
Compte rendu #6

Avancement sur le traitement vidéo et l'assignation sans répétition

*CANHOTO Mickaël, BELDJILALI Maxime
Master 1 Imagine, groupe #2.2*

Avancement sur le traitement vidéo et l'assignation sans répétition	1
I - Introduction	1
II - Traitement vidéo	1
III - Assignation sans répétition	1
IV - Conclusion	1

I - Introduction

Au cours de la semaine écoulée, nous avons poursuivi nos efforts dans l'amélioration et l'optimisation de notre programme de mosaïque d'image. Nous avons concentré nos efforts sur deux aspects clés : l'amélioration du système de recherche sans répétition et le développement d'une fonctionnalité de lecture de vidéo. Dans ce rapport, nous détaillerons les progrès accomplis dans chacun de ces domaines et les défis rencontrés en cours de route.

II - Traitement vidéo

Pour le traitement vidéo, il faut repenser l'organisation du programme afin de réaliser une mosaïque pour chaque images d'une vidéo. De plus, il faut ajouter la possibilité de lire une vidéo, la décomposer en images, et la reconstruire pour obtenir la vidéo finale.

Pour cela, nous allons nous aider de la librairie [OpenCV](#). Elle va nous permettre de lire une vidéo et de la décomposer en image.

```
# Get video properties
cap = cv2.VideoCapture(self.image_in_location)
self.frameCount = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
self.frameWidth = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
self.frameHeight = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
self.fps = int(cap.get(cv2.CAP_PROP_FPS))

while(cap.isOpened()):
    ret, frame = cap.read()
    if ret:
```

```
self.buffer[i] = frame
```

De plus, elle va nous permettre de recréer la vidéo à partir de nos images traitées.

```
# Define the codec and create VideoWriter object
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
out = cv2.VideoWriter(self.image_out_location, fourcc, self.fps,
(self.frameWidth, self.frameHeight))

# Write every frame in the output video
for i in range(self.frameCount):
    out.write(self.buffer[i])
```

L'objectif est de maintenant créer une image mosaïque pour chaque frame dans la vidéo. Cependant, de nombreux problèmes apparaissent lorsque nous essayons de l'implémenter. Par exemple, la résolution entre la vidéo originale et la mosaïque diffère. Il faut aussi prendre en compte l'évolution de notre programme pour l'adapter à la prise en charge vidéo. C'est pourquoi notre prise en charge vidéo n'est pas encore fonctionnelle.

III - Assignment sans répétition

Nous avons aussi optimiser l'algorithme faisant l'assignation sans répétition. En effet, il était possible d'accélérer légèrement cette phase en permettant une recherche moins profonde dans le KD Tree sur une première assignation. Cela a permis d'obtenir des temps d'exécution environ 20% plus courts mais n'a pas affecté la complexité temporelle. Par conséquent, cet algorithme d'assignation n'est toujours pas satisfaisant si l'on souhaite augmenter le niveau de segmentation de l'image finale et l'on doit toujours implémenter des algorithmes plus rapides comme le CEP décrit il y a deux semaines.

IV - Conclusion

En conclusion, cette semaine a été marquée par l'ajout de fonctionnalités à finaliser. Grâce à nos efforts pour améliorer le système de recherche sans répétition. De plus, l'ajout de la fonctionnalité de lecture de vidéo élargit considérablement les possibilités créatives offertes par notre programme.

Cependant, des défis subsistent, notamment en termes d'optimisation des performances et de correction pour la lecture de vidéos.

Dans les semaines à venir, nous nous concentrerons sur la résolution des problèmes restants pour finaliser les objectifs actuels.