smooth inpainting

(remplissage lisse ?)

La diapositive la plus prétentieuse de ma vie

Le problème général du traitement d'image

À partir d'une image

$$f : A \to \mathbb{R}$$

trouver une image

$$g : B \to \mathbb{R}$$

de la façon la plus simple possible.

Cas particuliers : ($A = B$) débruitage, approximation, lissage, ...

($A \cap B = \emptyset$) interpolation, extrapolation, zoom-in, échantillonage irrégulier, *multiple landmark warping*, synthèse de texture, colorisation, ...

($A \subseteq B$) régression, interpolation approximée, interpolation robuste, ...

Cas particulier $A = B$

Le problème général pour $A = B$

À partir d'une image

$$f : A \to \mathbb{R}$$

trouver une image

$$g : A \to \mathbb{R}$$

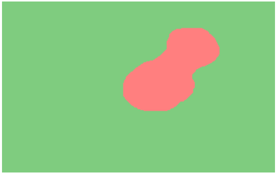
la plus simple possible.

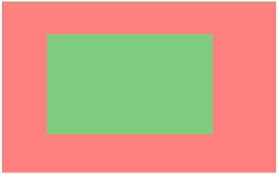
Cas particuliers :

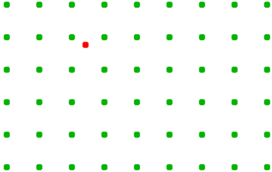
- **simple = nette :** débruitage
- **simple = peu informative :** approximation, compression
- **simple = régulière :** lissage

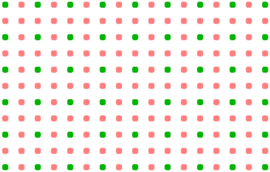
Cas $A \cap B = \emptyset$

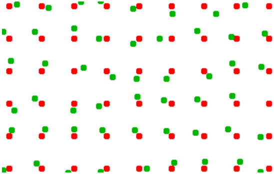
problème général : $f : A \to \mathbb{R} \implies g : B \to \mathbb{R}$

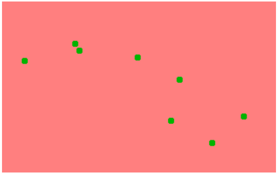

inpainting


extrapolation


interpolation


zoom-in

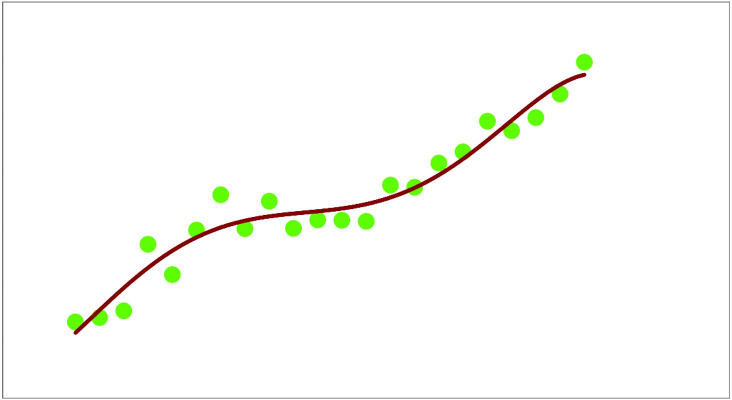

échantillonage irregulier


warping

Cas particulier $A \subseteq B$

$\boxed{\text{problème général : } f : A \to \mathbb{R} \implies g : B \to \mathbb{R}}$

- régression
- interpolation aproximmée
- interpolation robuste

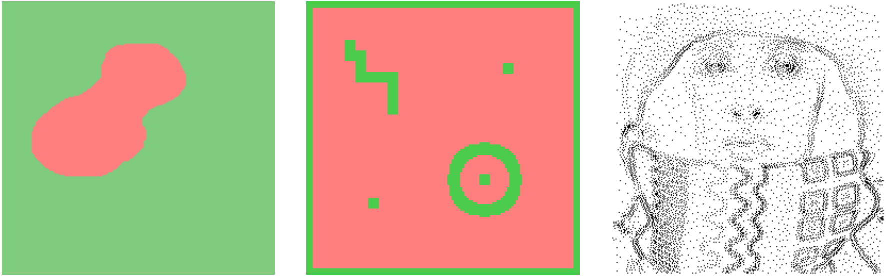


Smooth inpainting

problème général : $f : A \to \mathbb{R} \implies g : B \to \mathbb{R}$

Le *smooth inpainting* est un cas particulier du problème général avec la forme suivante :

- $A \cap B = \emptyset$
- $A \cup B =$ rectangle
- A fermé
- g dépends seulement des valeurs de f sur ∂A (et de f', f'', etc.)



- Observation : Fourier INTERDIT
- Observation : exemplar-based inpainting INTERDIT

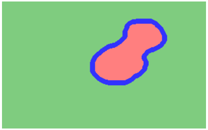
Méthodes de solution

$\boxed{\text{problème général : } f : A \to \mathbb{R} \implies g : B \to \mathbb{R}}$

$\boxed{\mathbf{S} := \partial A = \partial B}$

Voisin plus proche :

$$g(b) = f\left(\arg\min_{a \in \mathbf{S}} d(a, b)\right)$$



Combinaison linéaire de voisins :

$$g(b) = \sum_{a \in \mathbf{S}} w(a, b) f(a)$$

Critère variationnel :

$$\min_{g|_{\mathbf{S}} = f} E(g)$$

Équation aux dérivées partielles

$$\begin{cases} F(g) = 0 & B \\ g = f & \mathbf{S} \end{cases}$$

Méthodes variationnelles

$\boxed{\min E(g) \iff E'(g) = 0}$

$E(g)$	$E'(g) = 0$	commentaire
$\int \lvert\nabla g\rvert^2$	$\Delta g = 0$	Laplace
$\int \lvert\nabla g\rvert$	$\mathrm{div}\left(\frac{\nabla g}{\lvert\nabla g\rvert}\right) = 0$	variation totale, lignes de niveau droites
$\sup \lvert\nabla g\rvert$	$D^2 g(\nabla g, \nabla g) = 0$	AMLE, pente minimale, Canny
$\int \lvert\nabla g\rvert^p$	$\mathrm{div}\left(\lvert\nabla g\rvert^{p-2}\nabla g\right)$	p-Laplacien
$\int \sqrt{1 + \lvert\nabla g\rvert^2}$	$\mathrm{mean\text{-}curv}(g) = 0$	surface minimale, courbure moyene
$\int \lvert\Delta g\rvert^2$	$\Delta\Delta g = 0$	double Laplacien, thin plates
$\int(\alpha + \beta k(g)^2)\, \lvert\nabla g\rvert$	$\cdots$	elastica
$\cdots$	$\cdots$	Cahn-Hilliard
$\cdots$	$\cdots$	$\cdots$

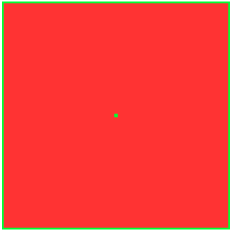
1D vs 2D

En dimension D=1 ces problèmes deviennent triviaux

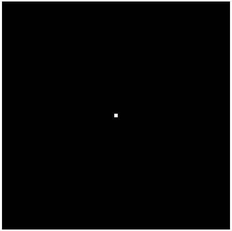
2D	1D
$\Delta g = 0$	$g'' = 0$
$\operatorname{div}\left(\frac{\nabla g}{\vert\nabla g\vert}\right) = 0$	$g'' = 0$
$D^2 g(\nabla g, \nabla g) = 0$	$g'' = 0$
$\operatorname{div}\left(\vert\nabla g\vert^{p-2}\nabla g\right)$	$g'' = 0$
$\text{mean-curv}(g) = 0$	$g'' = 0$
$\Delta\Delta g = 0$	$g'''' = 0$

$g'' = 0 \iff g =$ droites par morceaux
$g'''' = 0 \iff g =$ polynômes de degré 3 par morceaux

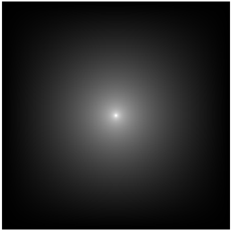
Réponse impulsionnelle



region



data



Laplace

AMLE

Le problème général en pixels

```c
// in: input image, with NAN values to be filled-in
// w: image width
// h: image height
// out: output image
void smooth_inpainting(float *out, float *in, int w, int h);
```

Équation de Laplace discrétisée

Il faut résoudre

$$\begin{cases} \Delta g(x) = 0 & x \in B \\ g(x) = f(x) & x \in A \end{cases}$$

où

$$\Delta g(x,y) = -4 * g(x,y) + g(x+1,y) + g(x,y+1) + g(x-1,y) + g(x,y-1)$$

C'est un système linéaire du type $Ax = b$. En Matlab : $x = A \backslash b$.

Méthode iteratif :

$$u^{n+1} = u^n + \tau \Delta u$$

Gauss-Seidel, Jacobi, successive over-relaxation, gradients conjugués. . .

Exemples de la méthode iterative

(videos lena/claudia)

Méthode pyramidale

Idée 1 : un trou d'un seul pixel est facile a remplir par interpolation.
Idée 2 : vus de suffisamment loin, tous les trous sont d'un seul pixel.

Algorithme :

1. Faire zoom arrière jusqu'à le trou disparait
2. Faire zoom avant pour revenir à la taille originale
3. Copier les valeurs haute-résolution que on connaissait déjà

Méthode pyramidale

```c
// zoom-out by 2x2 block averages
// NANs are discarded when possible
static void zoom_out_by_factor_two(float *out, int out_w, int out_h,
                                   float *in, int in_w, int in_h)
{
        for (int j = 0; j < out_h; j++)
        for (int i = 0; i < out_w; i++)
        {
                float a[4], sum = 0;
                a[0] = in[ (2*j    )*in_w  +  2*i     ];
                a[1] = in[ (2*j    )*in_w  +  2*i + 1 ];
                a[2] = in[ (2*j + 1)*in_w  +  2*i     ];
                a[3] = in[ (2*j + 1)*in_w  +  2*i + 1 ];
                int count = 0;
                for (int k = 0; k < 4; k++)
                if (isfinite(a[k])) {
                        sum += a[k];
                        count += 1;
                }
                out[ j*out_w + i ] = count ? sum/count : NAN;
        }
}
```

Méthode pyramidale

(videos lena/claudia multi-échelle)

Travail futur

- Implémenter les autres méthodes (esp. thin plates)
- Comparer les résultats avec chaque application
- Cas non-homogène (e.g. $\Delta u = f$)