Projet CLANU 2016: un algorithme d'inpainting

E. Bretin, T. Grenier et O. Bernard

5 avril 2016

Cet énoncé donne les objectifs, questions et livrables attendus pour le projet CLANU. Ce projet a pour objectif principal de vous faire concevoir un logiciel en langage Matlab/C+ illustrant une méthode mathématique spécifique : l'*inpainting*. Les méthodes sont introduites dans cet énoncé et différentes implémentations sont à faire pour répondre aux questions.

La qualité des résultats et de l'analyse mathématique d'une part, ainsi que la qualité de la conception et de la démarche de conception d'autre part, seront prises en considération par les enseignants d'informatique et de mathématiques.

1 Introduction

Le but de ce projet est d'implémenter et d'étudier numériquement une méthode simple d'inpainting utilisant une approximation Spline et une résolution de système linéaire à partir d'algorithmes de descente (méthode du gradient et gradient conjugué). Plus précisément, les méthodes d'inpainting sont des techniques d'interpolation qui permettent de reconstruire des données manquantes dans une image. L'origine de ces pertes peuvent être par exemple la conséquence de la détérioration d'une photographie et la reconstruction de ces pixels perdus est communément appelé inpainting par analogie avec le travail d'un artiste peintre qui, par coups de pinceau, restaurerait les parties manquantes tout en assurant la cohérence avec le reste de l'image.

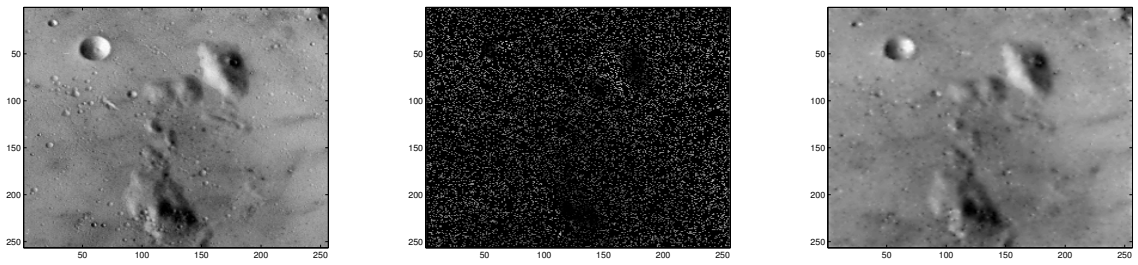

FIGURE 1: Exemple de reconstruction d'une image endommagée utilisant l'algorithme développé dans le cadre de ce projet avec : à gauche l'image originale ; au centre : l'image endommagée (les pixels perdus sont représentés en noir) ; à droite : l'image reconstruite

2 Un exemple d'algorithme d'inpainting

L'idée est tout d'abord de représenter l'image de taille $N \times N$ en niveaux de gris à l'aide d'une fonction

$$I : [1, N]^2 \to [0, 1].$$

Le masque des données manquantes est aussi représenté à l'aide de la fonction

$$m : [1, N]^2 \to \{0, 1\}.$$

Plus précisément, si $m(x) = 0$ (et respectivement $m(x) = 1$) alors l'image I est (et respectivement n'est pas) endommagée en $x = (x_i, x_j) \in [1, N]^2$.

L'objectif de l'inpainting est alors de trouver une bonne reconstruction de l'image originale I à partir de l'image corrompue $Ic(x) = m(x)I(x)$. Pour ce faire, nous allons déterminer une approximation $\tilde{I}$ définie par les contraintes suivantes :

Contraintes

$$(A) \quad \begin{cases} \tilde{I}(x) = I(x) & \text{si } m(x) = 1, \\ -\Delta\tilde{I}(x) = 0 & \text{si } m(x) = 0. \end{cases}$$

Ici, l'opérateur $\Delta\tilde{I}$ représente le laplacien de l'image $\tilde{I}$ en x et cette dernière contrainte permet d'imposer une certaine régularité sur l'image reconstruite.

D'un point de vue discret, l'image $I_{i,j}$ pixelisée est un échantillonnage de la fonction I aux noeuds $x_{i,j} = (i, j)$ pour tous $(i, j) \in \{1, 2 \cdots, N\}^2$. Nous allons de plus déterminer la fonction $\tilde{I}$ par l'intermédiaire de son interpolation spline aux noeuds $x_{i,j}$:

$$\tilde{I}(x) = \sum_{i=1}^{N} \sum_{j=1}^{N} \beta_{i,j} \varphi_{i,j}(x),$$

où $\varphi_{i,j}(x) = \varphi(x_i - i)\varphi(x_j - j)$ et où

$$\varphi(s) = \begin{cases} (\frac{2}{3} - s^2(2 - |s|)/2 & \text{si } |s| \leq 1, \\ \frac{1}{6}(2 - |s|)^3 & \text{si } |s| \in [1, 2], \\ 0 & \text{si} |s| > 2. \end{cases}$$

L'idée est alors de déterminer les coefficients $\beta_{i,j}$ en résolvant un système linéaire issu de la version discrétisée des contraintes (A) :

Discrétisation des contraintes

$$(B) \quad \begin{cases} \tilde{I}(x_{i,j}) = I_{i,j} & \text{si } m(x_{i,j}) = 1, \\ -\Delta\tilde{I}(x_{i,j}) = 0 & \text{si } m(x_{i,j}) = 0, \end{cases}$$

En remarquant que

$$\varphi(0) = 4/6, \quad \varphi(1) = \varphi(-1) = 1/6, \quad \text{et} \quad s(n) = 0 \quad \forall n \in \mathbb{Z} \setminus \{-1, 0, 1\},$$

la première équation de (B) montre que les coefficients $\beta_{i,j}$ vérifient

Equation 1

$$\begin{aligned} I_{i,j} \quad = \quad & \frac{16}{36}\beta_{i,j} + \frac{4}{36}\left(\beta_{i-1,j} + \beta_{i+1,j} + \beta_{i,j+1} + \beta_{i,j-1}\right) \\ & + \frac{1}{36}\left(\beta_{i-1,j-1} + \beta_{i-1,j+1} + \beta_{i+1,j-1} + \beta_{i+1,j+1}\right). \end{aligned}$$

De même avec

$$\varphi''(0) = -2, \quad \varphi''(1) = \varphi(-1) = 1, \quad \text{et} \quad \varphi''(n) = 0 \quad \forall n \in \mathbb{Z} \setminus \{-1, 0, 1\},$$

la deuxième équation de (B) implique que

Equation 2

$$\begin{aligned} 0 \;=\; & 8\beta_{i,j} - (\beta_{i-1,j} + \beta_{i+1,j} + \beta_{i,j+1} + \beta_{i,j-1}) \\ & - (\beta_{i-1,j-1} + \beta_{i-1,j+1} + \beta_{i+1,j-1} + \beta_{i+1,j+1})\,. \end{aligned}$$

Au final, les coefficients $\beta_{i,j}$ sont donc solution du système linéaire

$$A\beta = b,$$

où la matrice A est définie par

$$(A\beta)_{i,j} = \begin{cases} \frac{16}{36}\beta_{i,j} + \frac{4}{36}\left(\beta_{i-1,j} + \beta_{i+1,j} + \beta_{i,j+1} + \beta_{i,j-1}\right) \\ \quad + \frac{1}{36}\left(\beta_{i-1,j-1} + \beta_{i-1,j+1} + \beta_{i+1,j-1} + \beta_{i+1,j+1}\right) \text{ si } m(x_{i,j}) = 1 \\ 8\beta_{i,j} - (\beta_{i-1,j} + \beta_{i+1,j} + \beta_{i,j+1} + \beta_{i,j-1}) \\ \quad - (\beta_{i-1,j-1} + \beta_{i-1,j+1} + \beta_{i+1,j-1} + \beta_{i+1,j+1}) \text{ si } m(x_{i,j}) = 0, \end{cases}$$

et où le second membre vérifie

$$b_{i,j} = \begin{cases} I_{i,j} & \text{si } m(x_{i,j}) = 1 \\ 0 & \text{si } m(x_{i,j}) = 0. \end{cases}$$

En pratique, la solution du système linéaire $A\beta = b$ sera obtenue avec une méthode de gradient puis une méthode de gradient conjugué. Voici un code matlab (figures 2 et 3) qui implémente l'algorithme décrit précédemment et qui utilise une méthode de gradient. Ce code est aussi disponible sous moodle.

3 Questions mathématiques

Voici une liste de questions qui serviront de support pour l'évaluation de projet pour la partie mathématiques.

1. Montrer que les contraintes discrètes (B) impliquent bien les équations 1 et 2.
2. Expliquer soigneusement chacune des lignes du codes *matlab* des figures 2 et 3.
3. Dans la fonction *matrice_A*, l'équation 2 est multipliée par $N1N2$. Etudier l'influence de ce terme sur la convergence de la méthode du gradient.
4. Afin de gagner en efficacité, implémenter la méthode de gradient conjugué pour la résolution du système linéaire $A\beta = b$.
5. Afin de tester les limites de cet algorithme d'inpainting, proposer d'autres exemples de reconstruction basés sur de nouvelles images et d'autres masques.
6. Proposer et tester une méthode d'inpainting qui permet de traiter les images en couleur.
7. Quel enseignant de GE se cache dans l'image de la figure 4 (fichier *professeur.tiff* dans moodle) ?

```
function [J] = matrice_A(I,Masque),
[N1,N2] = size(I);
J = zeros(size(I));

for i=1:N1,
     bool_iplus = i<N1;
     iplus = min(i+1,N1);
     bool_imoins = i>1;
     imoins = max(i-1,1);

    for j=1:N2,

     bool_jplus = j<N2;
     jplus = min(j+1,N2);
     bool_jmoins = j>1;
     jmoins = max(j-1,1);

        if (Masque(i,j)>0)
            J(i,j) = 1/36*(16*I(i,j) + 4*(bool_iplus*I(iplus,j) + bool_imoins*I(imoins,j) + ...
                      bool_jplus*I(i,jplus) +  bool_jmoins*I(i,jmoins)) + ...
                     (bool_iplus*bool_jplus*I(iplus,jplus) + bool_imoins*bool_jplus*I(imoins,jplus) + ...
                      bool_imoins*bool_jmoins*I(imoins,jmoins)) + bool_iplus*bool_jmoins*I(iplus,jmoins));
        else
            J(i,j) =  -N1*N2*(-8*I(i,j) + 1*(I(iplus,j) + I(imoins,j) + I(i,jplus) + I(i,jmoins)) + ...
                (I(iplus,jplus) + I(imoins,jplus) + I(imoins,jmoins)) + I(iplus,jmoins));
        end
    end
end

```

FIGURE 2: Exemple de fonction *Matlab* qui permet de calculer le produit $A\beta$

```
clear all;

%%%%%%%%%%%%%%%%%%% Lecture  image
I = imread('moon.tiff');
I = double(I)/256;
I = sum(I,3)/3;
[N1,N2] = size(I);

%%%%%%%%%%%%%%%%%%% Exemple de masque
Masque = max(min(round(0.6*rand(size(I))),1),0);
Ic = I.*Masque;


%%%%%%%%%%%%%%% Algorithme d'inpainting avec une méthode de gradient
  res = 1;
  b = Ic;
  xk = b;

  while res > 10^(-2),
     dk = b - matrice_A(xk,Masque);
     Adk = matrice_A(dk,Masque);
     alphak = sum(sum(dk.*dk))/sum(sum(Adk.*dk));
     xk = xk + alphak*dk;
     res = dk(:)'*dk(:)/(N1*N2)

    %%%%%%%%%%%%% Plot image au cours du calcul
    imagesc(matrice_A(xk,ones(size(I))));
    colormap('gray')
    pause(0.01)
  end

```

FIGURE 3: Exemple de script *Matlab* qui permet de reconstruire une image corrompue avec l'algorithme d'inpainting décrit précédemment et utilisant une méthode de gradient pour la résolution du système linéaire $A\beta = b$.

4 Questions Informatiques

La réalisation se fera en C/C++ avec l'environnement *QtCreator*. Les canevas sont disponibles sous moodle. Seuls les fichiers `clanu_process.cpp` et `clanu_process.h` sont à modifier. Les images seront représentées par des matrices de flottants : `float **`.

1. Prendre en main le canevas `Inpainting_Console` et le compiler. Comprendre le système de passage

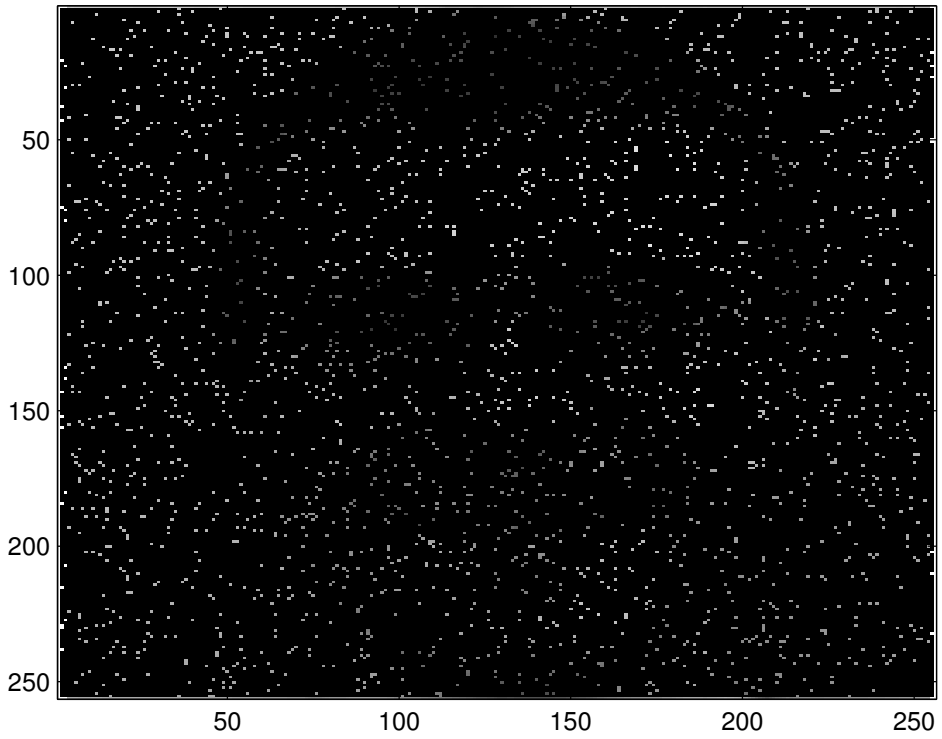

FIGURE 4: Quel enseignant de GE se cache dans cette image ?

des paramètres en ligne de commande. A partir des `float **` issus de deux images lues (une image et un masque), programmer un traitement qui inverse la valeur des niveaux de gris de l'image (attention les valeurs sont entre 0 et 255) **uniquement** là où les pixels du masque ont une valeur supérieure à 0.

2. Dans un nouveau projet basé sur `Inpainting_Console`, écrire les méthodes en C/C++ réalisant l'inpainting avec le gradient conjugué.
3. Compléter le code précédent pour pouvoir faire de l'inpainting sur des images en couleurs.
4. Intégrer votre code (les deux fichiers modifiés) dans le canevas `inpaiting_IHM`.

5 Évaluations et rendus

5.1 Organisation

Ce projet rentre dans le cadre des modules IF2 et MA2 et s'effectue en binôme. Il comporte en particulier une note de projet en binôme ainsi que 2 évaluations individuelles.

Les binômes sont à établir par les étudiants sous moodle avant le lundi 25 avril minuit.

Une soutenance sur la partie mathématique est prévue mi-mai.

Les rendus (le rapport (pdf) et les sources (zip)) sont à remettre **au plus tard le vendredi 10 juin minuit**.

Attention, les rendus se faisant sous *moodle*, aucun délai supplémentaire n'est accordé.

Les détails pour la soutenance, les rendus et évaluations individuelles sont donnés ci-après.

5.2 Soutenance Mathématiques

Une soutenance en binôme sur la partie mathématiques sera prévue à la mi mai. A cette occasion, vous remettrez un mini rapport de votre projet (1 page recto-verso) comportant les codes *matlab* et les réponses aux questions. Il s'agira alors de présenter en 5 minutes une synthèse de votre travail puis de répondre à quelques questions qui vous seront posées. Vous aurez alors une vingtaine de minutes pour préparer vos réponses.

5.3 Rendus

Chaque binôme déposera sur *moodle* :

1. le rapport au format pdf,
2. la réalisation informatique : un fichier zip des sources.

Le fichier **.zip** sera constitué uniquement des fichiers `Matlab` et C++ nécessaires à la compilation, des fichiers de projet **.pro** et de quelques exemples d'images à traiter.

- Ne mettez pas vos exécutables dans l'archive : votre projet sera recompilé.
- Assurez-vous que les fichiers présents dans l'archive permettent la création des exécutables.
- Respectez bien les consignes de programmation et les canevas fournis.
- De même, ne pas mettre dans cette archive les résultats.

Le nom du fichier des sources déposé sera : `BXX_NOM1_NOM2_Source.zip` où vous aurez remplacé les XX par votre numéro de binôme et les $NOM1$ et $NOM2$ par vos noms.

5.4 Structure du rapport

Le rapport comportera les parties suivantes :

1. une introduction présentant l'étude et l'organisation de votre rapport.
2. une partie d'environ trois pages avec vos propres mots dans laquelle vous expliquerez la solution d'inpainting et vos propositions.
3. une partie *Analyse Numérique* dans laquelle les réponses aux questions mathématiques seront données ainsi que quelques résultats pertinents.
4. une partie *Informatique* qui détaillera les algorithmes implémentés, l'organisation du code, les réponses aux questions ainsi que quelques résultats pertinents
5. une conclusion résumant le travail réalisé, l'organisation du projet (Gantt) et la répartition du travail dans le binôme.
6. au besoin, une partie bibliographie.

Les qualités de rédaction et de présentation ainsi que de votre pertinence d'analyse des résultats seront grandement prises en compte dans l'évaluation du rapport.

Afin d'éviter certaines dérives du travail en binôme, il vous est demandé de préciser le partage des tâches dans le rapport et l'auteur des sources (cf. conclusion).

Le nom du fichier déposé sera : `BXX_NOM1_NOM2_Rapport.pdf` où vous aurez remplacé les XX par votre numéro de binôme et les $NOM1$ et $NOM2$ par vos noms.

5.5 Évaluations individuelles

Les évaluations individuelles portent respectivement sur les parties mathématiques et informatique et auront lieu pendant les épreuves de IF2 et de MA2. Ces deux évaluations sont obligatoires et s'effectueront sans document. Leur durée sera au minimum 30 minutes pour la partie informatique et de 15 minutes pour la partie mathématiques.