Projet de réalité augmentée :

Au cours des mois de mars à mai, nous avons travaillé sur un projet de réalité augmentée par traitement d’image. La problématique du sujet consistait à remplacer dans une courte vidéo une feuille vierge bleu clair, dont les quatre coins étaient visualisés par 4 picots bleu foncé par une image de notre choix. Pour cela, nous avons utilisé Matlab, créant nos propres fonctions ainsi qu’utilisant celles proposées par le logiciel. Nous allons au cours de ce rapport décrire notre processus de réflexion au cours de la réalisation de ce projet, ainsi que son implémentation sous Matlab.

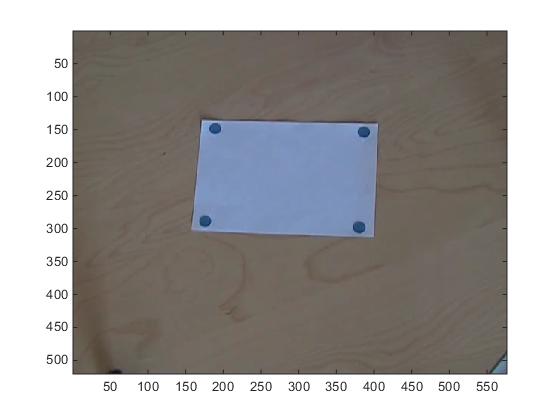
**Démarche algorithmique :**

Afin d’atteindre notre objectif de remplacer la feuille par une image, l’algorithme général peut-être décomposé en 4 étapes principales :

* Phase d’initialisation
* Repérage de la position des picots
* Incrustation de l’image dans la vidéo
* Prise en compte des doigts de la main tenant la feuille dans l’incrustation

Nous allons maintenant détailler ces étapes :

La vidéo fournie pour le projet représente une main faisant glisser une feuille bleue sur une table, les 4 coins de la feuille étant délimités par des points plus sombres :



Afin de pouvoir remplacer la feuille par une image différente, il faut pouvoir détecter les pixels de l’image correspondant à la feuille. Mais déterminer tous ces pixels par la couleur ne donnerait pas des bords de feuille droits, il est plus simple de détecter les 4 picots bleu foncé et d’en déduire la feuille. Ces picots sont détectés par leur couleur. La phase d’initialisation permet de détecter la couleur des picots dès la première image de la vidéo, afin de ne plus avoir à effectuer cette démarche pour le reste des images (ce qui serait très couteux en temps). Il s’agit de sélectionner sur cette première image la zone de la couleur nous intéressant, ici l’intérieur d’un picot, puis d’en déduire le modèle colorimétrique de cette zone et enfin la distance en couleur de chaque pixel de l’image à la couleur sélectionnée (distance de Mahalanobis).

Une fois l’initialisation faite, le modèle colorimétrique reste valable pour toute les images de la vidéo ; nous pouvons alors détecter la position des picots pour chacune de ces images. Pour cela, il faut d’abord effectuer un seuillage des images de distance de Mahalanobis. Ce seuillage nous fournit alors une image noire avec pour seules zones blanches les picots. Par traitement nous obtenons les coordonnées des barycentres des picots, qui sont ensuite ordonnancés afin qu’un picot pour une image corresponde au même picot dans l’image suivante.

Une fois les positions des 4 picots connues, il est désormais possible d’incruster notre image sur la vidéo par homographie : les coins de notre image sont placés aux picots. C’est ici que l’ordonnancement des barycentres des picots rentre en jeu : il est essentiel pour que d’une image à l’autre dans la vidéo, notre image rajoutée ne change pas de sens. On agrandit ensuite l’image rajoutée afin qu’elle remplace toute la feuille (les picots ne peuvent pas être placés aux coins exacts de la feuille).

Il reste cependant encore un problème : notre incrustation a remplacé tous les pixels de la feuille par ceux de notre image ajoutée, mais elle a aussi remplacé les pixels correspondant à la main tenant la feuille. Si nous voulons que la vidéo soit réaliste, il faut faire en sorte que la main soit visible par-dessus notre image rajoutée. Pour cela, il faut d’abord détecter quels sont les pixels correspondants à la main sur une image. On utilise une méthode similaire à celle employée pour détecter les picots : on détecte la couleur de la main, on image la distance de Mahalanobis à cette couleur pour l’ensemble de l’image, puis on seuille. Les pixels non nuls restants correspondent à la main, on peut alors les prendre en compte lors de l’incrustation.

**Implémentation :**

Après avoir expliqué le principe général de chacune des étapes de l’algorithme, détaillons maintenant le processus d’implémentation de cet algorithme sous Matlab.

Au cours de la phase d’initialisation, on travaille sur la première image de la vidéo. Sur cette image, on voit la feuille avec ses 4 picots, sans la main. On commence tout d’abord par sélectionner la zone de pixels de la couleur qui nous intéresse (ici le bleu foncé). Cette sélection peut se faire manuellement ou automatiquement. La zone devient alors une image (très) réduite de notre image de départ. On applique alors à cette image une fonction permettant de déterminer la matrice M correspondant au modèle colorimétrique de notre image réduite (Cette fonction sera réutilisée par la suite pour la détection de la main).

# Résultats

Le programme final prend donc en entrée une vidéo montrant la feuille avec ses quatre picots bleus, ainsi que l’image servant pour le remplacement. Il y a également deux paramètres, le nom du fichier pour la vidéo de sortie, après remplacement ; et un paramètre ‘inception’ qui détermine le nombre de fois que le traitement doit être effectué sur une même image (l’image de remplacement devient la vidéo elle-même).

On obtient donc plusieurs vidéos possibles en sortie :

* Sortie1.avi correspond à un traitement simple. Le traitement de la présence des doigts a été optimisé, mais on observe quand même certains problèmes de couleur.
* Sortie2.avi à un traitement ‘inception=2’. Le traitement est plus long pour obtenir ce résultat, car aucune optimisation n’a été faite pour ce traitement. On pourrait améliorer grandement le temps de traitement en diminuant la résolution des premiers traitements (image intérieurs, très petites sur l’image finale)
* Sortie3.avi correspond à un traitement dans lequel la couleur de la main a été mal choisie.