TP7 – Photomontage

Principe du photomontage par collage

Le *photomontage par collage* consiste à remplacer une zone d'une image « cible », notée c, par des données issues d'une image « source », notée s. Lancez le script `exercice_0`, qui demande à l'utilisateur de sélectionner un polygone p dans s, puis de sélectionner un rectangle r dans c (en cliquant sur deux sommets opposés). La fonction `imresize` de Matlab permet de définir la transformation affine $t : s \to c$ telle que $t(e) = r$, où e est le rectangle englobant de p (cf. figure 1). Le résultat est une image u telle que $u(x, y) = c(x, y)$ partout, sauf pour les points $(x, y) \in t(p)$, pour lesquels $u(x, y) = s(t^{-1}(x, y))$. Ce « collage » doit être effectué canal par canal. Vous constatez que le résultat n'est pas réaliste, ce qui n'est guère surprenant avec un algorithme aussi naïf.

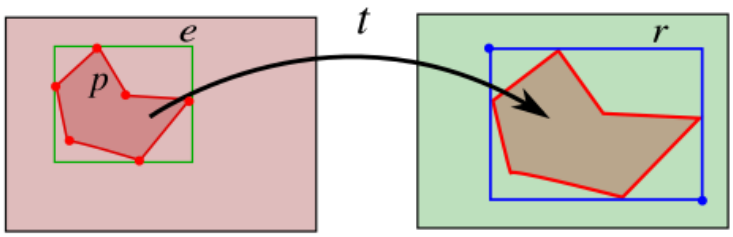


Image source s. Image cible c.

FIGURE 1 – À gauche : sélection d'un polygone p dans l'image source s. À droite : sélection d'un rectangle r dans l'image cible c. Les points cliqués par l'utilisateur sont indiqués en rouge dans s, en bleu dans c.

Plutôt que d'insérer dans c des données issues de s, mieux vaut définir le champ vectoriel $\mathbf{g}$ égal à ∇c sur $r \backslash t(p)$ et à ∇s sur $t(p)$, puis résoudre l'équation $\nabla u = \mathbf{g}$ sur r, en imposant $u = c$ sur le bord de r. Ce problème n'ayant en général aucune solution exacte, on le résout de manière approchée, au sens des moindres carrés :

$$\min_{u:\, \mathbb{R}^2 \to \mathbb{R}} \iint_{(x,y) \in r} \|\nabla u(x, y) - \mathbf{g}(x, y)\|^2 \, dx \, dy \tag{1}$$

L'équation d'Euler-Lagrange qui découle de (1) est une *équation de Poisson* :

$$\Delta u(x, y) = \nabla \cdot \mathbf{g}(x, y) \tag{2}$$

qui doit être résolue canal par canal. La discrétisation de (2) par différences finies s'écrit :

$$\mathbf{A}\, \mathbf{u}_k = \mathbf{b}_k, \qquad k \in \{\mathrm{R}, \mathrm{V}, \mathrm{B}\} \tag{3}$$

Dans (3), la matrice $\mathbf{A}$ est une version discrète de l'opérateur laplacien (cf. TP5), de taille $N \times N$, où N est le nombre de pixels contenus dans r, et les vecteurs $\mathbf{u}_k$ et $\mathbf{b}_k$ sont obtenus par concaténation, respectivement, des valeurs de u (inconnues) et des valeurs de $\nabla \cdot \mathbf{g}$ (connues) pour l'ensemble des pixels de r, dans le canal k.

Exercice 1 : photomontage par collage

Faites une copie du script `exercice_0`, de nom `exercice_1`, que vous modifierez de manière à remplacer l'appel à la fonction `collage_naif` par un appel à la fonction `collage`, que vous devez écrire. Ce script vous permettra, par exemple, d'incruster la randonneuse du TP5 dans un tableau de Van Gogh (cf. figure 2), ou de réaliser d'autres photomontages avec les images de votre choix.

Image source s.

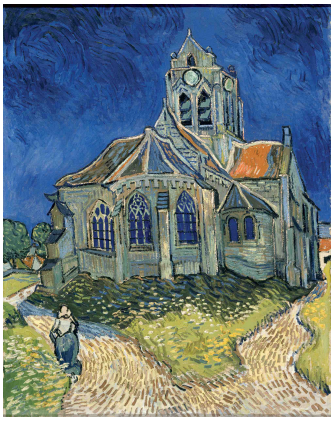
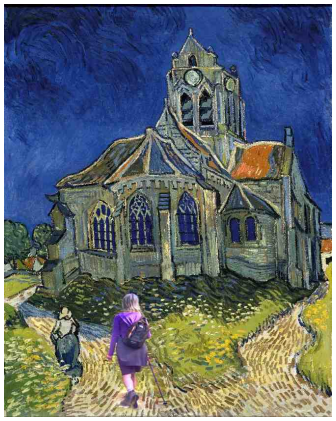


Image cible c.



Résultat du photomontage.

FIGURE 2 – Exemple de photomontage.

Pour l'écriture de la fonction `collage`, censée mettre en œuvre cette technique de photomontage, il est fortement recommandé de procéder comme suit :

- Convertissez les paramètres d'entrée `r` et `s` au format double.
- Pour le calcul de la matrice $\mathbf{A}$, reportez-vous au TP5.
- Pour imposer la condition au bord $u = c$, vous devez modifier les lignes de la matrice $\mathbf{A}$ correspondant aux pixels du bord de r. Ces lignes ne doivent comporter que des éléments nuls, à l'exception d'un élément égal à 1. Cela peut être réalisé grâce à la fonction `sparse` de Matlab (`doc sparse`), en utilisant la syntaxe suivante, où `indices_bord_r` contient la liste des indices des pixels du bord de r, et où `n_bord_r` et `n_r` désignent, respectivement, le nombre de pixels du bord de r et le nombre de pixels de r :

  ```
  A(indices_bord_r,:) = sparse(1:n_bord_r,indices_bord_r,ones(n_bord_r,1),n_bord_r,n_r);
  ```

- N'oubliez pas de modifier également les lignes du vecteur $\mathbf{b}_k$ correspondant aux pixels du bord de r.
- Enfin, après résolution de l'équation (3), n'oubliez pas de redimensionner $\mathbf{u}_k$.

Remarques :

- Dorénavant, contrairement à l'algorithme naïf qui ne modifie que les points $(x, y) \in t(p)$, tous les points $(x, y) \in r$ sont modifiés. L'intérêt d'utiliser un domaine r de forme rectangulaire est que cela donne à la matrice $\mathbf{A}$ une structure très régulière (cf. TP5).
- Une astuce permettant d'affecter à la variable `indices_bord_r` la liste des indices des pixels du bord de r consiste à créer une matrice `bord_r` constituée de 1, ayant le même nombre de lignes et le même nombre de colonnes que r, puis à mettre à 0 les éléments de cette matrice non situés sur le bord.
- Attention : il serait faux de remplacer le laplacien de c par le laplacien de s à l'intérieur de $t(p)$; il faut remplacer le gradient de c par le gradient de s à l'intérieur de $t(p)$, puis calculer la divergence $\nabla \cdot \mathbf{g}$ du champ vectoriel $\mathbf{g}$ ainsi formé.

Exercice 2 : décoloration partielle d'une image

La fonction `rgb2gray` de Matlab permet de transformer une image couleur en image en niveaux de gris. Cette transformation est également réalisable en utilisant le format LAB, dont le premier canal est celui de la *luminance*. Les fonctions `rgb2lab` et `lab2rgb` de Matlab permettent de passer du format RGB au format LAB.

L'exemple de la figure 3 illustre la *décoloration partielle d'une image*, où l'image originale constitue la source s, et où le canal de luminance de cette même image constitue la cible c. Ce qui est remarquable, avec cette autre technique de photomontage, c'est qu'elle ne nécessite pas de segmentation préalable de la rose. Il suffit que le polygone p, qui est affiché sur l'image de gauche de la figure 3, détoure très grossièrement la rose sur l'image originale. Mais n'allez pas croire que cette technique effectue, comme par magie, la segmentation de la rose !

Image originale. Image résultat.

FIGURE 3 – Exemple de décoloration partielle d'une image sans segmentation préalable.

Faites une copie du script `exercice_1`, de nom `exercice_2`, que vous modifierez de manière à initialiser la source s et la cible c par l'image `rose.jpg` convertie au format LAB, puis à annuler les deuxième et troisième canaux de l'image c. La fonction `collage` reste inchangée. Là encore, une fois mis au point, testez le script `exercice_2` sur d'autres images que `rose.jpg`.

Exercice 3 : autres techniques de photomontage (facultatif)

D'autres techniques de photomontage sont décrites dans un article de Pérez, Gangnet et Blake datant de 2003, dont ce TP s'inspire en grande partie, et qui vous est fourni dans les données. En guise d'exercice facultatif, il vous est proposé de lire cet article, d'identifier une application autre que celles des exercices 1 et 2, et d'implémenter cette application sous la forme d'un script de nom `exercice_3`. L'application illustrée sur la figure 8 de cet article semble particulièrement intéressante, mais vous avez le champ libre !