Equation de Poisson et traitement d'image

Groupe de travail thématique

4TQMS801S

Equation de Poisson et traitement d'image

Virginie MONTALIBET Sébastien Eyzat

March 6, 2020

Contents

1 Présentation du problème

Le traitement d'images est un ensemble de méthodes permettant d'étudier et de transformer une ou plusieurs images à l'aide de moyens mathématiques et numériques. Le principe du traitement d'images consiste à extraire certaines informations de celles-ci, afin de les étudier ou de les modifier.Il est utilisé dans beaucoup d'applications telles que l'amélioration du contraste, l'application d'un filtre(flou, lissage, changement de couleurs), ou encore les détections et identifications d'objets par exemple.

Aujourd'hui, nous nous intéressons à l'incrustation d'images. A partir de deux images, comment sélectionner une partie de la première et l'incruster de la manière la plus naturelle possible dans la seconde ?
Afin d'éclaircir nos propos et d'identifier les problèmes que nous devons résoudre, voici un exemple de ce que nous souhaitons faire.
Nous disposons des deux images présentées ci-dessous, l'image T(arget) et l'image S(ource).

Figure 1: Image T

Figure 2: Image S

L'objectif est d'incruster toute ou partie de l'image S dans l'image T. En terme de manipulations, cela consiste à effectuer un copier/coller ou encore un clônage de la seconde image dans la première. Le résultat que nous attendons pour une incrustation "réussie" est un résultat comme celui présenté ci-dessous :

Figure 3: Image finale attendue

Cette image est extraite d'une simulation effectuée sur 'ipol.im/...'.Nous comparerons nos images obtenues avec celle-ci à la fin de ce rapport.
Ce résultat semble naturel, les marques et bords de l'image qui a été collée sont très peu visibles, les oiseaux semblent faire partie de l'image finale. Mais comment obtenir un tel résultat ?
Reprenons nos deux images séparées, T et S. Commençons par effectuer un simple copier/coller. Pour faire cela, il suffit de prendre une partie de l'image S, puis de la transférer à l'endroit voulu dans l'image T. En effectuant cette manipulation voici l'image que nous devrions obtenir :

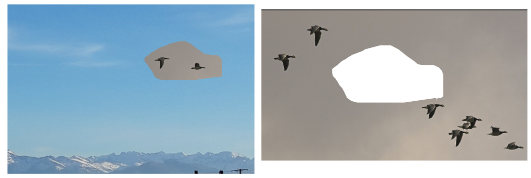


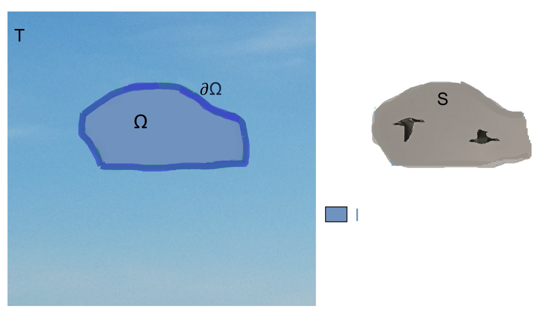
Figure 4: Simple copier/coller

Ce résultat n'est bien entendu pas convenable et bien loin de l'image finale attendue. Le découpage et la démarcation entre image originale et image collée sont beaucoup trop visibles, les couleurs ne sont pas les mêmes et incohérentes. Nous souhaitons obtenir un résultat beaucoup plus naturel, comme écrit plus haut. Cette simple manipulation n'est donc pas suffisante pour effectuer le clônage cohérent d'une image dans une autre.

Il semble donc évident de vouloir modifier l'image finale obtenue afin qu'elle paraisse la plus naturelle possible. Nous verrons tout au long de ce rapport, quels sont les changements a effectuer, et comment la résolution d'une équation aux dérivées partielles, nous permet d'obtenir un bien meilleur résultat.

1.1 Traduction du problème sous formes mathématiques.

Comme écrit plus haut nous souhaitons apporter des modifications au précédent collage afin qu'il corresponde au mieux à nos attentes. Pour ce faire, considérons I, la partie de l'image finale à modifier. Nous pouvons ainsi représenter le problème sous forme schématique.



Ici :

- T est l'image "Target", l'image destination, l'image sur laquelle s'effectuera le collage, l'arrière plan.
- S est l'image "Source", l'image que nous souhaitons coller.
- Ω est le domaine dans lequel se trouvent nos inconnues.
- $\partial\Omega$ est la frontière de Ω.
- I est notre inconnue, la partie de l'image que nous ne connaissons pas et que nous voulons remplir.

Nous souhaitons en effet, trouver une fonction I qui satisfasse un certain nombre de critères, afin de correspondre au résultat voulu. IL faut donc déterminer la fonction I.Cette fonction représente notre image modifiée. Mais quelles sont les conditions qu'elle doit remplir pour que le rendu soit le meilleur possible? Notre nouvelle image I doit-elle être plus proche de l'image collée S, ou de l'image d'arrière-plan T ?
Pour répondre à cette question, reprenons le problème de départ.

Pour que l'image obtenue paraisse naturelle,il faut que l'image obtenue dénature le moins possible les deux images sélectionnées au départ. En effet, nous devons garder les détails de l'image collée S, ne pas modifier les variations qu'elle pourrait posséder comme les contours des objets lui appartenant, par exemple.Nous devons pouvoir retrouver les informations présentes dans celle-ci. Il faut donc que les variations présentes dans I soient presque identiques à celles de S. Mathématiquement, cela est équivalent à dire que les gradients de I et S soient très proches, voir identiques. Donc à résoudre cette équation :

$$min \iint_{\Omega} ||\nabla I_{x,y} - \nabla S_{x,y}||^2 dxdy$$

Mais les démarcations entre l'image collée et l'image d'arrière-plan T, ne doivent pas non plus être visibles, il faut donc que les pixels se situant sur cette partie là, i.e $\partial\Omega$, soient le plus proches possible de T . Nous voulons donc, mathématiquement, chercher :

$$I_{(x,y)} = T_{x,y} \ sur \ \partial\Omega$$

Répondre au problème implique donc de résoudre un problème variationnel classique auquel des conditions sur le bord de Dirichlet sont ajoutées. Réécrivons le problème mathématique que nous cherchons à résoudre :

$$\begin{cases} min \iint_{\Omega} ||\nabla I_{x,y} - \nabla S_{x,y}||^2 dxdy \\ \qquad\qquad I_{(x,y)} = T_{x,y} \ sur \ \partial\Omega \end{cases}$$

1.2 Résolution dans le cas discret

Notons $g(\nabla f, v)$ la fonction :

$$g(\nabla f, v) = \int ||\nabla f - v||^2$$

Rappelons qu'ici ∇f est le gradient de f que l'on peut noter $\nabla f = (\frac{\partial f}{\partial x} \frac{\partial f}{\partial y})^T$ et v est un vecteur que l'on notera $v = (v_x, v_y)$. Ainsi nous pouvons réecrire g sous la forme suivante :

$$g(\nabla f, v) = \int ((\frac{\partial f}{\partial x} - v_x)^2 + (\frac{\partial f}{\partial y} - v_y)^2)$$

Mais la fonction f qui minimise le problème posé, satisfait l'équation d'Euler-Lagrange ci-dessous :

$$\frac{\partial g}{\partial f} - \frac{d}{dx}\frac{\partial g}{\partial f} - \frac{d}{dy}\frac{\partial g}{\partial f} = 0$$

$$\frac{\partial g}{\partial f} - \frac{d}{dx}\frac{\partial g}{\partial f} - \frac{d}{dy}\frac{\partial g}{\partial f} = 0$$

$$2 \times ()$$

EXPLICATIONS
NEEDED

$$2(\frac{\partial^2 f}{\partial x^2} - \frac{\partial v_x}{\partial x} + \frac{\partial^2 f}{\partial y^2}\frac{\partial v_y}{\partial y}) = 0?$$

$$\Delta f = div v \tag{1}$$

Résoudre ce problème variationnel est équivalent à résoudre l'équation de poisson avec conditions aux bords de Dirichlet, suivante :

$$\begin{cases} \Delta I = \Delta S \; sur \; \Omega \\ I = T \; sur \; \partial\Omega \end{cases}$$

Nous verrons deux approches afin de résoudre cette équation, la première utilisant des discrétisations, la seconde la méthode de Fourier.

2 Résolution

Nous verrons dans un premier temps, comment résoudre cette équation à l'aide de discrétisation et de différences finies. Puis nous résoudrons celle-ci à l'aide des transformées de Fourier. Rappelons que nous souhaitons résoudre l'équation de Poisson suivante :

$$\begin{cases} \Delta I = \Delta S \; sur \; \Omega \\ I = T \; sur \; \partial\Omega \end{cases}$$

2.1 Discrétisation

Avant de discrétiser les Laplaciens de nos deux images, il faut définir le maillage que nous allons utiliser.

2.1.1 Qu'est qu'une image

Une image peut être représentée comme une succession de pixels. En traitement d'image, ce sont d'ailleurs sur ces pixels que le traitement est effectué. Leur modification entraîne la modification de l'image globale.Nous verrons donc une image comme la succession de pixels et nous pouvons la représenter comme une grille, dans laquelle chaque carré représente un pixel. Schématiquement, nous pourrions donc découper notre image comme ci-dessous Nous

Figure 5: Maillage d'une image

numéroterons les pixels suivant la règle suivante. Le premier pixel est situé en haut en gauche, puis il suffit de parcourir la grille comme ci-dessous :

Les pas d'espaces sont donc égaux et valent 1. Dans la suite nous considérons que notre image est de taille $N \times M$.

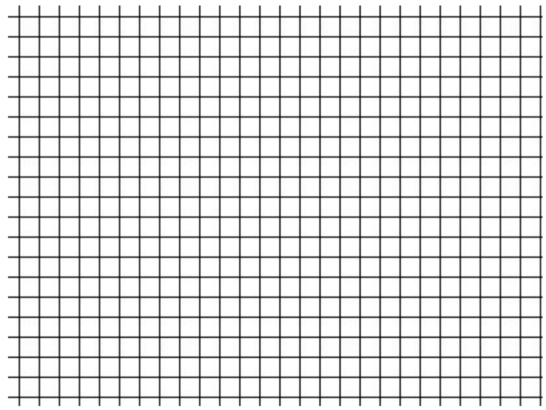


Figure 6: Parcours d'une grille

2.2 Méthode des différences finies

Nous cherchons à résoudre :

$$\begin{cases} \Delta I = \Delta S \ sur \ \Omega \\ I = T \ sur \ \partial\Omega \end{cases}$$

En utilisant la méthode des différences finie, commençons par discrétiser le Laplacien. Rappelons qu'il peut s'écrire de la forme :

$$\Delta I(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Pour discrétiser celui-ci, il faut donc commencer par discrétiser $\frac{\partial^2 I}{\partial x^2}$ et $\frac{\partial^2 I}{\partial y^2}$.

Discrétisation des dérivées secondes : A l'aide des formules de Taylor à l'ordre 2 ci-dessous :

$$I(x+h,y) = I(x,y) + h \times \frac{\partial I(x,y)}{\partial x} + h^2 \times \frac{\partial^2 I(x,y)}{\partial x^2} + o(h^3)$$
$$I(x-h,y) = I(x,y) - h \times \frac{\partial I(x,y)}{\partial x} + h^2 \times \frac{\partial^2 I(x,y)}{\partial x^2} + o(h^3)$$

Et en effectuant la somme de ces deux équations, nous obtenons une discrétisation possible de $\frac{\partial^2 I(x,y)}{\partial x^2}$:

$$\frac{\partial^2 I(x,y)}{\partial x^2} = \frac{1}{h^2} \times I(x+h,y) + I(x-h,y) - 2 \times I(x,y)$$

2.2.1 Discrétisation du Laplacien

Comme le Laplacien n'est autre que la somme de $\frac{\partial^2 I(x,y)}{\partial x^2}$ et $\frac{\partial^2 I(x,y)}{\partial y^2}$, une discrétisation possible de celui-ci s'écrit de la forme :

$$\Delta I(x,y) = \frac{I(x+h,y) + I(x-h,y) - 2 \times I(x,y)}{h^2} + \frac{I(x,y+k) + I(x,y-k) - 2 \times I(x,y)}{k^2}$$

Les pas d'espaces h et k étant égaux à 1, nous pouvons écrire une discrétisation du Laplacien :

$$\Delta I(x,y) = I(x+1,y) + I(x-1,y) + I(x,y+1) + I(x,y-1) - 4 \times I(x,y)$$

Application à une image Afin de calculer le Laplacien du pixel I(i,j),il est donc nécessaire d'avoir la connaissance de ses pixels voisins que nous nommerons par la suite U(p), D(own), L(eft), R(ight) pour les pixels I(i-1,j), I(i+1,j), I(i,j-1), I(i,j+1).

	1	
1	-4	1
	1	

2.2.2 Résolution de l'équation de Poisson

Résoudre $\Delta I(x,y) = \Delta S(x,y)$ sur Ω est équivalent à résoudre : Soit $g(x,y) = S(x+1,y) + S(x-1,y) + S(x,y+1) + S(x,y-1) - 4 \times S(x,y)$

$$\begin{cases} I(x+1,y) + I(x-1,y) + I(x,y+1) + I(x,y-1) - 4 \times I(x,y) = g(x,y) \\ \hfill sur\ \Omega \\ \hfill I = T\ sur\ \partial\Omega \end{cases}$$

Nous devons donc résoudre un système à $n \times m$ inconnues.

Afin de résoudre ce système, il est plus facile de l'écrire sous forme matricielle. Nous devons donc résoudre un système de la forme AI = b et sa solution est : $I = A^{-1} \times b$. Avec :

- A une matrice carrée de taille (mn, mn)
- I un vecteur colonne de taille (mn,1)
- b un vecteur colonne de taille (mn,1)

Transformation de l'image Afin de pouvoir résoudre ce système, I doit être un vecteur colonne. Or dans notre cas, I est une matrice de taille (m,n). Le nouveau vecteur I, est la concaténation des colonnes de l'image I, comme ci dessous.

Ecriture de la matrice du système

Ecriture de vecteur Après avoir calculé le Laplacien de l'image S, le vecteur b est obtenu en concaténant les colonnes de l'image ΔS.

Exemple simple

2.3 Fourier

3 Implémentation

3.1 Présentation de l'interface

Création d'une interface permettante de (1) choisir l'image Source qui sera coupée puis collée. (2) Choisir l'image Target qui sera l'image de fond. En cliquant sur l'une ou l'autre des images, il est possible de sélectionner une zone de l'image, afin de la récupérer. Des masques s'affichent alors dan (3) ou (4). Indique que la sélection s'est bien faite. Enfin en cliquant sur "Paste", l'image finale apparait dans (5). Les fonctions :
La fonction createmask Elle prend en paramètre (L'image sur laquelle le découpage va s'effectuer. Elle commence par utiliser imfreehand ce qui permet de dessiner et sélectionner une zone de l'image. La région d'intérêt "roi1". Une fois cette zone récupérée elle créée un masque "mask1" dans lequel tous les pixels situés à l'intérieur de "roi1", sont marqués comme true. ON récupère la position de la région d'intéret et la taille du masque. On modifie le masque afin de pouvoir trvaailler avec, en réalité on remplace juste les "0s" et "1s" par des int (0 ou 1). Ensuite pour afficher uniquement la sélection on multiplie le masque obtenu par l'image de départ. AInsi on obtient uniquement la zone sélectionnée et tout le reste de l'image vaut 0.
La fonction clonagev1. Elle prend en paramètre maskS, PosT et PosS et l'image source2. L'objectif ici est de coller les deux images l'une sur l'autre. ON récupère le masque qui a été créée sur l'image Source, la position de la région d'intérêt sur l'image source, la position où l'on doit coller le masque dans l'image T. ET l'image T. On veut pouvoir coller l'image 1 sur l'image 2. POur ça on veut additionner les deux masques obtenus. Le premier masque est celui de l'image source, le second masque et celui de l'image Target qui sont complémentaires. Pour pouvoir additionner les deux masques il faut qu'ils soient de la même taille. ON commence donc par ajuster la taille des images en agrandissant l'image la plus petite des 2. Puis on Bouge la zone qui nous intéresse à la position voulue dans l'image T. On créée le second masque lié à l'image T qui doit être l'inverse de celui crée pour l'image S. Et on additionne les deux masques pour obtenir l'image finale.
La fonction moveroi permet de bouger la région d'intérêt à la position voulue. On calcule la distance entre les deux points voulus et on circshift.
La fonction adjustsize Permet d'agrandir le masque le plus petit afin qu'il soit de la même taille que la plus grande des deux images.
La fonction invertmask inverse le masque tous les 0 deviennent 1 et inversement.

4 Résultats obtenus

5 Comparaison

6 Optimisation

6.1 Méthode de Douglas

7 Conclusion