

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

1	Hypothèses	3
2	Détection de contours et extraction de lignes	3
2.1	Détection de contours	3
2.2	Extraction de lignes - transformation de Hough	5
3	Identification d'un grillage	5
3.1	Reconnaissance du grillage dans la transformée de Hough	5
3.1.1	Hypothèses	5
3.1.2	RANSAC dans la transformée de Hough	5
3.2	Validation du grillage obtenu	8
3.3	Raffinement du grillage	8
3.3.1	Utilisation des couleurs de l'image	9
3.4	Utilisation de la périodicité du grillage dans la transformée de Hough	10
4	Reconstruction de l'image	10
4.1	Inpainting par régularisation variationnelle	11
4.1.1	Inpainting avec norme Sobolev	11
4.1.2	Inpainting avec norme TV	12
4.2	Inpainting par régularisation parcimonieuse	12

Introduction

1 Hypothèses

La complexité de supprimer des grillages dans des photos réside notamment dans la variété des grillages existant, dont découle la difficulté de leur détection. Nous commençons donc par présenter les hypothèses que nous avons formulées pour diriger la construction de notre algorithme. Nous testerons en conclusion notre algorithme à l'aune des limites de ces hypothèses

Grillage non flou Si la photo est focalisée sur l'animal rendant le grillage flou, la détection de contours menant au grillage est perdue. De manière générale la détection du grillage nous semble très difficile dans ce cas étant donné la perte d'information due au flou.

Photo prise approximativement face au grillage Si la photo est prise de biais, les lignes du grillage ne seront plus espacées périodiquement sur la photo et leurs angles varieront largement. Ce phénomène met en difficulté la caractérisation du grillage par sa périodicité spatiale, bien qu'on puisse la retrouver en connaissant la transformation homographique liée à la prise de vue. Comme de nombreuses photos ne sont pas prises parfaitement face au grillage, nous avons tenté de prendre en compte cette déformation géométrique

Grillage sans rupture de direction ou double grillage Si le grillage a une rupture de direction et se trouve donc dans deux plans différents ou si deux grillages sont présents sur l'image, notre algorithme ne détectera qu'un grillage. En effet pour palier la détection de faux grillages nous nous sommes restreints à détecter le grillage le plus probable dans l'image.

Grillage total Si le grillage est partiel, c'est à dire qu'il n'est que sur une partie de l'image, le masque obtenu pour l'inpainting étendra les lignes du grillage. L'idée est d'éliminer plus que nécessaire de manière à éviter de garder des morceaux de grillage non détectés.

Grillage sans croisillons Si le grillage est fait de croisillons, les lignes de ce dernier seront décalées à chaque noeud, complexifiant donc la détection de ces dernières en tant que lignes traversant toute l'image.

Nombre minimal de barreaux Nous supposons que la photo contient au moins 2 barreaux dans chaque direction. En effet le grillage perd toute caractéristique de périodicité sinon.

2 Détection de contours et extraction de lignes

2.1 Détection de contours

La première étape de notre processus de reconnaissance de grillage est de détecter les contours d'une image. En effet, les grillages provoquent naturellement une notion de contours dans l'image de par leur superposition sur le paysage.

Il existe de nombreuses manières d'obtenir un détecteur de contours, nous avons opté pour la méthode de Canny. Celle-ci a l'avantage de procéder par seuillage par hysteresis ce qui permet d'obtenir de bons résultats - au prix d'une recherche des paramètres optimaux. A COMPLETER.

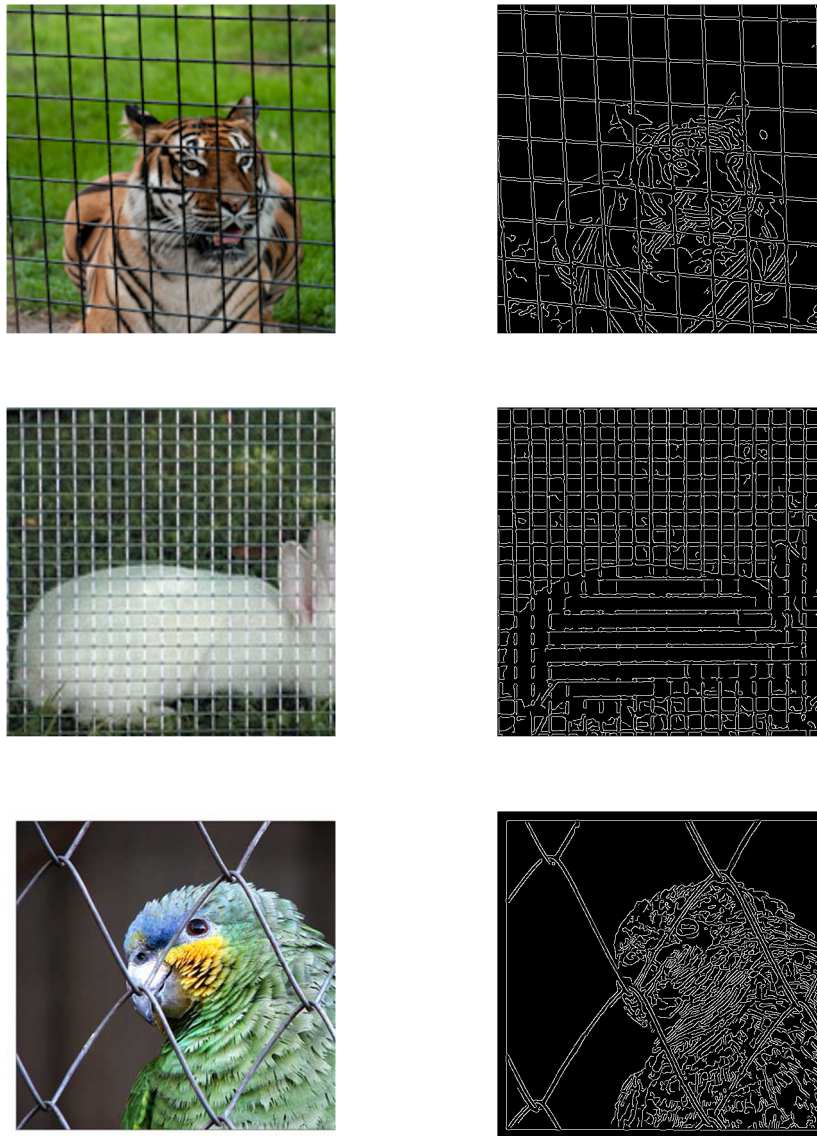


FIGURE 1 – Dans les colonnes de gauche, l'image initiale et dans celles de droite les résultats de la détection de contour avec l'algorithme de Canny. Toutes les images sont redimensionnées en 512×512 .

2.2 Extraction de lignes - transformation de Hough

A ECRIRE.

3 Identification d'un grillage

Si un grillage est un ensemble de lignes, il possède d'autres propriétés particulières qui aident à sa reconnaissance. Sa périodicité spatiale ainsi que l'angle formé entre les deux directions du grillage en font partie. Nous nous sommes donc tout d'abord intéressés à la reconnaissance de ces éléments dans la transformée de Hough.

3.1 Reconnaissance du grillage dans la transformée de Hough

3.1.1 Hypothèses

Tout d'abord on peut supposer que les deux directions du grillage sont approximativement perpendiculaires (à $90^\circ \pm 30^\circ$). Deux paquets de piques doivent donc être visibles avec cet écart d'angle sur la transformée de Hough.

En outre supposons le grillage pris de face comme en figure 4, ses lignes détectées auparavant possèdent alors la particularité d'avoir toutes le même angle θ et de varier périodiquement en ρ comme on le voit sur sa transformée de Hough en figure 4. Si le grillage est observé de biais les angles des lignes varient comme on peut le voir sur l'image d'exemple 4. Leur représentation dans la transformée de Hough en figure 4 n'est plus une droite verticale ni même une droite et les valeurs de ρ correspondantes ne sont plus périodiques.

Nous supposons que le grillage n'est pas pris trop de biais comme présenté avant, on peut alors espérer détecter des lignes dans Hough, ou plus précisément deux lignes dont les abscisses (l'angle moyen des lignes détectées dans l'image) sont séparées d'environ 90° .

3.1.2 RANSAC dans la transformée de Hough

Tout d'abord nous centrons la transformée de Hough de manière à ce que les lignes que l'on désire y détecter ne soient pas coupées. En effet la transformée de Hough détecte les lignes dans l'image dont l'angle est compris entre -90° et 90° . Si le grillage comprend une direction horizontale déformée par la prise de vue, les lignes détectées selon cette direction peuvent avoir des angles à la fois légèrement supérieurs à -90° et légèrement inférieurs à 90° . Or elles forment ensemble un potentiel grillage que l'on aimerait détecter dans la transformée de Hough.

Concrètement nous réalisons une permutation de la transformée de Hough selon l'axe des abscisses jusqu'à ce que la zone coupée (au niveau de l'abscisse 0 et de l'abscisse maximale) contienne un minimum de piques de la transformée de Hough. Ce centrage est illustré en figure 2.

Nous avons observé dans la transformée de Hough des artéfacts au niveau en abscisse des angles $\theta = -90^\circ, -45^\circ, 45^\circ, 90^\circ$. Il s'agit en quelque sorte d'un décalage de la transformée de Hough à ces angles menant à la détection de faux piques à leurs niveaux. Nous éliminons donc tous les piques pouvant être détec-

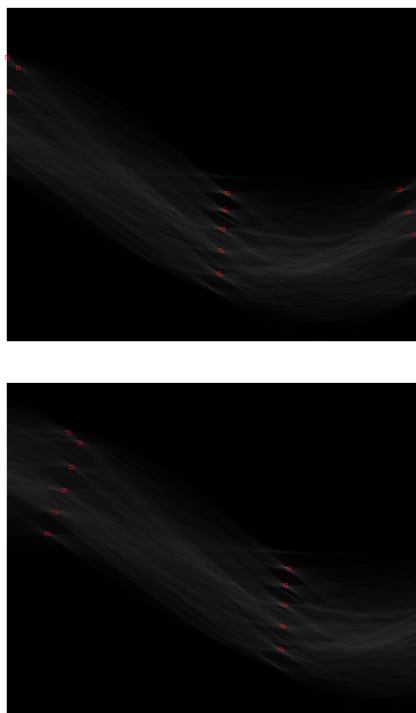


FIGURE 2 – Centrage de la transformée de Hough, à gauche la transformée de Hough non centrée, à droite la transformée de Hough centrée.
La ligne coupée dans la transformée de Hough ne l'est plus après centrage et peut être détectée.

tés à ces abscisses avant de procéder à la suite.

Nous avons tout d'abord tenté de reconnaître des lignes dans la transformée de Hough par une seconde transformée de Hough sur la transformée de Hough. Cependant cette méthode était trop rigide car elle ne permettait pas de reconnaître le grillage si celui-ci était pris trop en biais et ne s'illustrait donc pas tout à fait comme une droite sur l'image.

Nous avons donc affiné notre méthode en essayant de reconnaître ces droites à l'aide d'un algorithme RANSAC appliqué sur des fenêtres de la transformée de Hough. RANSAC a en effet l'avantage d'offrir plus de souplesse qu'une transformée de Hough sur la transformée de Hough tout en gardant l'idée de détecter des lignes et non de chercher un simple agglomérat de piques dans la transformée de Hough. Cette amélioration est illustrée en figure ??

Concrètement nous prenons deux larges fenêtres dans la transformée de Hough dont la largeur horizontale est de 30° (couvrant ainsi les lignes de l'image dont les directions sont semblables à 30° près) et la largeur verticale est celle de la transformée de Hough. Ces fenêtres sont espacées de 90° illustrant la détection des deux directions du grillage et non de simples lignes parallèles dans l'image.

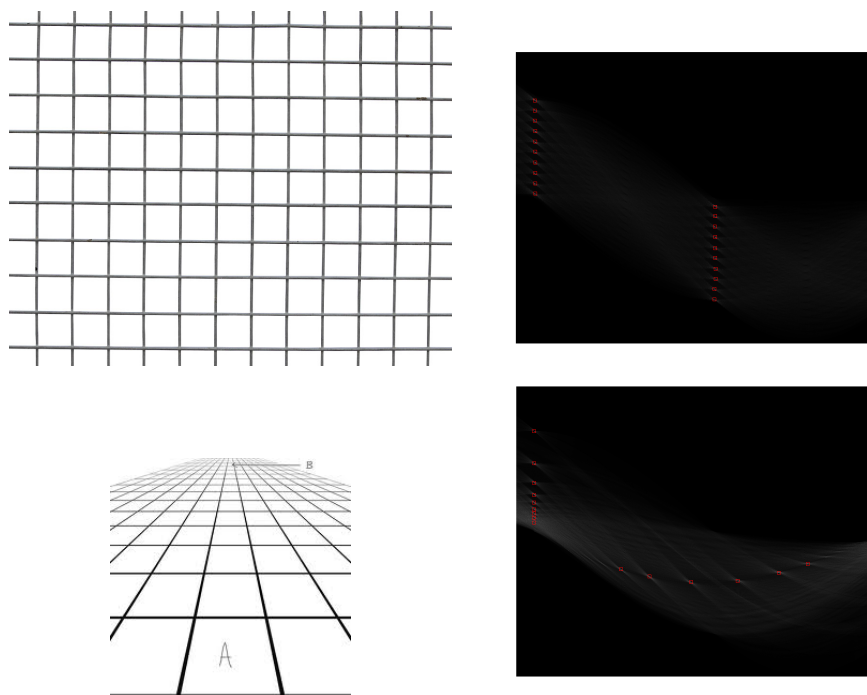


FIGURE 3 – Illustration de l'impact de la perspective sur les piques troués dans la transformée de Hough.
A gauche les images, à droite leurs transformée de Hough respectives.

Dans chacune de ces fenêtres nous utilisons un algorithme RANSAC afin de détecter les lignes dans la transformée de Hough qui s'y trouvent. Nous utilisons RANSAC pour éviter de prendre en compte des piques de la transformée de Hough qui n'ont rien à voir avec le grillage dans la fenêtre prise en compte et qui fausseraient la détection de la ligne.

Nous faisons ensuite glisser cette paire de fenêtres pour couvrir la totalité de la transformée de Hough. Pour chacune des paires de fenêtres nous calculons le nombre de piques se trouvant près de la ligne détectée par RANSAC dans chacune des deux fenêtres. Cela nous donne un premier score. Si la ligne trouvée dans la transformée de Hough est trop horizontale on considère que c'est une erreur et nous donnons un score de 0. Nous choisissons finalement la paire de fenêtres au score maximal.

Cette étape nous donne ainsi un premier grillage potentiel comme illustré en figure 5 parfois complet dans des cas simples comme celui présenté ici. En prenant de larges fenêtres nous pallions le problème de la prise de vue en biais qui déformerait les directions des lignes du grillage. L'utilisation de RANSAC dans ces larges fenêtres permet ensuite de détecter des lignes sans être dérangé par des piques ne décrivant pas des lignes du grillage.

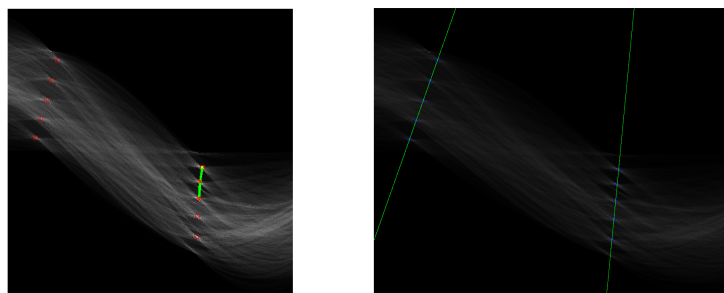


FIGURE 4 – Detections de droites dans la transformée de Hough grâce à gauche à une seconde transformée de Hough et à droite avec RANSAC
On observe que les points ne sont pas tout à fait alignés pour qu'une seconde transformée de Hough les détecte alors qu'un RANSAC y parvient mieux.

3.2 Validation du grillage obtenu

Une fois un premier grillage trouvé nous aurions aimé le valider grâce à l'image en utilisant des descripteurs différents de la détection de contours par méthode de Canny.

Une première solution était de vérifier s'il existait une périodicité de couleur au niveau du grillage détecté. En effet si on se place selon une direction du grillage et en dehors de ce dernier, la seconde direction devrait apparaître périodiquement par ses lignes qu'on reconnaît à leur couleur

Cependant cette méthode peut être mise à mal pour deux raisons. Tout d'abord il faudrait que le grillage ait une couleur relativement unie, or l'éclairage de la photo met en danger cette hypothèse. En outre, comme dit précédemment, la prise de vue de la photo peut faire perdre la périodicité spatiale du grillage rendant cette méthode caduque. C'est pourquoi nous n'avons finalement pas mis en place cette méthode.

Une deuxième solution pouvait être d'utiliser des transformées morphologiques. Nous avons notamment pensé à réaliser un gradient morphologique avec pour élément structurant un segment de direction l'une de celles du grillage.

Nous n'avons finalement pas validé les grillages obtenus car ils étaient dans une grande majorité des cas bons dès la première approche. Ceci est notamment dû à la qualité de notre premier détecteur et donc du choix de la résolution adoptée pour l'image.

3.3 Raffinement du grillage

Le premier grillage obtenu est le plus souvent incomplet : les deux directions du grillage sont connues ainsi qu'un certain nombre de barreaux mais pas tous. Nous avons donc essayé plusieurs méthodes pour raffiner ce premier grillage.

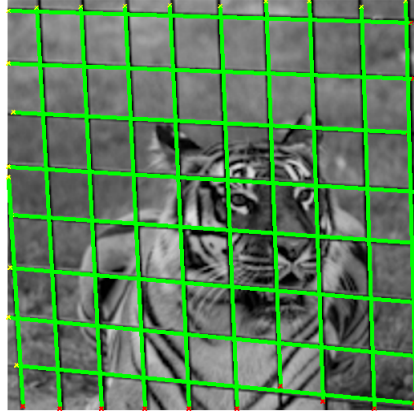


FIGURE 5 – Exemple de grillage obtenu avec la première méthode.

3.3.1 Utilisation des couleurs de l'image

Connaissant une partie des barreaux nous avons essayé de trouver la couleur du grillage. Pour cela nous avons sélectionné les pixels de l'image situés près des lignes que nous avons trouvées puis nous avons réalisé un K-means couleur sur l'ensemble de ces pixels en espérant que la classe majoritaire du K-means corresponde ainsi à la couleur du grillage.

Malheureusement la classe majoritaire ne correspond pas à la couleur du grillage. Cela peut s'expliquer notamment par le fait que nous détectons les contours du grillage et les lignes ainsi trouvées n'étant pas centrées sur le grillage, les pixels sélectionnés sont souvent hors du grillage. Pour mieux le comprendre nous avons affichés les valeurs des couleurs dans l'espace RGB de ces pixels en figure 6 : aucun paquet précis ne semble s'y former.

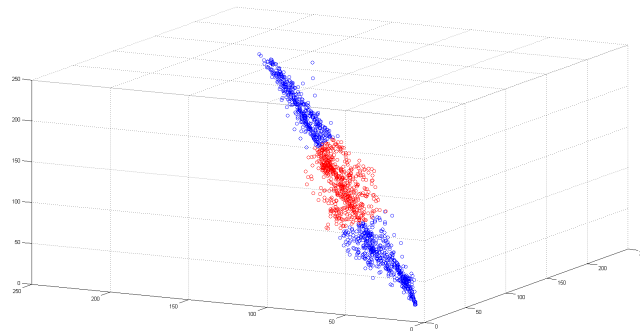


FIGURE 6 – Couleurs des pixels proches des lignes détectées dans l'espace RGB. En rouge le paquet le plus gros trouvé.

3.4 Utilisation de la périodicité du grillage dans la transformée de Hough

Comme dit précédemment, en supposant la photo prise relativement face au grillage l'écart en les piques d'une direction du grillage est sensiblement toujours le même : la périodicité du grillage se retrouve ainsi dans la transformée de Hough.

Pour chaque ligne dans la transformée de Hough trouvée précédemment nous calculons l'écart entre les pics sur la ligne. Nous calculons ensuite l'écart moyen en évitant de prendre en compte des valeurs aberrantes à l'aide d'un RANSAC 1D. En effet n'ayant pas tous les barreaux nous espérons en avoir suffisamment pour mettre en évidence la période, si entre deux barreaux, 5 n'ont pas été trouvés l'écart entre leurs piques sur la transformée de Fourier ne sera pas pertinent c'est pourquoi nous évitons de le prendre en compte.

Une fois cette période trouvée dans la transformée de Hough, nous prenons un pique de départ. Pour cela nous remontons selon la ligne trouvée sur la transformée de Hough pas à pas, la pas étant égal à la période trouvée, ceci jusqu'à être en dehors de la transformée de Hough. Nous redescendons ensuite selon la ligne selon ce même pas. A chaque pas, nous "zoomons" sur des petites fenêtres centrées en là où devrait se trouver un pique. Si un pique déjà trouvé s'y trouve nous le prenons et nous repartons de sa position sinon nous cherchons le maximum dans cette fenêtre comme pique. En repartant de points déjà trouvés nous évitons de propager l'erreur due à l'approximation de la période. En effet comme on l'a déjà vu avec la grille en perspective, dans le cas de grillages pris en biais les piques ne sont pas séparés tout à fait régulièrement.

Cette méthode permet de trouver de nombreux nouveaux piques. Cependant il reste parfois quelques rares barreaux non détectés. Pour cela nous pourrions repasser sur l'image, calculer l'écart successif entre les barreaux selon chacune des directions du grillage et chaque fois que l'écart double approximer l'existence d'un barreau au milieu de cet écart. Nous n'avons pas mis en place ce dernier raffinement car il est possible que jouer sur les informations recueillies à différentes résolutions ou utiliser plus intelligemment les transformées morphologiques donnent un résultat plus précis. En outre dans de nombreux cas ce raffinement suffit.

Nous présentons en figure 7 le résultat avant et après raffinement selon la méthode proposée ici. On constate que de nombreux nouveaux barreaux ont été trouvés mais qu'il en manque encore et que certains ont été ajoutés.

4 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.

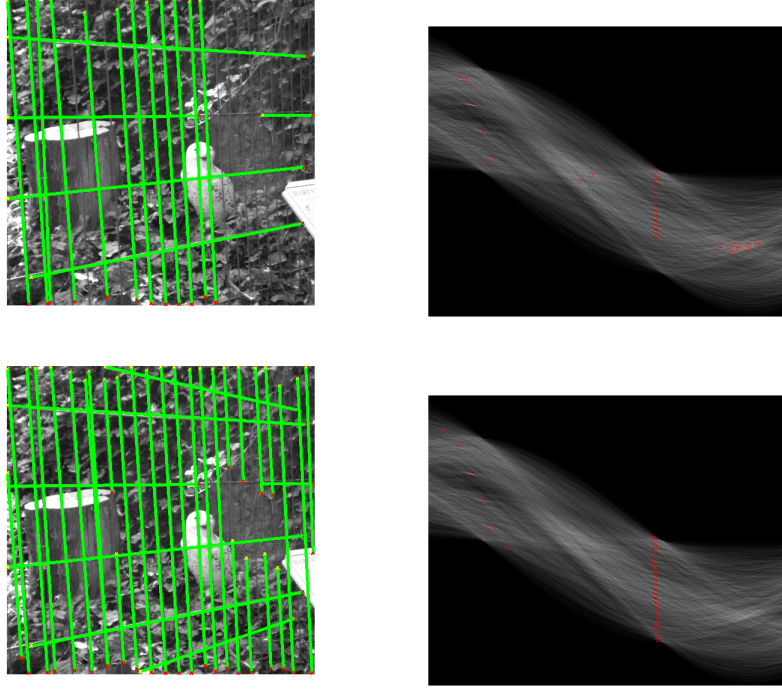


FIGURE 7 – Raffinement du grillage en utilisant la périodicité du grillage dans Hough
A gauche on observe les images, à droite leurs transformées de Hough et les piques détectés. En haut il s'agit du premier grillage détecté avec la méthode précédente, en bas avec le raffinement.

4.1 Inpainting par régularisation variationnelle

4.1.1 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 Φ 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 - Ω 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 $-\text{div}$). 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 présentés en figure (A METTRE). INTERPRETATION.

4.1.2 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).

4.2 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 notés $a = (a_m)_m$, on part 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 λ correspond à la régularisation pour le bruit, comme nos images ne sont pas bruitées, nous prenons un λ 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 Ψ . 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 :

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

Cet algorithme s'adapte dans le cas d'ondelette non orthogonales mais les dernières simplifications ne sont plus licites.

(COMPARAISON METHODES + IMAGES).

Conclusion