TIPE

Notes pour l'oral

Intro *1 min*

— Vision informatique = **gain** de temps + efficacité

— **Domaine médical** : compteurs de cellules automatisés ⇒ rassembler plus rapidement des informations

— **QUESTION** : Comment peut-on ainsi reconnaitre un objet?

— Objet qcq = **trop général**

— Système **HawkEye** → rendu 3D de la trajectoire d'une balle de tennis ⇒ arbitrage vidéo

— D'où choix de la balle pour la **simplicité** de sa forme

But du TIPE : Elaborer une méthode pour effectuer une telle détection

J'ai procédé suivant **4 axes** pour **établir la méthode**, et implémenté tous les algorithmes en CAML :

1. Suivi de la balle
2. Détection par sa forme
3. Détection par sa couleur
4. Localisation dans l'espace

1^e Partie : Suivi de la balle *1 min 30 s*

— Plus simple à traiter avant une détection réelle

— Inspiration d'une **publication sur HawkEye**

— Suivi de trajectoire en détectant la balle **image par image** sur une vidéo prise depuis une **caméra fixe**

— J'ai par ailleurs élaboré **2 filtres** améliorant nettement le temps de détection

A) Filtre prédictif

— Supposons que $\vec{v}$ varie peu entre 2 images ⇒ la balle ne peut parcourir qu'une distance $d = v\,\partial t$ au maximum

— Elle se trouvera donc probablement dans une zone bcp plus petite ⇒ **amélioration du temps de calcul**

Montrer Doc (ball_pos_forsee)

B) Filtre passe-haut

— Seuls les points de l'image qui changent p/r à l'image précédente (ceux dont la pulsation est la plus grande dans la transformée de Fourier) sont susceptibles d'appartenir à la balle

— **Réduction considérable** du nombre de points à analyser

Montrer Doc (ball_reference, ball_test, ball_testspeedfilter)

C) Algorithme

— **2 1^e images** servent à initialiser le filtre prédictif : besoin d'une **détection sur la totalité de l'image**

— Ensuite, **pour chaque image** :

- Charger la zone prédite
- Appliquer le filtre passe-haut
- Détection sur l'image résultante

Montrer Doc (schéma algorithme)

2^e Partie : Recherche par la forme *2 min 30 s*

— A priori, balle sphérique $\Rightarrow$ **forme circulaire** sur l'image

— J'ai donc pensé à **détecter les contours de l'image** et à trouver les **contours circulaires**

— Cet algorithme est **inspiré** d'une version présentée dans [3]

A) Détection de contours

— Contours = zone où les **propriétés de l'image changent brutalement**

— Dans les images converties en noir et blanc, j'utilise l'**intensité lumineuse**
Sur transparent : Formule P contour $\Leftrightarrow \|\overrightarrow{\text{grad}}P\| \geqslant g_0$

— La normale à un contour $\leftrightarrow$ **direction du gradient**
Montrer Doc (test contour)

B) Formes circulaires

— Contour circulaire : normales dirigées vers le **centre du cercle**
Montrer doc (circ_edge)

— Tracé en chaque point d'un contour de l'image des droites **dirigées par le vecteur gradient**

— Le centre d'un cercle = le nombre de droites passant par ce point est max local
Montrer exemple

C) Performances et limites

— Exécution en quelques secondes sur une image de 800×600

— Un seul paramètre : valeur limite du gradient

— Problèmes :

- Taille de la balle
- "Bruit" exterieur
- Autres formes circulaires de l'image
- Surtout, **temps de pose de la caméra** : déformation de la balle

Montrer exemples

— C'est pourquoi j'ai essayé de **construire un algorithme** pour **détecter la balle par sa couleur**

3^e Partie : Recherche par la couleur *3 min*

— La balle est d'une couleur jaune particulière : se détache bien sur l'arrière-plan

— Mon idée : détecter les zones de l'image d'une **couleur uniforme** et d'une **taille** correspondant à une balle

A) Filtrage des couleurs

— **Elimination des points superflus** pour réduire le temps d'exécution

— J'ai remarqué que les couleurs de la balle se trouvaient dans une zone plane donnée dans l'espace (r, g, b), paramétrisée à l'aide de MAPLE

MONTRER DOC (graphe plan)

— Au lieu de choisir d'éliminer les points à trop grande distance d'une **couleur moyenne** (trop peu précis)

— Couleurs situées à trop grande distance **de cette zone**

— Renvoi en sortie du filtre des couleurs valides, projetées sur le plan jaune, ou d'une couleur tampon dans le cas d'un point non valide
MONTRER DOC (tests)

B) Composantes connexes

— Rechercher une zone particulière $\Leftarrow$ **regroupement en composantes connexes** d'une couleur uniforme
SUR TRANSPARENT : ALGORITHME

— Initialisation : **graine** (*seed*)

— Puis extension aux **points adjacents** : ajout progressif de points

— Rajout de la condition d'**uniformisation de la couleur** : un point n'est ajouté que si sa couleur est assez proche de la couleur de la composante

— Graines de départ = grille
MONTRER DOC (segmentation)

C) Elimination

— Composantes **trop petites ou trop grandes** éliminées

— Je considère que la balle se trouve à la position de la composante connexe dont la couleur moyenne est **la plus proche d'une couleur donnée**, correspondant à la couleur trouvée dans la plupart des cas

MONTRER DOC (élimination + valeur γ)

D) Performances et limites

— Complexité **quadratique** $\mathcal{O}(h\,w)$

— 8 paramètres : deux pour la **taille de la grille**, trois pour la **taille de la balle** attendue, un pour la **distance maximale au plan jaune**, un pour la **couleur moyenne de la balle** et un pour l'**uniformisation** des composantes

— Quasi-totalité des balles détectées

— Temps d'exécution similaire à l'algorithem précédent

MAIS

— Intervention plus importante de l'utilisateur (plus de paramètres)

— Détecte n'importe quelle zone jaune

En combinant avec les techniques de suivi, ce dernier point est en majeure partie résolu, la balle étant souvent la seule zone jaune en mouvement

CONCLUSION

1 min

— Présentation du résultat : on voit que la trajectoire est mal détectée quand la balle est loin de la caméra

— Améliorations :

- Vision Stereo
  ⇒ Une caméra ↔ 2 paramètres : il faut 2 caméras pour avoir 3 paramètres d'espace et localiser la balle en 3D
  ⇒ A l'aide de la position de la balle sur l'image de chaque caméra, 2 angles directeurs (plus simples car la position en pixels/pouces/cm est dépendante du matériel)
  ⇒ Avec ces deux angles, intersection de plans
- Meilleure détection des petites balles
  ⇒ Peut-être une extension du filtre passe-haut évoqué plus haut?
- Traitement statistique
  ⇒ Si on ne détecte pas la balle mais autre chose, la distribution des positions au cours du temps est aléatoire