INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

Rapport technique

AUTOMATISATION D'UNE CHAÎNE DE TRI

Projet réalisé par

Thomas DUVIGNACQ
Soizic HELLO
Othmane MESKINE
Solal THOMAS
Groupe 1

Année 2019-2020

I. Problématique

De plus en plus d'industriels souhaitent automatiser leur chaîne de production en ajoutant des options de tri de pièces ou d'objets. SOLEC souhaiterait investir sur ce marché en plein essor en proposant une solution de tri d'objets en fonction de leur couleur et/ou leur forme. Ce système serait basé sur un convoyeur entraîné par un moteur pas-à-pas, un détecteur de couleur et un détecteur de présence de pièces. La problématique explicite est donc la réalisation d'une chaîne de tri d'objet en fonction de la couleur ou de la forme de l'objet tout en ayant la possibilité d'agir sur les modalités de tri (forme et couleur donc mais aussi vitesse et contrôle).

II. Architecture du prototype

1. Du fonctionnement de base

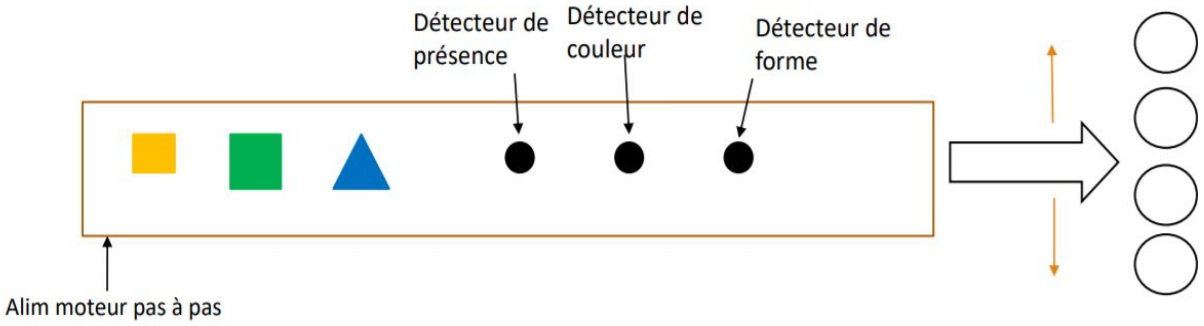

fig 1 : schéma de la fonction de base du montage soit le tri de pièce selon la forme ou la couleur

2. Au schéma fonctionnel

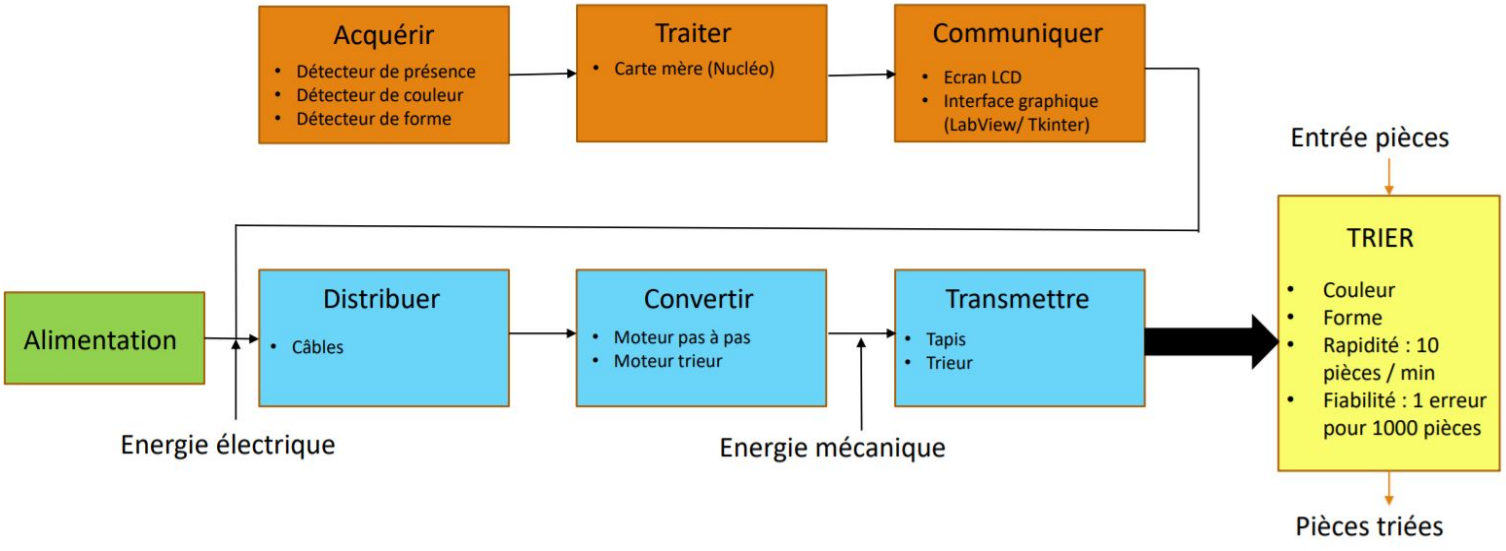

fig 2 : schéma fonctionnel du montage

3. Architecture des codes

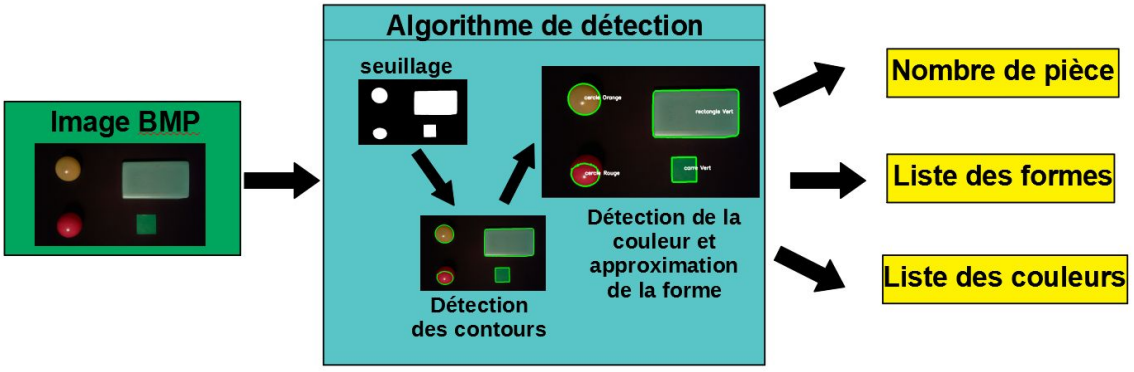

fig 3 : schéma de fonctionnement du programme de détection de forme et de couleur

III. Fonction réalisées

La réalisation du projet est dans un premier temps séparée en deux parties. Il y a une partie interface graphique, qui permet d'afficher le nombre de pièces triées en fonction de leur couleur ou de leur forme et permet à l'utilisateur d'interagir avec celle-ci, et une partie traitement d'image qui permet la détection de la forme et de la couleur des pièces.

1. *Interface graphique*

L'interface graphique a été conçue avec *Tkinter* et permet à l'utilisateur de vérifier le nombres de pièces triées et le nombre de pièces par couleur et par forme, qui seraient disposées sur un convoyeur. Elle permet également à l'utilisateur de changer le mode de tri des pièces, qui se fait en fonction de la couleur ou de la forme de la pièce.

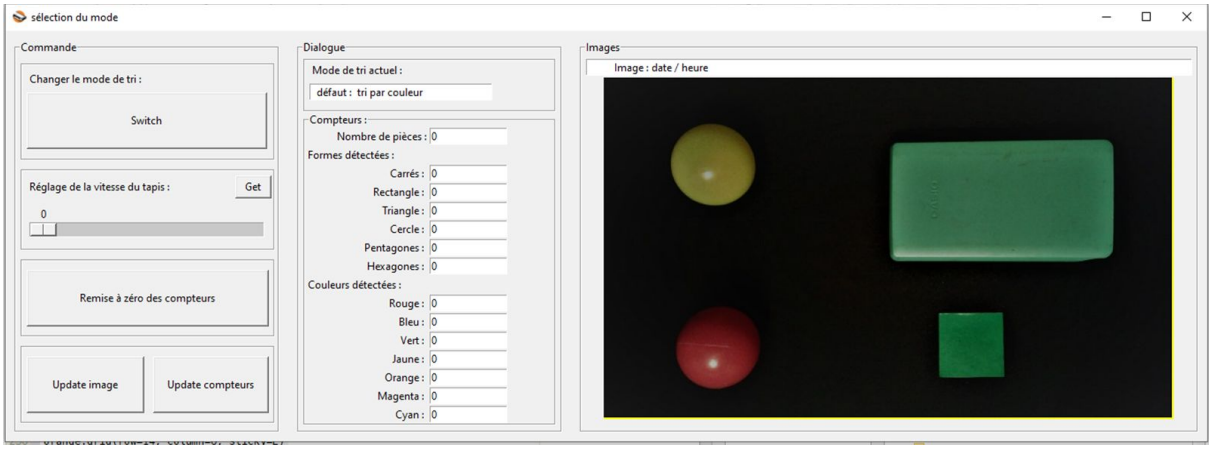

fig 4 : Interface graphique pour l'utilisateur

Cette interface graphique est décomposée en 3 parties. Il y a une partie *commande*, qui permet à l'utilisateur de directement interagir avec l'interface, une partie *dialogue* qui donne

les informations concernant le tri et une dernière partie *image*, qui affiche l'image que l'on est en train de traiter. Ces trois parties sont dans trois *frame* différentes, de sorte que les parties soient bien séparées et que la lecture de l'interface soit facile pour l'utilisateur.

Dans la partie *commande*, plusieurs boutons donnent la possibilité à l'utilisateur d'interagir avec l'interface et de faire des réglages.

- Bouton Switch : change le mode de tri des pièces
- Bouton Get : permet à l'utilisateur, grâce au curseur, de régler la vitesse du tapis
- Bouton Remise à zéro : remet tous les compteurs à 0
- Bouton Update compteurs : met à jour les valeurs affichées
- Bouton Update Image : met l'image à jour et affiche la date et l'heure

Ensuite, dans la partie *dialogue*, l'interface affiche le nombre de pièces total triées ainsi que le nombre de pièces triées par couleur et par couleur. On a ainsi accès au nombre de formes détectées et au nombre de couleurs détectées. Lorsqu'une pièce est triée, le compteur ajoute +1 à la case correspondante à sa forme ou à sa couleur. L'interface affiche également le mode de tri actuel.

Enfin, dans la partie *image*, on affiche l'image qui est en train d'être traitée, qui correspond à celle utilisée dans la partie traitement d'image.

2. *Traitement d'image*

La partie traitement d'image a été réalisée en utilisant la bibliothèque graphique OpenCV. C'est l'une des bibliothèques les plus utilisées dans le domaine du traitement d'image. Ses algorithmes permettent de réaliser la plupart des opérations classiques en traitement d'images.
Le programme de détection contient deux fonctions : une fonction de reconnaissance de forme et une fonction de détection des couleurs. Pour obtenir le traitement d'une image, il faut lancer la fonction **reco(**"*Nom_Image.bmp*"**)**. Celle-ci renvoie alors 3 variables : le nombre de pièce détectée, une liste des formes et une liste des couleurs de chaque pièce. La fonction de reconnaissance des couleurs est appelée par la fonction **reco**.

2-1. Détection de forme

Etape 1 : On convertit l'image en niveau de gris à l'aide de la fonction **cv2.cvtColor()** pour pouvoir effectuer ensuite un seuillage avec **cv2.threshold()**. Le seuillage d'image est une technique qui permet de transformer une image en niveau de gris en une image dont les valeurs de pixels ne peuvent avoir que la valeur 1 ou 0. On obtient ainsi une image où les formes sont en blanc et le fond en noir. On peut extraire de cette image seuillée les contours de chaque pièce de l'image avec **cv2.findContours()**.

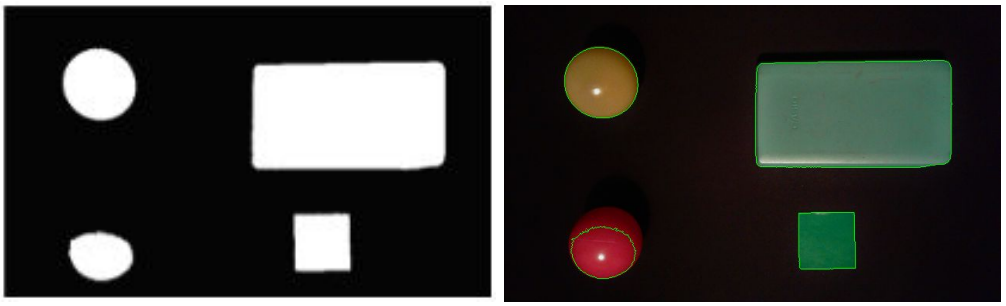

fig 5 : Image seuillée à gauche et contours détectés à droite

Etape 2 : Après avoir détecté les contours, on va réaliser une approximation de ces contours via la fonction **approxPolyDP()**. Cette approximation consiste à comprimer les contours qui sont sous la forme de segments horizontaux, verticaux ou diagonaux. Autrement dit, on ne laisse que les extrémités du segment au lieu de garder tous les points qui le constituent. Par exemple, un rectangle sera représenté par 4 points. Ensuite, en comptant le nombre de sommets, on peut déterminer la forme. On a fixé le nombre maximal à 5 sommet ce qui correspond à un hexagone. Au delà de ce nombre, on approxime la forme à un cercle. On teste chaque contour un par un. Il se peut que des artefacts soit détectées, on teste donc le périmètre du contour pour qu'il est une valeur minimale et soit bien celui de l'une de nos pièces.

2-2. Détection de couleur

Concernant la détection de couleurs, on convertit tout d'abord notre image dans le repère HSV pour Hue Saturation Value (en français TSV pour Teinte Saturation Valeur). L'intérêt de passer du repère RGB au repère HSV est qu'il est plus simple de classifier les couleurs. En effet, en RGB, il est nécessaire d'établir les conditions sur le triplet de valeurs alors que dans le cas du HSV, nous pourrons utiliser uniquement les informations de teinte (H).

La teinte est codée suivant l'angle qui lui correspond sur le cercle des couleurs :

- **Rouge** : [0°,20°] et [330°,360°]
- **Orange** : [20°,40°]
- **Jaune** [40°,80°]
- **Vert** : [80°,160°]
- **Cyan** : [160°,200]
- **Bleu** : [200°,260°]
- Magenta: [260°,330°]

Cercle chromatique

La **saturation** représente l'intensité de la couleur et la **valeur** représente la brillance de la couleur. Dans notre cas, on ne s'intéresse qu'à la valeur de la **teinte**. Il est ainsi plus simple de travailler qu'avec une seule variable.

Ensuite, on applique à notre image une sorte de masque qui est en réalité une image noire contenant les formes en blanc détectées auparavant. Grâce à ce masque, on va pouvoir accéder directement aux valeurs des pixels qui se trouvent dans la forme détectée et calculer par la suite la valeur moyenne de tous ces pixels pour déterminer la couleur.

IV. Fonctions à réaliser

Pour avoir un montage réel complètement fonctionnel il nous resterait plusieurs choses à réaliser :

- Premièrement l'étude et la mise en fonctionnement des moteurs pas à pas que nous n'avons pas pu faire fonctionner. Ceci dans le but de faire tourner le tapis et cela à une vitesse réglable via un paramètre normalisé dans l'interface graphique que l'on vient récupérer et appliquer via le bouton get.

- Il nous faut aussi dans ce sens réunir tous les codes réalisés. En effet dans le but de bien faire fonctionner le montage, il est essentiel de réunir le code de l'interface avec celui du traitement d'image car ceux-ci fonctionnent de pair. Par exemple, l'interface affiche des compteurs et potentiellement une image qui sont obtenus via le programme qui traite l'image des objets qui passent. Il nous faudrait de plus fusionner la partie code du moteur pour être en mesure de pouvoir le contrôler via l'interface.

- Ensuite pour prendre les photos et pouvoir les traiter dans un environnement optimal (éclairement constant et source contrôlée) il nous faudrait mettre en oeuvre une sorte de boîte noire dans laquelle les objets entrent et sont pris en photo avant de ressortir pour être trié.

- Il nous faudrait également tester les capteurs à notre disposition individuellement pour savoir si nous ne faisons pas fausse route… En bref toute la partie test et mise en oeuvre du montage reste en suspend avant d'avoir un convoyeur trieur effectif.