



ÉCOLE
CENTRALELYON

ÉCOLE CENTRALE LYON

UE ELC
TRAITEMENT D'IMAGES
RAPPORT

Template Matching-Image Transformations

Élèves :

Lucie DRAPIER
Paul HOAREAU
Alexandra MICHON
Cécile SOUDÉ

Enseignant :

Shaifali PARASHAR

Table des matières

1	Introduction	2
2	Template Matching	2
3	Homographie	6
3.1	Partie 1 : Trouver les coordonnées des quatre coins	7
3.2	Partie 2 : Transformation du poster des Simpsons vers la publicité sur le bus	7
3.3	Partie 3 : Transformation de la publicité vers le poster des Simpsons . . .	8
4	Conclusion	9

1 Introduction

Le traitement d'images est un domaine d'étude qui trouve de multiples applications, comme l'analyse d'image dans le domaine médical, ou la détection d'un motif par une caméra de surveillance par exemple.

Nous étudions dans ce projet deux aspects du traitement d'images : dans un premier temps, on s'intéresse à la détection du profil de "Waldo" dans une image très riche en détails, couleurs et motifs.

Ce travail comporte une difficulté principale : il faut parvenir à proposer un algorithme fonctionnel, mais surtout réalisable en un temps réaliste, sans quoi on ne pourrait appliquer la méthode dans un autre domaine. Cela pose par conséquent la question de la méthode d'identification de Waldo.

Dans un second temps, nous chercherons à remplacer la publicité sur le côté d'un bus dans une photographie.

Cet objectif peut être décomposé en étapes intermédiaires : pour transposer une nouvelle image (de Simpsons) sur un emplacement donné, il faut détecter l'espace alloué pour l'image de Simpsons. Ensuite, il faut appliquer les transformations requises à l'image pour qu'elle ait la dimension choisie (homographie), puisque les effets de perspective modifient la perception de l'image. Enfin, il faut remplacer la publicité initialement présente par l'image de Simpsons.

2 Template Matching

Dans cette partie, on cherche tout d'abord à détecter la figure de Waldo non bruité [2a] dans l'image du park d'amusement [1], puis le template de Waldo bruité [2b].



FIGURE 1 – Image encombrée de divers motifs, comportant Waldo



(a) Non bruité



(b) Bruité

FIGURE 2 – Template Waldo

Pour ce faire, nous avons utilisé la technique de la somme des distances au carré (SSD en anglais) entre le template et une portion de l'image [1] à une certaine position. Cette technique nous est utile pour évaluer la similitude entre le modèle et ses emplacements possibles dans l'image. Chaque pixel du template est alors comparé à un pixel de la portion d'image correspondante. Le code associé est le suivant [3] :

```
Entrée [12]: def calculate_ssd(pos, template, img):
    x,y = pos

    mat_ext = img[y:y+template.shape[0], x:x+template.shape[1]]
    mat_dist = (mat_ext-template)**2
    val_ssd = np.sum(mat_dist)
    return val_ssd
```

FIGURE 3 – Code : calcul de la SSD entre un template une portion de l'image à une certaine position

Puis, lorsque une portion de l'image du park présente des similitudes avec le template, la valeur de la SSD est faible. Nous cherchons alors à trouver la portion de l'image qui présente la valeur SSD la plus faible possible. Le code suivant vise à trouver ce minimum dans l'image du park et donne la position trouvée du coin supérieur gauche du template dans l'image du park [4] :

Entrée [7]:

```

def find_template(template, img_mat) :
    ssdmin = np.inf
    lim_x = img_mat.shape[0] - template.shape[0]
    lim_y = img_mat.shape[1] - template.shape[1]

    for y, ligne_img in enumerate(img_mat) :
        if y < lim_x :
            for x, pix_img in enumerate(ligne_img) :
                if x < lim_y :
                    threshold = 70
                    if distance(pix_img, template[0][0]) <= threshold:
                        ssd_calculated = calculate_ssd((x,y), template, img_mat)

                    if ssdmin > ssd_calculated :
                        ssdmin = ssd_calculated
                        coo = (x,y)

    return coo

```

FIGURE 4 – Code : Fonction pour chercher un template dans une image

Pour raccourcir le temps de calcul, un seuil a été introduit : si la valeur de similitude entre le premier pixel de Waldo et celui de la portion considérée du park est suffisamment faible, on calcule le SSD pour l'ensemble de la portion. Dans le cas où ces deux premiers pixels sont déjà trop éloignés pour être des candidats de matching, on ne prend pas le temps de calculer leur SSD.

Ceci est effectué pour l'ensemble du park. A chaque fois qu'un nouveau SSD est calculé, il est comparé au précédent et conservé seulement s'il est plus faible. Ceci est réalisé afin de trouver la valeur de SSD la plus basse à l'issue de l'algorithme (position de Waldo).

Après itération sur chaque portion de l'image du park, le code entoure la position trouvée de Waldo dans l'image du park [5a] et [5b] :



(a) Non bruité



(b) Bruisé

FIGURE 5 – Recherche de Waldo dans l'image

L'image non bruitée est bien repérée par l'algorithme de détection de template dans l'image du park.

En outre, le bruit ajouté dans l'image [2b] n'empêche pas pour autant l'algorithme de détecter le template de Waldo dans l'image. En effet, le bruit modifie de manière aléatoire la couleur associée à certain pixels de l'image, et augmente ainsi la valeur finale de la SSD,

mais en comparaison avec les autres valeurs, la position de Waldo reste celle présentant le plus de similitudes (SSD minimale). On remarquera cependant que pour obtenir ce résultat dans le second cas, il a été nécessaire d'augmenter le threshold (seuil) d'admissibilité du premier pixel afin de prendre en compte le bruit (et ne pas éliminer à tort la position de Waldo). Cependant, celui-ci ne devait pas être non plus trop élevé pour ne pas allonger le temps de calcul.

3 Homographie

Dans ce problème, nous allons examiner comment utiliser les transformations homographiques pour manipuler des images. L'objectif sera ici de remplacer la publicité Sprite présente sur le bus [6] par le poster des Simpsons [7].



FIGURE 6 – Bus - Publicité Sprite



FIGURE 7 – Poster des Simpsons

Pour ce faire, nous allons tout d'abord déterminer les coordonnées des quatre coins de l'affiche du bus que nous souhaitons couvrir. Puis, nous utiliserons une matrice homographique afin de transposer le poster des Simpson sur la publicité du bus. Enfin, effectuerons la transformation précédente dans le sens inverse, c'est-à-dire que nous remplaceons chaque pixel de l'affiche avec le pixel du poster afin de voir si les résultats obtenus sont identiques.

3.1 Partie 1 : Trouver les coordonnées des quatre coins

Notre matrice d'homographie a 8 inconnues, nous avons donc besoin de trouver 8 paramètres connus, soit 4 points nous donnant ainsi 8 coordonnées. Ces quatre points sont tout simplement les quatre coins de l'affiche du bus que nous souhaitons couvrir. Pour trouver les coordonnées de ces quatre points avec une bonne précision et assez efficacement nous avons fait le choix d'utiliser le module Tkinter via une fonction que vous pouvez retrouver dans notre code.

3.2 Partie 2 : Transformation du poster des Simpson vers la publicité sur le bus

Pour transposer le poster des Simpson vers l'affiche sur le bus, il est nécessaire d'utiliser la matrice homographique entre notre première image et la portion de notre deuxième image qui nous intéresse.

Pour ce faire, nous avons choisi d'utiliser la méthode des moindres carrés en assumant que le dernier paramètre vaut 1 (ce qui est possible, car l'homographie n'a que 8 degrés de liberté). Pour trouver nos inconnues nous avons donc fait se correspondre les coins du poster dont les coordonnées sont évidentes avec les coins de l'affiche sur le bus dont nous

avons pu récupérer les coordonnées précédemment.

La résolution du système d'équations est alors assez simple avec un calcul matriciel.
On obtient finalement l'image suivante :



FIGURE 8 – Image avec les poster des Simpsons sur le bus par la méthode 1

3.3 Partie 3 : Transformation de la publicité vers le poster des Simpsons

Après avoir fait la transformation dans le sens le plus évident, nous l'avons aussi réalisée au sens inverse. C'est à dire que nous avons calculé les coordonnées homogènes de chaque pixel faisant partie de l'affiche du bus en utilisant l'inverse de la matrice utilisée précédemment. De là, nous avons remplacé chacun des pixels de l'affiche avec le pixel du poster correspondant. De cette manière, nous avons obtenu un résultat relativement identique au précédent comme le montre l'image suivante.



FIGURE 9 – Image avec les poster des Simpsons sur le bus par la méthode 2

Le code pour faire cette opération a une complexité supérieure au premier, tout simplement parce qu'il faut vérifier que chaque pixel appartient bien au parallélogramme (à l'affiche du bus). Ceci-dit, cette partie pourrait être améliorer en considérant un rectangle inscrit dans le parallélogramme dans lequel on sait que les pixels font partie de l'affiche, mais cette optimisation a un impact minime pour l'utilisation qu'on fait de ce code.

Cependant, malgré cette complexité plus élevée, le temps de calcul est inférieur à celui constaté lors de l'utilisation de la première méthode, tout simplement car le nombre de pixels parcourus est beaucoup plus faible dans ce cas ! En effet, le poster des Simpsons est composé de beaucoup plus de pixels que l'affiche sur l'image du bus, il n'est donc pas nécessaire de parcourir tous les pixels du poster. Sur l'un de nos ordinateurs, on obtient en moyenne 3.5s avec la première méthode contre 0.5s avec la seconde, un écart qui n'est pas négligeable.

Attention, ceci n'est bien sûr pas valable dans tous les cas, si l'affiche du bus avait une définition supérieure à celle du poster alors ce ne serait plus vrai. Dans ce cas il faudrait aussi remplacer les pixels manquants par interpolation...

4 Conclusion

Grâce aux deux projets décrits dans ce rapport, nous avons pu mettre en évidence plusieurs aspects essentiels du traitement d'images.

En effet, dans la partie de recherche de Waldo, nous avons étudié une méthode d'identification de motif : en particulier, nous avons utilisé la méthode des moindres carrés pour trouver la zone du poster qui avait la correspondance de couleur la plus forte avec la tête de Waldo. Lorsque la tête de Waldo était bruitée, le calcul de ssd (somme des différences au carré) renvoyait des valeurs différentes, pourtant la position finale obtenue (minimum de tous les calculs) est à nouveau celle de Waldo : cela témoigne de la robustesse de la méthode.

Dans la seconde partie, on s'est intéressés à un procédé de remplacement d'un objet dans une image par un autre.

Deux méthodes ont été explorées : la première consiste à appliquer une matrice homographique au poster des Simpsons, de sorte à donner à ce poster les bonnes dimensions pour être ensuite affiché sur le bus. La seconde approche, plus coûteuse, prend comme point de départ l'affiche à remplacer, lui applique l'inverse de la matrice homographique (pour le "remettre à plat") et enfin le fait correspondre avec le pixel associé du poster des Simpsons.

Cela a montré que plusieurs méthodes permettent d'obtenir un même résultat, mais leurs complexités doivent être prises en compte pour distinguer leur efficacité : on a ici un facteur 7 en écart de rapidité entre l'une et l'autre.