

# Projet Image, Signal, Simulations

Anne-Sophie Dusart

Guilhem Fouilhé

Chloé Michel

Damien Ricci

## Recalage d'image - $\ell^1 - \ell^2$



# 1 Introduction

Le recalage d'image a pour objectif d'établir une correspondance entre les points représentant un même objet dans deux images d'une même scène. Cette technique est utilisée pour retrouver des détails sur un film de mauvaise qualité, stabiliser un film ou pour comparer une image à une image de référence en imagerie médicale notamment.

Dans notre projet, nous allons étudier une méthode d'optimisation qui consiste à minimiser une énergie somme d'un terme d'attache aux données et d'un terme de régularisation.

Soit  $\varepsilon > 0$ , soit  $t \in \mathbb{R}^{N^2} \times \mathbb{R}^{N^2}$ . Notre terme d'attache aux données est :

$$DT_\varepsilon(t) = \sum_{m,n=1}^N \varphi_\varepsilon(|\tilde{v}^1((m,n) + t_{m,n}) - v^2_{m,n}|^2)$$

où  $\tilde{v}^1(\lfloor (m,n) + t_{m,n} \rfloor) = v^1_{m,n} + \langle \nabla v^1_{m,n}, t_{m,n} \rangle$  et  $\varphi_\varepsilon(x) = \sqrt{x + \varepsilon}$

Le terme de régularisation du champs de vecteurs  $t$  est défini par :

$$R(t) = \sum_{m,n=1}^N |\nabla t^1_{m,n}|^2 + |\nabla t^2_{m,n}|^2$$

Ce qui nous donne l'énergie suivante à minimiser :

$$E(t) = \sum_{m,n=1}^N \sqrt{|\tilde{v}^1((m,n) + t_{m,n}) - v^2_{m,n}|^2 + \varepsilon} + \lambda(|\nabla t^1_{m,n}|^2 + |\nabla t^2_{m,n}|^2)$$

Avec  $\lambda > 0$  un paramètre appelé poids. On modélise ce problème de recalage par le problème d'optimisation sans contrainte suivant :

$$(\mathcal{P}) : \begin{cases} \text{minimiser } E(t) \\ \text{où } t \in \mathbb{R}^{N^2} \times \mathbb{R}^{N^2} \end{cases}$$

## 2 Existence d'une solution

Tout d'abord, nous devons montrer que le problème  $\mathcal{P}$  admet une solution. Pour cela, on va montrer que la fonction  $E$  est continue et coercive. On justifiera également sa différentiabilité, qui nous sera utile pour la prochaine section.

## 2.1 Coercivité

On montre d'abord que la fonction énergie  $E$  est coercive.

Soit  $t \in \mathbb{R}^{N^2} \times \mathbb{R}^{N^2}$ , soit  $\varepsilon > 0$ . On a  $\varphi_\varepsilon(t_{m,n}) = \sqrt{t_{m,n} + \varepsilon}$ , donc  $\varphi_\varepsilon$  est coercive car :

$$\text{Or, } \lim_{\|x\| \rightarrow +\infty} \sqrt{x} = +\infty \Rightarrow \lim_{\|t\| \rightarrow +\infty} \varphi_\varepsilon(t) = +\infty$$

Car  $x \mapsto \sqrt{x}$  est coercive. Montrons que cela entraîne la coercivité de  $E$ .

$$\begin{aligned} \forall t \in \mathbb{R}^{N^2} \times \mathbb{R}^{N^2}, \quad E(t) &\geq \sum_{m,n}^N \varphi_\varepsilon((v^1_{m,n} + \langle \nabla v^1_{m,n}, t_{m,n} \rangle - v^2_{m,n})^2) \\ &\geq \sum_{m,n}^N \underbrace{v^1_{m,n} + \langle \nabla v^1_{m,n}, t_{m,n} \rangle - v^2_{m,n}}_{\substack{\rightarrow +\infty \\ \|t\| \rightarrow \infty}} \end{aligned}$$

Lorsque  $t$  est constant, car on a alors  $t_{m,n}$  qui ne peut pas avoir tous les produits scalaires avec  $\nabla v^1_{m,n}$  nuls, si ceux-ci changent de direction (il s'agit donc d'une hypothèse implicite pour justifier la coercivité de cette fonction).

Lorsque  $t$  n'est pas constant, c'est le terme de régularisation qui est coercif, car au moins une valeur en  $(m, n)$  va faire diverger  $E$ .

Nous avons donc montré que :

$$\lim_{\|t\| \rightarrow +\infty} E(t) = +\infty$$

Ce qui signifie que  $E$  est coercive.

## 2.2 Continuité et différentiabilité

La fonction objectif  $E$  est continue en  $t$  comme composition de fonctions continues. Elle ne pose également pas de problème de différentiabilité comme en témoigne le calcul du gradient ci-dessous.

Notre fonction  $E$  étant continue et coercive, elle admet un minimiseur  $t_0$  sur  $\mathbb{R}^{N^2} \times \mathbb{R}^{N^2}$ .

### 3 Convergence de l'algorithme

#### 3.1 Calcul du gradient

Premièrement, déterminons  $\nabla E$  en 2 étapes. D'abord nous allons calculer le gradient de l'attache aux données  $\nabla DT_\varepsilon$ , puis celui du terme de régularisation  $\nabla R$ .

Calcul de  $\nabla DT_\varepsilon$  en  $(m, n)$  par un développement de Taylor :

$$\begin{aligned}\varphi_\varepsilon(\tilde{v}^1((m, n) + t_{m,n} + h_{m,n})) &= \varphi_\varepsilon(\underbrace{\langle \nabla v^1_{m,n}, t_{m,n} \rangle}_{=: Ht_{m,n}} + \underbrace{\langle \nabla v^1_{m,n}, h_{m,n} \rangle}_{=: Hh_{m,n}} - (v^2_{m,n} - v^1_{m,n})|^2) \\ &= \varphi_\varepsilon((Ht_{m,n} - (v^2_{m,n} - v^1_{m,n}))^2 \\ &\quad + 2(Ht_{m,n} - (v^2_{m,n} - v^1_{m,n}))Hh_{m,n} + (Hh_{m,n})^2)\end{aligned}$$

Or,

$$(Hh_{m,n})^2 = (\langle \nabla v^1_{m,n}, h_{m,n} \rangle)^2 = o(\|h_{m,n}\|)$$

Et :

$$\varphi_\varepsilon(t + h) = \varphi_\varepsilon(t) + \varphi'_\varepsilon(t)h + o(\|h\|)$$

Donc, en posant  $X_{t_{m,n}} = 2\varphi'_\varepsilon((Ht_{m,n} - (v^2_{m,n} - v^1_{m,n}))^2)(Ht_{m,n} - (v^2_{m,n} - v^1_{m,n}))$ , on a :

$$\begin{aligned}DT_\varepsilon(t + h) &= \sum_{m,n=1}^N \varphi_\varepsilon(\tilde{v}^1((m, n) + t_{m,n} + h_{m,n})) \\ &= \sum_{m,n=1}^N \varphi_\varepsilon((Ht_{m,n} - (v^2 - v^1)_{m,n})^2 + 2(Ht_{m,n} - (v^2 - v^1)_{m,n})Hh_{m,n} + o(\|h_{m,n}\|)) \\ &= \sum_{m,n=1}^N \varphi_\varepsilon(\tilde{v}^1((m, n) + t_{m,n})) + X_{t_{m,n}}Hh_{m,n} + o(\|h_{m,n}\|) \\ &= DT_\varepsilon(t) + \langle H^*X, h \rangle + o(\|h\|)\end{aligned}$$

Où  $H^*$  est l'opérateur suivant :

$$\begin{array}{ccc} H^* & : & \mathbb{R} \rightarrow \mathbb{R}^{N^2} \times \mathbb{R}^{N^2} \\ & & t \mapsto t \cdot \nabla v^1 \end{array}$$

Le gradient du terme d'attache aux données est donc  $\nabla DT_\varepsilon = H^*X$ . On va maintenant calculer le gradient du terme de régularisation.

$$\begin{aligned}
|\nabla t^1_{m,n} + \nabla h^1_{m,n}|^2 + |\nabla t^2_{m,n} + \nabla h^2_{m,n}|^2 &= |\nabla t^1_{m,n}|^2 + 2\langle \nabla t^1, \nabla h^1 \rangle_{m,n} + |\nabla h^1_{m,n}|^2 \\
&\quad + |\nabla t^2_{m,n}|^2 + 2\langle \nabla t^2, \nabla h^2 \rangle_{m,n} + |\nabla h^2_{m,n}|^2 \\
&= |\nabla t^1_{m,n}|^2 + 2\langle \nabla t^1, \nabla h^1 \rangle_{m,n} + |\nabla t^1_{m,n}|^2 \\
&\quad + 2\langle \nabla t^2, \nabla h^2 \rangle_{m,n} + o(\|h_{m,n}\|) \\
&= |\nabla t^1_{m,n}|^2 + |\nabla t^2_{m,n}|^2 + 2\langle \nabla t^1, \nabla h^1 \rangle_{m,n} \\
&\quad + 2\langle \nabla t^2, \nabla h^2 \rangle_{m,n} + o(\|h_{m,n}\|)
\end{aligned}$$

On en déduit que le gradient du terme de régularisation est  $\nabla R(t) = 2(\nabla^* \nabla t^1, \nabla^* \nabla t^2)$ . Enfin, on se souvient que si  $\nabla = (D_x, D_y)$  avec :

$$\begin{cases} (D_x w)_{m,n} = (w_{m,n} - w_{m-1,n}) \\ (D_y w)_{m,n} = (w_{m,n} - w_{m,n-1}) \end{cases}$$

On a alors :

$$\begin{cases} (D_x^* w)_{m,n} = (w_{m,n} - w_{m+1,n}) \\ (D_y^* w)_{m,n} = (w_{m,n} - w_{m,n+1}) \end{cases}$$

Et enfin  $\nabla^* w = D_x^* w_1 + D_y^* w_2$ . On peut maintenant déterminer le gradient de  $E$  :

$$\nabla E(t) = H^* X t + 2\lambda(\nabla^* \nabla t^1, \nabla^* \nabla t^1)$$

## 3.2 Propriétés de la fonction objectif

Nous savons que notre problème  $\mathcal{P}$  admet une solution. On va maintenant montrer les propriétés de la fonction objectif qui nous permettent de dire que l'algorithme converge, à savoir :

- $E$  est propre SCI
- $E$  est convexe
- $E$  est  $L$ -lisse

### 3.2.1 Propre SCI

La fonction  $E$  est propre, car elle est positive et non identiquement égale à  $\infty$ .  $E$  est également SCI, car elle est continue.

### 3.2.2 Convexité

On remarque que le terme d'attache aux données  $DT_\varepsilon$  est en chaque terme de la somme convexe comme composée de la fonction convexe  $x \mapsto \sqrt{x^2 + \varepsilon}$  avec la fonction affine  $t_{m,n} \mapsto \tilde{v}^1((m, n) + t_{m,n}) - v^2_{m,n}$ .

Concernant le terme de régularisation, il s'agit également d'une somme composition d'une fonction convexe ( $x \mapsto \|x\|^2$  ou  $\|\cdot\|$  désigne la norme euclidienne) et d'une fonction affine (l'opérateur de différence finie). La fonctionnelle  $E$  est donc convexe.

### 3.2.3 Gradient $L$ -Lipschitzien

Les deux termes de la fonctions sont en fait Lipschitziens. On a :

$$\begin{aligned}
\|DT_\varepsilon(t) - DT_\varepsilon(t')\|_2^2 &= \|H^*Xt - H^*Xt'\|_2^2 \\
&= \sum_{m,n}^N (H^*(Xt_{m,n} - Xt'_{m,n}))^2 \\
&= \sum_{m,n}^N ((Xt_{m,n} - Xt'_{m,n}) \nabla v^1_{m,n})^2 \\
&\leq \frac{1}{\varepsilon} \sum_{m,n}^N ((t_{m,n} - t'_{m,n}) \nabla v^1_{m,n})^2 \\
&\leq \frac{1}{\varepsilon} \|\nabla v^1\|_2^2 \|t - t'\|_2^2
\end{aligned}$$

Et :

$$\begin{aligned}
\|R(t) - R(t')\|_2^2 &= \sum_{m,n}^N (\nabla R(t_{m,n}) - \nabla R(t'_{m,n}))^2 \\
&= 2^2 \sum_{m,n}^N (\nabla^* \nabla(t_{m,n}^1 - t_{m,n}^{1'}) + \nabla^* \nabla(t_{m,n}^2 - t_{m,n}^{2'}))^2 \\
&\leq 2^2 \|t - t'\|_2^2
\end{aligned}$$

D'où  $E$  est de Gradient  $L$ -Lipschitzien avec  $L = \frac{\|\nabla v^1\|_2}{\sqrt{\varepsilon}} + 2\lambda$

### 3.2.4 Convergence de l'algorithme de la descente de gradient

On utilise le Théorème 2 page 18 du cours, qui nous permet de conclure que l'algorithme de la descente de gradient appliqué à la minimisation de la fonction

$E$  converge pour un pas fixe  $\alpha = \frac{1}{L}$  où  $L$  est la constante de Lipschitz du gradient de  $E$ .

## 4 Résultats expérimentaux

### 4.1 Implémentation

On a implémenté de manière classique l'algorithme suivant :

---

**Algorithm 1** Gradient Descent

---

```
1: procedure GRADIENT DESCENT( $v^1, v^2, \alpha, t_0, ItMax$ )  
2:    $\mathbf{t} \leftarrow \mathbf{t}_0$ .  
3:   for  $i \in \{1, \dots, ItMax\}$  do  
4:      $\mathbf{t} \leftarrow \mathbf{t} - \alpha \nabla E(t)$   
5:   end for  
6:   Return  $\mathbf{t}$ .  
7: end procedure
```

---

en implémentant  $\nabla E$  à partir des calculs adjoints calculés plus haut.

Nous avons également, dans le but d'améliorer les performances de nos algorithmes, utilisé un algorithme de recherche de point fixe pour le calcul d'une approximation du pas de plus grande descente.

### 4.2 Résultats qualitatifs

Nous allons tester notre algorithme sur plusieurs images de différentes tailles et avec plus ou moins de détails. Les paramètres utilisés sont :  $\lambda = 0.1$  ,  $\epsilon = 0.01$  et  $pas = 0.1$ . Ces paramètres ont été choisis après plusieurs tests visibles sur le notebook.

#### 4.2.1 Image 1

Dans un premier temps, on teste notre algorithme sur une petite image  $100 \times 100$  pixels.

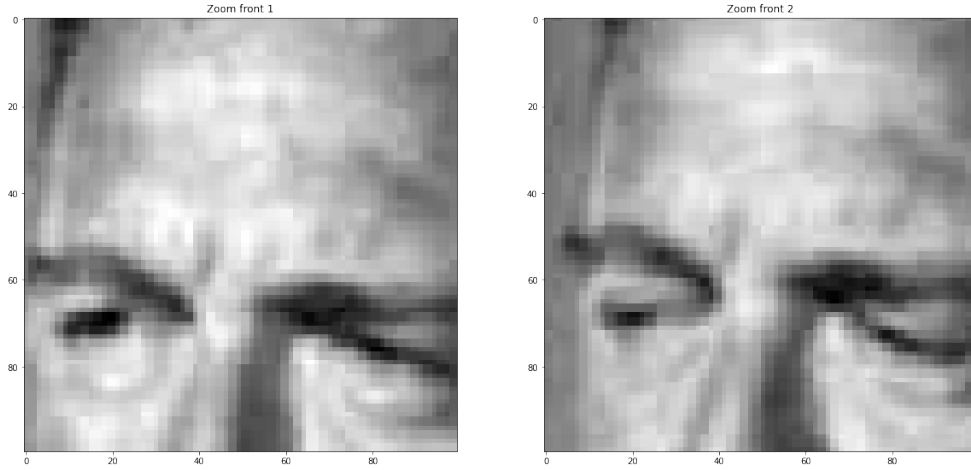


FIGURE 1 – On voit que l’homme fait un mouvement des yeux vers le bas à gauche, et que son visage tourne également dans cette direction.

On va maintenant afficher le champ de vecteurs par-dessus l’image 1, afin d’illustrer le mouvement des pixels calculé par notre algorithme. Le champ de vecteur affiché a été trouvé en 10 itérations avec le pas optimal.

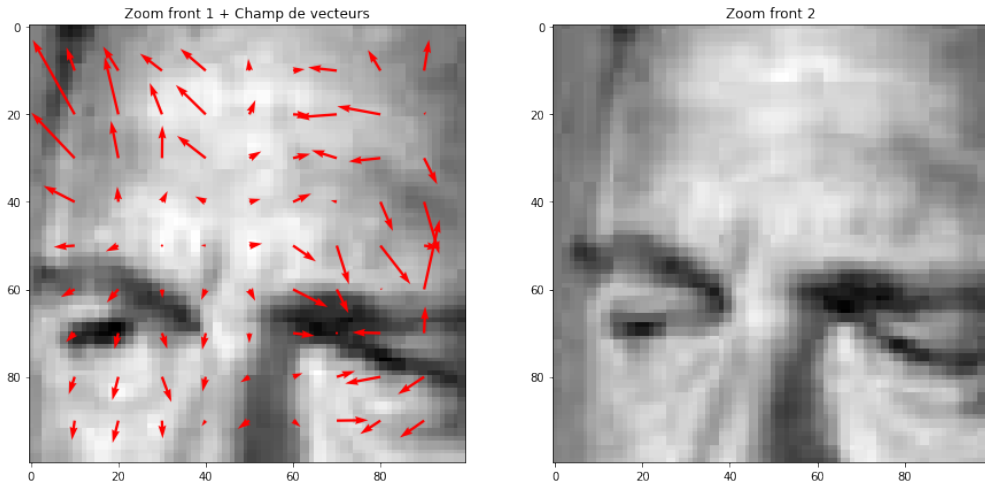


FIGURE 2 – On voit que les vecteurs au niveau des yeux correspondent bien au mouvement réel. En ce qui concerne les vecteurs présents sur le haut de la tête, ils illustrent le mouvement de rotation de la tête (indiquent donc une direction vers le haut/derrière la tête).

Nous avons essayé d’afficher l’image finale recalée, c’est-à-dire  $\tilde{v}^1$  qui vaut  $\tilde{v}_{m,n}^1 = v^1(\lfloor (m,n) + t_{m,n} \rfloor)$  en  $(m,n)$ , mais cela n’a pas donné un rendu satisfaisant. On



ne le montrera donc pas pour les autres couples d'images testés.

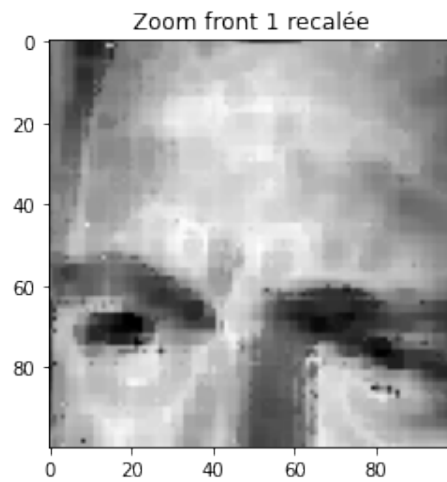


FIGURE 3 – Image recalée  $\tilde{v}^1$

A titre de comparaison, on a fait tourner nos algorithmes avec les mêmes paramètres avec l'algorithme du pas constant.

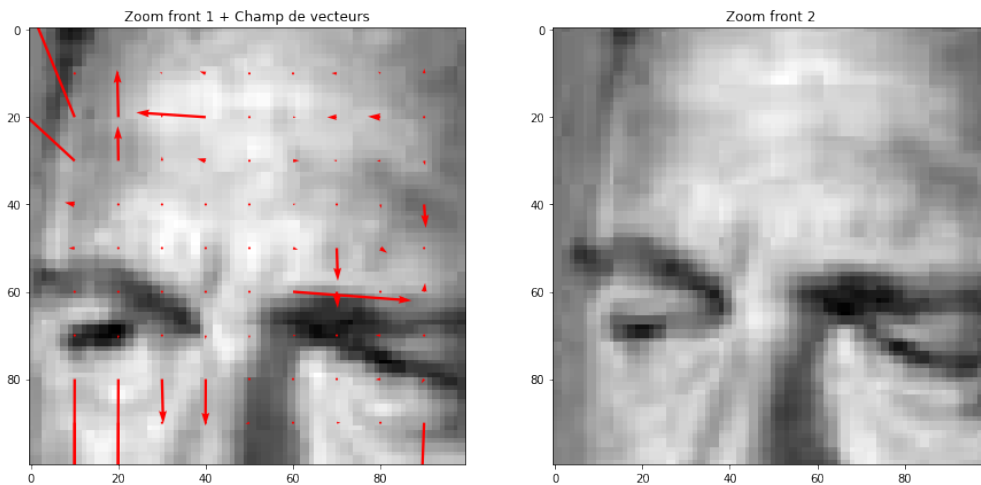


FIGURE 4 – On voit bien que le champ de vecteur obtenu n'est pas satisfaisant.

On remarque que l'algorithme du pas constant est plus rapide mais moins efficace en terme de champ de vecteur. Pour les autres images nous ferons seulement le test avec l'algorithme du pas optimal.

### 4.2.2 Image 2

Testons maintenant notre algorithme sur une image plus grande de 350 x 350 pixels qui n'est autre que le dé-zoom de l'image précédente.

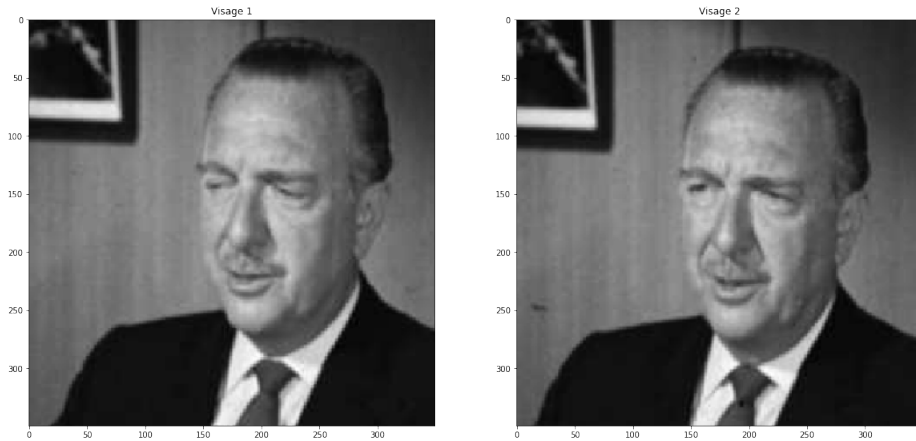


FIGURE 5 – On voit que le décor ne bouge pas, seul l'homme effectue le même mouvement de tête que précédemment.

Affichons le champ de vecteurs par-dessus la première image calculée avec 10 itérations.

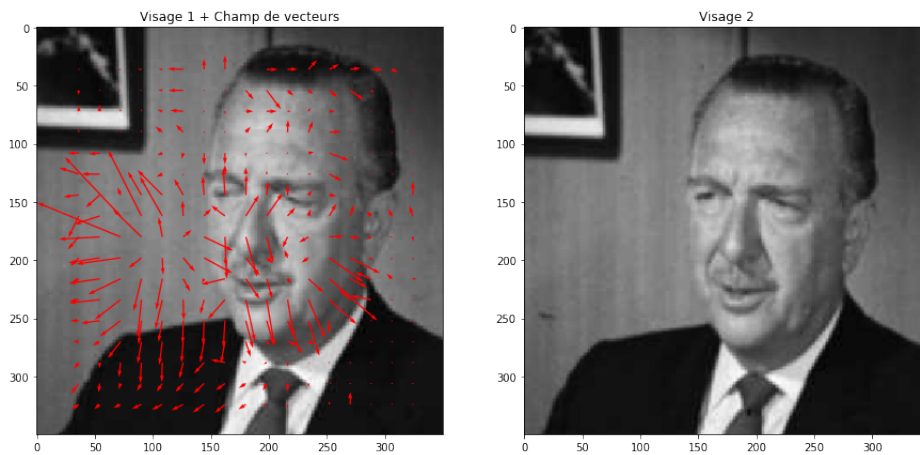


FIGURE 6 – Les vecteurs sur le visage illustrent bien le mouvement de tête, cependant ceux dans le décor sont biaisés. Le fond de l'image uni a perturbé le fonctionnement de l'algorithme.

### 4.2.3 Image 3

La troisième image choisie comporte moins de détails et de nuance de couleurs. C'est une image 200 x 200 pixels.

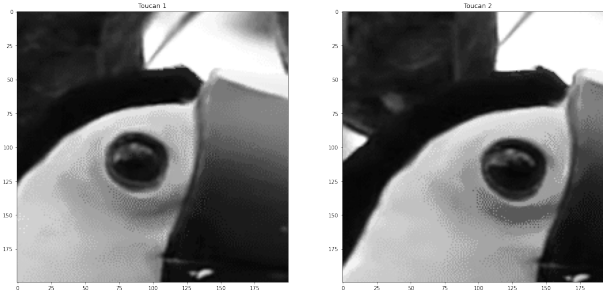
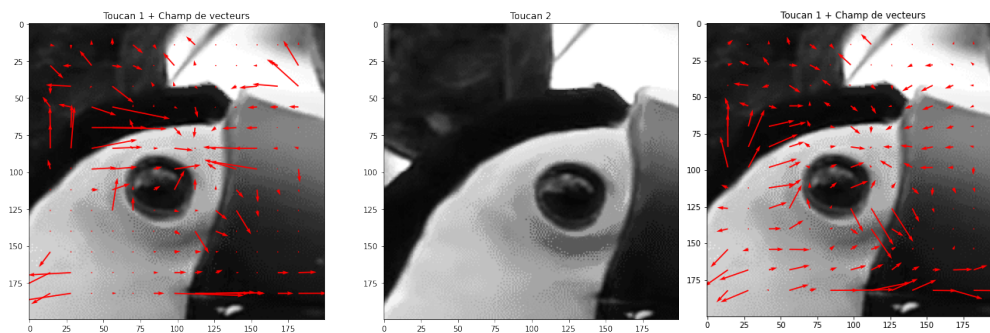


FIGURE 7 – On remarque que le toucan avance sa tête vers la droite.

Comparons maintenant les résultats de l'algorithme du pas optimal pour 10 itérations (image à gauche) et 100 itérations (image à droite).



Concernant le résultat pour 10 itérations, on observe que les vecteurs de mouvements sont soit très grand soit pratiquement inexistant alors que pour le champ de vecteur après 100 itérations, la taille des vecteurs semble plus raisonnable. Dans les deux cas, on observe bien un mouvement vers la droite de l'oeil et de bec mais aussi du pelage pour le champ de vecteur affiché après 100 itérations. L'algorithme de recalage semble encore ici moins efficace sur les aplats de couleurs unis même si le fait de faire 100 itérations améliore cela.

### 4.2.4 Image 4

Comme dernière image on a choisie une grande image de 500 x 500 pixels avec beaucoup de nuance de gris et de détails.

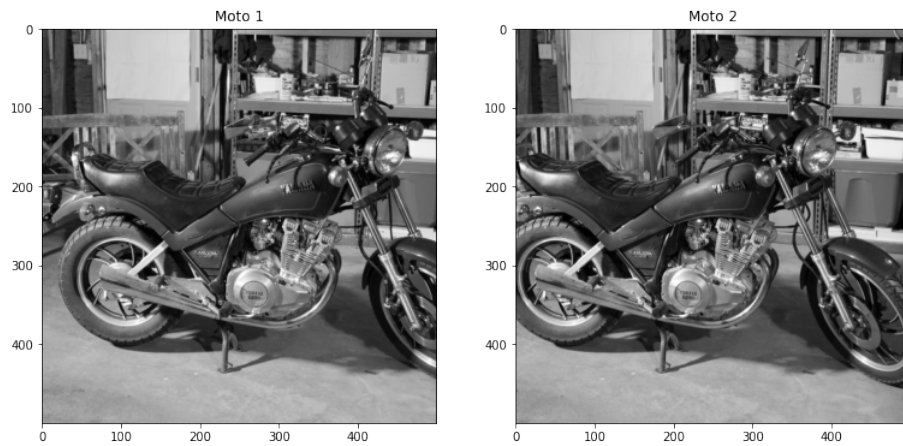


FIGURE 8 – On remarque que la moto recule avec un mouvement vers la gauche de l'image.

Ci-dessous est affiché le champ de vecteurs par-dessus la première image calculée avec 10 itérations.

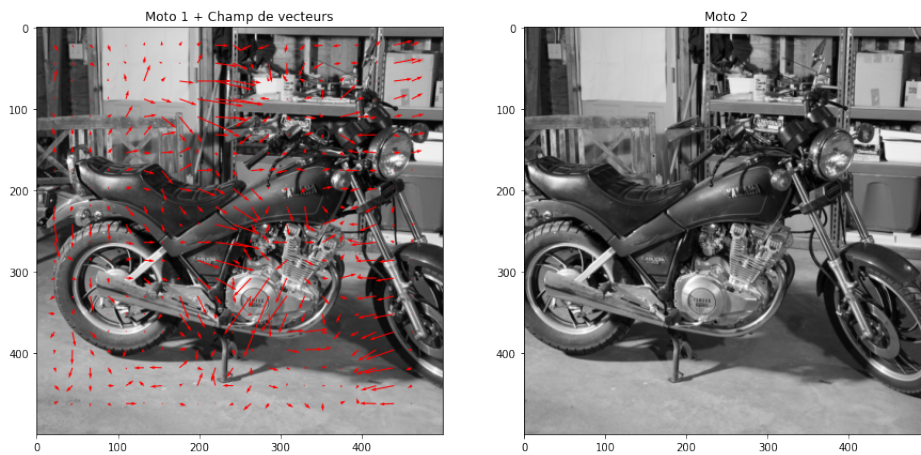
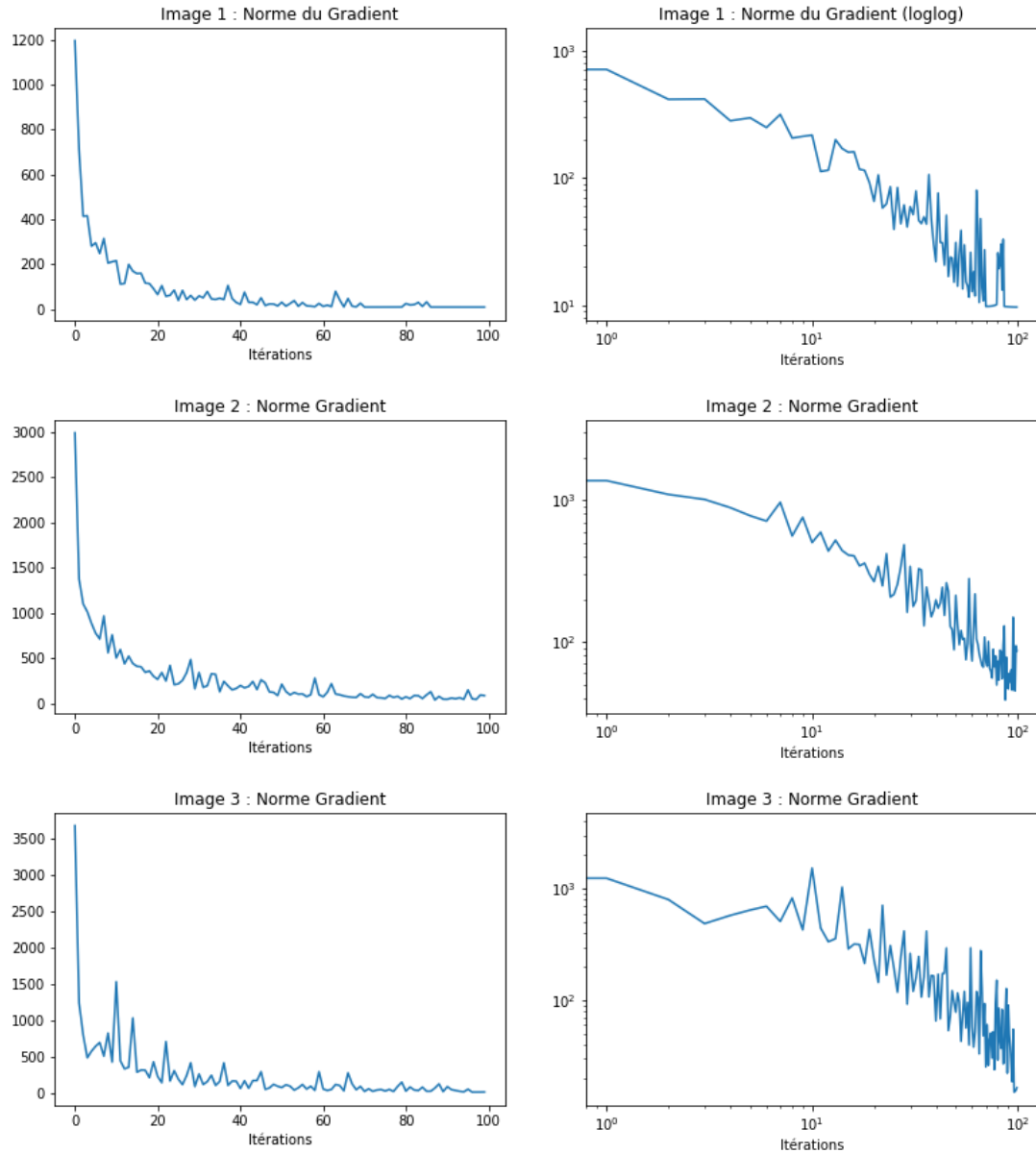


FIGURE 9 – On voit que les vecteurs du devant de la moto, la roue avant et le guidon, indique bien un mouvement vers la gauche. Ceux du moteurs également. Mais les vecteurs du fond montre un mouvement vers la droite qui n'existe pas sur les images.

Un résultat avec 100 itérations rendrait le champ de vecteurs plus précis mais de part la taille et le niveau de détail de l'image, l'algorithme du pas optimal est beaucoup trop lent (20 minutes pour 10 itérations).

### 4.3 Résultats quantitatifs

Pour être certain que nos algorithmes sont efficaces, on a affiché l'évolution de la norme du gradient de la fonction objectif au cours des itérations. De plus, on l'a également affiché en échelle log log pour mieux observer son évolution.



On peut voir que la norme du gradient diminue énormément au cours des itérations, pour les 3 premières images testées. Nous n'avons pas affiché de graphique pour la dernière image car, comme dit plus haut, le calcul de 100 itérations est bien trop long et 10 itérations ne suffisent pas à avoir un graphique de norme décroissante

suffisamment satisfaisant.

Ces résultats nous confirment bien que notre implémentation est correcte et efficace.