# Trouver la méthode à utiliser pour détecter un panneau

1. Seuillage de l’image

Changer l’espace de couleur : passer en HSV : fonction Imgproc.Color\_BGR2HSV

Faire une ouverture (renforce l’apparition de l’objet). Attention : ouverture basée sur les pixels noirs :

* Faire une érosion (permet l’extraction unitaire des objets donc de les séparer : test si un des pixels du masque est noir)
* Faire une dilatation (pour restaurer les objets et donc détection de contour)

Afficher dans une fenêtre la couleur rouge et dans un fenêtre la couleur bleue de l’image en mode HSV (utiliser core.inRange : reprendre les fonctions du TP avec Song pour afficher une matrice dans une fenêtre graphique : slide 13, 21)

Se concentrer sur la forme la plus grande entre la bleue et la rouge : Calculer la surface de l’espace délimité par le contour : Imgproc.contourArea

1. Détection de contour

Appliquer le filtre de Canny : Imgproc.Canny: détection de segments dans une image, Imgproc.findContours: détection de contours, Imgproc.drawContours: affichage de contours (slide 24)

1. Reconnaitre la forme : MatOfPoint

Calculer la surface de l’espace délimité par le contour : Imgproc.contourArea

Calculer le plus petit cercle qui enveloppe le contour avec la fonction Imgproc.minEnclosingCircle

Si Surface est proche de PI\*R2 alors c’est un cercle

Sinon calculer le polygone approximatif du contour

Si nombre de segments est 3 alors c’est un triangle

Si nombre de segments est >= 4 et <= 6 alors il s’agit d’un rectangle, ou d’un carré ou d’un polygone.

1. Template matching

Extraire de l’objet depuis l’image

Convertir l’objet en niveaux de gris

Extraire les caractéristiques et les points d’intérêts de l’objet

Pour chaque Template i :

Extraire les caractéristiques et les points d’intérêts du Template (FeatureDetector.create(FeatureDetector.XXX) puis DescriptorExtractor.create(DescriptorExtractor.XXX))

Calculer la correspondance entre le Template i et l’objet

Choisir le Template j tel que j = Min(score(i))