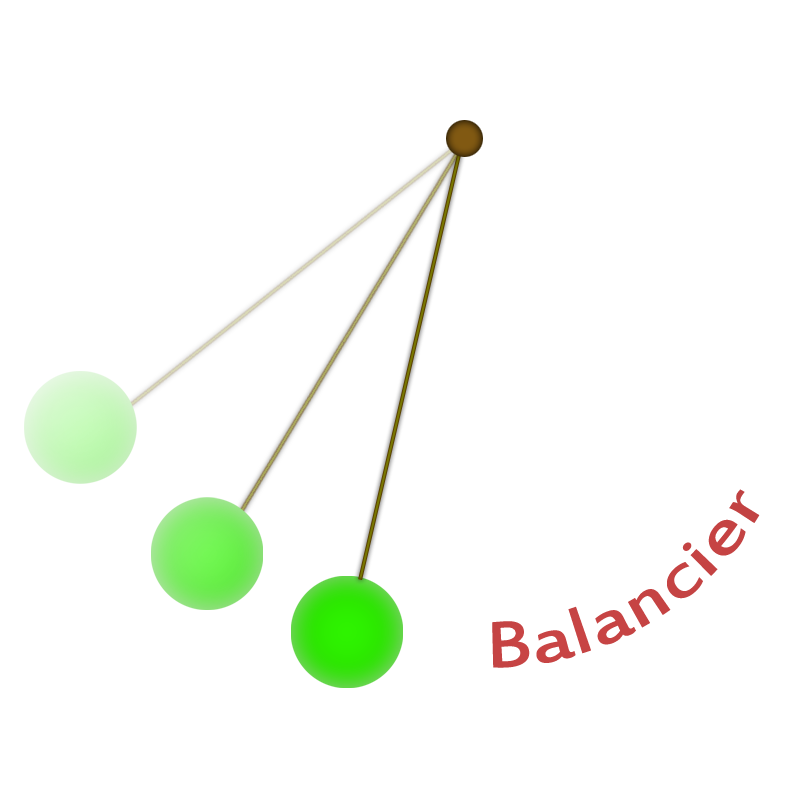
|  |
| --- |
| Détection et calcul d’un mouvement de balancier |
| Projet OpenCV dans le cadre du cours de traitement d’images |
| Christophe Bolinhas et Mathieu Rosser |



# Introduction

Dans le cadre du cours de traitement d’image, il fût décider de réaliser un projet OpenCV permettant de capturer puis d’analyser un mouvement de balancier afin d’en retirer des données et d’effectuer des comparaisons avec le modèle physique.

La problématique était la suivante :

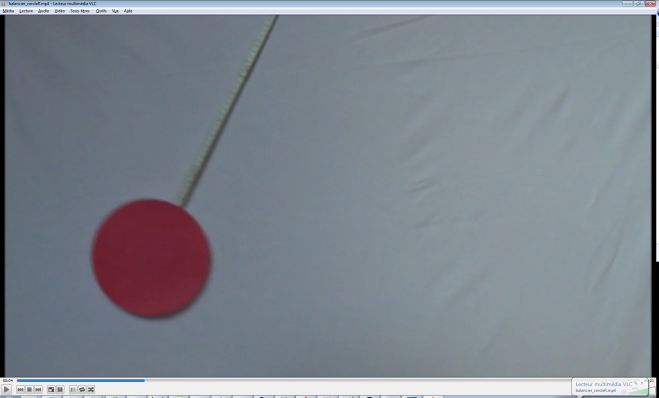
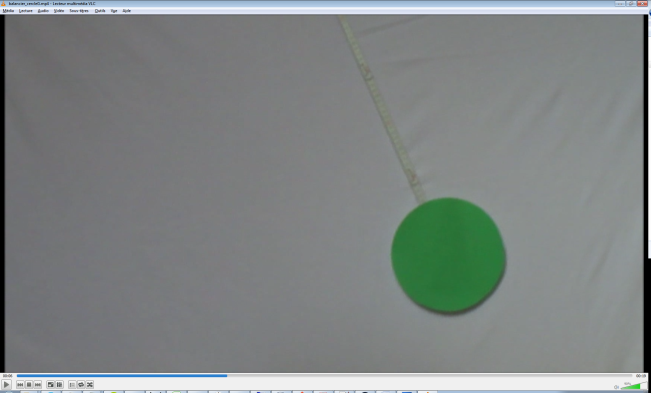
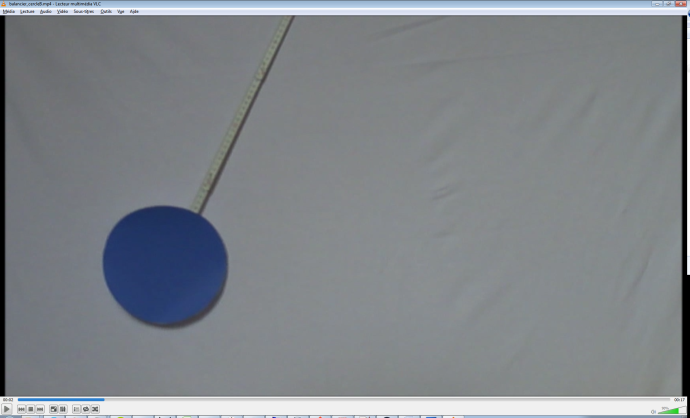
Trouver une solution pour détecter des objets en balancement et en retirer des valeurs. Celle-ci met notamment en avant les cas de la détection de forme, de motifs ou de mouvement au sein d’une vidéo.

# Capture des objets

La première étape dans l’analyse de ce mouvement est d’extraire l’élément en mouvement ainsi que ses positions dans l’espace de la vidéo. Pour ceci 3 méthodes furent appliquées.

## Détection de cercle avec Hough

Partant de vidéos utilisant un cercle de couleur se balançant au bout d’une règle.



Afin d’extraire les différents cercles de ces images, il est d’abord nécessaire de les isoler de leur fond. Pour ce faire, un split de l’image source est réalisé puis une sélection d’une des images ayant des pixels différents de ceux de la sphère est sélectionnée. Ceci va nous permettre d’avoir une image avec une bonne différence avec le fond de l’image.

[Screenshot avant threshold]

En se basant sur cette dernière image, il est relativement simple de réaliser un threshold et de ainsi faciliter l’analyse pour la suite.

[Screenshot threshold]

Une problématique liée à la règle utilisée pour balancer ces cercles cause quelques erreurs au niveau de leur définition. Afin de palier à ce problème il est effectué une ouverture sur l’image ce qui va permettre de supprimer ces imperfections.

Enfin, le cercle est detecté à chaque instance de l’image en utilisant la fonction HoughCircle d’OpenCV

cv2.HoughCircles(dilatation, cv2.cv.CV\_HOUGH\_GRADIENT, 1, 500, np.array([]), 200, 10, 60, 3000)

Cette dernière va permettre de détecter les centres à chaque moment T de la vidéo afin de pouvoir ensuite en faire une analyse mathématique.

## Détection d’image avec Sift

## Détection d’image avec Orb

# Analyse des données

Après avoir récupérer les données sur les vidéos au travers de la détection de centre ou au travers de la détection des points morts, il est possible d’effectuer une analyse et d’avoir ainsi les informations nécessaires au calcul des périodes. Celle-ci va ensuite être utile au calcul de diverses valeurs relatives au balancier.

## Analyse par extremum à partir de liste de points

## Analyse par période provenant de l’analyse des points morts

# Comparaisons au modèle physique

Blabla

## Calculs

## Résultats

# Conclusion

Bla bla

Meilleur algo : x

## Amélioration et idées d’autres approches

## Bilan

Avec ça on a apprit ça et ça et la bla bla =)