

Projet: Image - Signal - Simulation

Recalage d'image $l^1 - l^2$

Walid KACIMI, Yusuf MAKAS, Maria Sofia LOPEZ, Aya BOUKAREF

25 avril 2024

Table des matières

1	Introduction	2
2	Problème à résoudre	2
3	Propriétés du problème	3
3.1	Différentiabilité	3
3.2	Coercivité	3
4	Calculs pour la mise en place de l'algorithme de résolution	3
4.1	Adjoint ∇^*	4
4.2	Adjoint H^*	4
4.3	Gradient ∇E	5
5	Application du Recalage d'Images	6
5.1	Méthodologie de Recalage	6
5.1.1	Préparation des données	6
5.1.2	Paramètres utilisés	6
5.1.3	Application de l'Algorithme	7
5.2	Résultats et Analyses	8
5.2.1	Matrice de Vecteurs Résultante	8
5.2.2	Superposition des Vecteurs sur l'Image Initiale	8
5.2.3	Evaluation visuelle	8
5.3	Discussion des Résultats	8
5.4	Autres images testées	9

1 Introduction

Dans ce projet nous nous intéresserons à un concept important en traitement d'images : au recalage d'images. Il fait référence au motif de mouvement des objets entre les deux images successives, comme par exemple dans un film. Le but principal du recalage est de suivre et d'analyser le déplacement des pixels d'une image à une autre, ce qui permet de comprendre le mouvement et la dynamique de la scène observée.

2 Problème à résoudre

On dispose de deux images v^1 et $v^2 \in (\mathbb{R}^2)^{N^2}$ et on cherche un champs de vecteurs $t = (t_{m,n})_{1 \leq m,n \leq N} = (t_{m,n}^1, t_{m,n}^2)_{1 \leq m,n \leq N} \in (\mathbb{R}^2)^{N^2}$ tel que l'objet représenté dans v^1 en $(m, n) + t_{m,n}$ soit le même que l'objet représenté dans v^2 au pixel (m, n) .

Il existe un grand nombre de méthodes pour approcher le recalage. Dans ce cas on va s'intéresser encore à une méthode d'optimisation pour minimiser une énergie E définie pour $t \in (\mathbb{R}^2)^{N^2}$. On définit E comme une somme d'un terme d'attache aux données et $v_{(m,n)}^2$ et d'un terme de régularisation de t .

Dans notre cas, pour le terme d'attache aux données entre v^1 au point $(m, n) + t_{(m,n)}$ et $v_{(m,n)}^2$, on utilise une approximation affine de v^1 au voisinage de (m, n) . On pose alors :

$$\tilde{v}^1((m, n) + t_{m,n}) = v_{m,n}^1 + \langle \nabla v_{m,n}^1, t_{m,n} \rangle$$

Pour $\varepsilon > 0$, on définit alors notre terme d'attache aux données par :

$$\sum_{m,n=1}^N \sqrt{|v^1((m, n) + t_{m,n}) - v_{m,n}^2|^2 + \varepsilon}$$

On peut alors définir une fonction $\varphi_\varepsilon(t) = \sqrt{t + \varepsilon}$ qui nous donne le terme d'attache aux données de la forme :

$$\sum_{m,n=1}^N \varphi_\varepsilon(|v_{m,n}^1 + \langle \nabla v_{m,n}^1, t_{m,n} \rangle - v_{m,n}^2|^2)$$

Finalement on définit le terme de régularisation du champs de vecteurs t défini par :

$$\sum_{m,n=1}^N |\nabla t_{m,n}^1|^2 + |\nabla t_{m,n}^2|^2.$$

On utilisera l'expression du gradient suivante :

$$\nabla v_{m,n} = (v_{m,n} - v_{m-1,n}, v_{m,n} - v_{m,n-1})$$

Notre énergie s'écrit alors de la façon suivante :

$$E(t) = \sum_{m,n=1}^N \varphi_\varepsilon(|v_{m,n}^1 + \langle \nabla v_{m,n}^1, t_{m,n} \rangle - v_{m,n}^2|^2) + \lambda(|\nabla t_{m,n}^1|^2 + |\nabla t_{m,n}^2|^2)$$

avec $\lambda \in \mathbb{R}$ un paramètre.

On définit enfin notre problème à résoudre :

$$\min_{t \in (\mathbb{R}^2)^{N^2}} E(t)$$

3 Propriétés du problème

3.1 Différentiabilité

On s'intéresse d'abord à la différentiabilité de notre énergie $E(t)$.

Pour notre terme d'attache aux données, le produit scalaire, la somme sont différentiable, de même que la fonction $x \mapsto x^2$. Cependant, on sait que $|.|$ et $\sqrt{\cdot}$ ne sont pas différentiables en 0. Le fait d'utiliser la fonction φ_ε nous permet alors de contourner le problème. En effet en ajoutant $\varepsilon > 0$ le terme $|v^1((m,n) + t_{m,n}) - v_{m,n}^2|^2 + \varepsilon$ ne va jamais s'annuler pour tout $t_{m,n}$. Donc par composition, notre terme d'attache aux données est différentiable sur $(\mathbb{R}^2)^{N^2}$.

Pour notre terme de régularisation, on sait que ∇ est différentiable en tant qu'opérateur linéaire. De même la norme euclidienne $|.|$ est différentiable. Donc notre terme de régularisation est aussi différentiable sur $(\mathbb{R}^2)^{N^2}$.

On a donc

$$t \mapsto \sum_{m,n=1}^N \varphi_\varepsilon \left((\langle \nabla v_{m,n}^1, t_{m,n} \rangle + v_{m,n}^1 - v_{m,n}^2)^2 \right) + \sum_{m,n=1}^N \lambda \left(|\nabla t_{m,n}^1|^2 + |\nabla t_{m,n}^2|^2 \right)$$

differentiable sur $(\mathbb{R}^2)^{N^2}$.

3.2 Coercivité

Il est possible de montrer que ce problème a une solution dès que ∇v^1 peut prendre plusieurs directions. En effet, cette fonctionnelle est différentiable donc elle est continue. Il n'est pas difficile de se convaincre qu'elle est aussi coercive car t ne peut pas être à la fois constant et avoir des produits scalaires nuls avec tous les $\nabla v_{m,n}^1$ si ceux-ci changent de direction.

La différentiabilité et la coercivité nous donnent alors l'existence d'une solution de notre problème :

$$\min_{t \in (\mathbb{R}^2)^{N^2}} E(t)$$

4 Calculs pour la mise en place de l'algorithme de résolution

On va mettre en place un algorithme de descente du gradient, on calcule alors le gradient de notre énergie.

Pour simplifier les calculs on utilise l'opérateur H suivant :

$$\begin{aligned} H : (\mathbb{R}^2)^{N^2} &\longrightarrow \mathbb{R}^{N^2} \\ t &\longmapsto (\langle \nabla v_{m,n}^1, t_{m,n} \rangle)_{1 \leq m, n \leq N}. \end{aligned}$$

On a alors

$$E(t) = \sum_{m,n=1}^N \varphi_\varepsilon \left((H t_{m,n} + v_{m,n}^1 - v_{m,n}^2)^2 \right) + \lambda \|\nabla t^1\|^2 + \lambda \|\nabla t^2\|^2$$

On calcule alors l'adjoint de notre opérateur H . Pour le calculer on aura besoin de notre opérateur ∇ ainsi que son adjoint ∇^* . Calculons alors ∇^* :

$$\begin{aligned} \nabla : \mathbb{R}^{N^2} &\longrightarrow (\mathbb{R}^2)^{N^2} \\ v &\longmapsto \begin{pmatrix} \partial_x v_{m,n} \\ \partial_y v_{m,n} \end{pmatrix}_{1 \leq m, n \leq N} \end{aligned}$$

4.1 Adjoint ∇^*

On cherche ∇^* tel que $\langle \nabla v, t \rangle = \langle v, \nabla^* t \rangle$

$$\begin{aligned}\langle \nabla v, t \rangle &= \sum_{m,n=1}^N \partial_x v_{m,n} t_{m,n}^1 + \sum_{m,n=1}^N \partial_y v_{m,n} t_{m,n}^2 \\ &= \langle \partial_x v, t^1 \rangle + \langle \partial_y v, t^2 \rangle \\ &= \langle v, \partial_x^* t^1 \rangle + \langle v, \partial_y^* t^2 \rangle \\ &= \langle v, \nabla^* t \rangle,\end{aligned}$$

avec $\nabla^* t = \partial_x^* t^1 + \partial_y^* t^2$.

On va alors chercher ∂_x^* et ∂_y^* de la même façon :

$$\begin{aligned}\langle \partial_x v, w \rangle &= \sum_{m,n=1}^N (\partial_x v)_{m,n} w_{m,n} \\ &= \sum_{m,n=1}^N (v_{m,n} - v_{m-1,n}) w_{m,n} \\ &= \sum_{m,n=1}^N v_{m,n} w_{m,n} - \sum_{n=1}^N \sum_{m'=0}^{N-1} v_{m',n} w_{m'+1,n} \\ &= \sum_{m,n=1}^N v_{m,n} w_{m,n} - \sum_{m,n=1}^N v_{m,n} w_{m+1,n} \\ &= \sum_{m,n=1}^N v_{m,n} (w_{m,n} - w_{m+1,n}) \\ &= \langle v, \partial_x^* w \rangle,\end{aligned}$$

en faisant un changement d'indice $m' = m - 1$ et avec la périodicité de w .

Donc $(\partial_x^* w)_{m,n} = w_{m,n} - w_{m+1,n}$.

En suivant le même raisonnement on a aussi : $(\partial_y^* w)_{m,n} = w_{m,n} - w_{m,n+1}$.

Donc si on revient à l'expression de notre adjoint ∇^* on a :

$$\nabla^* t_{m,n} = \partial_x^* t^1 + \partial_y^* t^2 = t_{m,n}^1 - t_{m+1,n}^1 + t_{m,n}^2 - t_{m,n+1}^2$$

4.2 Adjoint H^*

On continue alors à chercher notre adjoint H^* :

$$\begin{aligned}
\langle Ht, u \rangle &= \sum_{m,n=1}^N (Ht)_{m,n} u_{m,n} \\
&= \sum_{m,n=1}^N \langle \nabla v_{m,n}^1, t_{m,n} \rangle u_{m,n} \\
&= \sum_{m,n=1}^N (\partial_x v_{m,n}^1 t_{m,n}^1 + \partial_y v_{m,n}^1 t_{m,n}^2) u_{m,n} \\
&= \sum_{m,n=1}^N t_{m,n}^1 (\partial_x v_{m,n}^1 u_{m,n}) + \sum_{m,n=1}^N t_{m,n}^2 (\partial_y v_{m,n}^1 u_{m,n}) \\
&= \langle t, H^* u \rangle,
\end{aligned}$$

Donc $(H^* u)_{m,n} = (\partial_x v_{m,n}^1 u_{m,n}, \partial_y v_{m,n}^1 u_{m,n})$.

4.3 Gradient ∇E

Calculons finalement ∇E :

Soit $t := (t^1, t^2), h := (h^1, h^2) \in (\mathbb{R}^2)^{N^2}$.

$$\begin{aligned}
E(t+h) &= \sum_{m,n=1}^N \varphi_\varepsilon \left(((Ht)_{m,n} + (Hh)_{m,n} + v_{m,n}^1 - v_{m,n}^2)^2 \right) + \lambda \|\nabla t^1 + \nabla h^1\|^2 + \lambda \|\nabla t^2 + \nabla h^2\|^2 \\
E(t+h) &= \sum_{m,n=1}^N \varphi_\varepsilon \left(((Ht)_{m,n} + v_{m,n}^1 - v_{m,n}^2)^2 \right) + \varphi'_\varepsilon \left(((Ht)_{m,n} + v_{m,n}^1 - v_{m,n}^2)^2 \right) \times 2 ((Ht)_{m,n} + v_{m,n}^1 - v_{m,n}^2) (Hh)_{m,n} \\
&\quad + o(|h_{m,n}|) + \lambda \|\nabla t^1 + \nabla h^1\|^2 + \lambda \|\nabla t^2 + \nabla h^2\|^2
\end{aligned}$$

car $\forall a \in \mathbb{R}_+$ et $\forall b \in \mathbb{R} : \varphi(a+b) = \varphi(a) + \varphi'(a)b + o(|b|)$, en prenant $\varphi = \varphi_\varepsilon(\cdot^2)$ et $b = (Hh)_{m,n}$.

De plus on a :

$$\|\nabla t^1 + \nabla h^1\|^2 = \|\nabla t^1\|^2 + \|\nabla h^1\|^2 + 2 \langle \nabla t^1, \nabla h^1 \rangle$$

et

$$\|\nabla t^2 + \nabla h^2\|^2 = \|\nabla t^2\|^2 + \|\nabla h^2\|^2 + 2 \langle \nabla t^2, \nabla h^2 \rangle$$

D'où

$$\begin{aligned}
E(t+h) &= \sum_{m,n=1}^N \varphi_\varepsilon \left(((Ht)_{m,n} + v_{m,n}^1 - v_{m,n}^2)^2 \right) + \varphi'_\varepsilon \left(((Ht)_{m,n} + v_{m,n}^1 - v_{m,n}^2)^2 \right) \times 2 ((Ht)_{m,n} + v_{m,n}^1 - v_{m,n}^2) (Hh)_{m,n} \\
&\quad + \lambda \left(\|\nabla t^1\|^2 + \|\nabla h^1\|^2 + 2 \langle \nabla t^1, \nabla h^1 \rangle + \|\nabla t^2\|^2 + \|\nabla h^2\|^2 + 2 \langle \nabla t^2, \nabla h^2 \rangle \right) + o(\|h\|)
\end{aligned}$$

On pose alors : $X = 2\varphi'_\varepsilon \left(((Ht)_{m,n} + v_{m,n}^1 - v_{m,n}^2)^2 \right) ((Ht)_{m,n} + v_{m,n}^1 - v_{m,n}^2) \in \mathbb{R}^{N^2}$.

et donc :

$$\begin{aligned} E(t+h) &= E(t) + \langle X, Hh \rangle + 2\lambda \langle \nabla t^1, \nabla h^1 \rangle + 2\lambda \langle \nabla t^2, \nabla h^2 \rangle + o(\|h\|) \\ &= E(t) + \langle H^*X, h \rangle + 2\lambda \langle \nabla^*\nabla t^1, h^1 \rangle + 2\lambda \langle \nabla^*\nabla t^2, h^2 \rangle + o(\|h\|) \end{aligned}$$

D'où on a finalement :

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

5 Application du Recalage d'Images

5.1 Méthodologie de Recalage

Dans cette section, nous détaillons l'application et les résultats pour le recalage d'images. Nous expliquons les étapes clés, de la préparation des données à l'application de l'algorithme du gradient, en mettant l'accent sur les paramètres utilisés et la validation des résultats.

5.1.1 Préparation des données

Dans cette étape, nous commençons par rassembler les données nécessaires pour le recalage d'images. Pour illustrer le processus, nous utiliserons deux images en niveaux de gris : car1_bw.jpg et car2_bw.jpg. Il s'agit de deux images successives qui sont proches. Pour les figures 1 et 2 on voit le déplacement de quelques voitures.

Une fois les images acquises, nous procédons à une étape de prétraitement pour les mettre dans un format approprié pour le recalage. Dans notre cas les deux images étaient en couleur, on a donc fait une conversion en niveaux de gris numérique puis une opération de redimensionnement. On va se concentrer sur une partie de l'image, avec une des voitures où le mouvement est le plus évident, afin de limiter le nombre de calculs.

L'objectif de cette étape de préparation est de garantir que les deux images sont comparables et que le processus de recalage peut être appliqué de manière appropriée.



FIGURE 1 – Image car1_bw.jpg



FIGURE 2 – Image car2_bw.jpg

5.1.2 Paramètres utilisés

Dans cette section, nous détaillons les différents paramètres utilisés dans notre algorithme de recalage d'images.

- **Paramètre epsilon (ε) :** Le paramètre epsilon est utilisé pour assurer la différentiabilité de notre énergie, on le choisit petit avec $\varepsilon = 0, 1$.



FIGURE 3 – Image redimensionnée de car1_bw.jpg

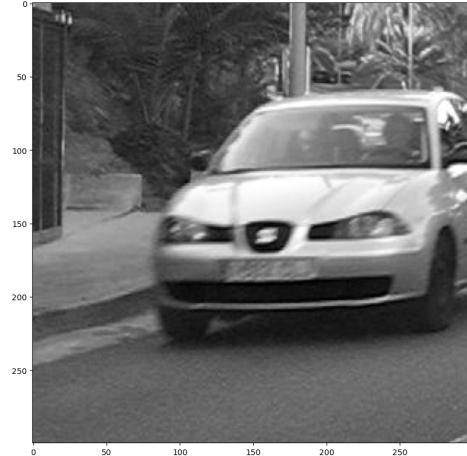


FIGURE 4 – Image redimensionnée de car1_bw.jpg

- **Paramètre lambda (λ) :** Le paramètre lambda est utilisé pour contrôler l'importance de la régularisation par rapport aux données dans l'énergie de l'algorithme de recalage. Plus λ est grand, plus le terme de régularisation a d'importance (et donc t est régulier). Plus λ est petit, plus le terme d'attache aux données a d'importance (au risque d'avoir des problèmes pour recaler les objets dont l'intensité a changé). Nous avons choisi $\lambda = 1$ dans nos expériences, ce qui nous a permis d'obtenir un équilibre satisfaisant entre régularisation et fidélité aux données.
- **Pas de l'optimisation :** Le pas de l'optimisation contrôle la taille des étapes prises lors de la descente de gradient dans notre algorithme d'optimisation. Un pas trop grand peut entraîner une divergence de l'algorithme, tandis qu'un pas trop petit peut ralentir la convergence. Nous avons choisi un pas de 0.01 dans nos expériences, ce qui a assuré une convergence rapide tout en évitant les problèmes de divergence.
- **Nombre d'itérations :** Le nombre d'itérations détermine le nombre de fois que l'algorithme d'optimisation est appliqué pour ajuster les paramètres du modèle de recalage. Un nombre d'itérations insuffisant peut entraîner une convergence prématûre et des résultats sous-optimaux, tandis qu'un nombre d'itérations excessif peut entraîner un surajustement aux données d'entraînement. Nous avons choisi 100 itérations dans nos expériences, ce qui nous a permis d'obtenir des résultats satisfaisants sans surajustement.

5.1.3 Application de l'Algorithme

1. **Application de l'algorithme :** Nous appliquons notre algorithme de recalage d'images en utilisant les images prétraitées et les paramètres choisis. L'algorithme du gradient est utilisé pour minimiser la fonction d'énergie E , qui mesure la différence entre les images alignées et les images cibles. Mathématiquement, l'algorithme du gradient fonctionne comme suit :

Algorithme 1 Algorithme du gradient

Entrée: Les entrées nécessaires au calcul de E et de ∇E

Sortie: Une approximation d'un minimiseur de E : w

```

Initialisation de  $w$ 
Tant que l'algorithme n'a pas convergé faire
    Calculer  $d = \nabla E(w)$ 
    Calculer un pas  $t \geq 0$ 
    Mettre à jour :  $w \leftarrow w - t d$ 
Fin tant que

```

FIGURE 5 – Algorithme du gradient pour la minimisation de la fonction d'énergie E

Ce processus est répété jusqu'à ce qu'une convergence satisfaisante soit atteinte.

2. **Évaluation des résultats :** Une fois l'algorithme de recalage terminé, nous évaluons les résultats obtenus en utilisant des mesures de performance telles que l'énergie de recalage et la qualité visuelle des images alignées.

5.2 Résultats et Analyses

5.2.1 Matrice de Vecteurs Résultante

Nous commençons par afficher la matrice de vecteurs obtenue après la minimisation de la fonction d'énergie E avec l'algorithme du gradient. Cette matrice de vecteurs est représentée graphiquement dans l'image `vecteurcar1a2.png`.

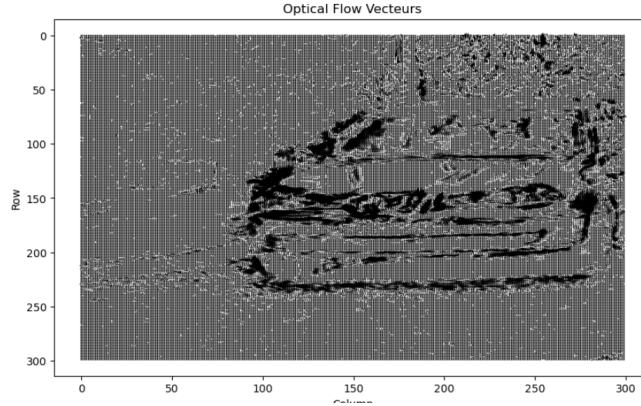


FIGURE 6 – Matrice de vecteurs résultante après minimisation de E

5.2.2 Superposition des Vecteurs sur l'Image Initiale

Ensuite, nous superposons cette matrice de vecteurs sur l'image initiale (`car1_bwaa.png`) afin de visualiser le décalage obtenu avec notre algorithme. Cela nous permet de vérifier visuellement si le décalage correspond bien à l'image cible.

À partir de cette superposition, nous pouvons analyser la qualité du recalage et déterminer si les décalages obtenus correspondent bien à l'image cible, permettant ainsi une évaluation visuelle de la performance de notre algorithme.

5.2.3 Evaluation visuelle

Il est clair que les vecteurs en rouge correspondent au mouvement de la deuxième image de voiture. Pour construire une deuxième image intermédiaire à partir de nos images, nous utiliserons la méthode d'interpolation au plus proche voisin, où les coordonnées de chaque pixel de l'image résultante seront obtenues en ajoutant les composantes entières des vecteurs de la solution approximative à celles de l'image initiale :

5.3 Discussion des Résultats

Après 100 itérations, l'algorithme semble converger efficacement, mais sa compilation prend du temps en raison du grand nombre de calculs de produits scalaires et de normes sur les images, notamment en raison du nombre élevé de pixels (ce qui explique pourquoi nous avons redimensionné les images). Cependant, malgré cette contrainte, nous obtenons des vecteurs de petite taille pour chaque pixel qui semblent bien suivre le mouvement vers la deuxième image.



FIGURE 7 – Image initiale (`car1_bwaa.png`) avec superposition des vecteurs de recalage

5.4 Autres images testées

Dans cette section, nous présentons les résultats de notre algorithme de recalage sur d'autres séquences d'images.

Ensuite, nous superposons cette matrice de vecteurs sur la première image (`frame11.png`) pour visualiser le décalage obtenu par l'algorithme de recalage.



FIGURE 8 – Image initiale décalé à l'aide d'interpolation au plus proche voisin



FIGURE 9 – Image frame11.png

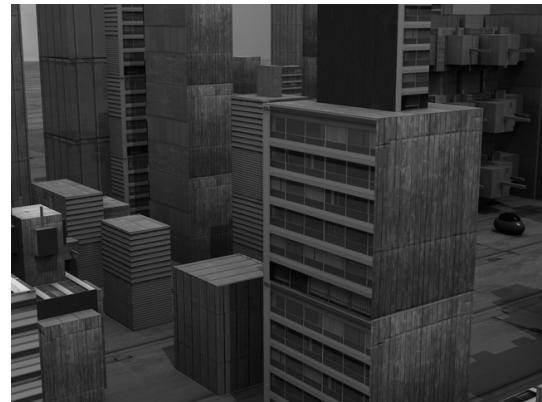


FIGURE 10 – Image frame12.png

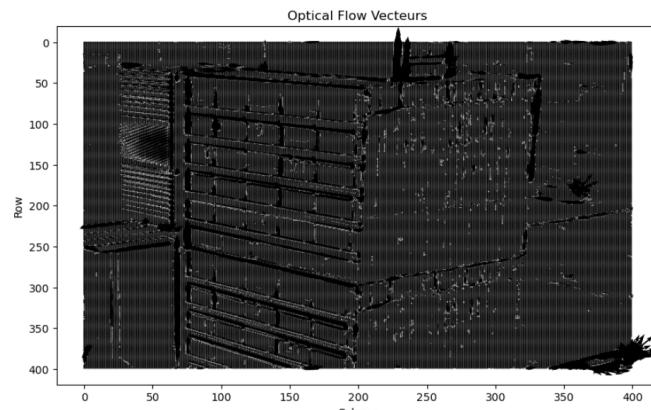


FIGURE 11 – Matrice de vecteurs résultante pour la paire d'images frame11.png et frame12.png

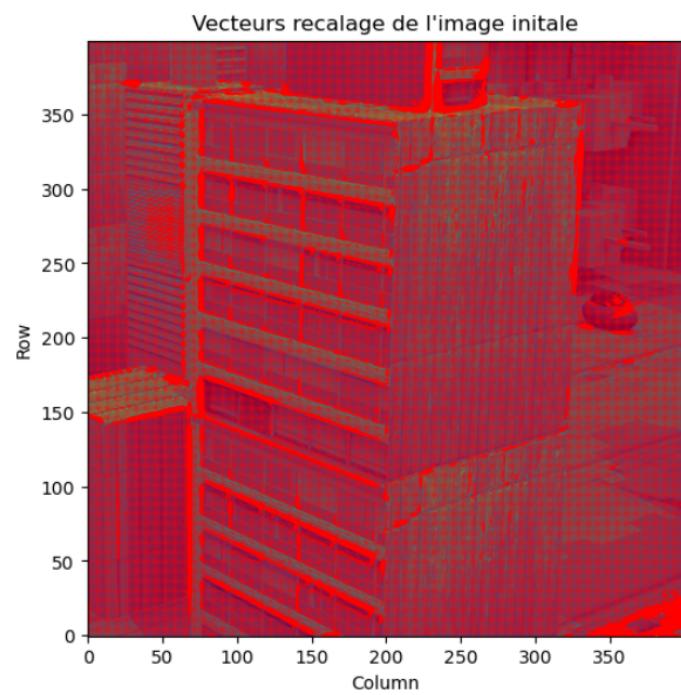


FIGURE 12 – Superposition des vecteurs de recalage pour la paire d'images `frame11.png` et `frame12.png`