

# Reconstruction d'un tableau sans reflets à partir de photos prises sous des angles différents

Théophile DALENS et Jean CAILLÉ

13 Mars 2014

## 1 Introduction

Il devient aujourd'hui facile d'obtenir des photos de chef-d'oeuvre dans un musée. Toutefois la présence de lumière couplés aux vernis et aux cadre en verres entraine sur la plupart des photos prisent des reflets peu plaisants à l'oeil. En multipliant les points de vue, nous allons voir qu'il est possible de supprimer ces reflets et d'obtenir une photo proche de celle prise dans des conditions idéales.

## 2 Méthode

La méthode proposée par l'article ne s'attache pas à traiter le tableau sous l'angle de la reconnaissance d'image (reconnaitre par exemple les aplats de couleurs et supprimer les reflets à partir de cette information), mais uniquement grâce à des techniques génériques, permettant leur application sur la plupart des tableaux. L'idée consiste à prendre plusieurs photos du même tableau sous différents angles. Les reflets ne dégraderons pas alors la même zone du tableau sur les différentes images et nous pourrons reconstruire à partir de ces images une image "virtuelle" sans reflets.

L'algorithme suggéré par l'article se décompose en deux parties. Tout d'abord, les images sont "recalées" afin d'obtenir une perspective commune. Ainsi, les pixels des différentes images correspondent aux mêmes zones sur le tableaux, et il devient plus facile de reconstruire l'image corrigée. La seconde partie correspond justement à cette reconstructions. Plusieurs méthodes sont suggérées et nous nous sommes attachés à implémenter et comparer les différentes méthodes de fusion d'image proposées.

### 2.1 Recalage des images

Comme décrit plus haut, la première partie revient à "recaler" les images entre elles, c'est à dire simuler une perspective commune entre les photos. Sois nous pouvons nous attacher à modifier toutes les photos pour recaler l'image du tableau sur un rectangle donné, mais cela nécessite une connaissance *a priori* des dimensions du tableau, que nous ne connaissons pas. Pour palier à ce problème, nous choisissons parmi les images de la peinture une photo dite de *référence* et nous recadrerons les autres pour les afficher sous cette perspective.

Dans le cas où la caméra est idéale (modèle sténopé, pas de déformation dues à la lentille), et le tableau est parfaitement plan, la transformation permettant de recaler un tableau est une perspective. Nous n'avons pas eu besoin lors de nos traitement d'hypothèses supplémentaires.

### 2.1.1 Recalage manuel

Pour trouver la perspective liant deux images données, la solution la plus simple est de demander à l'utilisateur de cliquer sur 4 points communs entre les deux photos (par exemple les quatre coins du tableau), et de trouver la transformation permettant de passer de  $\{p_i\}_{i=1\dots 4}$  à  $\{q_i\}_{i=1\dots 4}$ . C'est cette solution que nous avons tout d'abord implémenté. Les résultats étaient satisfaisant au premier abord, mais la précision de la méthode n'était pas suffisante pour la suite du traitement. En effet, l'utilisateur ne peut cliquer qu'avec une précision de quelques pixels. Nous avons alors dû nous orienter vers une méthode automatisée faisant appel à des descripteurs locaux.

### 2.1.2 Recalage automatique avec utilisation des SIFT

Pour obtenir une précision sous-pixelique, nous extrayons des points d'intérêt de type SIFT [4] de manière automatique sur les deux images. En supposant que la plupart des points d'intérêt sont communs sur les deux images, nous pouvons rechercher des correspondances point-à-point en se servant des descripteurs (Figure 1). Nous pouvons alors chercher la meilleure homographie permettant de faire correspondre les descripteurs de la première image à ceux de la seconde image. Comme l'algorithme SIFT peut fournir des points qui ne sont pas parfaits (correspondance entre deux zones très éloignées dans la réalité par exemple), nous avons utilisé la méthode RANSAC [2] pour filtrer les bonnes correspondances et obtenir le recalage (Figure 2 et 3). Ce recalage s'est avéré suffisant pour la suite de l'algorithme.



Figure 1: Paires de descripteurs SIFT entre les deux images avant filtrage par RANSAC

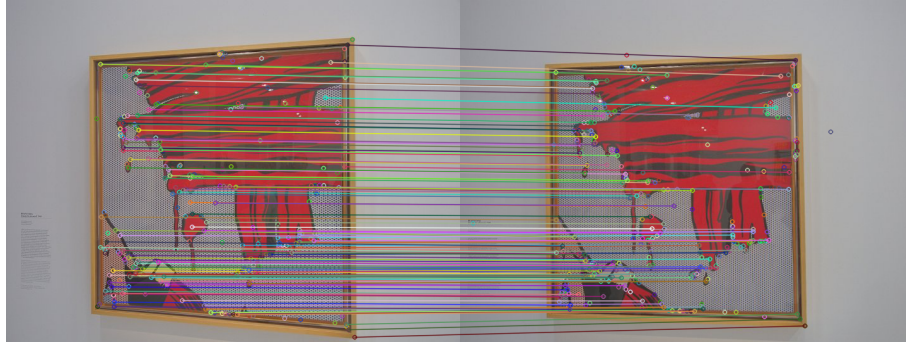


Figure 2: Paires de descripteurs SIFT estimés correctes par l'algorithme RANSAC et utilisés pour reconstruire la perspective



Figure 3: A gauche : image de travail, au centre : image de référence, à droite : image de travail après recalage par rapport à l'image de référence. Noter la différence de position des reflets entre l'image de référence et l'image recalée

## 2.2 Fusion des images

Une fois les images recalées, l'étape suivante est de fusionner les images entre elles pour éliminer les reflets. Plusieurs approches sont possibles ; dans les premières approches développées ci-après, nous testons des approches qui fusionnent les images pixels par pixels, c'est-à-dire que chaque pixel de l'image reconstitué dépend uniquement des pixels correspondant dans les différentes images recalées. Ainsi, si les images recalées sont notées  $\tilde{I}_1, \dots, \tilde{I}_n$ , l'image fusionnée  $J$  est définie par

$$J(x, y) = f \left( \tilde{I}_1(x, y), \dots, \tilde{I}_n(x, y) \right)$$

De nombreux choix sont possibles pour la fonction  $f$ . Dans la section 2.2.1, nous proposons une fusion par minimum sur chaque canal de couleur. Nous proposons ensuite des fusions par médiane ; elle se fait par canal RGB dans la section 2.2.2, par médiane vectorielle dans la section 2.2.3, ou par une méthode légèrement différente dans un autre espace de couleur, détaillée dans la section 2.2.4.

Il est également possible de travailler dans l'espace gradient au lieu de l'espace de l'image. Dans ce cas, l'image fusionnée  $J$  est définie par

$$\nabla J(x, y) = f \left( \nabla \tilde{I}_1(x, y), \dots, \nabla \tilde{I}_n(x, y) \right)$$

Ce cas est détaillé dans la section 2.2.5.

### 2.2.1 Fusion par minimum

Pour cette méthode, nous faisons l'hypothèse que les reflets ne font qu'ajouter de la luminosité aux différents pixels de l'image, et que les reflets ne peuvent se retrouver dans toutes les images. Dès lors, il suffit de prendre le minimum de chaque canal de couleur pour parvenir à éliminer les reflets.

### 2.2.2 Fusion par médiane par canal

### 2.2.3 Fusion par médiane vectorielle

Une autre façon consiste à voter, pixel par pixel, pour ce qui devrait être le pixel sans reflet. On suppose ici que le reflet se trouve dans moins de la moitié des images à un endroit donné. Deux méthodes sont alors possibles. On peut prendre la médiane des pixels, canaux de couleurs par canaux de couleurs. Cette méthode, simple et intuitive, a le désavantage, comme la méthode précédente, de mélanger les canaux couleurs d'images différentes, ce qui peut conduire à des couleurs non naturelles. Pour pallier à ce problème, nous avons également implémenté un équivalent de la médiane dans l'espace vectoriel des couleurs ; il s'agit du vecteur minimisant la somme de ses distances  $L_1$  aux autres vecteurs. En pratique, nous n'observons pas de différences entre ces deux médianes.

### 2.2.4 Fusion par médiane dans l'espace Lab

### 2.2.5 Fusion dans l'espace gradient

Enfin, une méthode décrite dans [3] permettant de réduire plus les reflets consiste à prendre la médiane vectorielle dans l'espace gradient des images. Pour ce faire, nous calculons les images gradients des images recalées, converties en niveau de gris, et nous en calculons la médiane vectorielle en chaque point. Ensuite, nous construisons les gradients de chaque canal de couleur de l'image fusionnée en prenant, pour chaque point et chaque canal, le gradient de l'image dont la médiane est issue. Ce sont donc bien les gradients des canaux de couleurs qui sont considérés.

Lorsque l'on connaît le gradient  $F$  d'une image, il est possible de la retrouver en minimisant  $\min ||\text{grad}I - F||$ . Ceci se fait grâce à l'équation de Poisson  $\Delta I = \text{div}F$ . Cette équation est alors résolue, canal par canal, en passant par la transformée de Fourier, s'inspirant de [5]. Rencontrant des difficultés dans cette dernière étape, nous n'avons pas pu terminer cette méthode. Elle donne cependant d'excellents résultats dans [3], éliminant certains reflets apparaissant dans une majorité d'images.

## 3 Implémentation

Pour l'implémentation de la méthode décrite ci-dessus, nous avons choisi des outils C++ accompagné de la librairie OpenCV [1] de traitement d'image. La collaboration a été réalisée grâce au gestionnaire de version git. La totalité du code et ses révisions antérieures sont donc disponible sur Github : <http://github.com/jcaille/CVCanvas>

## 4 Résultats

Les résultats varient peu selon les méthodes (figure 4). Les médianes permettent de retirer de nombreux artéfacts de reflets, mais ne fonctionnent pas là où une majorité d'images présentent des artéfacts. Les méthodes de médiane, canal par canal ou globale, donnent les mêmes résultats sur les exemples que nous avons essayés.

## 5 Améliorations possibles

Plusieurs améliorations de l'algorithme sont envisageables. Si les images obtenues sont de bonne qualité, l'algorithme ne sait pas faire la différence entre les zones de la peinture et les zones extérieures (cadre, mur, paysage, ...), dans lesquelles la fusion est effectuée alors qu'elle n'a pas de sens. Une possibilité pour détecter ces zones et de segmenter l'image de référence (par exemple en utilisant des snakes quadrangulaires) entre un "intérieur" sur lequel la fusion serait effectuée et un extérieur sur lequel on garderait le contenu d'origine. Pour s'aider dans cette segmentation, nous pourrions utiliser les informations des autres photos du tableau.

De plus, l'algorithme que nous avons développé (en particulier la partie de fusion) n'est pas de fait résistant aux changements de contrastes en général. Une idée pour corriger ce défaut serait d'égaliser les histogrammes des différentes images (de préférence en limitant cette égalisation à la zone "intérieur" du tableau).

## References

- [1] G. Bradski. *Dr. Dobb's Journal of Software Tools*.
- [2] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [3] Gloria Haro, Antoni Buades, and Jean-Michel Morel. Photographing paintings by image fusion. *SIAM Journal on Imaging Sciences*, 5(3):1055–1087, 2012.
- [4] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.
- [5] Jean Michel Morel, Ana Belen Petro, and Catalina Sbert. A pde formalization of retinex theory. *Image Processing, IEEE Transactions on*, 19(11):2825–2837, 2010.

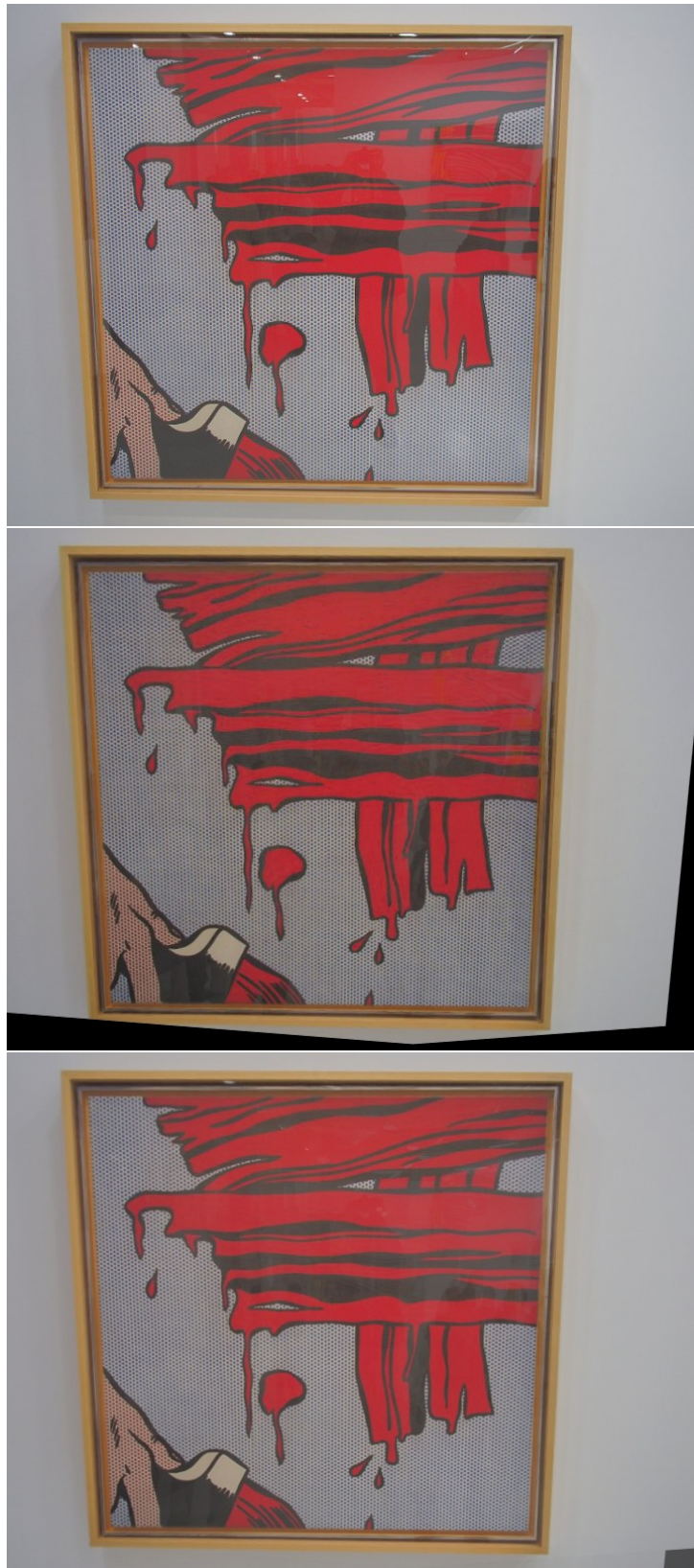


Figure 4: Correction des artefacts. De haut en bas : image de référence, puis corrigée avec les méthodes du minimum et de la médiane.