

Détection et reconnaissance de panneaux

Bureau d'étude Twizy ISN 2A

S8 2021-2022

ENSEM

Objectifs du bureau d'étude

- Appliquer la pédagogie par projet
- Formation par l'expérience et le développement de l'autonomie
- Concevoir, développer et expérimenter une solution numérique
- Application par la pratique des cours de S5, S6 et S7
- Apprendre par la pratique la gestion d'un projet
- Savoir faire et compétences acquises
 - Connaissance des techniques de la vision par ordinateur (computer vision) et le traitement d'images
 - Savoir développer une solution basée sur la bibliothèque OpenCV

Contexte du sujet

- Les véhicules et les robots autonomes ont besoin de percevoir leurs environnements
 - Autres véhicules, objets, piétons, signalisation, ...
 - Percevoir – Analyser - Agir
- La détection et la reconnaissance de panneaux de signalisation routière est indispensable pour le déplacement des véhicules sur les routes
- Plusieurs capteurs embarqués d'acquisition de données
 - Caméras embarquées
 - Télémètre laser 3D : LIDAR
- Analyse des images capturées pour détecter les formes et les objets dans l'environnement du véhicule

Sujet du bureau d'étude

- Détection des objets et des formes dans des images 2D
 - Détection et reconnaissance de panneaux de signalisation routière



Plateforme expérimentale

- Véhicule électrique Twizy
- Caméras GoPro HERO 4 (12MP)
- Panneaux de signalisation



Travail demandé

- Etude des algorithmes de détection et de reconnaissance d'objets
 - avec Matlab
 - avec la bibliothèque OpenCV (programmation Java)
- Acquisition des images d'une signalisation routière avec la plateforme expérimentale
- Implantation en Java d'une solution de détection et de reconnaissance des panneaux avec OpenCV
- Démonstration de la solution
 - Acquisition des images depuis un GoPro via le Wifi
 - Traitement des images par l'application Java pour détecter et reconnaître les panneaux

Exemple de résultat attendu

Java - DetectionSignalisation/src/ImageProcessing.java - Eclipse - /Users/AbdelkaderLahmadi/Documents/workspace

Package Explorer

- 3nodes
- ActionBarSh
- android-map
- appcompat
- bidon
- BlagueDuJou
- BlagueJour
- Chrome_Hac
- convertisseu
- DetectionSig
- src
 - (default) Seuils couleur
 - Ima Min max
 - Seu 0 6
 - Tabl 160 179
 - UI.j
 - Utili
- JRE Syste
- opencv-2
- .settings
- .classpath
- .DS_Store
- .project

- flowoidExporter
- google-play-services_lib

Charger Image

Convertir couleurs

Extraire couleur

Détecter contours

Extraire panneaux

Ajouter Supprimer

Valeurs par défaut

Match

Match

Masque

Contours

Conversion

tring sign) {

Writable Smart Insert 294 : 9

Mise en œuvre du module

- Définition du cahier de charges
- Spécifications « Traitement d'images »
 - Méthodes de détection et extraction d'un panneau
 - Méthodes de reconnaissance de type des panneaux
- Spécifications « développement logiciel »
 - Fonction d'acquisition des images
 - Fonction de détection
 - Fonction de reconnaissance
- Développement et tests unitaires
- Intégration
- Tests, validation et évaluation de la solution
- Démonstration

Gestion du module

- N équipes de deux binômes (4-5 étudiants)
 - Traitement d'images : binôme
 - Développement logiciel : binôme
 - Intégration, tests, validation et documentation : toute l'équipe
- Rôle des enseignants
 - Soutien scientifique et technologique
 - Veille au plan de développement de la solution logicielle

Partie traitement d'images

- Introduction au traitement d'images
- Filtrage, débruitage
- Segmentation et seuillage (couleurs)
- Détection de contours (formes géométriques)
- Extraction de caractéristiques pour la reconnaissance de formes
- Introduction aux algorithmes de template Matching (SIFT, ORB)

Expérimentation et tests sous Matlab (4h) puis développement sous OpenCV

Partie programmation OpenCV sous java

- Lecture d'un fichier d'une image couleur
- Conversion des images vers différents espaces de couleurs (RGB, BGR, HSV, HSI)
- Extraction d'un objet depuis une image
 - Technique basée sur la couleur
 - Technique basée sur la forme
- Détection de contours : filtre de Canny
- Détection d'une forme géométrique : cercle, rectangle, carré, polygone, triangle
- Reconnaissance d'un panneau depuis la forme détectée
- Programmation SWING : interface graphique de l'application

Pratiques développement logiciel

- Diagramme UML de classes
- Tests unitaires
- Documentation détaillée du code
- Guide utilisateur de l'application
- Scénarios de démonstration de la solution
- Mise en place d'un dépôt git de code source
 - Créer votre projet sous le gitlab de l'UL pour déposer votre code et travailler en équipe : <https://gitlab.univ-lorraine.fr/>
 - Nom du projet pour chaque groupe : *nom_groupe_twizy_2022*