



**INSTITUT
POLYTECHNIQUE
DE PARIS**

DÉPARTEMENT DE PREMIÈRE ANNÉE

REDIMENSIONNEMENT D'IMAGES PAR SEAM-CARVING

CHENG Léo – GARNAUD Léo – HOSSAINY Malak – SONDALI Soufiane

sous la direction de

M. MONASSE Pascal

IMAGINE/LIGM, Ecole nationale des ponts et chaussées, Université Paris-Est, France

Table de matière

1. Introduction
2. État de l'art
3. Implémentation et Méthodologie
4. Résultats et Expérimentations
5. Conclusion

Table de matière

- 1. Introduction**
2. État de l'art
3. Implémentation et Méthodologie
4. Résultats et Expérimentations
5. Conclusion

Contexte et enjeux

- Explosion des usages d'images (web, mobile, impression).
- Besoin d'adapter la taille des images **sans perdre de contenu**.
- Limites des méthodes classiques : redimensionnement global, recadrage...



a) Image originale



b) Recadrage



c) Redimensionnement global

Problématique et objectifs

Problématique :

Comment redimensionner une image tout en préservant ses régions visuellement importantes ?

Objectifs :

- Comprendre l'algorithme *Seam Carving* (**Avidan & Shamir**, 2007).
- Implémenter une solution performante en C++.
- Evaluer plusieurs fonctions d'énergie.
- Explorer des applications avancées : agrandissement, suppression d'objet, etc.

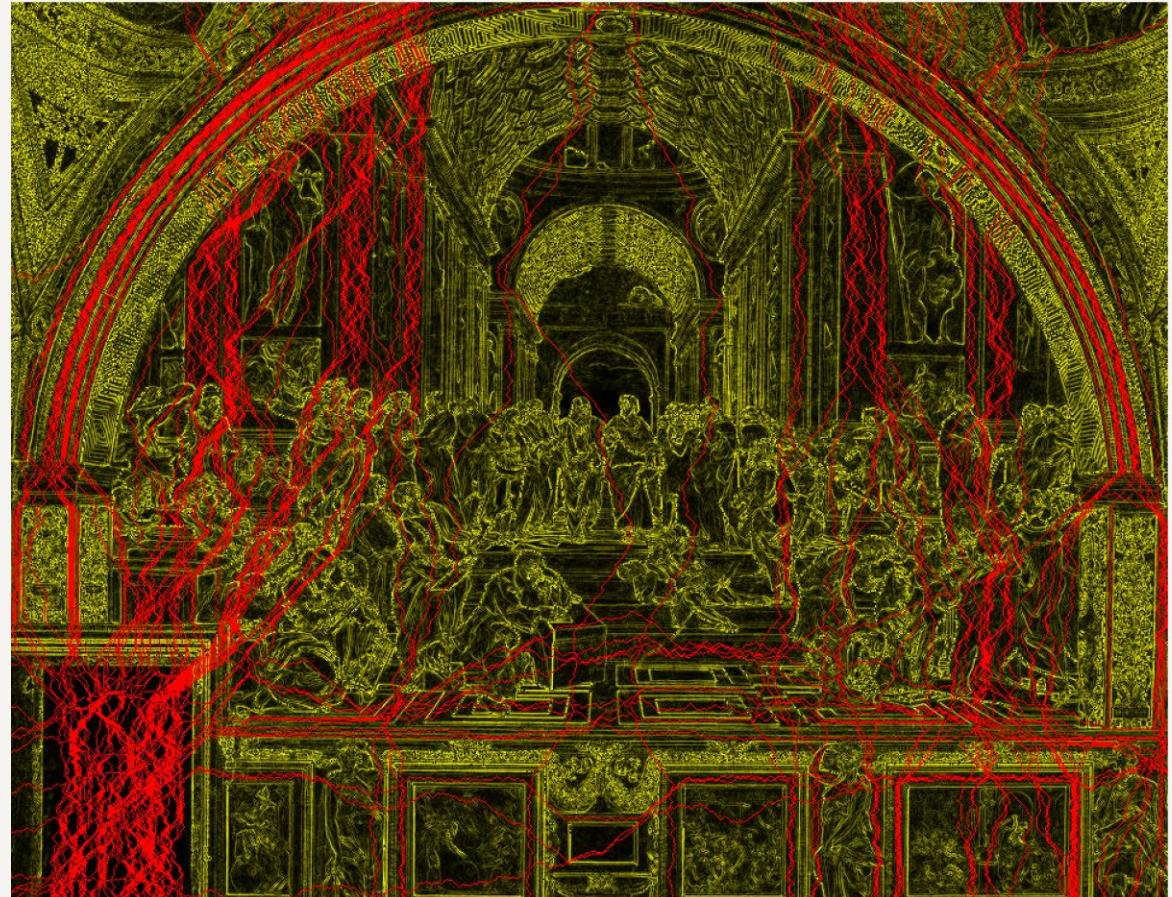
Table de matière

1. Introduction
- 2. État de l'art**
3. Implémentation et Méthodologie
4. Résultats et Expérimentations
5. Conclusion

Seam Carving – Redimensionnement d'Images Orienté Contenu

Principe :

Méthode de redimensionnement qui insère ou supprime des *coutures* (seams) selon l'importance visuelle locale, évaluée par une **fonction d'énergie**.



Seam Carving – Définitions clés

Couture :

Chemin continu de pixels (vertical ou horizontal) affectant le moins possible les zones importantes de l'image.

- Couture **verticale** : une suite $(i, c(i))$ telle que $|c(i)-c(i-1)| \leq 1$.
- Couture **horizontale** : une suite $(c(j), j)$ telle que $|c(j)-c(j-1)| \leq 1$.

Fonction d'énergie :

Fonction $E : [0, n-1] \times [0, m-1] \rightarrow \mathbb{R}^+$ qui mesure l'importance d'un pixel.

L'idée clé est donc de supprimer les coutures de plus faible énergie cumulée pour préserver les zones importantes.

Algorithme de sélection optimale – Couture verticale (DP)

Objectif :

Trouver la couture verticale minimisant l'énergie totale :

$$\min_c \sum_{i=0}^{n-1} E(i, c(i))$$

Étapes de la programmation dynamique (DP) :

1. Construction d'une matrice de coût cumulatif :

$$\forall j \in \llbracket 0, m - 1 \rrbracket, \begin{cases} M(0, j) = E(0, j) \\ M(i, j) = E(i, j) + \min [M(i - 1, j - 1), M(i - 1, j), M(i - 1, j + 1)] \end{cases}, \quad \forall i \in \llbracket 1, n - 1 \rrbracket$$

avec la convention $M(i - 1, -1) = M(i - 1, m) = +\infty$.

2. Remontée du chemin depuis le pixel de plus faible énergie sur la dernière ligne en suivant, à chaque ligne, le minimum local parmi les trois voisins de la ligne du dessus (si ceux-ci existent).

Algorithme de sélection optimale – Couture verticale (DP)

Algorithm 1: Recherche de la couture verticale optimale

Data: Image d'énergie E de taille $n \times m$
Réultat: Couture verticale optimale

```
1 for  $j \leftarrow 1$  to  $m$  do
2    $M(0, j) = E(0, j)$ 
3 for  $i \leftarrow 2$  to  $n$  do
4   for  $j \leftarrow 1$  to  $m$  do
5      $M(i, j) \leftarrow E(i, j) + \min [M(i - 1, j - 1), M(i - 1, j), M(i - 1, j + 1)]$  (en respectant les bords)
6    $j_{\min} \leftarrow \arg \min_j M(n, j)$ 
7    $seam[n] \leftarrow j_{\min}$ 
8 for  $i \leftarrow n - 1$  to  $1$  do
9    $j \leftarrow seam[i + 1]$ 
10   $seam[i] \leftarrow j'$  tel que  $M(i, j')$  est minimal parmi
11     $M(i, j - 1), M(i, j), M(i, j + 1)$  (indices valides uniquement)
12 return  $seam$ 
```

Table de matière

1. Introduction
2. État de l'art
- 3. Implémentation et Méthodologie**
4. Résultats et Expérimentations
5. Conclusion

Fonction d'énergie : Norme L¹ du gradient de l'intensité I

Définition (image en niveaux de gris) :

$$\forall i, j \in \llbracket 0, n - 1 \rrbracket \times \llbracket 0, m - 1 \rrbracket, \quad E_1(i, j) = \left| \frac{\partial I}{\partial x}(i, j) \right| + \left| \frac{\partial I}{\partial y}(i, j) \right|$$

Approximations discrètes (différences centrées) :

$$\forall i, j \in \llbracket 0, n - 1 \rrbracket \times \llbracket 0, m - 1 \rrbracket, \quad \frac{\partial I}{\partial x}(i, j) \simeq \frac{I(i, j + 1) - I(i, j - 1)}{2} \quad \text{et} \quad \frac{\partial I}{\partial y}(i, j) \simeq \frac{I(i + 1, j) - I(i - 1, j)}{2}$$

En bordure : le **voisin manquant** = **pixel lui-même**

Pour les images en couleurs : Calcul séparé sur les 3 canaux R, G, B

- Énergie globale = **somme des énergies des 3 canaux**

Fonction d'énergie : Norme L² du gradient de l'intensité I

On définit la fonction d'énergie L² comme suit :

$$E_2(i, j) = \sqrt{\left(\frac{\partial I}{\partial x}(i, j)\right)^2 + \left(\frac{\partial I}{\partial y}(i, j)\right)^2}$$

→ Norme euclidienne du **gradient de l'image** I.

- Utilise des **dérivées partielles** approximées numériquement (similaires à celles de la norme L¹).
- Même traitement pour les **pixels en bordure** de l'image que celui de la norme L¹.

Fonction d'énergie : *Histogram of Oriented Gradients* HoG

1. Définir une fenêtre (ex : 10×10) centrée sur chaque pixel (i, j) .

2. Calcul du gradient en chaque pixel de la fenêtre ainsi que:

- Orientation : angle du vecteur gradient.
- Magnitude : norme euclidienne (force du gradient).

3. Construction d'un histogramme local sur la fenêtre autour du pixel :

- Les orientations sont réparties dans des bins (ex : 9 bins) couvrant 360 degrés.
- Chaque pixel de la fenêtre contribue à son bin en fonction de sa magnitude.

4. Définition de l'énergie HoG au pixel (i, j) :

$$E_{\text{HoG}}(i, j) = \frac{E_1(i, j)}{\max [\text{HoG}(I(i, j))]}$$

où E_1 est la norme L^1 du gradient en (i, j) , et **HoG** l'histogramme local normalisé autour du pixel (i, j) .

Remarque : Pour les bords de l'image, on exclut les pixels dont la fenêtre sortirait des dimensions.

Fonction d'énergie : Entropie du gradient de I

1. Définir une fenêtre (ex : 10×10) centrée sur chaque pixel (i, j) .
2. Calcul du gradient, de l'orientation et de la magnitude en chaque pixel de la fenêtre.
3. Construction d'un histogramme des orientations sur la fenêtre autour du pixel :
 - Les magnitudes sont réparties dans N_b bins selon les orientations.
 - L'histogramme est ensuite normalisé (somme des bins = 1).
4. Calcul de l'entropie de l'histogramme.
où p_k est la probabilité d'occurrence associée au bin k .
5. Définir l'énergie totale du pixel :

$$H(i, j) = - \sum_{k=0}^{N_b-1} p_k \log(p_k)$$

$$E_{\text{entropie}}(i, j) = E_1(i, j) + H(i, j)$$

Remarque : Les pixels en bordure d'image sont traités comme dans HoG (fenêtre réduite si nécessaire).

Comparaison entre fonctions d'énergies - 1 -



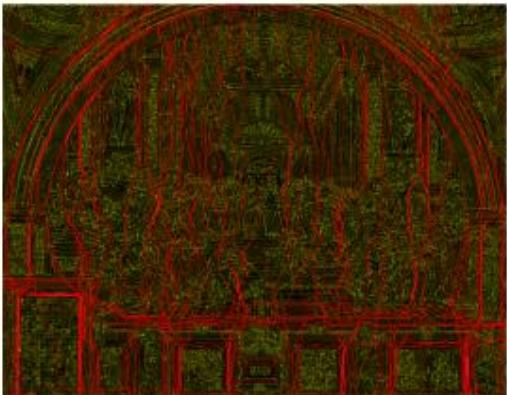
(a) Suppression de 100 coutures en utilisant la fonction d'énergie HoG.



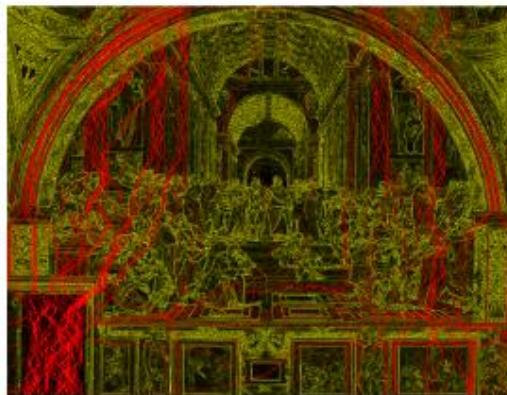
(b) Suppression de 100 coutures en utilisant la fonction d'énergie norme L^1 .



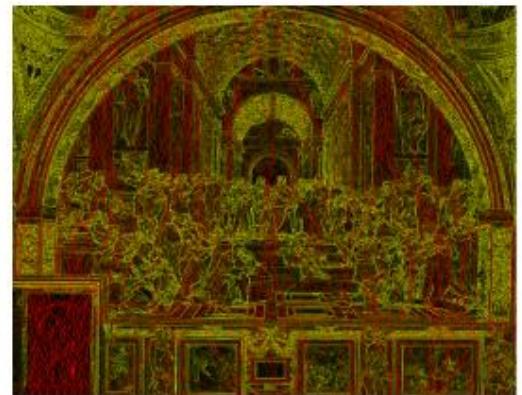
(c) Suppression de 100 coutures en utilisant la fonction d'énergie entropie.



(d) Affichage des coutures selon la fonction d'énergie HoG.



(e) Affichage des coutures selon la fonction d'énergie norme L^1 .



(f) Affichage des coutures selon la fonction d'énergie entropie.

Comparaison entre fonctions d'énergies - 2 -



(a) Suppression de 200 coutures en utilisant la fonction d'énergie entropie.



(b) Suppression de 200 coutures en utilisant la fonction d'énergie HoG.



(c) Suppression de 200 coutures en utilisant la fonction d'énergie norme L^1 .



(d) Suppression de 200 coutures en utilisant la fonction d'énergie norme L^2 .

Méthode	Résultat sur visages	Observations
L^1	Médiocre	Ignore les traits doux
L^2	Excellent	Bon compromis entre contraste et douceur
HOG	Très mauvais	Supprime les zones faciales importantes
Entropie	Bon	Protège les textures riches, complémentaire à L^2

Choix d'implémentation - 1 - : Méthode classique

On applique l'algorithme précédent pour chaque couture. Pour actualiser le tableau d'énergie, on modifie l'énergie des voisins des coutures. Soit (i,j) un pixel appartenant à une couture. On lui fixe une énergie négative arbitrairement. Puis on regarde l'énergie de chacun de ses voisins, on regarde si leur énergie est négative. Le premier pixel d'énergie positive devient le nouveau voisin du pixel (i,j) .

- Complexité temporelle : $O(k^2mn)$

Choix d'implémentation - 2 - : Méthode pointeurs

- Accès aux nouveaux pixels voisins en temps constant.
- Mise à jour du tableau d'énergie avec une complexité linéaire en la longueur de la couture.
- Mais occupation mémoire plus grande.
- Complexité temporelle : $O(kmn)$

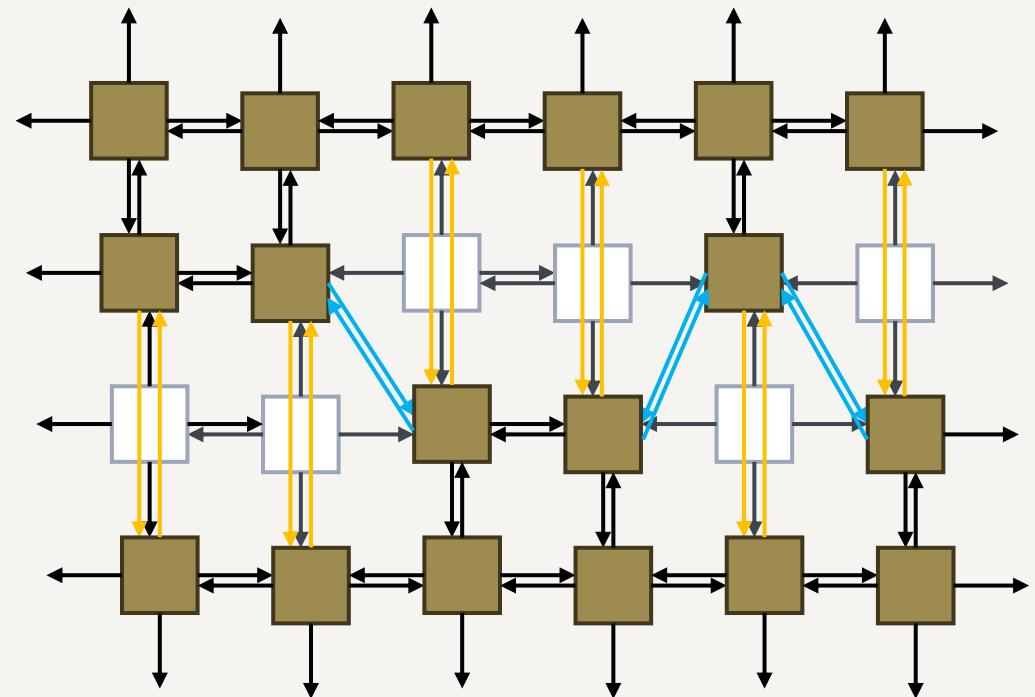


Table de matière

1. Introduction
2. État de l'art
3. Implémentation et Méthodologie
- 4. Résultats et Expérimentations**
5. Conclusion

Réduction d'images



a) Image originale

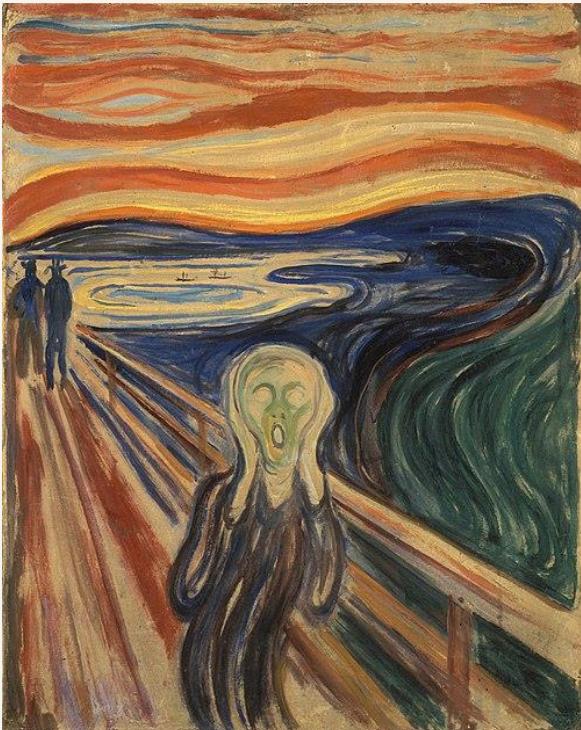


b) Résultat après réduction

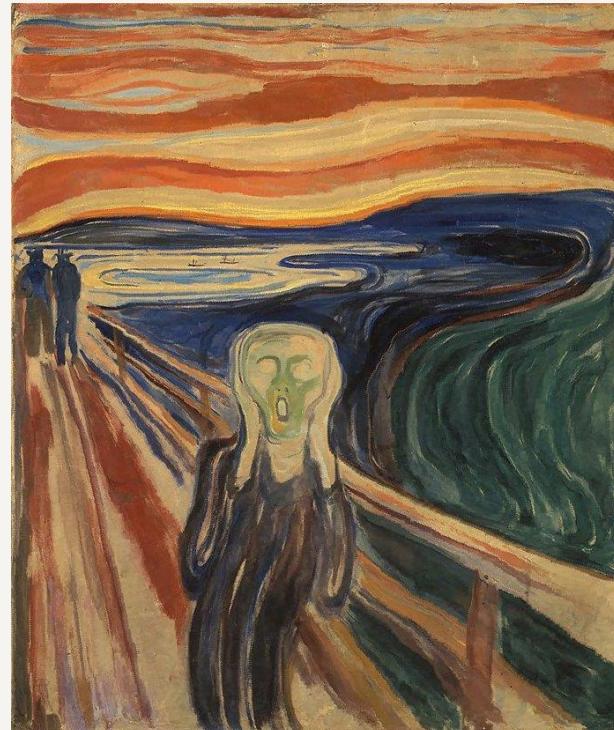
Agrandissement d'images

Identifier les **k premières coutures** à supprimer, ensuite procéder :

- **soit** par **duplication** directe des coutures identifiées.
- **soit** en insérant une couture calculée comme la **moyenne des pixels** de la couture et de ses voisins.



a) Image originale



b) Résultat après agrandissement

Amplification d'images

- Réduction d'abord de l'image, puis agrandissement du **même nombre** de coutures supprimées.
- Augmenter les proportions **des éléments saillants** de l'image.



a) Image originale



b) Résultat après amplification

Reciblage (ex : visages humains)

- Certaines zones importantes (ex : **visages humains**) peuvent être supprimées si elles ont un faible contraste.
- L'utilisateur sélectionne les zones importantes à préserver.
- Modification de la **matrice d'énergie** pour augmenter artificiellement le gradient dans ces zones.
- Les coutures sont ainsi contraintes à **contourner** ces régions clés.



a) Image originale



b) Réduction sans reciblage



c) Réduction avec reciblage

Suppression d'objets guidée par l'utilisateur

- L'utilisateur sélectionne la **zone à supprimer**.
- Modification de la matrice d'énergie pour **attirer les coutures : Gradient artificiellement négatif** dans la zone ciblée.



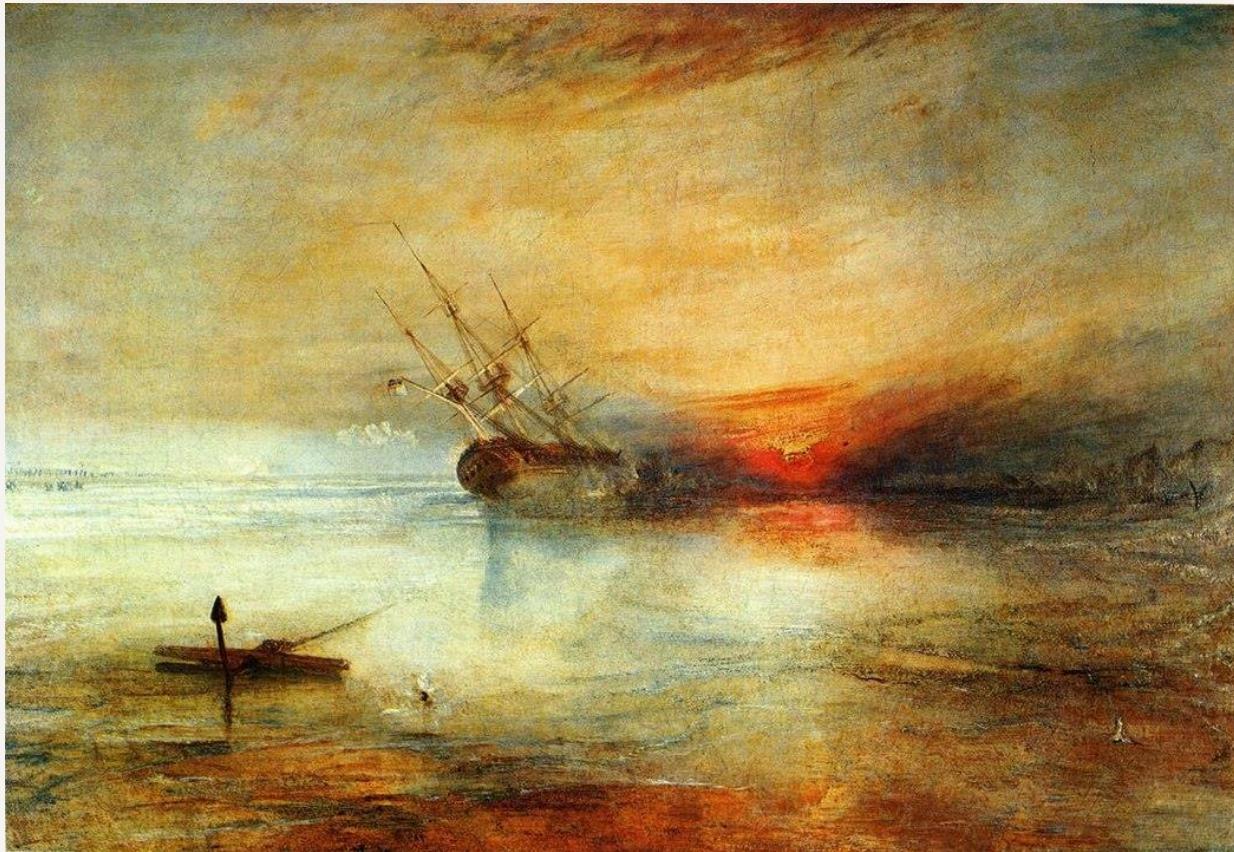
a) Image originale



b) Différentes images dont on a supprimé des personnages

Reconstruction de POISSON

- **Contraste plus faible** que celui laissé par le redimensionnement.
- Utile en **post-traitement correctif**, mais efficacité dépend des artefacts initiaux.



a) Image originale



b) Image réduite



c) Image réduite
puis reconstruite
à l'aide d'un
éditeur de Poisson

Gestion des multi-images

- Séparer calcul et affichage.
- Permettre une utilisation fluide et adaptative.

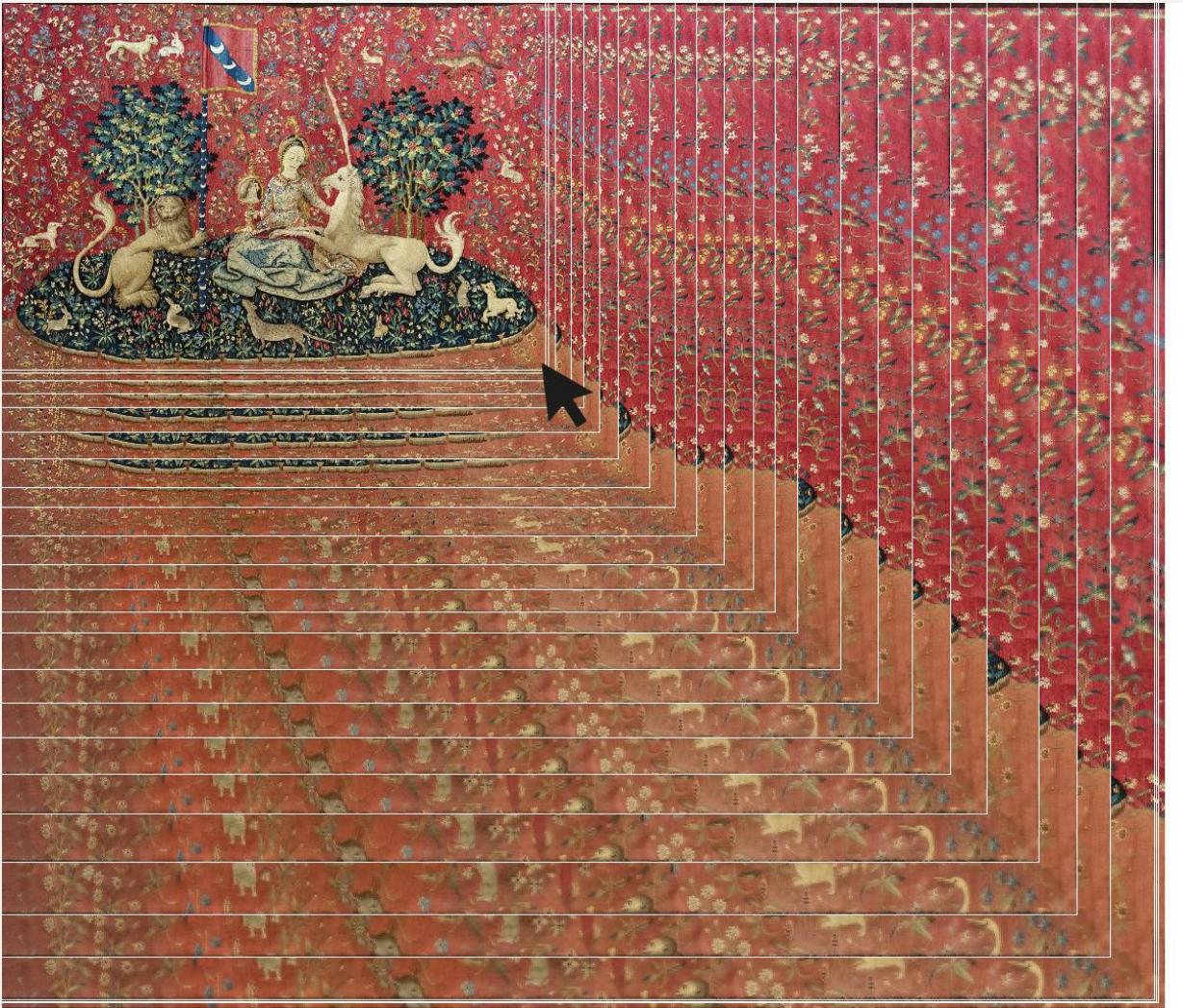


Table de matière

1. Introduction
2. État de l'art
3. Implémentation et Méthodologie
4. Résultats et Expérimentations
- 5. Conclusion**

Limites du Seam Carving - 1 -

Sur les images contenant des **motifs répétitifs** (carrelages, textures, etc.) :

- Les coutures peuvent alors **fragmenter visuellement** ces éléments.
- Résultat : perte de **cohérence spatiale** et dégradation perceptible.



a) Image originale

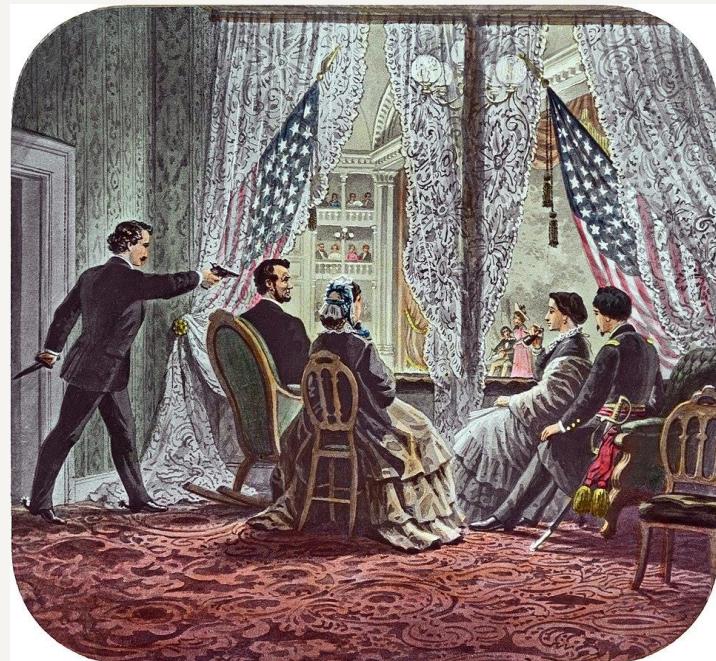


b) Résultat par amplification

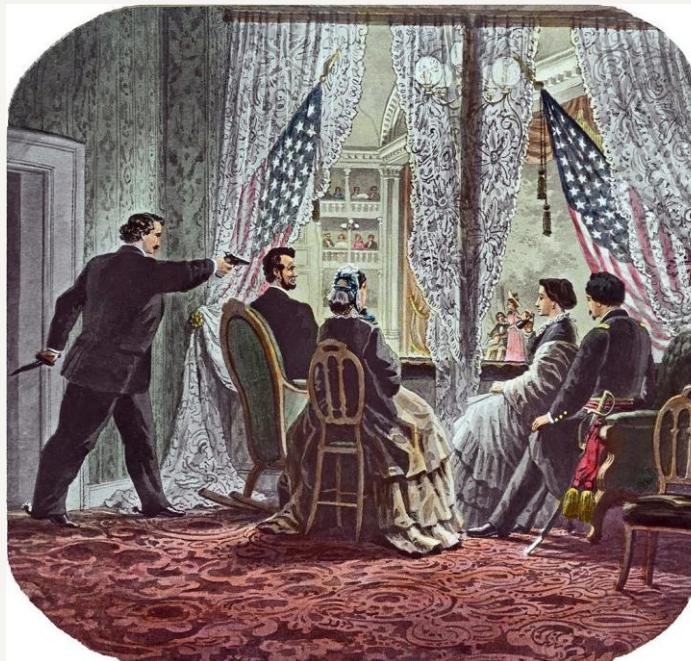
Limites du Seam Carving - 2 -

Une image est dite **trop condensée** lorsqu'elle **ne contient aucune zone visuellement négligeable**.

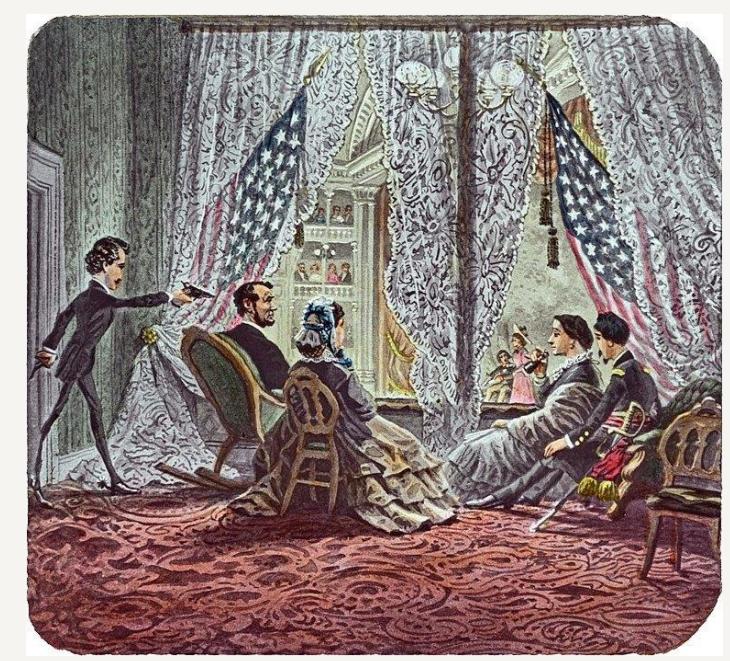
- Chaque région a une **importance perceptuelle forte**.
- Aucune couture ne peut être retirée sans affecter visiblement l'image.



a) Image originale



b) Résultat après agrandissement



c) Résultat après réduction

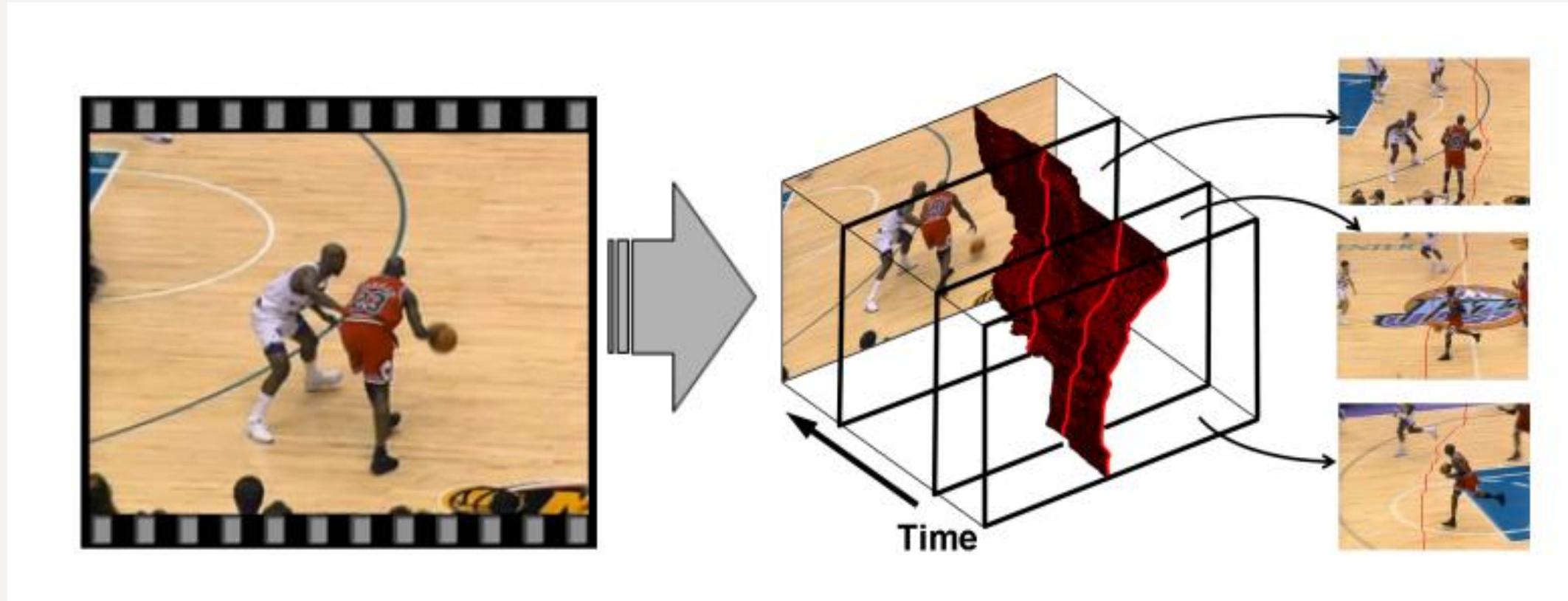
Extension - 1 -



Extension - 2 -



Extension - 3 -



Seam Carving appliqué pour les vidéos
(Rubinstein et al. en 2008)

MERCI POUR VOTRE ATTENTION