

# Rendu TP 3

## Traitement d'images

BOUTIN André  
DE ROECK Dimitri

Décembre 2023

### Exercice 1

Afin de choisir l'algorithme, nous avons effectué divers tests sur une portion des fragments à notre disposition. Nous avons notamment testé SIFT, FAST pour les points d'intérêts et BRISK pour les descripteurs. De même, nous nous sommes appuyés sur quelques articles universitaires comparant la performance des différents algorithmes.

Lors de ces tests, nous avons observé que ORB était peu performant, même sur des fragments de tailles importantes, là où SIFT et FAST étaient bien plus performant. Cependant, entre ces deux derniers, nous avons observé un matching plus performant avec FAST et BRISK avec un BruteForceMatcher qu'avec SIFT avec le même matcher. Les associations faites avec SIFT étaient très éloignées de la réalité de terrain contrairement à FAST qui était beaucoup plus précis.



FIGURE 1 – Exemple montrant le fait précédemment

## Exercice 2

K, le nombre minimum d'associations pour calculer une paramétrisation  $x$ ,  $y$ ,  $\theta$  serait de 2. Ceci reste une valeur minimale, car cela dépend de la précision souhaitée. De plus, une association ne permettrait pas de calculer la rotation.

Pour calculer les  $x$ ,  $y$  et  $\theta$  en fonction des associations :

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} xf \\ yf \end{pmatrix} + \begin{pmatrix} tx \\ ty \end{pmatrix}$$

Lors du développement de l'algorithme, nous avons remarqué qu'à de nombreuses reprises, les possibilités d'angles étaient parfois réparties dû à des petites différences. Nous avons donc ajouté une fonction permettant d'avoir le vote le plus similaire en testant notamment l'angle modulo une marge d'erreur afin de regrouper les votes pour un angle et avoir la bonne rotation finale.

## Exercice 4

L'avantage de l'utilisation de la distance euclidienne est que si on a déjà de bonnes associations, retrouver la position des fragments sera très précis et beaucoup plus rapide que RANSAC car ce dernier fait de l'aléatoire et a besoin de nombreuses itérations pour garantir un résultat satisfaisant.

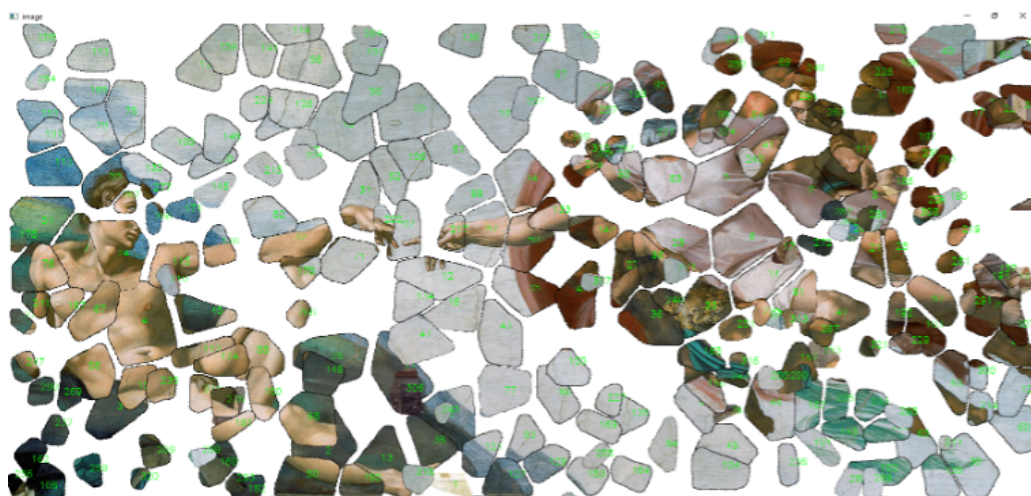


FIGURE 2 – Résultat obtenu en lançant le fichier `reconstruction_euclidian.py`

En jouant sur les paramètres du vote au niveau des angles et avec le contrôle des fragments avec la distance euclidienne, nous avons réussi à obtenir un résultat plutôt satisfaisant.

À titre de conclusion, nous pourrions dire que l'algorithme le plus efficace pour cette fresque avec ces fragments est l'algorithme FAST + BRISK. SIFT et ORB n'ont pas donnés de résultats très concluants, même s'ils ont leurs points forts.