

# Total Variation Inpainting

SOBUCKI Stéphane

**Abstract**—Ce document introduit l'inpainting qui permet de restaurer des images endommagées ou de retirer des éléments indésirables.

**Index Terms**—inpainting, traitement d'image, variation totale

## I. INTRODUCTION

L'inpainting est une technique de traitement d'image qui permet de réparer une zone définie par l'utilisateur, on appellera cette zone la zone à remplir. Le terme d'inpainting a commencé à être utilisé dans le domaine du traitement d'image dans [1]. Nous allons dresser l'état de l'art de cette technique et présenter quelques domaines d'applications. Enfin, nous proposerons une implémentation de la variation totale [2] pour le problème d'inpainting et nous analyserons les résultats obtenus.

## II. ÉTAT DE L'ART ET APPLICATIONS

L'inpainting est utilisée dans le cinéma et la photographie pour restaurer les médias, par exemple pour corriger les rayures, artéfacts, enlever les traces de poussières ou encore retirer des éléments indésirables. On peut retrouver des implémentations d'inpainting dans des logiciels de retouche photo ou de traitement de vidéo.

Par essence l'inpainting est un problème mal posé, dans le sens où il n'existe pas une solution unique à ce problème. Il y a donc plusieurs approches [3] pour essayer de résoudre au mieux ce problème. Dans toutes les méthodes utilisées, on définit la zone endommagée comme la zone à remplir. Dans cette zone la valeur des pixels est inconnue et on va chercher à l'interpoler en répondant à un critère. Le critère dépend des méthodes utilisées que nous allons maintenant introduire.

Nous avons tout d'abord les méthodes basées sur les équations dérivées partielles [4] comme Mumford-Shah [5], Cahn-Hilliard [6] etc. Nous nous intéressons à l'énergie contenue dans l'image, les zones bruitées dans une image correspondent à une grande énergie. Avec ce type de méthode, nous allons chercher à trouver les valeurs de pixels qui minimisent l'énergie.

La deuxième catégorie est celle basée sur les patch de pixel avec par exemple la méthode proposée par Criminisi [7], on va chercher à remplir chaque pixel à la frontière de la zone d'intérêt par la valeur du pixel dont le patch est le plus similaire à la zone d'intérêt. Ce processus est réitéré jusqu'à ce que toute la zone soit traitée.

## III. IMPLÉMENTATION

Nous allons implémenter une méthode basée sur l'équation de la chaleur. Nous allons utiliser chercher à minimiser

l'énergie de notre image, en effet une grande énergie signifie des disparités importantes dans l'image. Ce type de méthode a d'abord été utilisée pour débruitez des images [8]. Avec un raisonnement analogue, nous pouvons l'utiliser pour l'inpainting. Pour cela, nous allons utiliser la fonctionnelle de variation totale (TV) qui est définie comme ceci :

$$\int_{\Omega} ||\nabla u||^2 + D\lambda \int_{\Omega} (u - f)^2$$

Où  $\lambda \int_{\Omega} (u - f)^2$  est le terme de fidélité et  $\int_{\Omega} ||\nabla u||^2$  le terme de régularisation. L'équation de la chaleur tend à flouter l'image,  $\lambda$  le coefficient de fidélité permet de garder l'énergie proche de l'image originale et donc de réduire ce phénomène. On a pas d'informations sur les pixels dans la zone à restaurer, on annule le terme de fidélité en ajoutant un coefficient D qui est nul dans la zone à restaurer et est égal à 1 partout ailleurs. Ensuite, on calcule la première variation de la fonctionnelle définie par Lagrange qui correspond à la minimisation de l'énergie :

$$\nabla E = -\dot{\nabla} \left( \frac{\nabla u}{|\nabla u|} \right) + 2\lambda(u - f) \quad (1)$$

$$= -\frac{u_{xx}u_y^2 - 2u_xu_yu_{xy} + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{\frac{3}{2}}} + 2\lambda(u - f) \quad (2)$$

Il ne reste plus qu'à utiliser les différences finies pour obtenir les dérivées premières et secondes de  $u$  que l'on obtient grâce au développement de Taylor.

$$\begin{cases} u_{xx} = \frac{u(i+h,j) + u(i-h,j) - 2u(i,j)}{h^2} \\ u_{yy} = \frac{u(i,j+k) + u(i,j-k) - 2u(i,j)}{k^2} \\ u_{xy} = \frac{u(i+h,j+k) + u(i-h,j-k) - u(i-h,j+k) - u(i+h,j-k)}{4hk} \end{cases}$$

Dans le cas d'une image, les pas d'espace  $h$  et  $k$  sont définis tels que :  $h = k = 1$ .

---

### Algorithm 1 TV inpainting

---

```

1: procedure TV_INPAINTING( $f, \lambda, T, dt$ )
2:    $t \leftarrow 0$ 
3:    $u \leftarrow f$  ▷  $f$  : image endommagée
4:   while  $t < T$  do
5:     Compute  $u_{xx}, u_{yy}, u_{xy}$ 
6:      $\nabla E \leftarrow -\frac{u_{xx}u_y^2 - 2u_xu_yu_{xy} + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{\frac{3}{2}}} + 2\lambda(u - f)$ 
7:      $u^{n+1} \leftarrow u^n + dt(-\nabla E)$ 
8:      $t \leftarrow t + dt$ 
9:   end while
10:  return  $u$  ▷  $u$  : image restaurée
11: end procedure

```

---

#### IV. ANALYSE DES RÉSULTATS

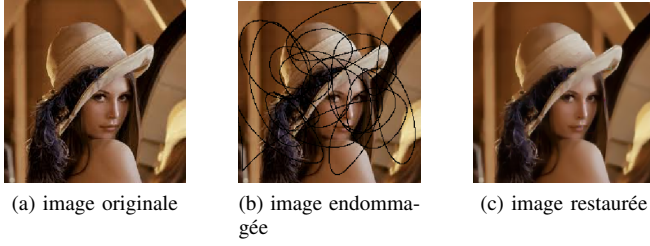


Fig. 1: Rayures fines

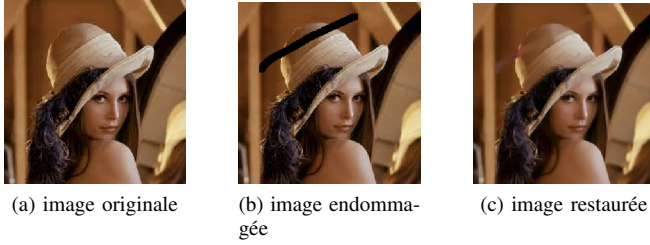


Fig. 2: Trait épais



Fig. 3: Zoom Fig. 2c

Pour des surfaces à traiter fines, on arrive rapidement à convergence et la correction est bonne (Fig. 1). Dès lors que les zones à remplir sont trop épaisses, la correction n'est pas satisfaisante et le temps de traitement est très long. En effet, le flou est très présent et on arrive à le distinguer facilement (Fig. 3).

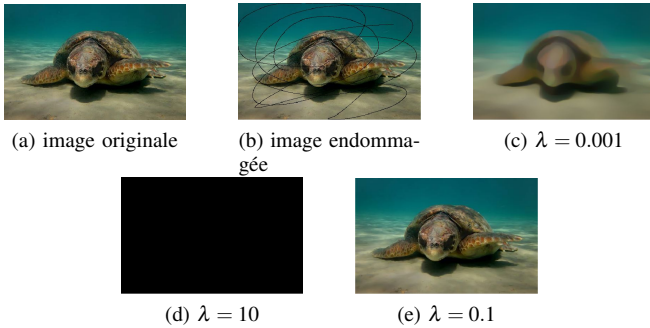


Fig. 4: Effet du coefficient de fidélité  $\lambda$

Il faut également faire attention au choix de  $\lambda$  (Fig. 4) puisqu'il peut conduire à des résultats indésirables. En effet, avec un  $\lambda$  trop faible (ici  $\lambda = 0.001$ ), l'image est très lissée et on perd beaucoup d'informations (Fig. 4c). Au contraire,

avec un  $\lambda$  trop grand (ici  $\lambda = 10$ ), les pixels de l'image vont tous converger vers la même valeur ce qui se traduit en une image de couleur uniforme (Fig. 4d). On détermine  $\lambda$  de façon empirique, on voit qu'avec  $\lambda = 0.1$  le résultat est satisfaisant et on se rapproche de l'image originale après inpainting (Fig. 4e). Nous pouvons donc conclure que pour des problèmes comme des rayures fines, salt&pepper ou des traces de poussière, cet algorithme est suffisant et satisfaisant. L'algorithme prend du temps à faire l'inpainting, l'utilisation d'un algorithme d'optimisation comme Split-Bregman [9] permet d'améliorer les performances en temps de calcul. Pour retirer des objets de l'image (Fig. 5), on voit bien que l'utilisation de la variation totale seule ne nous donne pas un résultat satisfaisant. En effet, on peut voir sur l'image que le flou créé est trop important (Fig. 5c). Pour aller plus loin dans l'utilisation, de cette méthode on peut utiliser des fonctionnelles d'un ordre plus élevée [10] ce qui permettrait de synthétiser les textures.

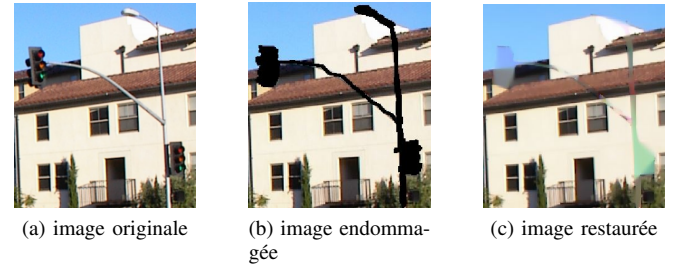


Fig. 5: Suppression d'objet

#### V. CONCLUSION

En conclusion, l'inpainting basée sur la variation totale permet de corriger des zones fines. Ce n'est pas une très bonne technique pour restaurer des grandes zones épaisses. La synthèse de texture est dans ce cas incontournable et c'est un domaine de recherche actif. Cette méthode est donc satisfaisante pour enlever du texte ou des rayures. Si on souhaite pouvoir retirer des objets de l'image, les méthodes par patch semblent plus adaptées.

#### REFERENCES

- [1] Marcelo Bertalmio, Guillermo Sapiro, Vicent Caselles, and Coloma Ballester. Image Inpainting.
- [2] Antonin Chambolle, Vicent Caselles, Matteo Novaga, Daniel Cremers, and Thomas Pock. An introduction to Total Variation for Image Analysis. November 2009.
- [3] Işık Barış Fidaner. A Survey on Variational Image Inpainting , Texture Synthesis and Image Completion. 2007.
- [4] Anis Theljani. Partial differential equations methods and regularization techniques for image inpainting.
- [5] Selim Esedoglu and Jianhong Shen. Digital inpainting based on the Mumford–Shah–Euler image model. *European Journal of Applied Mathematics*, 13(04), August 2002.
- [6] Andrea L. Bertozzi, Selim Esedoglu, and Alan Gillette. Inpainting of Binary Images Using the Cahn–Hilliard Equation. *IEEE Transactions on Image Processing*, 16(1), January 2007.
- [7] A. Criminisi, P. Perez, and K. Toyama. Region Filling and Object Removal by Exemplar-Based Image Inpainting. *IEEE Transactions on Image Processing*, 13(9), September 2004.
- [8] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. *Nonlinear Total Variation Based Noise Removal Algorithms*. 1992.

- [9] Pascal Getreuer. Total Variation Inpainting using Split Bregman. *Image Processing On Line*, 2, July 2012.
- [10] Moez Kallel, Maher Moakher, and Anis Theljani. Multiscale Fourth-Order Models for Image Inpainting and Low-Dimensional Sets Recovery. February 2016.