

Rapport sur le projet d'analyse d'image

Introduction

Le projet consiste à développer une application utilisant la bibliothèque OpenCV pour analyser une image, extraire des objets via les contours détectés, simplifier ces contours à l'aide de l'algorithme de Douglas-Peucker, et identifier les angles convexes et concaves. Ce travail inclut également la génération d'images séparées (vignettes) des objets détectés, en les enregistrant dans un dossier spécifique.

L'image analysée est une photo de puzzle sur un fond orange :



Figure 0 Image originale

1. Extraction de la couleur dominante

On a d'abord converti l'image en espace de couleurs HSV, qui permet de mieux gérer la couleur de l'image de manière séparée de la luminosité et de la saturation.

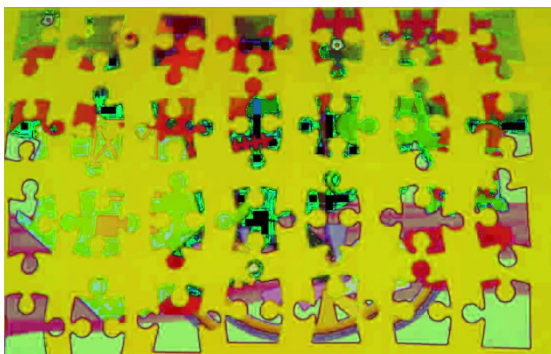


Figure 1 image puzzle converti en HSV

Ensuite, pour détecter la couleur dominante de l'image, on a utilisé l'histogramme de la composante teinte (H) de l'image HSV. Voici comment on a procédé :

Séparation des canaux HSV : On a utilisé `cv::split` pour séparer l'image en trois canaux : teinte (H), saturation (S) et valeur (V).

Création de l'histogramme de la teinte : On a créé un histogramme de 180 bins pour la teinte (H) en utilisant `cv::calcHist`, ce qui permet de connaître la fréquence de chaque valeur de teinte dans l'image.

Identification de la teinte dominante : On a utilisé `cv::minMaxLoc` pour trouver la valeur de teinte ayant la fréquence la plus élevée dans l'histogramme, ce qui représente la couleur dominante de l'image.

Calcul de la saturation et de la valeur moyennes : On a utilisé `cv::mean` pour calculer les moyennes de la saturation (S) et de la valeur (V), afin de mieux décrire la couleur dominante dans son ensemble.

Cela m'a permis de déterminer une couleur dominante et de créer un masque autour de cette couleur pour isoler les objets correspondants.

2. Création du masque basé sur la couleur dominante

Avec la teinte dominante obtenue, on a généré un masque pour extraire les zones de l'image ayant une couleur proche de cette teinte. Voici les étapes :

Définition de la plage de teinte acceptable : On a choisi une plage de teinte autour de la couleur dominante (avec une plage de 6.5 degrés pour être assez flexible).

Création du masque : En utilisant `cv::inRange`, on a créé un masque où les pixels dont la teinte (H) se situe dans la plage définie sont blancs (valeurs 255), et les autres sont noirs (valeurs 0).

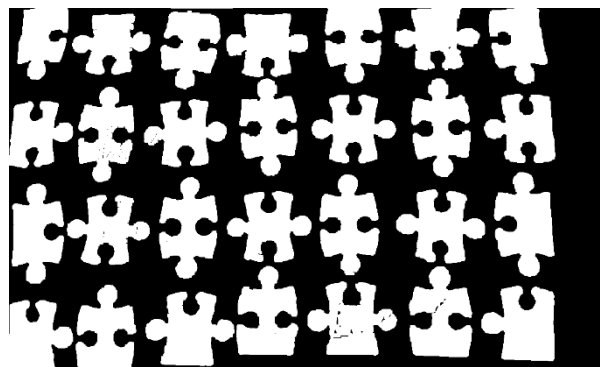


Figure 2 masque des pièces

Affinage du masque : Pour améliorer la précision du masque, on a utilisé des opérations morphologiques, comme l'érosion et la dilatation, avec un noyau de forme rectangulaire pour éliminer les petites imperfections et renforcer les objets détectés.

3. Détection des contours et simplification

Une fois le masque créé, on a utilisé la fonction `cv::findContours` pour détecter les contours dans l'image. Les étapes sont les suivantes :

Détection des contours : La fonction `cv::findContours` permet de détecter les contours dans l'image binaire générée par le masque.

Simplification des contours : Pour chaque contour détecté, on a appliqué l'algorithme de Douglas-Peucker via `cv::approxPolyDP`. Cela permet de simplifier les contours en réduisant le nombre de points tout en conservant la forme générale. On a ajusté la précision de la simplification en fonction de la longueur des contours, ce qui permet de garder un bon compromis entre précision et simplification.



Figure 3 contour des pièces approximé

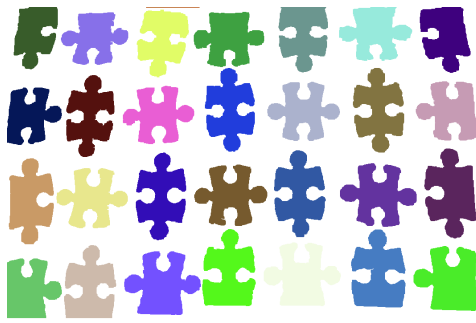


Figure 4 contour des pièces non approximé

4. Calcul des angles convexes et concaves

Pour chaque contour, on a calculé les angles entre les points successifs et on a identifié les points convexes et concaves :

Calcul des angles : On a utilisé une fonction Angle qui prend trois points (A, B, C) et calcule l'angle

signé entre eux en utilisant le produit scalaire et le déterminant. Le calcul est effectué en radians, puis converti en degrés pour une meilleure interprétation.

Identification des points concave : Si l'angle entre les points est supérieur à 30° , on a considéré que c'était un point concave. Dans ce cas, on a dessiné un point rouge pour le marquer.

Identification des points convexes : Si l'angle est inférieur à -30° , on a marqué ces points en vert pour les distinguer.

```
Convexe à 0 (angle = -84.7965°)
Convexe à 1 (angle = -95.9711°)
Concave à 2 (angle = 159.329°)
Convexe à 3 (angle = -77.1431°)
Convexe à 4 (angle = -51.3131°)
Convexe à 5 (angle = -121.024°)
Concave à 6 (angle = 88.0799°)
Convexe à 7 (angle = -95.4467°)
Convexe à 8 (angle = -81.7147°)
```

Figure 5 affichage des angles des différents points d'une pièce

5. Création de vignettes à partir des contours

Une partie essentielle de ce projet est l'extraction des objets détectés sous forme de petites images, ou "vignettes", et leur enregistrement dans un dossier dédié. Voici comment on a procédé :

Détection des contours : Pour chaque contour détecté, on a calculé un rectangle englobant (bounding box) avec la fonction `cv::boundingRect`. Ce rectangle délimite la zone de l'objet.

Extraction de l'objet : En utilisant ce rectangle, on a extrait la région d'intérêt (ROI) de l'image originale à l'aide de l'opérateur de découpe, `croppedImage(boundingBox)`.

Enregistrement des vignettes : On a ensuite sauvegardé chaque objet extrait dans un fichier PNG sous le nom `object_x.png`, où x est l'indice de l'objet. Chaque image est enregistrée dans le dossier data.



Figure 6 vignette de pièces

6. Affichage des résultats

Enfin, on a affiché plusieurs fenêtres à l'aide de `cv::imshow` pour visualiser l'image originale, le masque, et les contours simplifiés avec les lignes et les points marqués. Cela permet de valider visuellement les étapes du processus et de vérifier que tout fonctionne comme prévu.



Figure 8 image originale

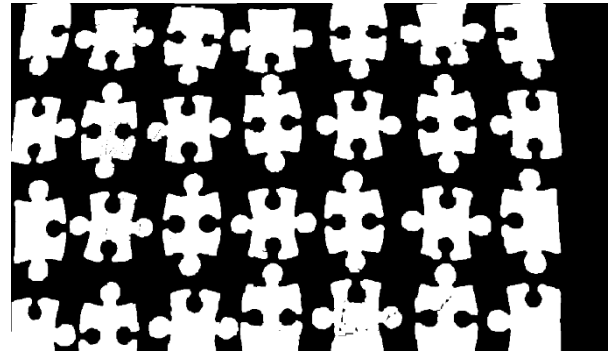


Figure 7 image du masque

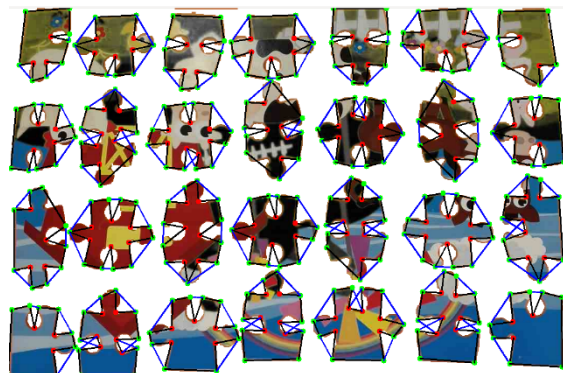


Figure 9 image des pièces avec les points convexes et concaves et leurs contours simplifiés

Les difficultés rencontrées

Au départ, on a eu des difficultés à comprendre les spécificités de l'espace de couleur HSV. La manipulation de la plage de teinte m'a demandé quelques essais avant de bien comprendre comment filtrer les couleurs efficacement. Cependant, après quelques ajustements, cette partie a fini par fonctionner correctement.

De plus, le calcul des angles signés m'a posé un problème. On a d'abord utilisé une méthode basée sur ce qu'on connaissait, mais elle ne retournait que des angles non signés. On a donc dû chercher sur Internet la formule adaptée à mon cas.

Enfin, on a aussi rencontré un souci pour représenter les points sur la figure, mais cela s'est vite réglé.

