



# TELLO TEMPO

## MAKE TELLO DRONE DANCE

### Students

MS I.A.

Louis-Gabriel Pouillot  
Mouhamadou Niane

### Tutor

M Alexandre Chapoutot

# Objectifs

- Reconnaître des gestes grâce à la caméra de Tello
- Piloter un drone avec ces gestes
- Extraire des features sonores
- Faire danser le drone suivant le rythme d'une musique

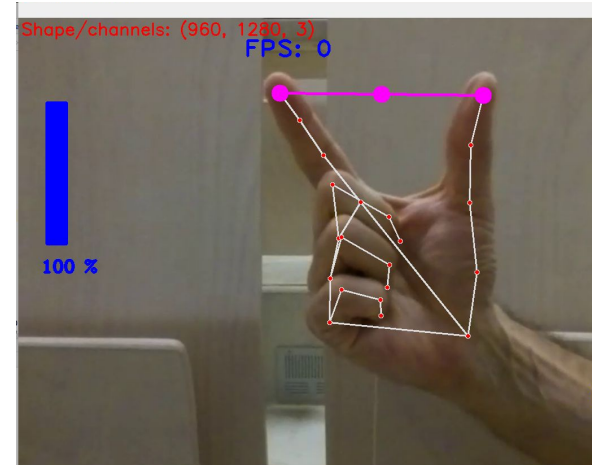
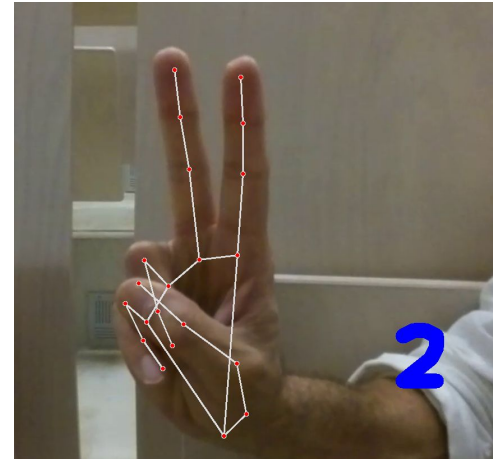
# Plan

- Hand Detection and Control
- Sound Control
- Dance Control
- Thread Control & Architecture
- Bilan

# Hand Detection and Control

# Tâches Principales

- Déclencher la musique (1 doigt levé)
- Arrêter la musique (2 doigts)
- Déclencher la danse/musique du drone et le beat (3 doigts)
- Arrêter la danse (4 doigts)
- Décollage (5 doigts)
- Ajuster le volume de la musique
  - distance entre pointes pouce/index
  - angle de la droite formée les extrémités inférieur à  $30^\circ$



# Utilisation de Mediapipe

- Bibliothèque open source
- Utilisation du machine learning pour déduire 21 points de repère 3D d'une main à partir d'une image
- Combinaison de deux modèles (détection de paumes et détection de points clés de la main)



# Palm Detection Model

- Permet de détecter l'emplacement initial des Mains
- Plus simple que la détection de mains ( estimation de boîtes englobantes d'objets rigides comme les paumes et les poings plus simple que la détection de mains aux doigts articulés)
- Modélisation de paumes à l'aide de boîtes de délimitation carrée
- Utilisation d'un auto-encodeur pour l'extraction de features caractéristiques
- Précision de 95.7%

# Hand Landmark Model

- Après la détection de la paume , notre modèle effectue une localisation précise des 21 coordonnées 3D par régression ( prédiction de coordonnées)
- Modèle entraîné sur 30.000 images du monde réel
- Liens avec nos tâches





# Sound control

# Sound Control - MIR & librosa

## MIR

- Music Information Retrieval
- Domaine de l'analyse du signal
- Beaucoup de recherche, en développement
- exemple de traitements: fingerprinting, genre recognition, transcription, recommandation, symbolic melodic similarity, source separation, instrument recognition, pitch tracking, beat tracking, ...

## Librosa

- bibliothèque d'extraction d'information musicale <https://librosa.org/>
- bien mise en valeur sur <https://musicinformationretrieval.com>



# Sound Control - Tello & BeatTracking (1)

- **Objectif :**

extraire des features d'intérêt transposables en mouvements

- **Solution :**

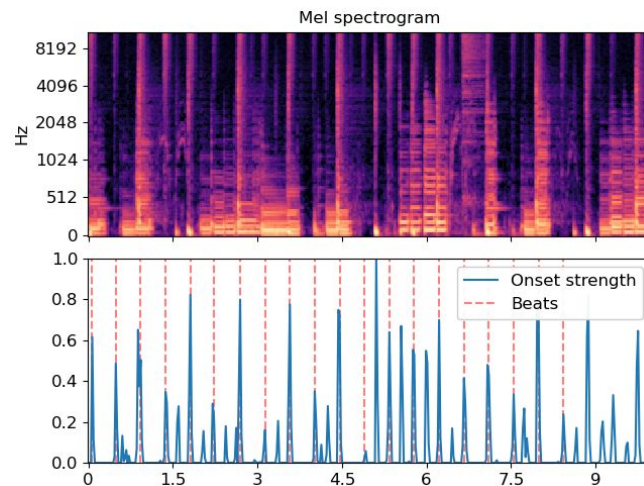
détection des beats audio (beat\_track)

- Measure onset strength
- Estimate tempo from onset correlation
- Pick peaks in onset strength approximately consistent with estimated tempo

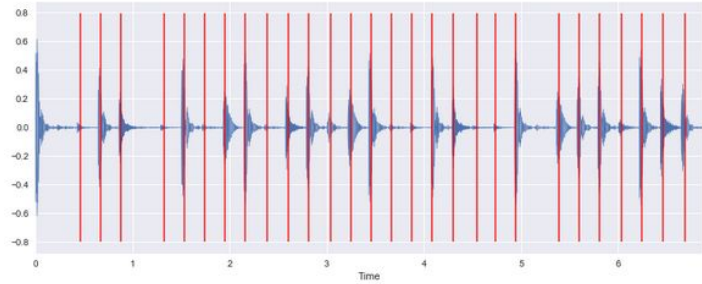
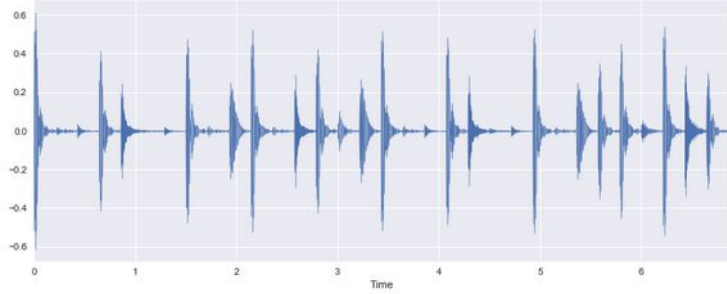
○

- **Problématiques :**

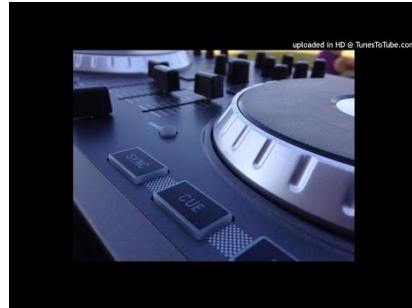
- Qu'est-ce qu'un beat ? Quel tempo est le bon ? Changements soudain de tempo ?



# Sound Control - Tello & BeatTracking (2)

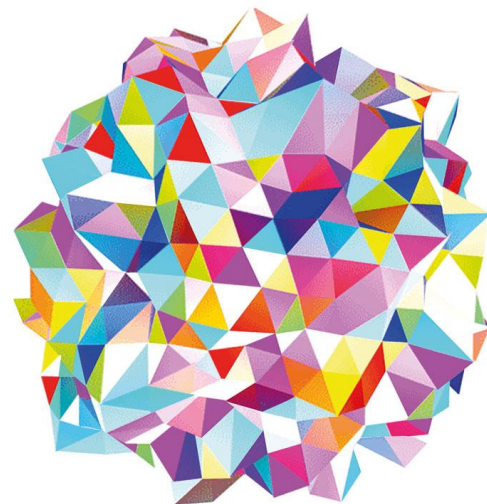


Volume up...



# Sound Control - Flow Machines

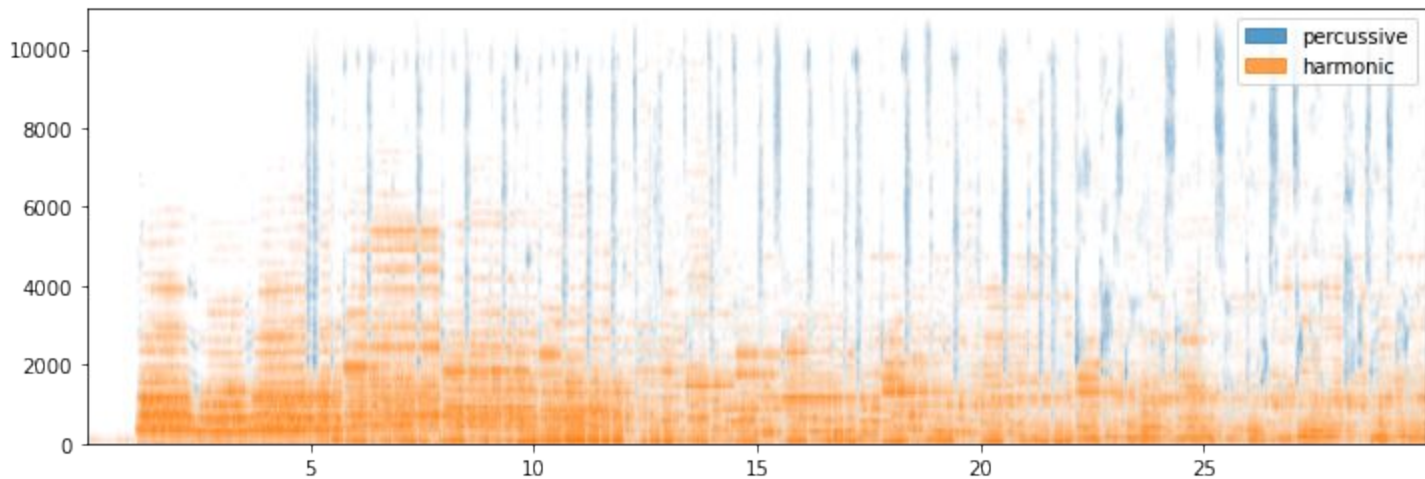
Guess who is this group ?



flowmachines

cf audio/daddys\_car.mp3 on github

# Sound Control - Exploration



cf `Daddys_Car_Exploration.ipynb` on github

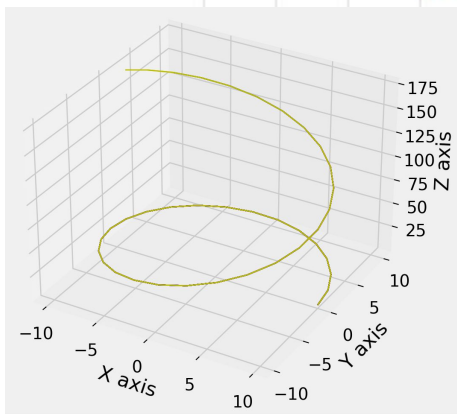
**Dance control**

# Dance control - Moves

- **Objectif** : implémenter des mouvements au rythme des beats
- Plusieurs types de mouvements

- Translation
  - mouvement de balancier (demo)
  - mouvement droite gauche, carrés, ...
- Rotation
  - mouvement circulaire
  - mouvement en spirale
    - cf 3d-view on github → trajectoire dynamique
- Patrol

```
def dance_right_left(self, _):  
    sign = 1  
    while True:  
        if self.bridge.changed_beat:  
            sign = -1 * sign  
            self.bridge.changed_beat = 0  
            self.drone.send_rc_control(  
                sign * self.speed_side, 0, 0, 0)
```



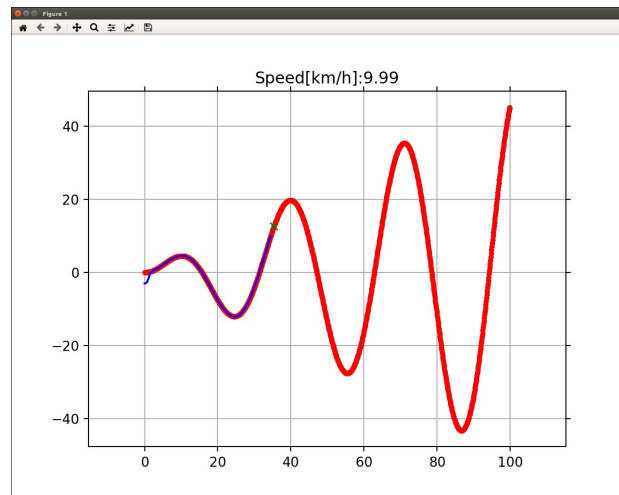


# Dance control - Difficultés

- interrompre des actions en cours
  - impossible avec les commandes forward/back, up/down, left/right, curve\_xyz\_speed, go\_xyz\_speed, ...
- position
  - utilisation de send\_rc\_control (paramètres vitesse)
  - difficulté de la localisation absolue du drone (changement de référentiel constant)
  - suivi de trajectoires
- instabilité drone
  - mouvements parasites
  - drift important (recalibrage)
  - drone escapes :-(
  - déplacements inattendus en cours de vol

# Dance control - Explorations

- projet d'implémentation d'un pure pursuit
  - nous avons envisager d'implémenter un algo de pure pursuit, le drone suivant les points d'une trajectoire calculée au fur et à mesure (cf pure-pursuit/ sur le repo github)
  - la commande send\_rc\_control prenant en paramètre des vitesses, il fallait implémenter
    - une conversion de référentiel en continu pour faire correspondre la position (0, 0) du robot dans son référentiel aux coordonnées de la trajectoire dans le référentiel d'origine
    - calculer la distance parcourue et PID p/r à la vitesse pour localiser le drone à chaque instant
- problématique similaire pour le mouvement en spirale



# Architecture & Thread Control

# Architecture & Thread control (1)

Principes mis en oeuvre :

- une classe TelloHandler qui gère
  - l'objet Tello(),
  - la boucle capture vidéo,
  - les interruptions,
  - le démarrage des sous-modules, ...
- une boucle de capture vidéo
  - capte les frames en continu
  - détecte les features de la main
  - récupère les saisies clavier
  - gère l'état courant
- ...

# Architecture & Thread control (2)

## Principes mis en oeuvre (suite)

- l'exécution parallèle (threads)
  - via la classe TelloThread
  - usage pour:
    - le timer de validation de la durée de maintien de la position des doigts (0.5 secondes)
    - la musique
    - les mouvements
- keyboard control
  - cf README.md

# Architecture & Thread control (3)

- Classes créées
  - TelloHandler : gère la connexion au drone, la boucle de capture, les états et les commandes
  - HandDetector: détecte les features de la main (MediaPipe)
  - TelloSound: gère la musique (play/stop, ajout beats, extraction beats, ..)
  - TelloDance: gère les mouvements
  - TelloThread: contrôle les actions parallèles
  - TelloBridge: gère la communication entre classes
- API utilisée
  - djitellopy

**Démos en vidéo**

# Vidéo statique





# Vidéo swing



**Bilan**

# Bilan

## Bénéfices

- usage du python dans les programmes scripts (vs notebook)
- approfondissement de la programmation objet et parallèle (Thread, Event)
- gestion des timeouts
- apprentissage des outils de vision (opencv et mediapipe)
- exploration de libs de son: librosa, alsa, vlc, pycaw, soundfile, sounddevice, playsound

## Difficultés

- compatibilité des libs entre linux et windows
- la réalité de la programmation robotique
  - problème de luminosité / delta entre ce qu'on espère et ce qu'on obtient
- debugging et tests : temps à allouer
- contraintes dues à l'API

**MERCI**