

Travaux pratiques

Séance 2 : flot optique

Traitement d'images et vision - M2 E3A SAM - UEVE

Exercice 1 : Flot optique

1. Etudier attentivement l'article : "Determining optical flow" de Horn et Schunk.
2. Expliquer comment les gradients spatiaux sont calculés.
3. Expliquer comment le gradient temporel est calculé.
4. Quelle approximation du Laplacien de u et v est utilisée ?
5. Expliquer le principe de la méthode de calcul du flot optique ainsi que ses limitations.
6. Expliquer comment les différents paramètres de la méthode peuvent être choisis.
7. Programmer sous Matlab l'algorithme.
8. Tester votre algorithme sur la séquence de votre choix :
 - D'abord sur la séquence Road (sans vérité terrain).
 - Ensuite sur l'une des séquences Middlebury fournies (répertoire **SEQUENCES_GT**).
9. Afficher le flot optique (voir annexe) : 1/comme une image en niveau de gris (une image par composante), 2/en dessinant des vecteurs à l'aide de la fonction **quiver_uv** (voir annexe) et 3/en couleur (fonction **computeColor** dans le répertoire **CODES**). Afin de visualiser la vérité terrain (répertoire **FLOTS_GT**), vous disposez, dans le répertoire **CODES**, d'un certain nombre de fonctions permettant la manipulation du format .flo.
10. Afin d'évaluer la précision du flot optique estimé sans disposer de vérité de terrain, il est possible de recalculer l'image $t + dt$ en utilisant les vecteurs estimés et comparer la nouvelle image ainsi obtenue avec l'image t . Ecrire une fonction calculant l'erreur résiduelle ainsi obtenue. Où se situent les plus fortes erreurs ? Conclure. Erreur résiduelle en chaque point :

$$\varepsilon_{x,y} = |I(x, y, t) - I(x + dx, y + dy, t + dt)|$$

11. Proposer une autre manière de calculer les gradients spatio-temporel. Comparez les résultats obtenus avec la méthode originale.
12. Les discontinuités du flot optique devraient correspondre à celles de l'image et des bordures séparant les différents objets. Proposer un moyen de déterminer les frontières entre mouvements différents et comparer avec les réelles frontières des objets (résultat d'une détection de contour).
13. Lorsque nous disposons d'une vérité terrain, l'évaluation consiste à comparer les flots optiques obtenus avec ceux de la vérité terrain en se basant sur les indicateurs AEPE et AAE. Mettre en place ce type d'évaluation sur les images de la base de données Middlebury (<https://vision.middlebury.edu/flow/>).
 - Représentation colorée de Middlebury. Teinte : orientation, Saturation : norme.
 - $\epsilon_{AE} = \arccos \left(\frac{\mathbf{w}^T \tilde{\mathbf{w}}}{\|\mathbf{w}\| \|\tilde{\mathbf{w}}\|} \right)$ avec \mathbf{w} : flot de vérité terrain et $\tilde{\mathbf{w}}$: flot estimé.
 - $\epsilon_{EPE} = \|\mathbf{w} - \tilde{\mathbf{w}}\|$
 - Ce sont ensuite les moyennes de ces erreurs qui sont calculées sur l'ensemble de l'image pour obtenir AEPE (Average End Point Error) et AAE (Average Angular Error).

14. Dans la méthode de Horn et Schunck originale, le processus itératif part d'un estimé des vecteurs vitesses égal à zéro. Modifier l'algorithme pour que l'estimé soit égal à la solution trouvée à l'instant précédent au même point.

Exercice 2 : Post-filtrage temporel et spatial du flot optique

1. Proposer un algorithme pour filtrer les vecteurs erronés en utilisant un filtrage spatial.
2. Proposer un algorithme pour filtrer les vecteurs erronés en utilisant un filtrage temporel.
3. Comparer les deux algorithmes.

ANNEXE : REMARQUES SUR L’AFFICHAGE DU FLOT OPTIQUE

- Pour visualiser les résultats sous forme de vecteurs, vous pourrez utiliser la fonction matlab "quiver" comme indiqué ci-dessous.

```
1 function quiver_uv(u,v)
2
3 % Resize u and v so we can actually see something in the quiver plot
4 scalefactor = 50/size(u,2);
5 u_=scalefactor*imresize(u,scalefactor,'bilinear');
6 v_=scalefactor*imresize(v,scalefactor,'bilinear');
7
8 % Run quiver taking into account matlab coordinate system quirks
9 % and scaling the magnitude of (u,v) by 2 so it is more visible.
10 quiver(u_(end:-1:1,:),-v_(end:-1:1,:),2);
11 axis('tight');
```

- Il est possible aussi de visualiser la composante u et la composante v séparément en les considérant chacune comme des images en niveaux de gris moyennant quelques approximations et arrondis.
- Enfin, une représentation courante est d'afficher le résultat en couleur en tenant compte du module et de l'orientation du vecteur estimé en chaque point. Utiliser la fonction **computeColor** pour cela.

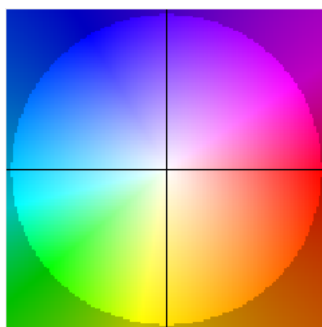


FIGURE 1 – Code couleur utilisé pour chaque module et chaque orientation des vecteurs vitesse estimés (en coordonnées polaire) pour une visualisation sur une seule image couleur.