

Extraction et analyse de données GoPro 360 max

Installation et préparation de l'environnement

Installer l'outil py-gpmf-parser pour extraire les données embarquées (accéléromètre, gyroscope, GPS...) contenues dans les vidéos .MP4 enregistrées par une GoPro Max 360.

1. Travailler dans un environnement Linux (Ubuntu conseillé) : soit via WSL (Windows Subsystem for Linux, plus simple), soit sur une machine Ubuntu native.
 2. Expliquer les gpmf de la goPro 360 max
 3. Chercher les caractéristiques de la caméra
 4. Créer un environnement virtuel : `python3 -m venv venv`, source `venv/bin/activate`
 5. Cloner et installer le dépôt : <https://github.com/urbste/py-gpmf-parser/tree/main>. suivre la méthode d'installation décrite dans le git.
 6. Copier la vidéo .MP4 dans un dossier du HOME Linux
- Essayer le code python `test.py` (joint).
 - Vous auriez un fichier json qui contient les données de la télémétrie.

Analyser les données json

Comprendre la structure des données issues de la vidéo GoPro. Visualiser et synchroniser les différentes sources (IMU, GPS, etc.).

1. copier le fichier json résultant dans un répertoire sur windows.
2. Je vous donne un fichier ipynb (Jupyter) ou vous pouvez le convertir en python et travailler avec anaconda pour traiter le fichier json résultant.
3. Inspecter la structure complète du fichier JSON, les dictionnaires et les clés.
4. Tracer toutes les données de télémétrie disponibles

```

libpostproc 57. 3.100 / 57. 3.100
[mov,mp4,m4a,3gp,3g2,mj2 @ 0x55b72a8cd080] All samples in data stream index: id [4:5] have zero duration, stream set to be
discarded by default. Override using AVStream->discard or -discard for ffmpeg command.
Input #0: mov,mp4,m4a,3gp,3g2,mj2, from 'GH010025.MP4':
  Metadata:
    major_brand      : mp41
    minor_version    : 538120216
    compatible_brands: mp41
    creation_time     : 2024-07-19T11:03:42.000000Z
    firmware         : H19.03.02.00.00
  Duration: 00:11:47.71, start: 0.000000, bitrate: 45277 kb/s
  Stream #0:0[0x1](eng): Video: h264 (High) (avc1 / 0x31637661), yuvj420p(pc, bt709, progressive), 1920x1080 [SAR 1:1 DAR 16:9], 45002 kb/s, 59.94 fps, 59.94 tbr, 60k tbn (default)
  Metadata:
    creation_time     : 2024-07-19T11:03:42.000000Z
    handler_name      : GoPro AVC
    vendor_id         : [0][0][0][0]
    encoder           : GoPro AVC encoder
    timecode          : 11:03:02:15
  Stream #0:1[0x2](eng): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 189 kb/s (default)
  Metadata:
    creation_time     : 2024-07-19T11:03:42.000000Z
    handler_name      : GoPro AAC
    vendor_id         : [0][0][0][0]
    timecode          : 11:03:02:15
  Stream #0:2[0x3](eng): Data: none (tmcd / 0x64636D74) (default)
  Metadata:
    creation_time     : 2024-07-19T11:03:42.000000Z
    handler_name      : GoPro TCD
    timecode          : 11:03:02:15
  Stream #0:3[0x4](eng): Data: bin_data (gpmd / 0x646D7067), 58 kb/s (default)
  Metadata:
    creation_time     : 2024-07-19T11:03:42.000000Z
    handler_name      : GoPro MET
  Stream #0:4[0x5](eng): Data: none (fdsc / 0x63736466), 13 kb/s (default)
  Metadata:
    creation_time     : 2024-07-19T11:03:42.000000Z
    handler_name      : GoPro SOS
  Unsupported codec with id 0 for input stream 2
  Unsupported codec with id 98314 for input stream 3
  Unsupported codec with id 0 for input stream 4

```

5. Essayer la commande « ffprobe GH010025.MP4 » (cela nécessite je pense d'installer ffmpeg) et regarder le résultat. Essayer de commander et documenter le résultat. Voici un exemple de sortie
6. Aller au site : (<https://goprotelemetryextractor.com/free/>) choisir free et essay d'extraire les données avec l'outil en ligne. Compare tes résultat
7. Convertir le fichier json en csv et faites un test sur Matlab.
8. Regarder les deux fichiers python dans le repertoire tests du git. Comprendre ce qu'il font et les tester, documenter.
9. À partir des données calculer le roulis, le lacet, et le tangage de la moto. Le calcul se fait par données brutes en utilisant les vitesses angulaires seules, les accélérations linéaires seules, ensuite en utilisant les filtres complémentaires et enfin si possible une fusion par filtre de Kalman. Je pense qu'il y a aussi les données de compass, ça permet de calculer l'angle du lacet.

Synchronisation des données

Problème : les capteurs n'ont pas le même taux d'échantillonnage (données à vérifier):

- Gyroscope : 200 Hz
 - GPS : 18 Hz
1. Même si les timestamps sont sur une échelle commune, l'alignement temporel doit être validé.
 2. Superposition sur vidéo : étudier <https://github.com/time4tea/gopro-dashboard-overlay>. Vous pouvez regarder à d'autres git aussi.

Analyse comparative entre deux conducteur sur le même circuit

Chaque scénario de conduite a été enregistré sur une GoPro MAX. En raison des limitations de taille de fichier sur la carte SD, chaque scénario est découpé automatiquement en deux fichiers vidéo de 4 Go maximum. Cela signifie qu'un même trajet est réparti sur deux vidéos consécutives par conducteur.

1. Comparer les comportements dynamiques (trajectoires, accélérations, roulis) de deux conducteurs ayant réalisé le même parcours.
2. Utiliser des bibliothèques Python (folium, geopandas, cartopy ou matplotlib) pour superposer les deux trajectoires sur carte satellite ou plan du circuit (Google Maps, OpenStreetMap, etc.).