéo en fonction de la méthode optimale. Pour le deuxième exercice, je me réfère à la méthode décrite dans la référence et utilise les deux fonctions fournies par l'enseignant pour générer le flux optique dans des directions différentes.

La difficulté de TP cette fois est beaucoup plus grande que la dernière fois, car de nombreux paramètres sont inconnus et je dois l'essayer moi-même. La difficulté de TP cette fois consiste à trouver la bonne combinaison de paramètres, y compris le nombre d'images traitées et divers paramètres utilisés dans l'algorithme, tels que V_{max} , V_{min} , a , etc. Pour la méthode de moyen-écart type, si le nombre de traitements simultanés est trop grand, toutes les images seront moyennées et la voiture ne pourra pas être distinguée. De plus, le choix de V_{max} et V_{min} nécessite également plusieurs tentatives. Je pense que tant que vous comprenez l'objectif et les étapes de TP, la programmation n'est pas difficile.

Quand j'ai vu ce sujet pour la première fois, mon idée était d'utiliser la méthode de streaming vidéo d'OpenCV pour gérer le flux optique. Mais après avoir complété les exigences de sujet étape par étape et surtout constaté que les étapes clés mentionnées dans le sujet ont fourni le code, j'ai trouvé qu'il n'était pas difficile de traiter un flux optique en utilisant MATLAB.

4 Référence

- [1] Berthold K.P. Horn and Brian G. Schunck, Determining Optical Flow, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.
- [2] L. Lacassagne & A. Manzanera, MOTION DETECTION: FAST AND ROBUST ALGORITHMS FOR EMBEDDED SYSTEMS, IEF/AXIS – Digiteo Labs – Université Paris Sud & UEI – ENSTA – Paris Tech