PROJET CODEVSI N°41

GÉOPOSITIONNEMENT DU TGV IRIS 320

HUBERT BAPTISTE
KHELLA ERIC
GATARD ANTOINE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

PARTIE 1
CONCEPTION

Formalisation du problème

Comment géolocaliser ce train avec une précision d'une milliseconde ?
3 types de fichiers vidéo disponibles :

- vidéo motrice avant
- vidéo motrice arrière
- vidéo vision périphérique

1 fichier XML

Solution globale proposée

Code qui analyse les données visuellement présentes sur les vidéos arrières et avant et qui les met en relation avec la référence temporelle que constitue la LED afin d'obtenir la précision souhaitée.

PARTIE 1
CONCEPTION

Les différents cas nominaux d'utilisation

cas n°1:

- L'utilisateur souhaite analyser chaque instant d'un ou plusieurs fichiers vidéo pour obtenir toutes les données associées

cas n°2:

- L'utilisateur veut à partir d'une date précise à la milliseconde obtenir les frames des vidéos associées à la date renseignée

PARTIE 1
CONCEPTION

Fonctionnalités principales nécessaires aux cas nominaux :

- générer un fichier (CSV, liste, dictionnaire) comportant toutes les informations nécessaires à la géolocalisation
- extraction des informations des fichiers vidéos par reconnaissance de caractère optique
- détection des états allumés et éteints de la LED
- extraction des informations du fichier XML pour accéder aux vidéos souhaitées

PARTIE 1
CONCEPTION

Contraintes à respecter:

- Temps d'analyse vidéo inférieur à la durée de la vidéo
- Précision à la milliseconde près dans la datation des mesures réalisées

Solutions envisagées :

- Mémoïsation, multiproccessing, vectorisation des opérations
- Interpolation, régression linéaire

PARTIE 2
RÉALISATION

Premiers codes

Objectifs :

- Poser les premières idées
- “**Débroussailler**” le problème
- Commencer le traitement vidéo (ouverture du fichier, OCR)

Problèmes :

- Codes **mal organisés** / implémentés (boucles imbriquées, valeurs sans noms dans le code)
- **Pas de liens** entre les codes (car chacun code à part)

Progression/amélioration des codes

- Disparition des variables globales
- Fonctions **plus efficaces** (pas de boucles for imbriquées = gain de temps de calcul)
- Modules importés dans un fichier à part, paramètres regroupés dans un fichier à part…

PARTIE 2
RÉALISATION

Structuration du code

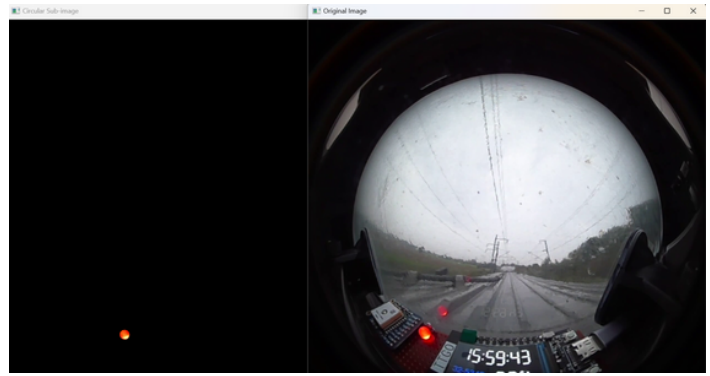
- **Tri** entre les fonctions utiles et celles qui sont obsolètes (car remplacées)
- On **renomme** les fonctions, les variables… pour **clarifier** leur utilité auprès des autres membres du groupe

Difficultés rencontrées 

De nombreuses difficultés ont été rencontrées :

- Les triplets RGB détectés sur la LED n'affichaient pas majoritairement du rouge

=> Solution ; afficher la zone analysée sur la LED

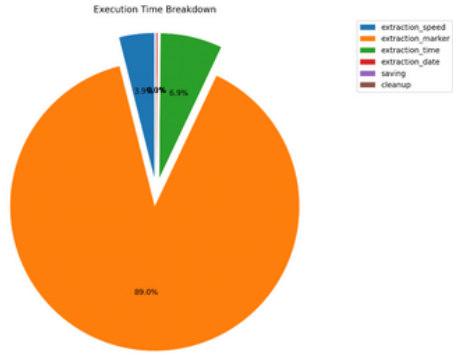


PARTIE 2
RÉALISATION

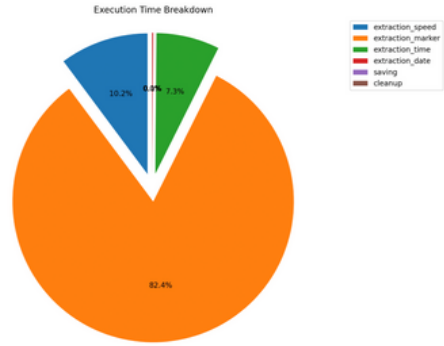
Difficultés rencontrées

- Temps de calcul global beaucoup trop long

=> Solution ; afficher le temps de calcul de chaque partie du code pour savoir où mettre les efforts



Pour M1



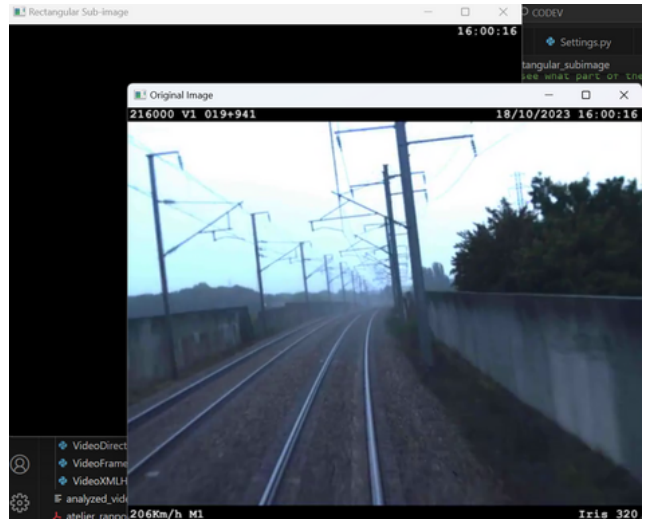
Pour M2

PARTIE 2
RÉALISATION

Difficultés rencontrées

- Il manquait quelques données lors de l'analyse par Pytesseract

=> Solution ; afficher la zone analysée sur les données



- L'analyse par Pytesseract prenait beaucoup trop de temps

=> Solution ; ne procéder à l'analyse de la donnée que si celle-ci a changé d'une frame à l'autre

PARTIE 2
RÉALISATION

Lien entre les différents codes

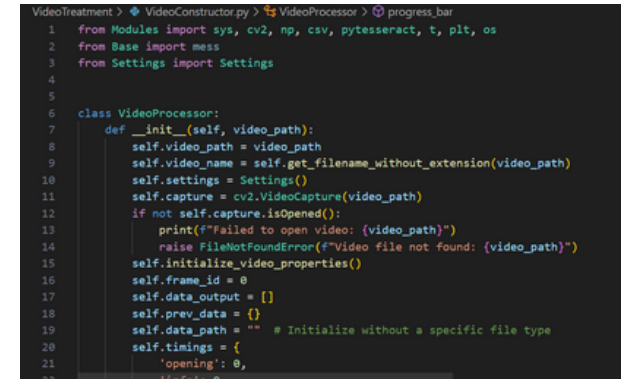
Il fallait encore **lier** les codes de notre projet:

- Détection de l'état de la LED
- Détection des données sur la vidéo
- Interpolation sur les 3 fichiers vidéo
- Production du fichier souhaité
- Retour du résultat à l'utilisateur

Codes orientés objet

- Remplacement des codes composés de fonctions mises à la suite en **codes orientés objet**

Objectif : Rendre **plus clair** la composition du code (produit final) et une **meilleure organisation** du code, de manière à rendre un produit **plus "professionnel"**.



PARTIE 3
VALIDATION

Solution concrète

-> Code qui permet à l'utilisateur de préciser une date et une heure précise à la milliseconde, et qui lui retourne les 2 frames correspondantes (des vidéos motrices M1 et M2, avant et arrière) à cet instant
Pas de fichier comme le xml où trouver les différents fichiers vidéo avec la LED mais en théorie, on pourrait récupérer une 3e frame.

Démonstration !

PARTIE 4
CONCLUSION

Limites / progressions à envisager sur le projet :

- Une meilleure communication dès le début du projet (retard au démarrage sur les compte rendus)
- Une meilleure répartition des tâches (parfois nous étions plusieurs sur un même code, ce qui n’est pas très productif)
- Plus d’anticipation concernant les échéances (prise de réunion en amont…)

PARTIE 4
CONCLUSION

Limites / progressions à envisager sur le code :

- Récupérer les distances entre la caméra arrière, la caméra avant, et la caméra périphérique (l'antenne de mesure), pour pouvoir coordonner les valeurs récupérées
- Faire une interface (même simpliste) pour que l'utilisateur utilise plus simplement le code
- Si on souhaite uniquement conserver le côté pratique, on pourrait supprimer les outils de débuggage et les outils qui nous ont servis pour préciser le traitement