

# Projet SI241

## Suppression de grillage sur des photos de zoo

Vincent ROULET & Vincent BODIN

---

### Résumé

Ce projet s'attelle à la tâche de suppression de grillage dans une photo de zoo par exemple. Les grillages sont souvent composé d'une structure particulière, tant au niveau de la forme géométrique que de la colorimétrie. Cette structure sera utilisée pour détecter le grillage au sein d'une photo. En seconde partie nous nous attacherons à reconstruire l'image masquée - *i.e.* où le grillage a été supprimé.

## Table des matières

<b>1</b>	<b>Hypothèses</b>	<b>3</b>
<b>2</b>	<b>Détection de contours</b>	<b>3</b>
<b>3</b>	<b>Extraction de lignes à partir des contours</b>	<b>3</b>
<b>4</b>	<b>Identification d'un grillage</b>	<b>3</b>
<b>5</b>	<b>Reconstruction de l'image</b>	<b>3</b>
5.1	Inpainting par régularisation variationnelle . . . . .	3
5.1.1	Inpainting avec norme Sobolev . . . . .	3
5.1.2	Inpainting avec norme TV . . . . .	4
5.2	Inpainting par régularisation parcimonieuse . . . . .	4

## Introduction

### 1 Hypothèses

### 2 Détection de contours

### 3 Extraction de lignes à partir des contours

### 4 Identification d'un grillage

### 5 Reconstruction de l'image

Une fois le masque appliqué à l'image, nous sommes ramené à un problème dit d'*inpainting*. Il s'agit alors de trouver une méthode permettant de combler les trous formés par le masquage en respectant la structure de l'image. Il existe de nombreuses méthodes pour cela, nous en avons implémenté quelques unes.

#### 5.1 Inpainting par régularisation variationnelle

##### 5.1.1 Inpainting avec norme Sobolev

La résolution du problème d'inpainting repose sur un problème de minimisation d'énergie - ici norme de Sobolev - sous des contraintes de correspondance. Ainsi si on note  $\Phi$  l'opérateur de masquage de l'image - avec le masque extrait - et  $y$  l'image masquée, l'*inpainting* par Sobolev se récrit :

$$f^* = \arg \min E(f) = \|\nabla f\|^2 \text{ sous contraintes } \Phi(f) = y \quad (1)$$

Cette fonction à minimiser est régulière et l'on peut utiliser un algorithme de descente de gradient projeté. Si l'on note -  $\Omega$  est le masque :

$$(\Pi f)_i = \begin{cases} y_i & \text{si } i \in \Omega \\ f_i & \text{si } i \notin \Omega \end{cases} \quad (2)$$

l'opérateur de projection orthogonal sur la contrainte  $\Phi f = y$ , alors l'algorithme de descente de gradient projeté consiste à itérer :

$$f^{(n+1)} = \Pi \left( f^{(n)} + \tau \Delta(f^{(n)}) \right) \quad (3)$$

où  $\Delta = -\nabla^* \circ \nabla$  (l'adjoint du gradient étant -div. Le terme de mise à jour est donc bien le terme classique de gradient dans la descente de gradient - ici c'est le gradient de l'énergie de Sobolev  $E(f)$ ). La condition de convergence de la méthode de descente de gradient projetée est que le pas de mise à jour vérifie :

$$\tau < \frac{2}{\|\Delta\|} \quad (4)$$

Les résultats de l'*inpainting* avec norme de Sobolev sont présenté en figure (A FAIRE). INTERPRETATION.

### 5.1.2 Inpainting avec norme TV

On peut aussi remplacer dans le problème d'optimisation précédent la norme de Sobolev par la norme de variation totale (TV). Nous verrons qu'elle a tendance à mieux reconstruire les contours que la norme Sobolev qui floute les images. Pour autant elle n'est pas différentiable en 0, on lui préfère donc sa version régularisée :

$$J_\epsilon(f) = \sum_x \sqrt{\|\nabla f(x)\|^2 + \epsilon^2} \quad (5)$$

De nouveau on peut réutiliser l'algorithme de descente de gradient projeté, il suffit donc de calculer le gradient de cette nouvelle énergie. Il vaut :

$$G_\epsilon = -\text{div} \left( \frac{(\nabla f)_i}{\sqrt{\|(\nabla f)_i\|^2 + \epsilon^2}} \right)_i \quad (6)$$

La contrainte à respecter sur le pas est cette fois-ci :

$$\tau < \frac{\epsilon}{4} \quad (7)$$

(RESULTATS + INTERPRETATION).

## 5.2 Inpainting par régularisation parcimonieuse

Cette méthode d'*inpainting* diffère de la première en ce sens qu'elle passe par une représentation parcimonieuse de l'image avant d'essayer de combler les trous. La représentation considérée est engendrée par une base de vecteurs  $\Psi = (\psi_m)_m$ , usuellement une base d'ondelette. Les coefficients dans cette base sont noté  $a = (a_m)_m$ , on par du principe que la représentation est parcimonieuse - i.e. les  $(a_m)_m$  le sont. On cherche un ensemble de coefficients  $a^*$  qui résout :

$$a^* \in \arg \min_a \frac{1}{2} \|y - \Phi \Psi a\|^2 + \lambda J(a) \quad (8)$$

Le paramètre  $\lambda$  correspond à de la régularisation pour le bruit, comme nos images ne sont pas bruitées, nous prenons un  $\lambda$  très faible. La notation  $\Psi a = \sum_m a_m \psi_m$  indique uniquement l'opération de reconstruction du signal. Souvent on prend  $J(a)$  comme la norme  $L^1$  :  $J(a) = \sum_m \|a_m\|$ .

On note  $S^\Psi(a)$  la fonction de seuillage - *soft* ou *hard* - dans la base  $\Psi$ . On peut prendre plusieurs bases d'ondelettes, nous présenterons des résultats pour une base orthogonale et une autre invariante par translation. L'algorithme utilisé pour résoudre le problème de minimisation précédent est le *forward-backward splitting* - aussi connu comme *iterative thresholding*. Il s'agit d'itérer :

$$f^{(n+1)} = S_{\tau, \lambda}^\Psi(f^{(n)} - \tau \Phi^*(\Phi f^{(n)} - y)) \quad (9)$$

L'avantage de prendre des ondelettes orthogonales est que  $\Phi^* = \Phi$ . Dans ce cas, pour qu'il y ait convergence le pas doit vérifier :

$$\tau < \frac{2}{\|\Phi^* \Phi\|} = 2 \quad (10)$$

Avec  $\tau = 1$ , et des ondelettes orthogonales toujours, l'étape de descente de gradient revient à projeter sur la contrainte  $\mathcal{C} = \{f | \forall \Omega(x) = 1, f(x) = y(x)\}$ . Ainsi dans le cas d'ondelettes orthogonales, l'algorithme revient à :

$$f^{(n+1)} = S_\lambda^\Psi(\text{Proj}_{\mathcal{C}}(f^{(n)})) \quad (11)$$

## Conclusion