

**IUT Informatique de Sophia-Antipolis**

2017/2018

**Rapport du projet Inpainting**

**M332 Modélisation Mathématiques**

**CABANES Matthieu, COURIEUX Alexi, FOURNIER Nathan et ZEKRI Meriam**





Introduction

Le terme d’Inpainting vient de la restauration d'art. L'art médiéval a commencé à être restauré dès la Renaissance, pour mettre des images médiévales "à jour" ou remplir des parties manquantes. La nécessité de retoucher des images de manière discrète a été très vite ressentie dans le domaine de la peinture dans un premier temps, puis la photographie jusqu’au cinéma. Les objectifs sont restés les mêmes : empêcher la détérioration (par exemple les rayures et les taches de poussières dans le film), ou pour ajouter ou supprimer des éléments. Dans le domaine numérique, le problème de l'Inpainting est apparu sous le nom "dissimulation d'erreur" dans les télécommunications, où le besoin était de remplir des blocs d'image qui ont été perdus pendant la transmission de données.

Sommaire

[I. Résumé 4](#_Toc506389378)

[II. L’Inpainting 5](#_Toc506389379)

[A. Description du problème 5](#_Toc506389380)

[B. Algorithme 5](#_Toc506389381)

# Résumé

Ce projet consiste à réaliser de la reconstruction d’images, aussi connu sous le nom d’Inpainting. Nous avons réalisé ce projet dans la cadre de notre formation à l’IUT Informatique de Sophia-Antipolis, durant le semestre trois.

Nous avons utilisé le langage JAVA pour y parvenir.

Dans un premier temps, nous détaillerons le concept de l’Inpainting. Nous expliquerons ensuite comment y parvenir et présenterons les différents cas tests. L’installation du projet sera également détaillée pour sa réutilisation.

Voici un exemple du résultat attendu :



Notre code prend donc en entrée une image (ex : celle du milieu), ainsi que la couleur du masque (ex : le vert), et donne en sortie une nouvelle image reconstruite en format bmp (ex : la dernière image).

# L’Inpainting

## Description du problème

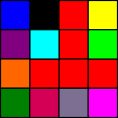
L’Inpainting correspond à la reconstruction d’images détériorées ou de remplissage des parties manquantes d’une image.

Pour reconstruire une image, les endroits de l’image à reconstruire doivent être recouverts d’une même couleur, que l’on appellera « masque ». Cette couleur est ensuite précisée pour la reconstruction. L’Inpainting consiste alors à modifier uniquement les parties colorées de cette couleur en se basant sur le reste de l’image pour un résultat réaliste

La difficulté de la reconstruction d’image est de déterminer le bout d’image à copier sur le masque pour obtenir un résultat satisfaisant.

## Algorithme

La reconstruction d’image se fait pixel par pixel. Nous avons donc besoin de créer une matrice correspondant à l’image. Cette matrice contient les dimensions de l’image et un tableau à deux dimensions de couleur dont chaque case représente la couleur de chaque pixel. Une couleur est représentée par un taux de rouge, de vert et de bleu (RGB).



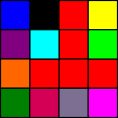
[0][0] : Color(0,0,255)

[0][1] : Color(0,0,0)

[0][3] : Color(255,0,0)

Une fois la matrice créée, nous créons le masque avec la couleur qui nous est donnée en entrée. Le masque contient un tableau à deux dimensions de booléens. Il parcourt la matrice de couleur, si elle est de la même couleur, la case du tableau du masque est mise à « true », sinon, à « false ».

Si la couleur du masque définie est le rouge (255, 0, 0)



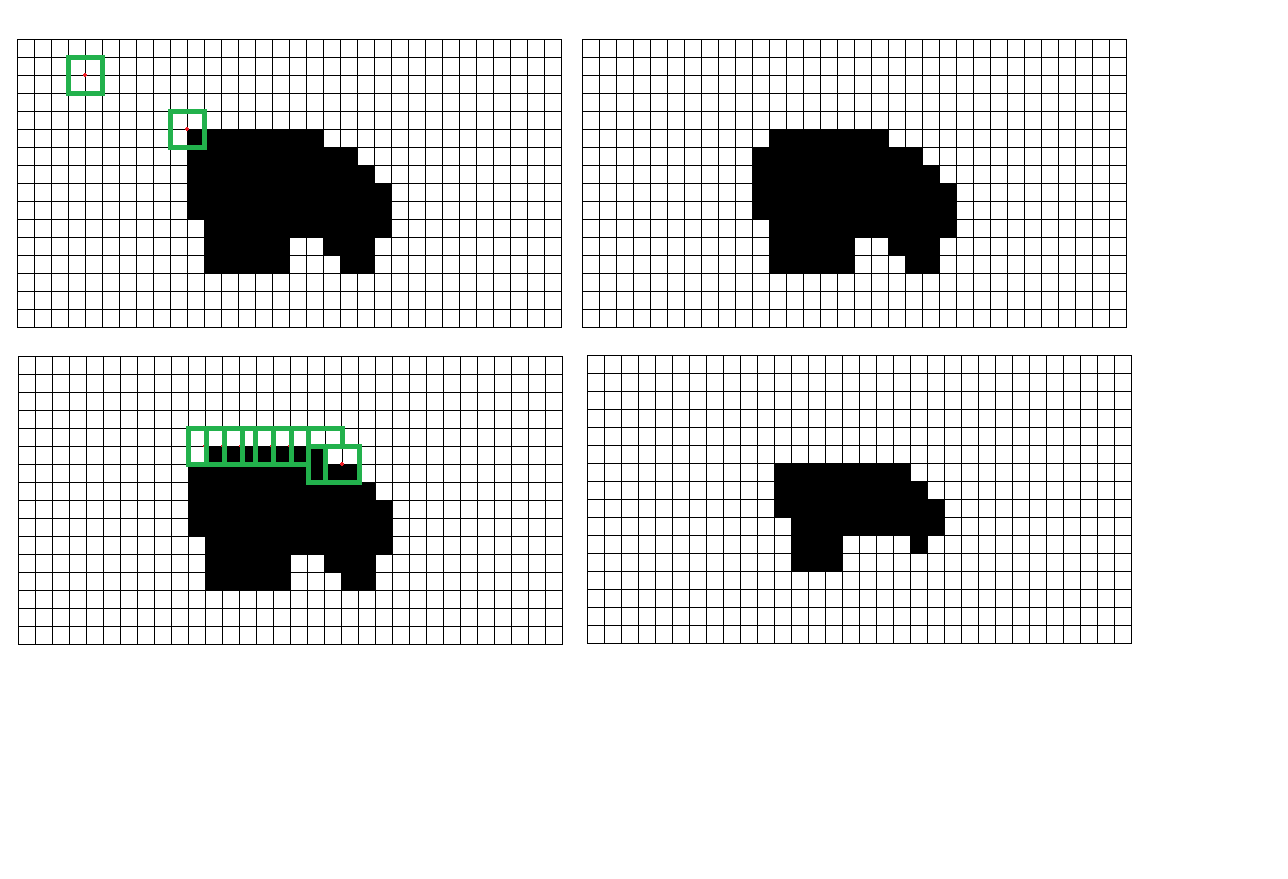
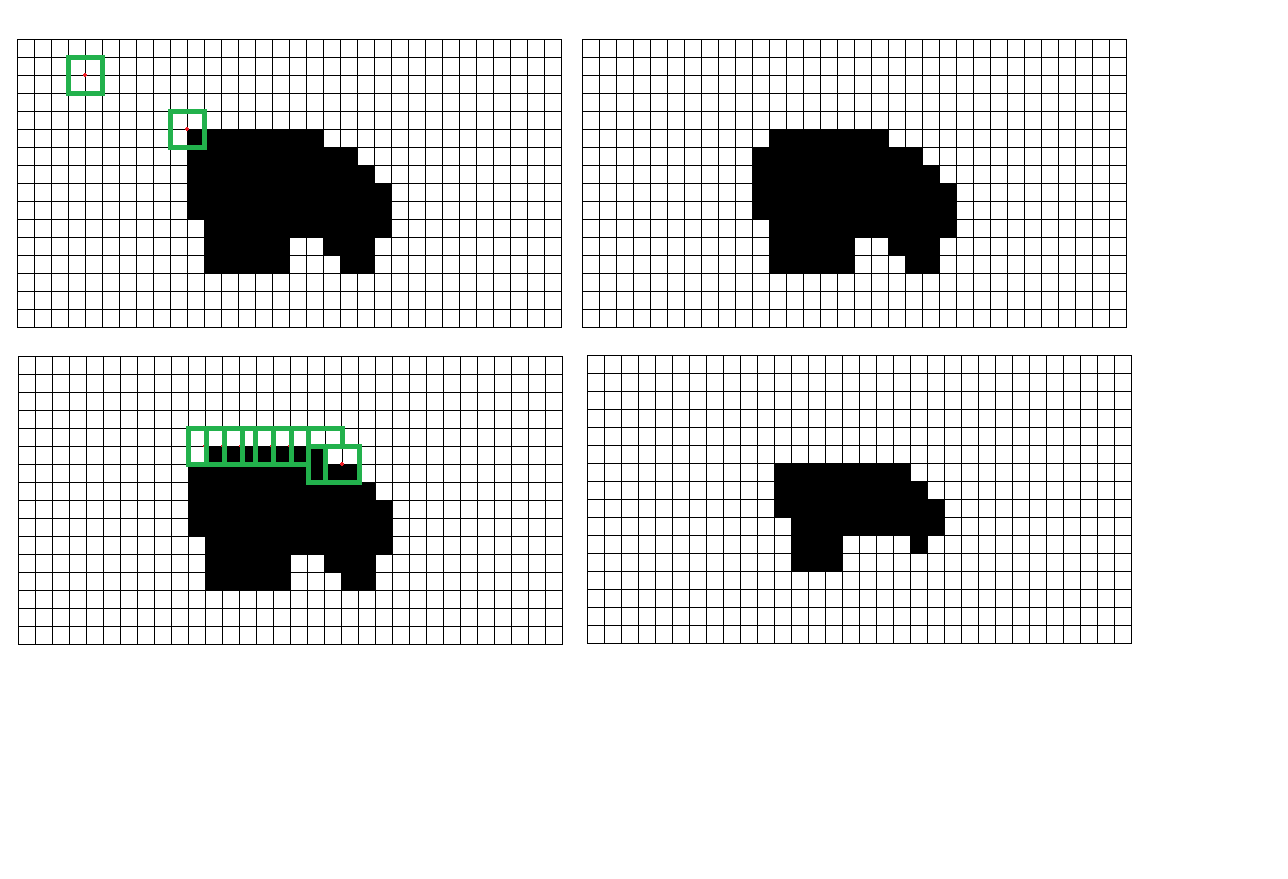
[0][0] : false

[0][1] : false

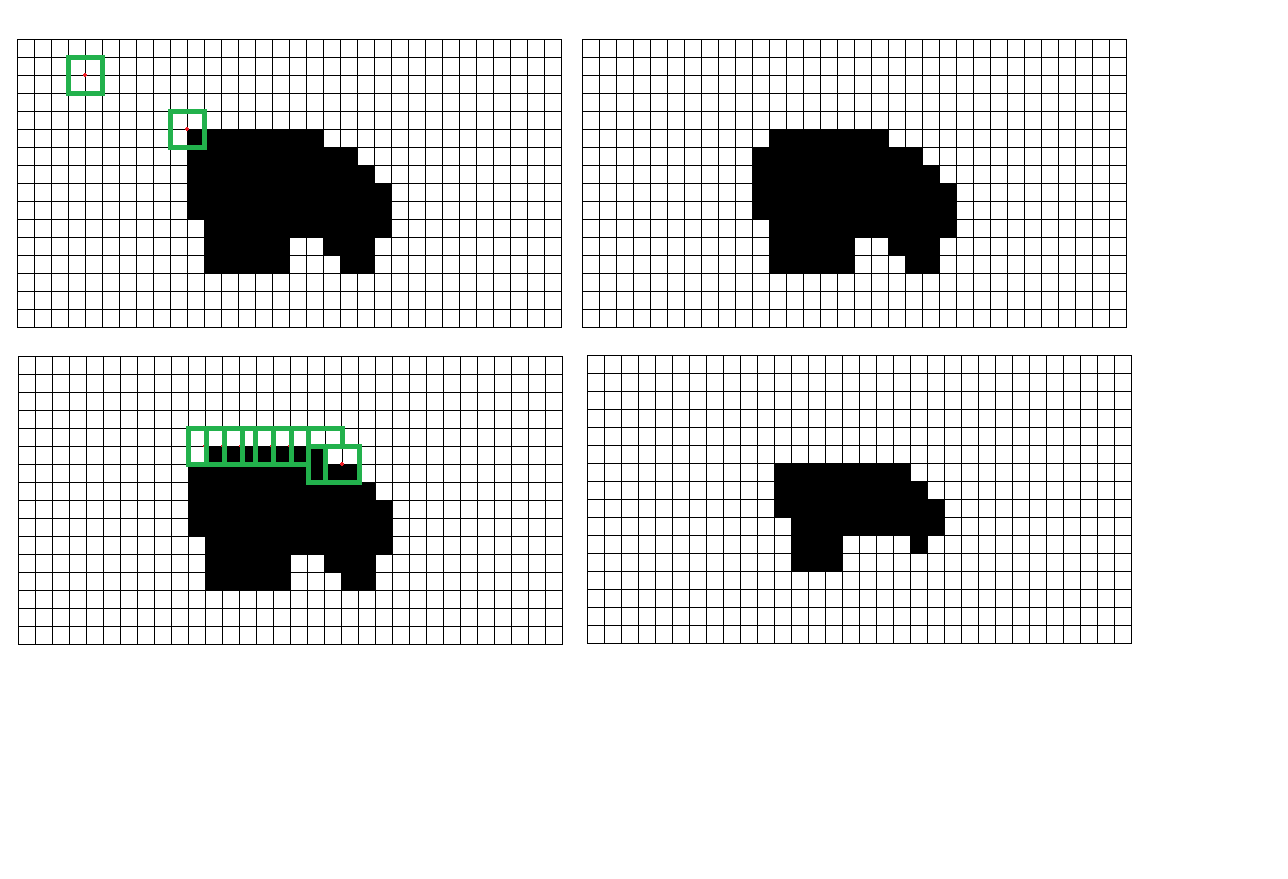
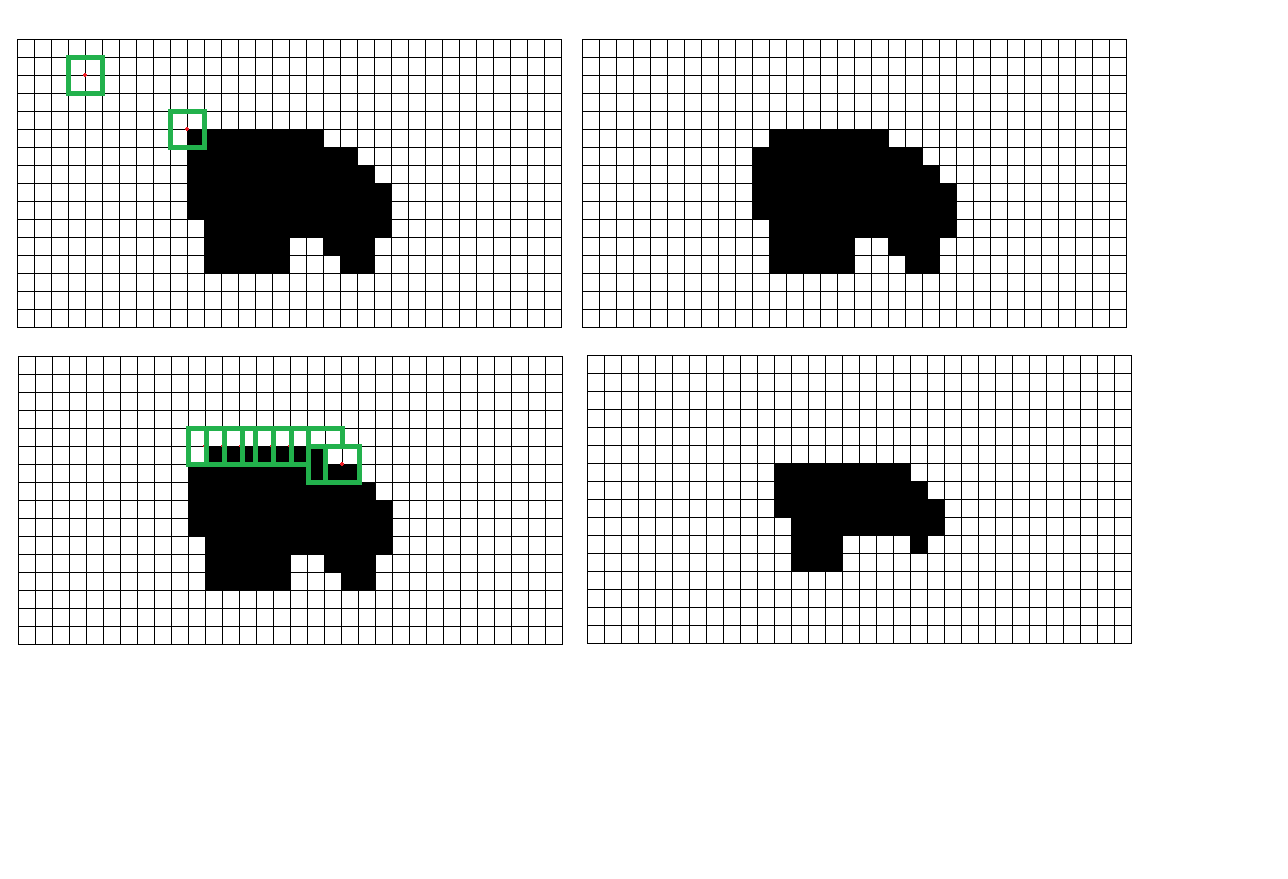
[0][2] : true

Avec la matrice et le masque, nous avons toutes les informations nécessaires pour la reconstruction de l’image. Nous savons à présent quels pixels sont à reconstruire.

La reconstruction se fait pixel par pixel. On applique un patch sur un point du contour du masque. On cherche ensuite un patch sur le reste de l’image qui correspond le mieux à celui-ci que l’on copie ensuite sur la partie du masque.

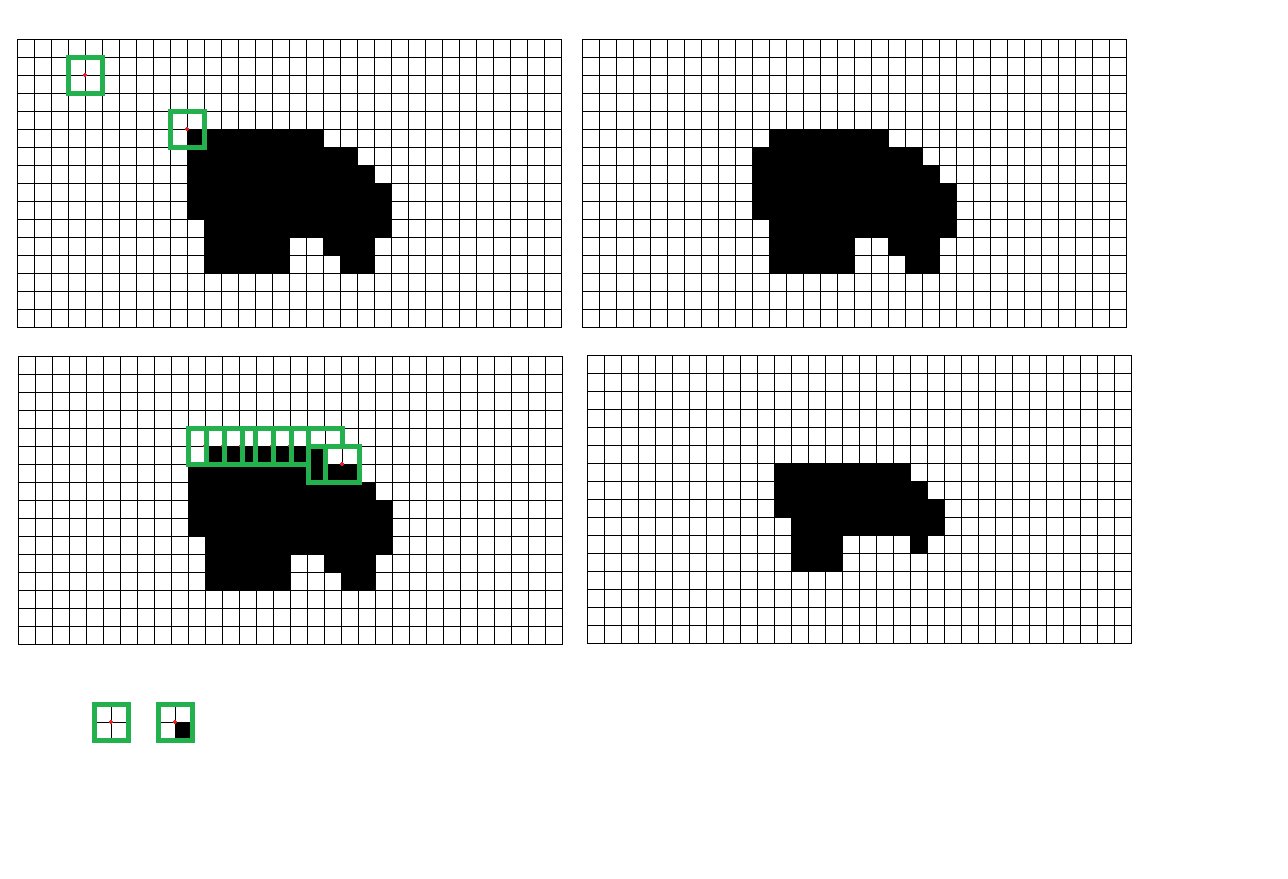
On parcourt le contour du masque de la même manière. On réitère l’action jusqu’à ce qu’il n’y ait plus de masque.

Pour trouver le patch adéquat, on calcule la différence de couleur entre les pixels des patchs :

d2

d1



d3

On additionne toutes les différences entre elles et le patch qui aura la plus petite différence sera copié. La zone parcourue pour trouver le bon patch est à déterminer, ainsi que la largeur du patch. On obtiendra alors des résultats différents en fonction de ces deux paramètres.