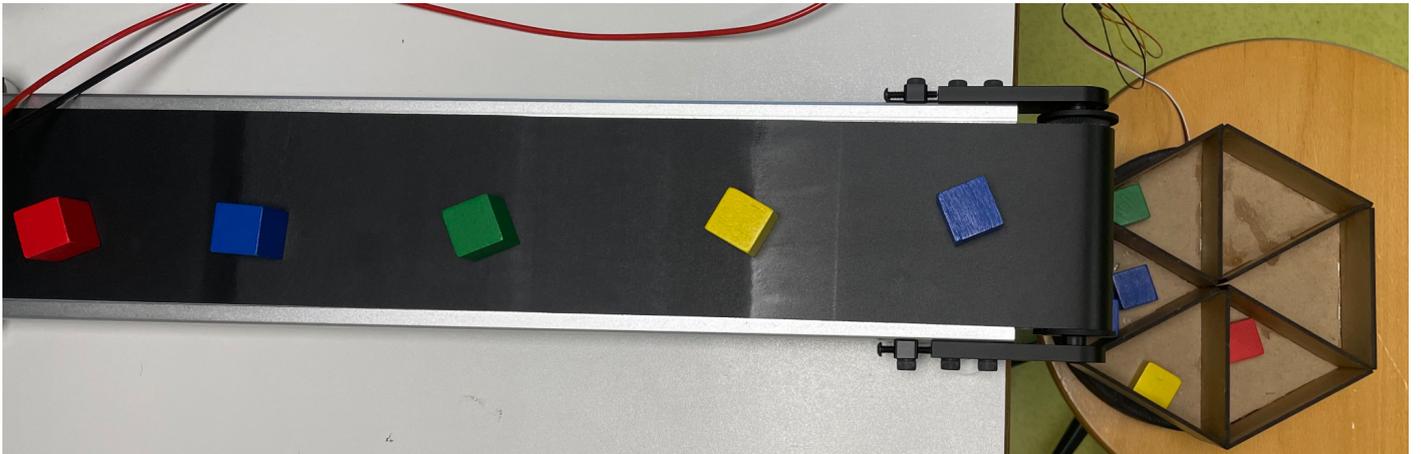


# VISION INDUSTRIELLE

## Pilotage d'un convoyeur



## Sommaire :

Introduction .....	1
I. Cahier des Charges .....	1
II. Schéma Fonctionnel .....	2
III. Détail de chaque étape .....	2
IV. Rétroplanning .....	6
V. Difficultés rencontrées .....	7
VI. Développement possible .....	7

## Introduction

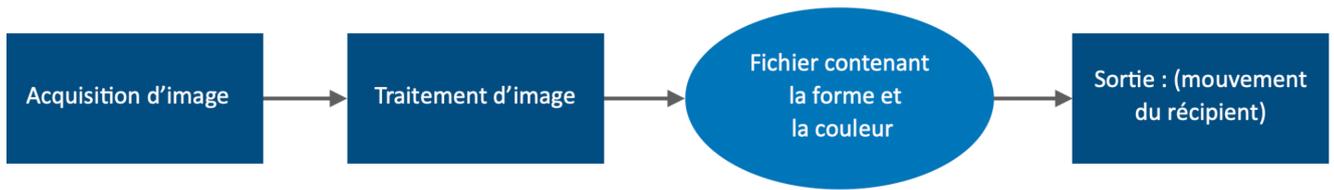
Dans le milieu industriel, le convoyage de pièces, la détection de certains défauts ou le tri par types d'objets sont très couramment utilisés. L'objectif de ce projet est de mettre en place un système de tri de pièces à l'aide d'un convoyeur et d'un système de traitement d'images.

### I. Cahier des Charges

La première version permet de détecter des cubes de 4 couleurs différentes et de les trier dans des conteneurs différents.

- **Rapidité** : Le système doit pouvoir analyser au minimum 10 pièces par minute.
- **Détection** : Le système doit pouvoir différencier les 4 couleurs de base suivantes : rouge, vert, jaune, bleu. La quantité de pièces triée et le temps moyen par pièce doivent pouvoir être affichés. Par la suite, un nouveau capteur pourra être développé pour augmenter la quantité de couleurs détectables. La forme pourra être un plus.
- **Fiabilité** : Une erreur d'une pièce sur 1000 est tolérée sur la détection des couleurs de base.
- **Ergonomie** : Une interface Humain-Machine, permettant de transmettre la couleur (ou forme) des pièces à trier, pourra être développée. Elle doit pouvoir être utilisée sans formation préalable.

### II. Schéma Fonctionnel



La mise en mouvement du convoyeur se fait par l'intermédiaire d'un moteur pas à pas. Le tri des pièces (par couleur ou forme) utilise une caméra. L'acquisition de l'image est réalisée par un code matlab. Ensuite l'image récupérée est traitée grâce à un programme matlab qui utilise différents filtres (ex : ..) in fine ce programme nous donne la couleur de l'objet sur le convoyeur. Cette information est ensuite envoyée à un programme Mbed qui commande le récipient de tri des objets. Le récipient est mis en mouvement grâce à un servomoteur (qui permet une rotation de 180°).

### III. Détail de chaque étape

#### 1. Acquisition de l'image

L'acquisition de l'image se fait via la webcam asservie par MATLAB par la toolbox Webcam Image Acquisition.

En ce qui concerne le hardware, la manipulation nécessite MATLAB et une webcam branchée à l'ordinateur via un port USB.

En ce qui concerne le software, nous avons besoin de MATLAB et de la toolbox *Image Acquisition*.

La première étape de l'acquisition consiste à ouvrir la webcam que l'on souhaite utiliser. Pour cela, nous devons connaître son nom. Afin de savoir quelle webcam est disponible à l'utilisation, on peut entrer la commande "webcamlist" qui retourne le nom des caméras disponibles. Dans notre cas, ici, à l'institut d'optique, notre webcam s'appelle Logitech HD Webcam C270. Nous pouvons définir la webcam que l'on souhaite utiliser grâce à la commande "cam=webcam('Logitech HD Webcam C270')".

Nous pouvons visualiser en direct ce que la webcam voit grâce à la commande : "preview(cam)" et prendre une photo grâce à "img=snapshot(cam)". La photo prise est dans la variable "img". Ensuite, les commandes de base du traitement d'image s'appliquent. "imagesc(img)" permet d'ouvrir l'image.

A la fin de l'utilisation, il ne faut surtout pas oublier la commande "clear('cam')" sans quoi nous pouvons rencontrer des problèmes dans les prochaines acquisitions.

## 2. Traitement de l'image

L'image précédemment acquise est directement récupérée dans MATLAB et traitée dans la foulée. Le traitement consiste en plusieurs étapes : un prétraitement visant à lisser un peu l'image, une seconde partie qui réalise une détection de contour et enfin une dernière partie pour la détermination de la couleur.

Pour ce qui est du lissage, cela se fait par un filtrage gaussien.

La partie détection de contours peut se faire à l'aide d'un filtre de sobel ou par la mesure de l'écart type local. S'ensuit un seuillage dont le paramétrage permet de rendre plus ou moins sensible la détection de contour. Enfin l'ensemble des objets détectés est placé dans un cellule appelée "DEFAUTS" avec d'une part le contour et d'autre part l'objet rempli.

Une fois les objets détectés et recensés dans la cellule DEFAUTS, on regarde à l'intérieur la couleur dominante en fonction de décomposition RGB de la couleur.

## 3. Communication Matlab - Mbed

Il suffit d'établir une connexion série entre Matlab et la carte Nucléo, les données qui transiteront seront simplement des caractères parmi "a", "b", "c", "d", "s" et "g".

## 4. Interface Homme Machine

Nous voulons avoir une interface afin de pouvoir programmer le tri et obtenir quelques informations. D'une part à l'entrée nous avons placé plusieurs box afin de définir quel objet va dans quel récipient et le bouton de démarrage du système. D'autre part, nous voulions savoir quel cube va dans quel bac et avoir un suivi en fonction du temps. Cela se fait par l'affichage d'un histogramme en direct. Enfin, nous avons installé un bouton stop afin d'arrêter le système.

Pour réaliser cette interface nous avons utilisé App Designer sur matlab.

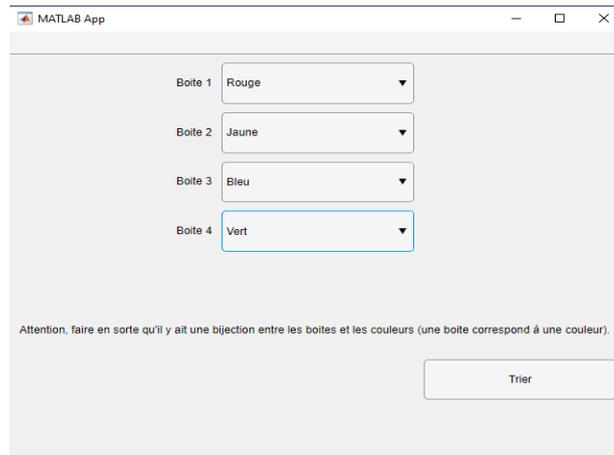


fig 1 : IHM avant le lancement du tri

La figure 1 est la première interface. L'utilisateur.rice doit entrer quelle boite correspond à quelle couleur. Mais attention il faut que chaque boîte corresponde à une couleur de cube existante. Ensuite en appuyant sur le bouton "trier", le tapis se lance et le tri commence.

Une deuxième fenêtre s'ouvre, nous pouvons voir en continue l'histogramme, les statistiques de quelle pièce à été trié, le nombre de couleur.

## 5. Commande du récipient et du tapis roulement

### a) Commande du récipient

Le récipient est monté sur un servomoteur ce qui permet de l'asservir en angle : le servomoteur est alimenté en 5V et reçoit sur une troisième borne un signal PWM dont le rapport cyclique correspond à une commande en angle et dont la période doit être de 20 ms. La documentation du servomoteur donne les rapports cycliques donne les correspondances rapports cycliques <-> angle. Mais celle-ci s'est avérée erronée, nous avons dû ré étalonner le servomoteur empiriquement pour les valeurs de différences d'angle qui nous intéresse. Ces valeurs sont disponibles dans le code fourni en annexe.

### b) Moteur pas à pas

Pour piloter un moteur pas à pas, on commande successivement deux bobines électriques orientées selon des directions orthogonales avec du courant tantôt positif, tantôt négatif. L'axe moteur, magnétique, s'oriente selon le champ. La répétition de cette séquence provoque la rotation de l'axe moteur (et donc du tapis roulant). Cependant, la

gestion de ces étapes une par une est fastidieuse, le LEnsE dispose de cartes électroniques facilitant la gestion d'un moteur pas à pas. Tous les branchements sont détaillés en annexes dans le document *branchements*. Il faut connecter les deux cartes l'une à l'autre via les deux PINs déjà présents dessus. La L297 sera connectée à la carte nucleo. Et la L298 au moteur pas à pas et à une alimentation d'environ 5V (astuce : si après configuration, le moteur pas à pas semble "vibrer" mais ne tourne pas, augmentez légèrement la puissance d'alimentation). Il faut piloter ces cartes de la manière suivante :

- un signal horloge sur le PIN *clock* qui sert à définir la vitesse de rotation du moteur
- un signal digital valant 0 ou 1 sur le PIN *cw* (pour *clockwork*) pour piloter le sens de rotation du moteur.
- un signal digital à 1 sur le PIN *reset*
- un signal digital à 1 sur le PIN *enable* (à 0 pour arrêter le moteur)
- le PIN *Vref* doit recevoir du 3,3V
- GND et 5V sont assez explicites

#### IV. Rétroplanning

Séance du	Avancées
26/01/2022	<ul style="list-style-type: none"> <li>- Prise en main du sujet</li> <li>- Présentation devant un membre de Solec</li> <li>- Répartition des tâches en fonction des savoirs faire / goût</li> </ul>
09/02/2022	<ul style="list-style-type: none"> <li>- Élaboration du récipient (plan..)</li> <li>- Commande du convoyeur</li> <li>- Recherche sur la détection d'objet</li> </ul>
02/03/2022	<ul style="list-style-type: none"> <li>- Commande du récipient</li> <li>- Programme de la détection</li> <li>- Acquisition de l'image</li> <li>- Recherche sur la communication MATLAB/Mbed. Réalisation d'un oscilloscope avec MATLAB</li> </ul>
09/03/2022	<ul style="list-style-type: none"> <li>- Acquisition de l'image</li> <li>- Communication entre Matlab et Mbed</li> <li>- Elaboration programme du traitement d'image</li> <li>- Contrôle qualité</li> </ul>

	<ul style="list-style-type: none"> <li>- Mise en commun des premiers codes. Test : on a mis des cubes sur le tapis, en rentrant manuellement la couleur sur le code matlab le récipient réceptionner correctement les cubes dans ses différentes cases.</li> </ul>
23/03/2022	<ul style="list-style-type: none"> <li>- Mise en commun des codes</li> <li>- Écriture IHM</li> <li>- Rédaction des livrables</li> </ul>
30/03/2022	<ul style="list-style-type: none"> <li>- Présentation</li> </ul>
06/04/2022	<ul style="list-style-type: none"> <li>- Rendu des livrables</li> </ul>

## V. Difficultés rencontrées

Nos premières idées étaient ambitieuses. Initialement nous désirions détecter les objets puis lancer l'acquisition des images.

Nous avons passé beaucoup de temps à essayer de faire fonctionner cette acquisition (réalisation de codes et montages électriques).

## VI. Développement possible

Nous pourrions améliorer notre système en

- Détectant plus de couleurs
- Détectant plus de formes
- Augmenter la vitesse du tapis

Nous devrions donc peut être réfléchir à une nouvelle méthode de tri.