

Didier Auroux

University of Nice Sophia Antipolis

France

auroux@unice.fr



---

# Topological asymptotic analysis for image inpainting

---

Second Workshop on Mathematical Analysis of Images in Bordeaux

LaBRI, Bordeaux

April 7-8th, 2014

# INTRODUCTION

**Motivations:** The topological gradient has been successfully used in electromagnetism, structural mechanics, fluid mechanics, ... for the identification of **optimal designs**, or optimal sub-domains.

Image processing techniques are often based on the identification of some specific parts: **edges, objects**, ...  $\rightsquigarrow$  **application of the topological gradient to image processing**.

**Advantages of the topological gradient:** a priori **fast** and **efficient** method  $\rightsquigarrow$  fast processing (real time) of the images and data, which is often a prerequisite e.g. for medical imaging.

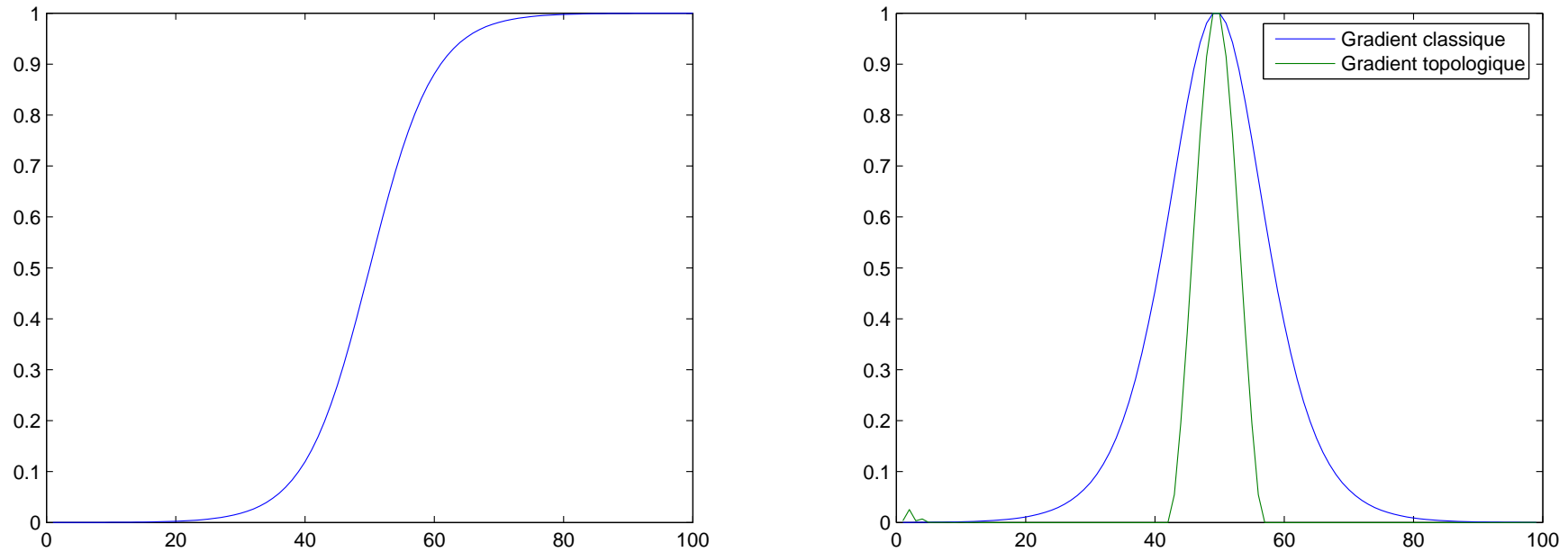
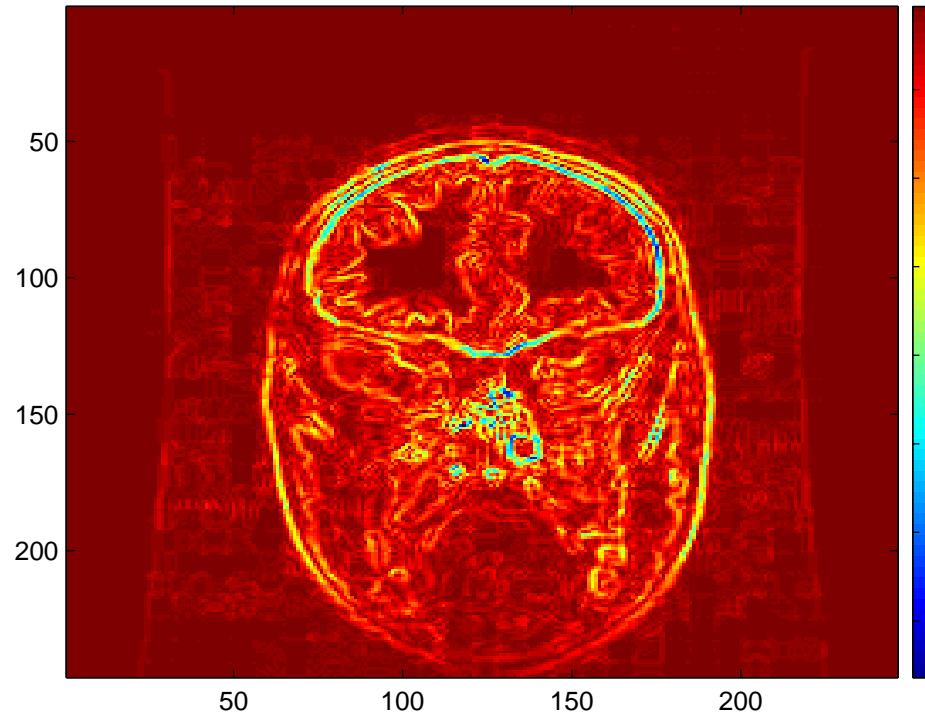
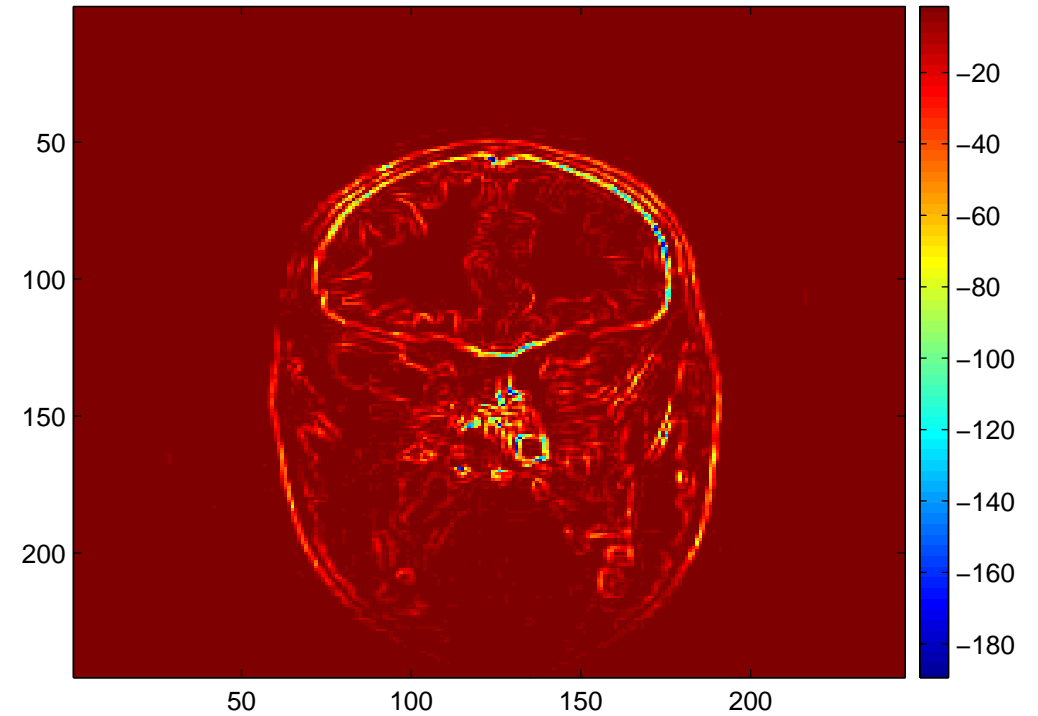


Figure 1: Gaussian probability density function, standard and topological gradients (normalized absolute value).



(a)



(b)

Figure 2: Standard gradient (a); Topological gradient (b).

# TOPOLOGICAL ASYMPTOTIC ANALYSIS

**Notations:** Let  $\Omega$  be a bounded **open set** of  $\mathbb{R}^n$ , and let us consider a **PDE** on  $\Omega$ :

$$\text{find } u \in \mathcal{V} \text{ such that } a(u, w) = l(w), \forall w \in \mathcal{V}.$$

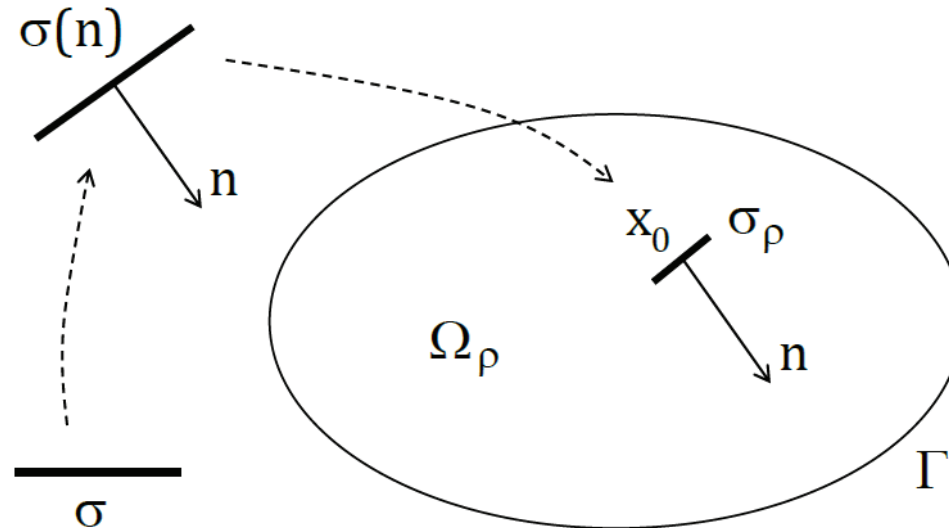
We also consider a **cost function**  $J(\Omega, u)$  measuring a criterion on the solution  $u$  of the PDE.

**Perturbation:** Insertion of a **small crack**  $\sigma_\rho = x_0 + \rho\sigma(n) \rightsquigarrow$  perturbed domain  $\Omega_\rho = \Omega \setminus \sigma_\rho$ . Perturbed PDE:

$$\text{find } u_\rho \in \mathcal{V}_\rho \text{ such that } a_\rho(u_\rho, w) = l_\rho(w), \forall w \in \mathcal{V}_\rho.$$

The cost function can be written as a function of only  $\rho$ :

$$j : \rho \mapsto \Omega_\rho \mapsto u_\rho \text{ solution of the perturbed PDE} \mapsto j(\rho) := J(\Omega_\rho, u_\rho).$$



**Topological gradient:** given by the asymptotic analysis when  $\rho$  goes to 0:

$$j(\rho) - j(0) = f(\rho)G(x_0) + o(f(\rho)),$$

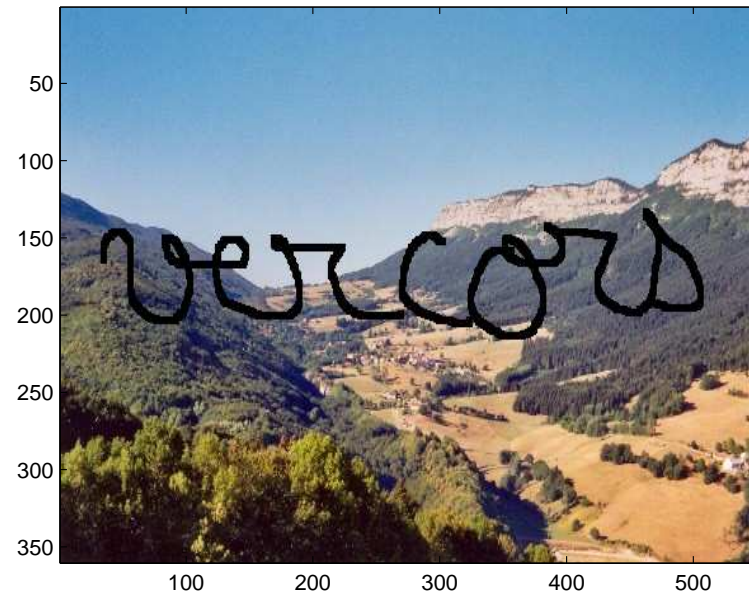
where  $f(\rho) > 0$ ,  $\lim_{\rho \rightarrow 0} f(\rho) = 0$ , and  $G(x_0)$  is called the **topological gradient** at point  $x_0$  [Masmoudi, 2001].

The minimization of  $j$  can be done by inserting cracks at the most negative values of the topological gradient.



# APPLICATION TO INPAINTING

**Goal:** identify the hidden part of the image.



**Standard approaches:** learning methods [Wen et al., 2006], minimization of the total variation [Chan-Shen, 2000-2005], morphological analysis [Elad et al., 2005-2006], ...

**Identification of hidden contours:** edge detection technique based on thermal diffusion [Perona-Malik, 1990; Morel et al., 1992; Weickert, 1994-2001]. Use of two measures on the boundary of the hidden part  $\rightsquigarrow$  Dirichlet and Neumann problems [A. and Masmoudi, 2006].

Let  $\Omega$  be a bounded domain in  $\mathbb{R}^2$ , we assume that there is a crack  $\sigma^*$  inside  $\Omega$ . We impose a flux  $\phi \in H^{-1/2}(\Gamma)$  on the boundary  $\Gamma$  of  $\Omega$ , and we try to localize a crack  $\sigma \subset \Omega$  such that the solution  $u \in H^1(\Omega \setminus \sigma)$  of

$$\begin{cases} \Delta u = 0 & \text{in } \Omega \setminus \sigma, \\ \partial_n u = \phi & \text{on } \Gamma, \\ \partial_n u = 0 & \text{on } \sigma, \end{cases} \quad (1)$$

satisfies  $u|_{\Gamma} = T$ , where  $T$  is a function of  $H^{1/2}(\Gamma)$ .

The compatibility condition (well-posed problem) is:

$$\int_{\Gamma} \phi \, ds = 0, \quad (2)$$

and

$$\int_{\Omega \setminus \sigma} u \, dx = 0. \quad (3)$$

Two boundary conditions  $\rightsquigarrow$  two problems, Dirichlet and Neumann:

$$\text{Find } u_D \in H^1(\Omega \setminus \sigma) \text{ s.t. } \begin{cases} \Delta u_D = 0 & \text{in } \Omega \setminus \sigma, \\ u_D = T & \text{on } \Gamma, \\ \partial_n u_D = 0 & \text{on } \sigma, \end{cases} \quad (4)$$

and

$$\text{find } u_N \in H^1(\Omega \setminus \sigma) \text{ s.t. } \begin{cases} \Delta u_N = 0 & \text{in } \Omega \setminus \sigma, \\ \partial_n u_N = \phi & \text{on } \Gamma, \\ \partial_n u_N = 0 & \text{on } \sigma. \end{cases} \quad (5)$$

For the actual crack, the two solutions  $u_D$  and  $u_N$  are equal. So we consider the cost function:

$$J(\sigma) = \frac{1}{2} \|u_D - u_N\|_{L^2(\Omega)}^2, \quad (6)$$

We introduce the two adjoint states, in  $H^1(\Omega)$ :

$$\begin{cases} -\Delta p_D = -(u_D - u_N) & \text{in } \Omega, \\ p_D = 0 & \text{on } \Gamma, \end{cases} \quad (7)$$

and

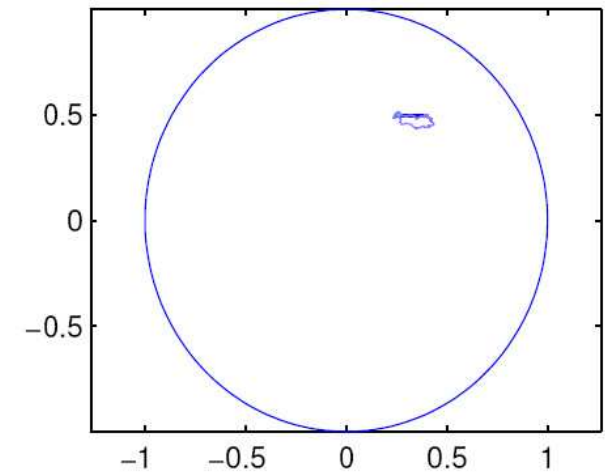
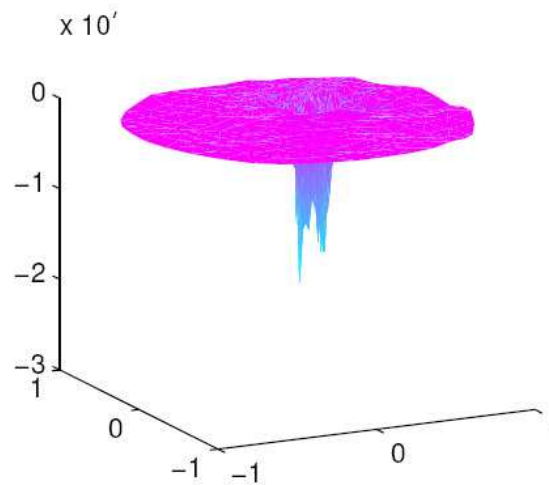
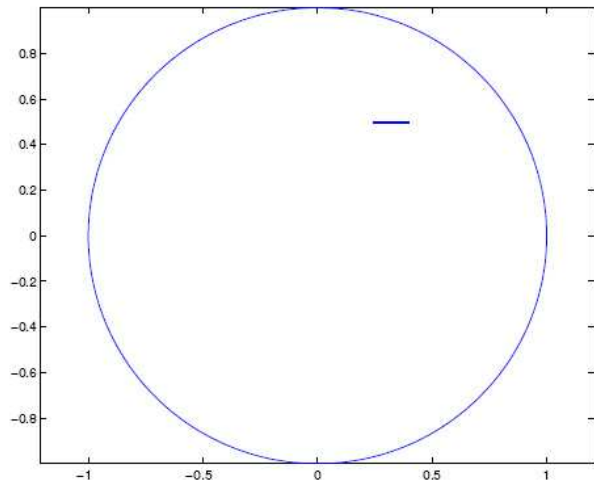
$$\begin{cases} -\Delta p_N = +(u_D - u_N) & \text{in } \Omega, \\ \partial_n p_N = 0 & \text{on } \Gamma. \end{cases} \quad (8)$$

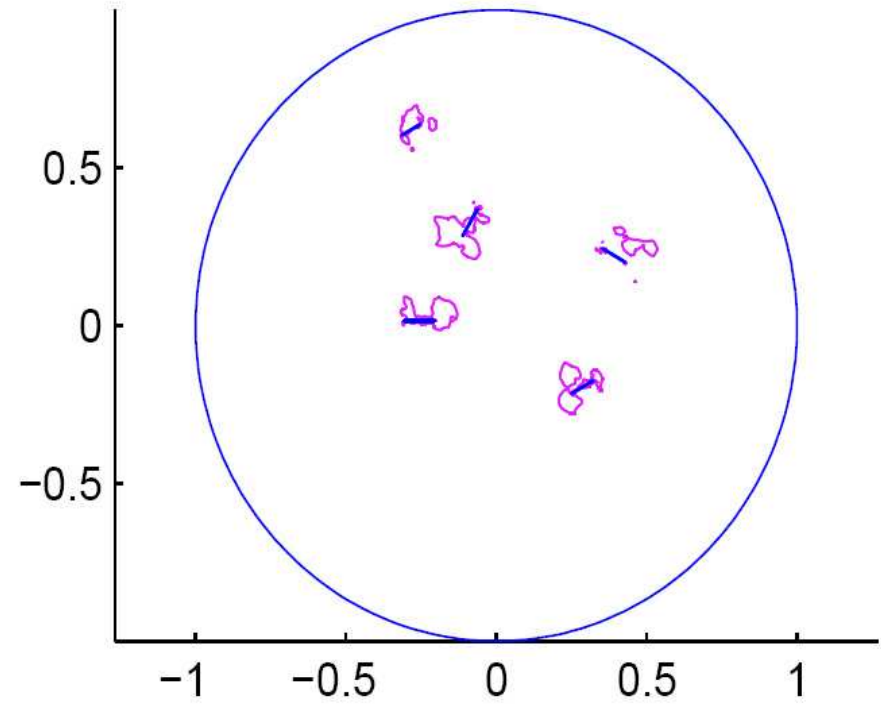
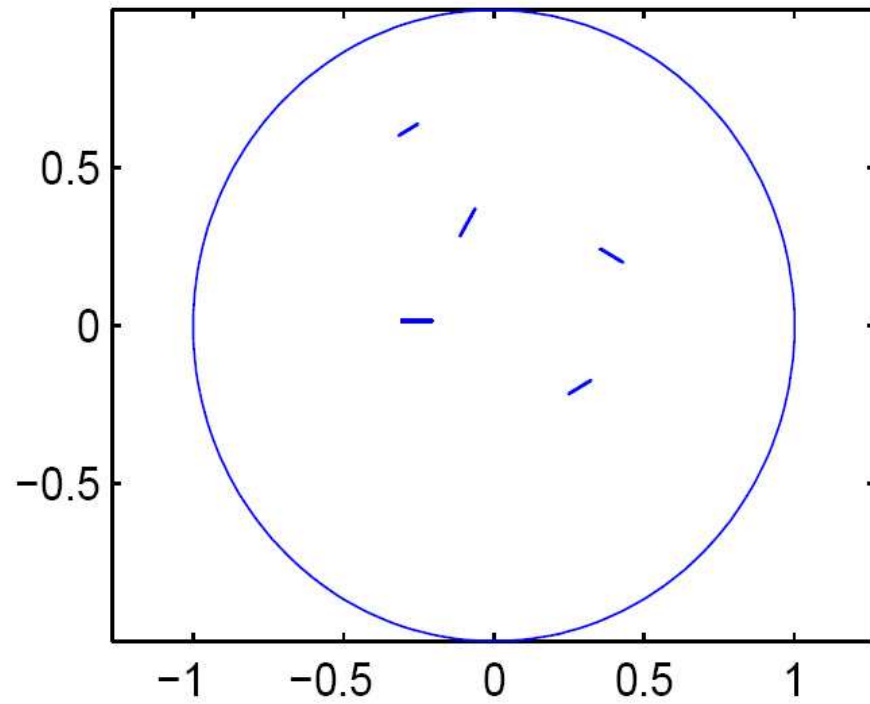
The variation of the cost function  $j(\rho) := J(u_{\sigma_\rho})$  when a small crack (size  $\rho$ ) is inserted, is given by the topological gradient:

$$j(\rho) - j(0) = f(\rho) g(x, n) + o(f(\rho)), \quad (9)$$

with

$$g(x, n) = - [(\nabla u_D(x) \cdot n)(\nabla p_D(x) \cdot n) + (\nabla u_N(x) \cdot n)(\nabla p_N(x) \cdot n)]. \quad (10)$$





Let  $\Omega$  be the image,  $\Gamma$  its boundary, and let  $\omega \subset \Omega$  be the unknown subset of the image, and  $\gamma$  its boundary. Let  $v$  be the original perturbed image, to be restored. Both Dirichlet and Neumann conditions are available on the boundary of  $\omega$ .

For a given crack  $\sigma$  inserted in  $\omega$ , let  $u_D(\sigma)$  and  $u_N(\sigma) \in H^1(\Omega \setminus \sigma)$  be the solutions of the following Dirichlet and Neumann problems:

$$\left\{ \begin{array}{l} \Delta u_D = 0 \quad \text{in } \omega \setminus \sigma, \\ u_D = v \quad \text{on } \gamma, \\ \partial_n u_D = 0 \quad \text{on } \sigma, \\ u_D = v \quad \text{in } \Omega \setminus \omega. \end{array} \right.$$

$$\left\{ \begin{array}{l} \Delta u_N = 0 \quad \text{in } \omega \setminus \sigma, \\ \partial_n u_N = \partial_n v \quad \text{on } \gamma, \\ \partial_n u_N = 0 \quad \text{on } \sigma, \\ u_N = v \quad \text{in } \Omega \setminus \omega. \end{array} \right.$$



**Goal:** find the cracks  $\sigma$  that minimize the difference between the two solutions:

$$J(\sigma) = \frac{1}{2} \|u_D - u_N\|_{L^2(\Omega)}^2.$$

If  $\sigma_\rho = x + \rho\sigma(n)$ , and  $j(\rho) := J(\sigma_\rho)$ , then

$$j(\rho) - j(0) = f(\rho)g(x, n) + o(f(\rho)),$$

where the topological gradient  $g$  is ([Amstutz et al., 2001]):

$$g(x, n) = - [(\nabla u_D(x) \cdot n)(\nabla p_D(x) \cdot n) + (\nabla u_N(x) \cdot n)(\nabla p_N(x) \cdot n)].$$

$p_N$  and  $p_D$  are the two adjoint states, corresponding to the two direct solutions  $u_N$  and  $u_D$  of the unperturbed problems:

$$\begin{cases} -\Delta p_D = -(u_D - u_N) & \text{in } \omega, \\ p_D = 0 & \text{on } \gamma, \\ p_D = 0 & \text{in } \Omega \setminus \omega, \end{cases} \quad \begin{cases} -\Delta p_N = +(u_D - u_N) & \text{in } \omega, \\ \partial_n p_N = 0 & \text{on } \gamma, \\ p_N = 0 & \text{in } \Omega \setminus \omega. \end{cases}$$

## Topological gradient:

$$g(x, n) = - [(\nabla u_D(x) \cdot n)(\nabla p_D(x) \cdot n) + (\nabla u_N(x) \cdot n)(\nabla p_N(x) \cdot n)] = n^T M(x) n,$$

where  $M(x)$  is a  $2 \times 2$  symmetric matrix (or  $3 \times 3$  in 3D problems or movies), defined by

$$M(x) = -\text{sym}(\nabla u_D(x) \otimes \nabla p_D(x) + \nabla u_N(x) \otimes \nabla p_N(x)).$$

Then the minimal value of  $g(x, n)$  is reached when  $n$  is the eigenvector associated to the **most negative eigenvalue**  $\lambda_{\min}(M(x))$  of  $M(x)$ .

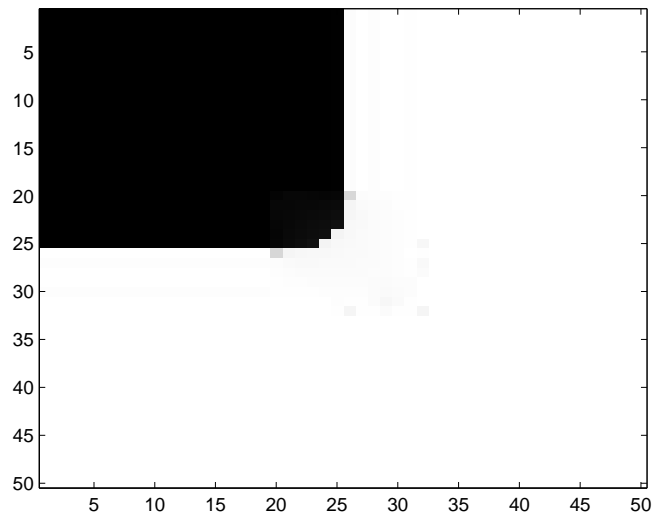
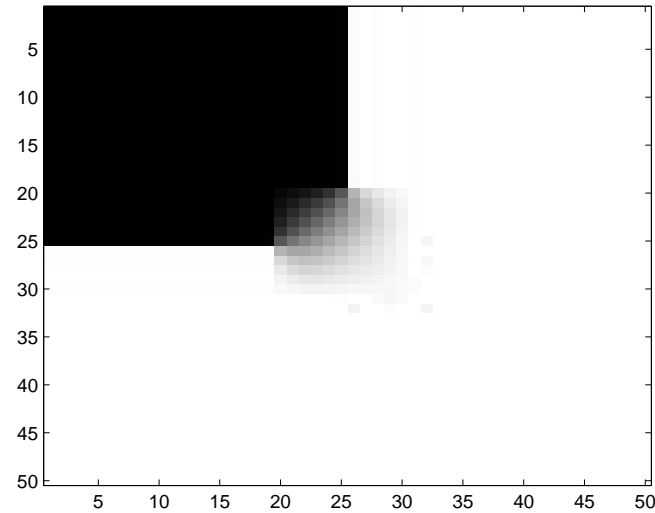
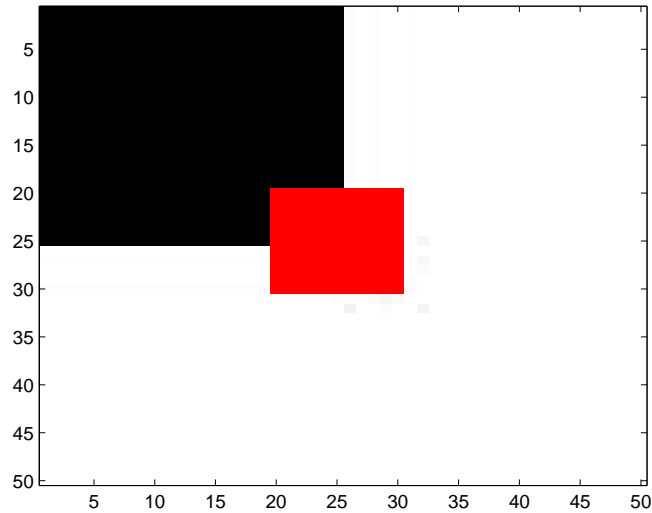
## Algorithm:

- Compute the solutions of the two direct problems (unperturbed:  $\sigma = \emptyset$ ).
- Compute the solutions of the two adjoint problems ( $\sigma = \emptyset$ ).
- Compute the matrix  $M(x)$  for each  $x \in \omega$ .
- Define the set of identified edges:

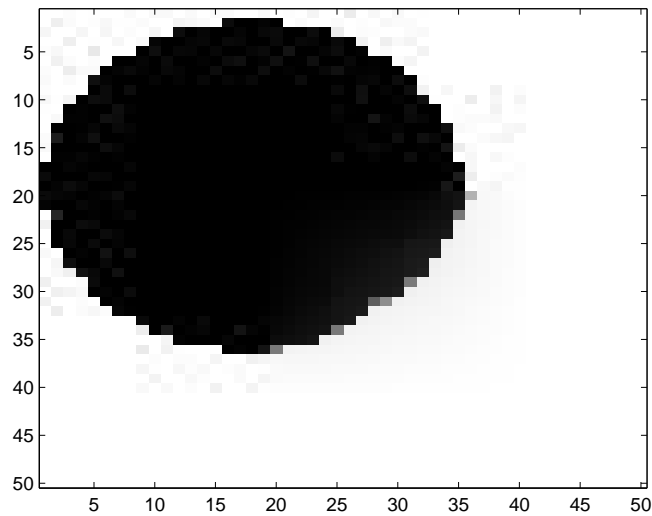
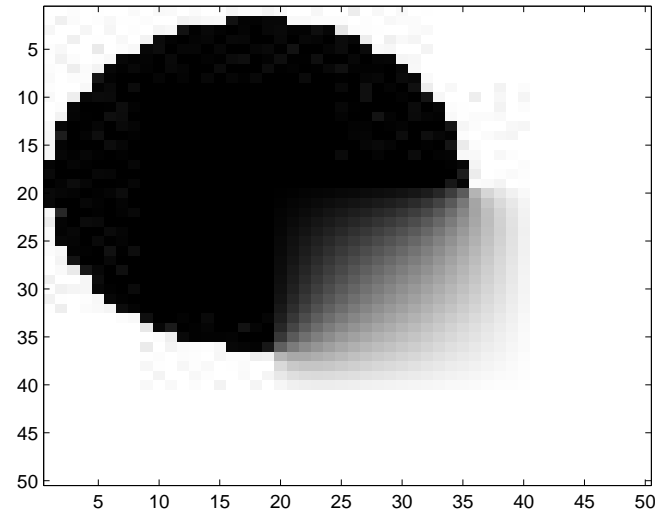
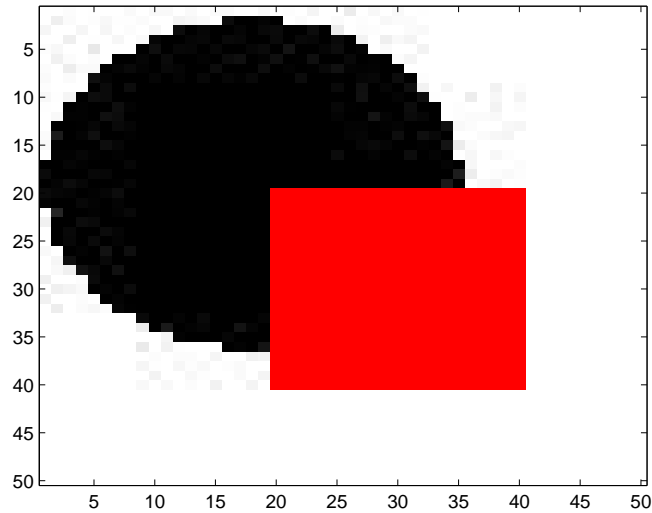
$$\sigma = \{x \in \omega; \lambda_{\min}(M(x)) < \delta < 0\},$$

where  $\delta$  is a negative threshold.

- Compute the solution of the direct Neumann (or Dirichlet) problem, perturbed by the insertion of  $\sigma$ .



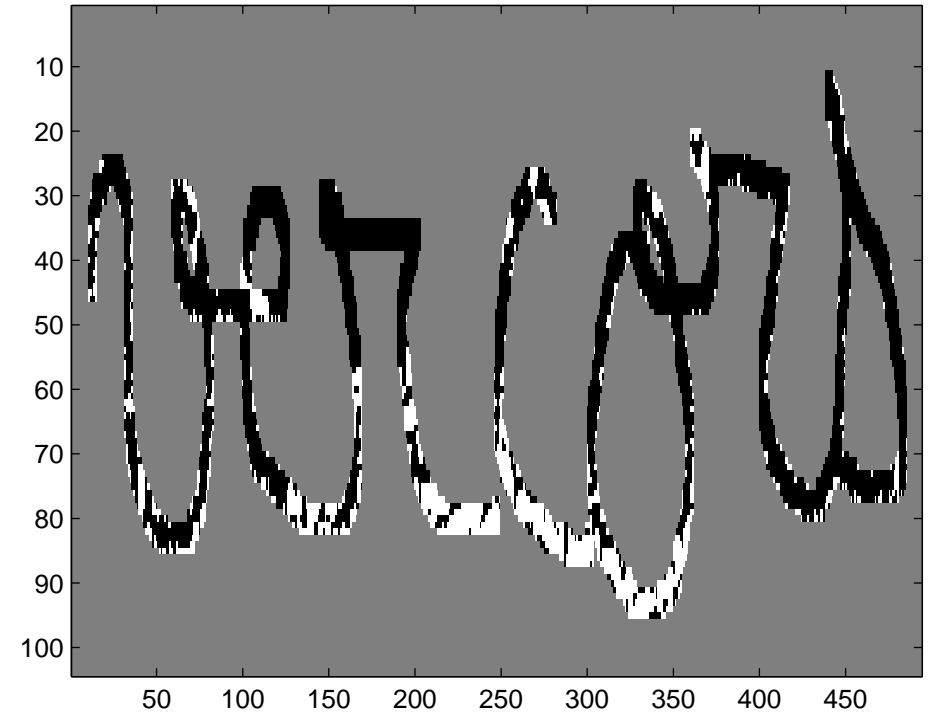
Original image; Inpainted image by linear diffusion; Inpainted image by the topological gradient.



Original image; Inpainted image by linear diffusion; Inpainted image by the topological gradient.

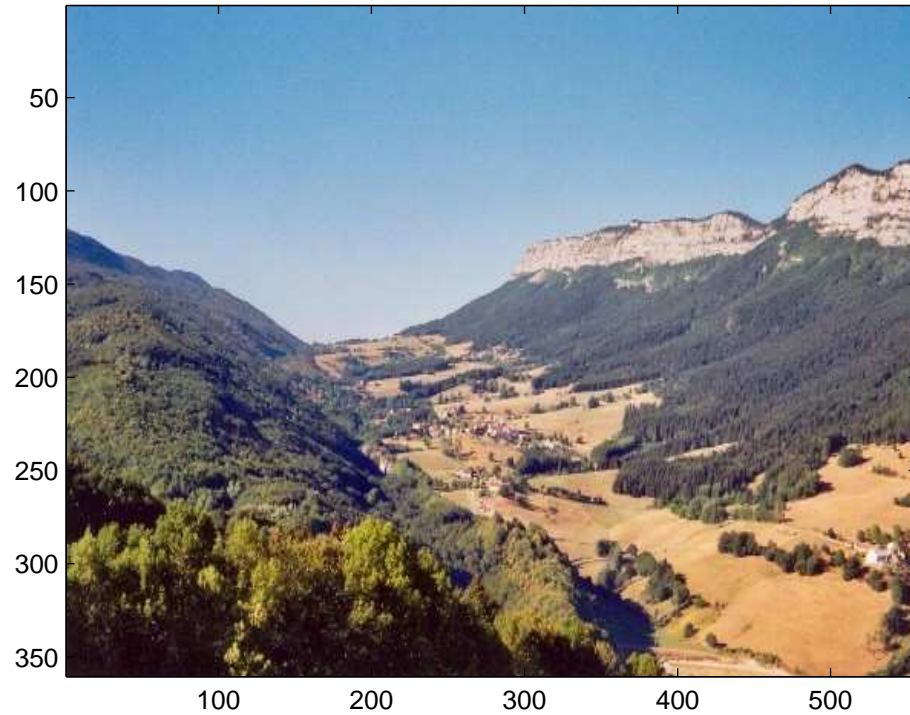


(a)

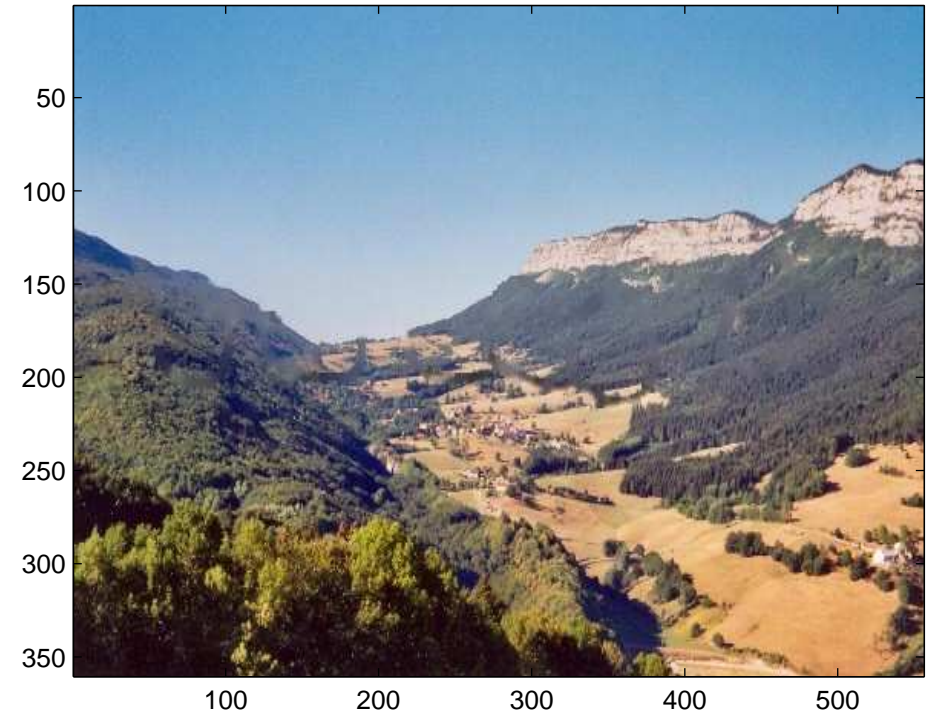


(b)

Figure 3: Original image (a); Hidden edges identified by the topological gradient (b).



(a)



(b)

Figure 4: Original image (a); Inpainted image (b).

# ALGORITHM COMPLEXITY



All algorithms consist in solving the following equation:

$$\begin{cases} -\operatorname{div}(c\nabla u) + u = v & \text{in } \Omega, \\ \partial_n u = 0 & \text{on } \partial\Omega, \end{cases}$$

with various values of  $c$ . In the first computations, the conductivity  $c$  is **constant**.

**Discrete cosine transform (DCT):**

$$\sum_{m,n} (1 + c(m\pi)^2 + c(n\pi)^2) u_{m,n} \phi_{m,n} = \sum_{m,n} v_{m,n} \phi_{m,n},$$

where  $\phi_{m,n} = \delta_{m,n} \cos(m\pi x) \cos(n\pi y)$  is a cosine basis, and  $(v_{m,n})$  are the DCT coefficients of the original image  $v$ .

The computations of the unperturbed problems are made as follows:

- compute the DCT coefficients  $v_{m,n}$  of the original image  $v$ .
- compute the DCT coefficients  $u_{m,n}$ :

$$u_{m,n} = \frac{v_{m,n}}{1 + c(m\pi)^2 + c(n\pi)^2}.$$

- compute the image  $u$  from its coefficients  $u_{m,n}$  (inverse DCT).

The complexity of a DCT is  $\mathcal{O}(n \cdot \log(n))$  operations, where  $n$  is the number of pixels of the image.

The perturbed problem can be written as follows:

$$A(c)u = B,$$

where  $u$  is the unknown (note that the equations are linear!!). As  $c$  is close to a constant (sparse edge set), we use a preconditioned conjugate gradient method:

$$[A(c_0)^{-1}A(c)] u = [A(c_0)^{-1}B].$$

$\rightsquigarrow$  the matrix  $A(c_0)^{-1}A(c)$  is now close to the identity matrix.

**Complexity of the algorithms:**  $\mathcal{O}(n \cdot \log(n))$  operations  $\rightsquigarrow$  real time processing [A. and Masmoudi, 2009].

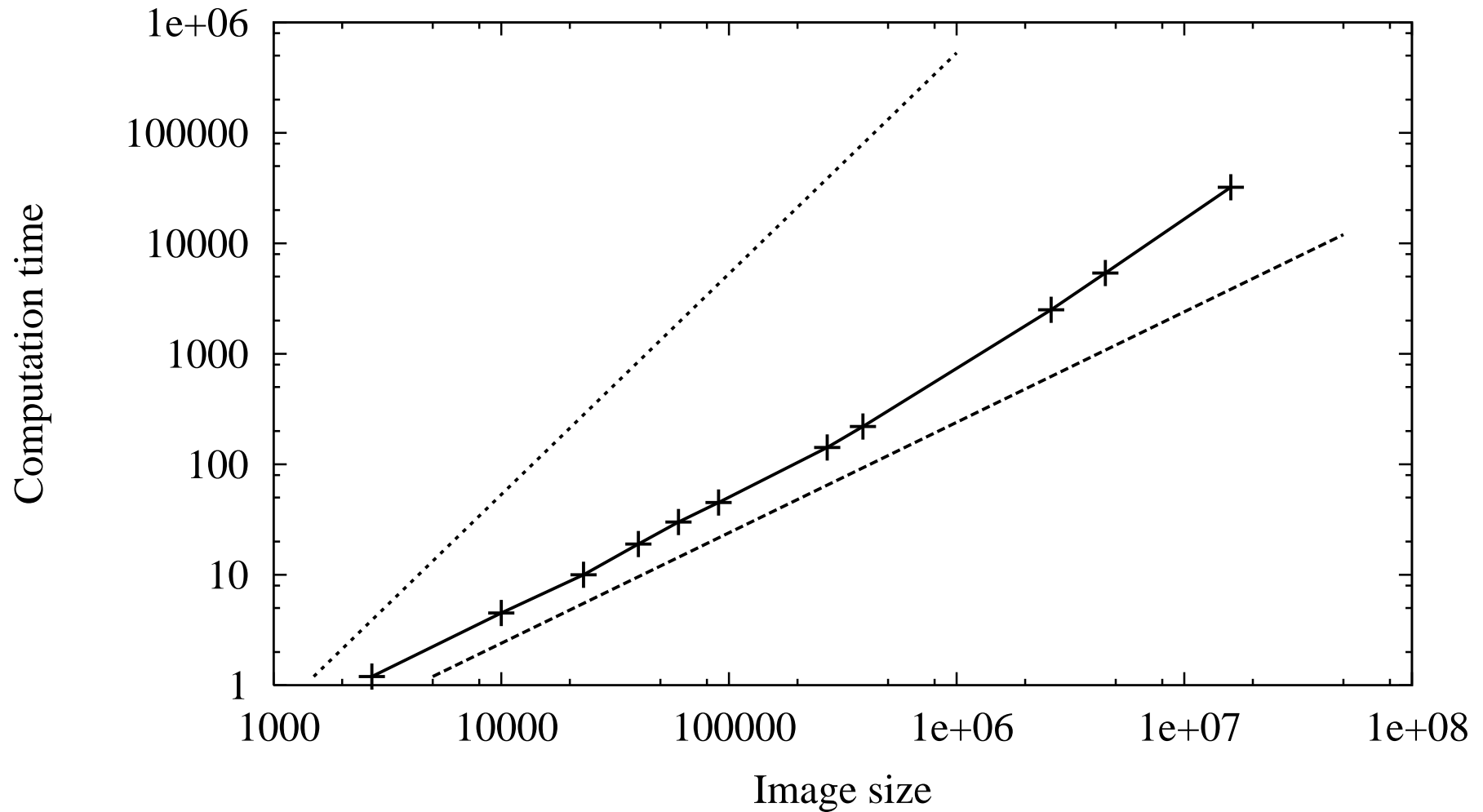


Figure 5: Computing time versus the image size (old computations, on a very old and tired laptop, with Matlab... imagine a C code for CUDA on a brand new GPU!).

# TOPOLOGICAL GRADIENT AND MINIMAL PATHS

**Main issue:** for several applications (segmentation, inpainting), it is crucial to have **connected edges**.

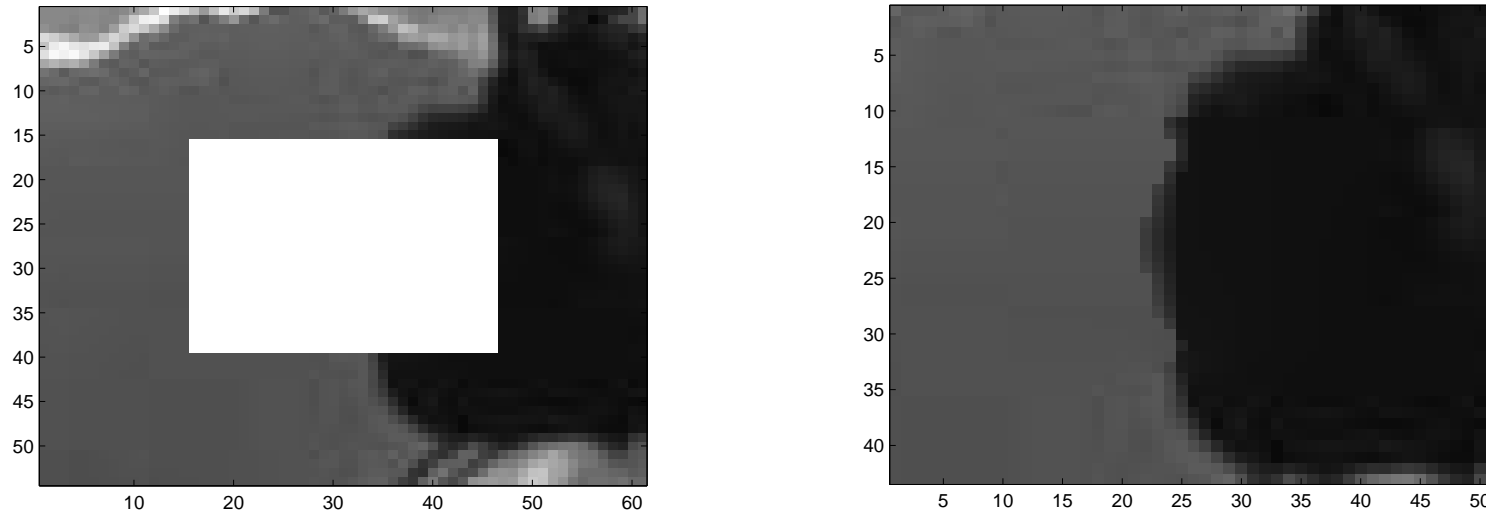


Figure 6: Inpainting with discontinuous edges.

**Idea:** instead of **thresholding** the topological gradient, we will look for the negative **valley lines** of the topological gradient  $\rightsquigarrow$  minimal paths and fast marching methods [Cohen et al., 1997-2005; Tian et al., 2003; Telea et al., 2002].

Let  $g$  be the topological gradient, the idea is to define a **potential function**, measuring for any point of the image the **cost** for a path to contain this point. And then we will find the **minimal path** between two edge points, in order to connect them. For positivity reasons:

$$P(x) = g(x) - \min_{y \in \Omega} \{g(y)\}.$$

Let  $C(s)$  be a path in  $\Omega$ , where  $s$  represents the arclength. The global cost of  $C$  is:

$$J(C) = \int_C (P(C(s)) + \alpha) ds,$$

where  $\alpha > 0$  is a regularization coefficient, measuring the real length of the path.

**Goal:** find the minimal path between two points  $\rightsquigarrow$  we define a **distance function**:

$$D(x; x_0) = \inf_{C \in \mathcal{A}(x, x_0)} J(C),$$

where  $\mathcal{A}(x, x_0)$  is the set of all paths connecting  $x$  and  $x_0$ .

**Front propagation:**

$$\frac{\partial C(s, t)}{\partial t} = \frac{1}{P(C(s, t)) + \alpha} \mathbf{n}_C(s, t),$$

where  $\mathbf{n}_C(s, t)$  is the outward unit normal to the front  $C$ . We initialize the propagation with  $C(s, 0)$ , an infinitely small circle centered at point  $x_0$ .

**Fast marching:** the distance function satisfies the following Eikonal equation:

$$\|\nabla_x D(x; x_0)\| = P(x) + \alpha,$$

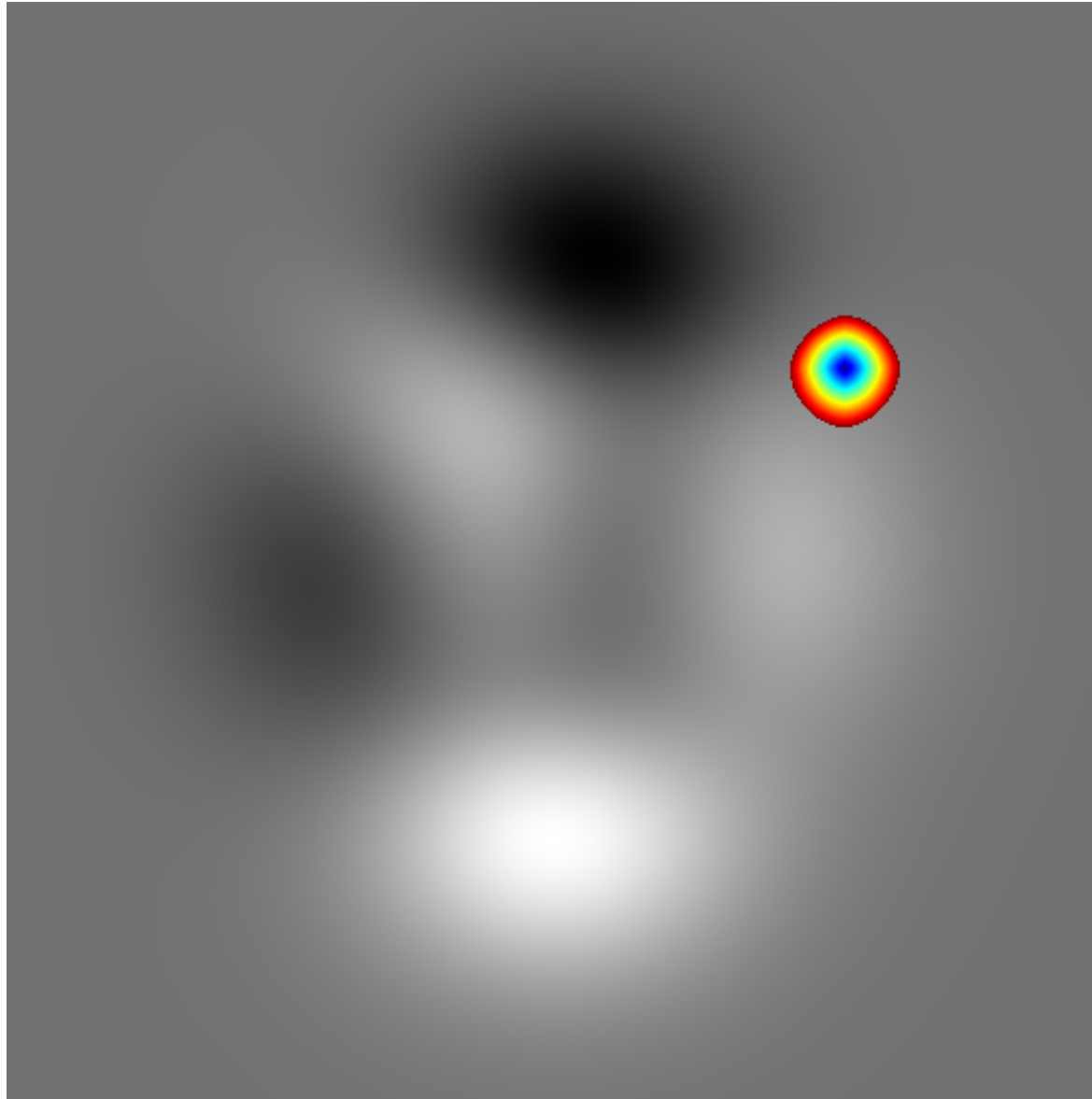
with the initialization  $D(x_0, x_0) = 0$ .

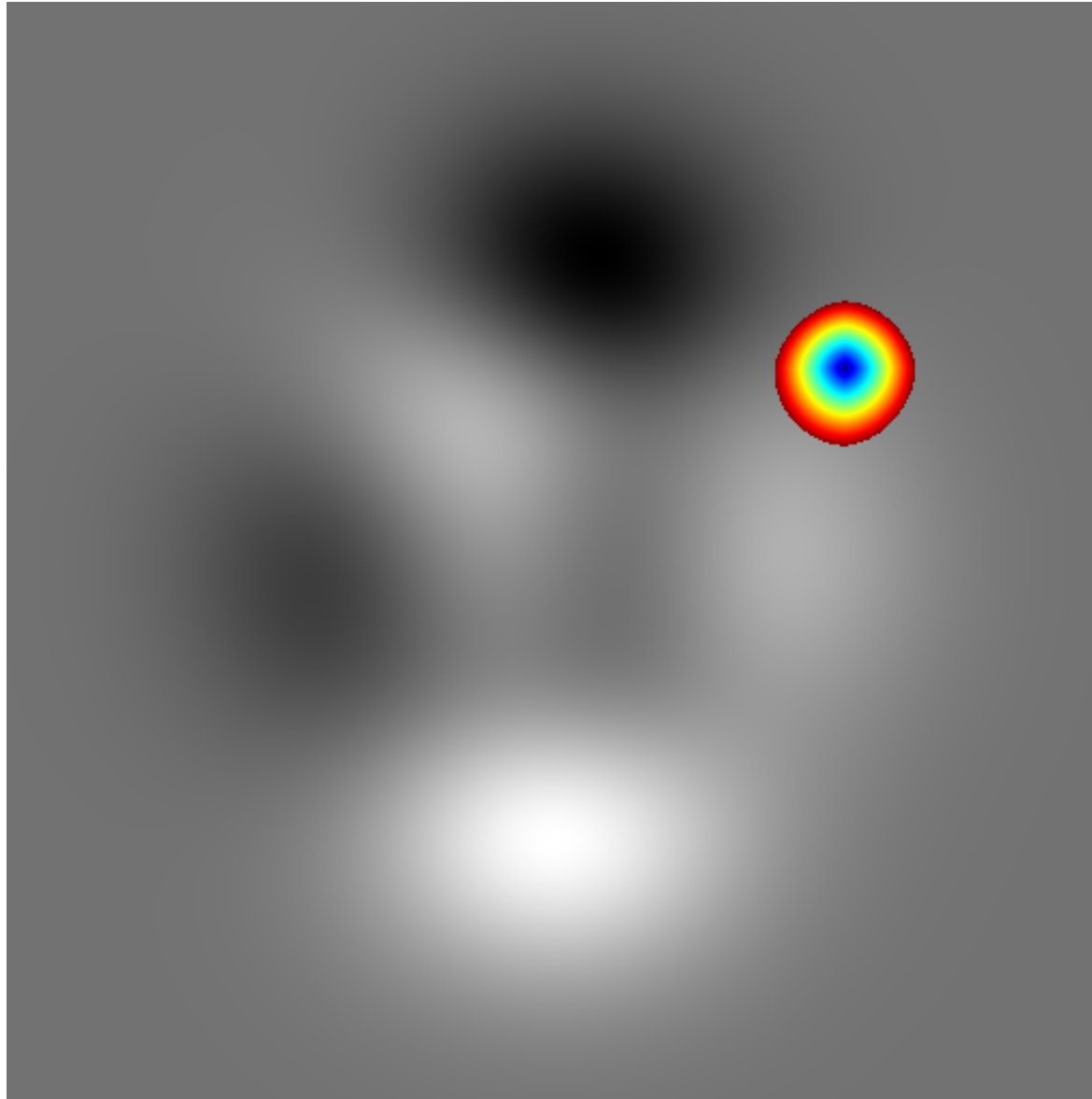
$\rightsquigarrow D(x; x_0)$  is exactly the **time  $t$  at which the front  $C$  initialized at point  $x_0$  reaches  $x$**  [Tian et al., 2003; Dicker, 2006].

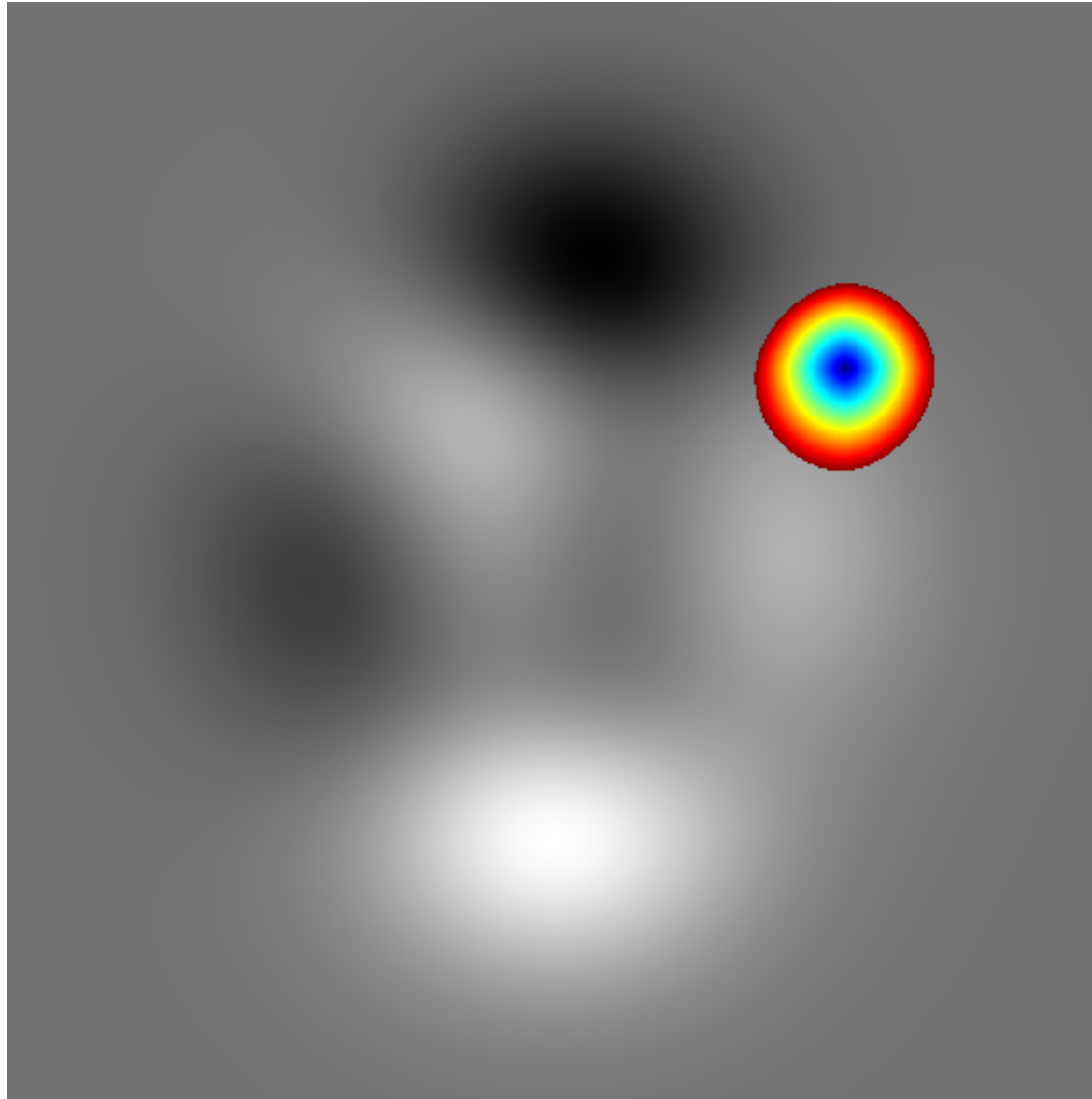


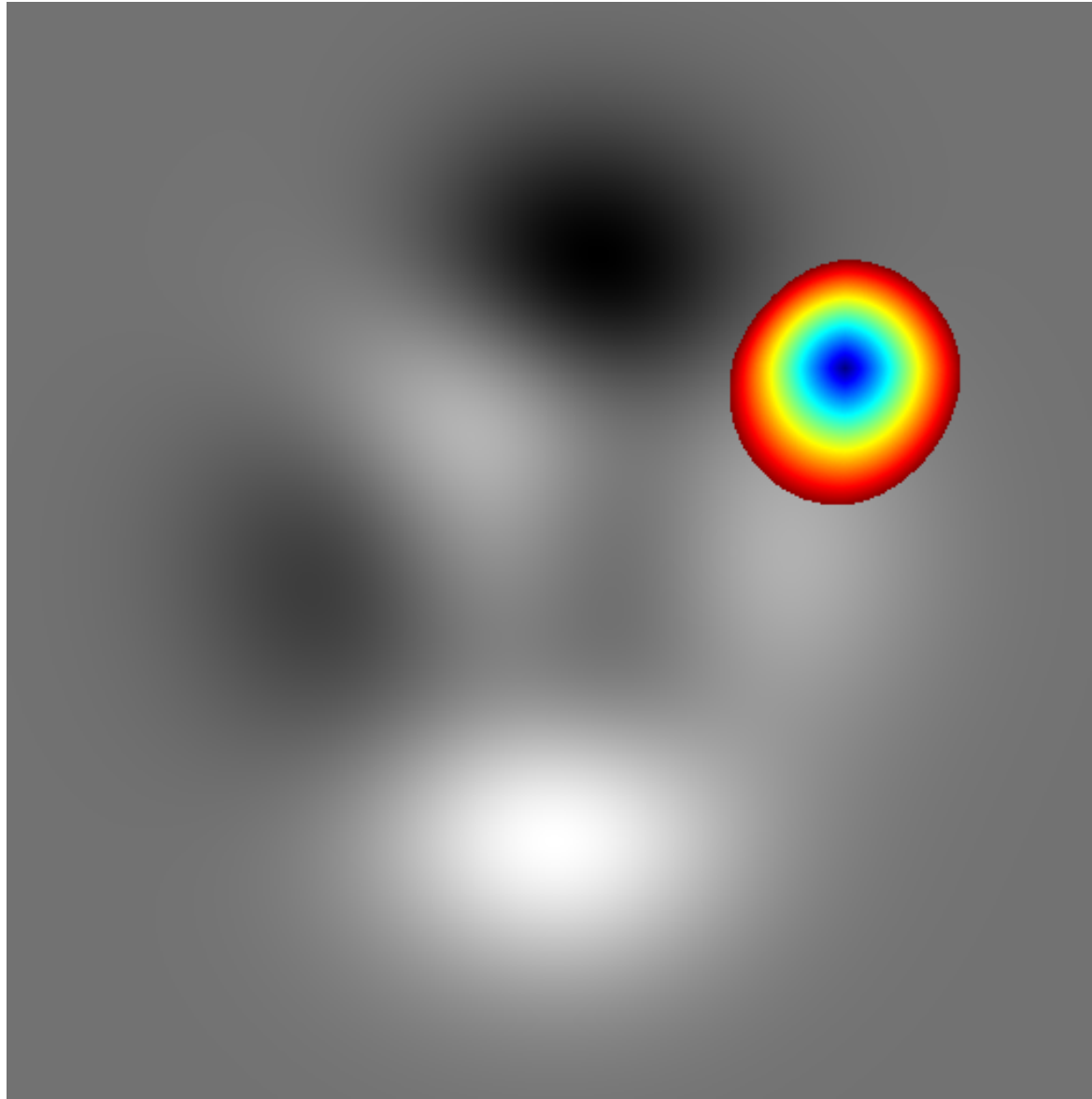


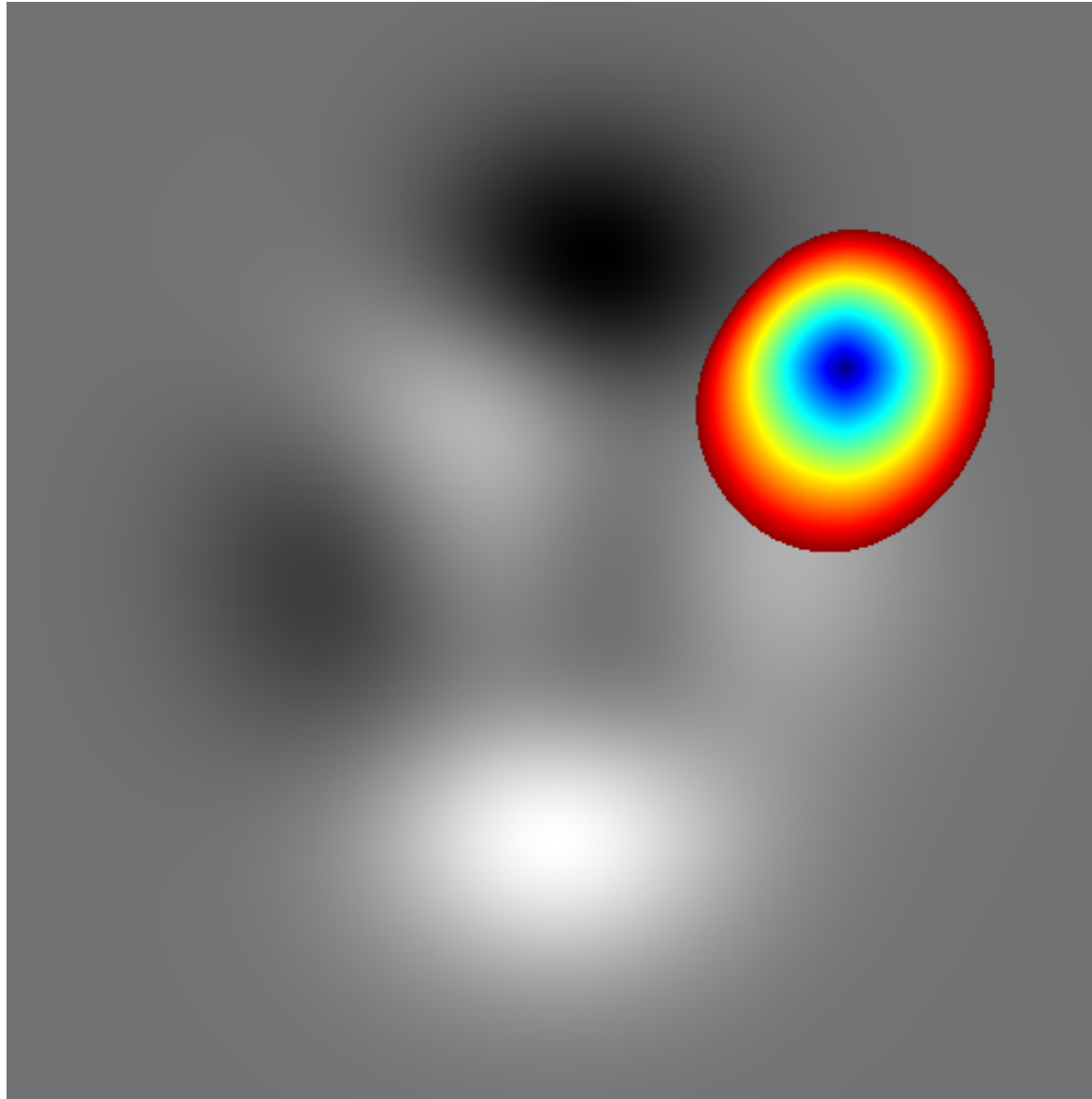


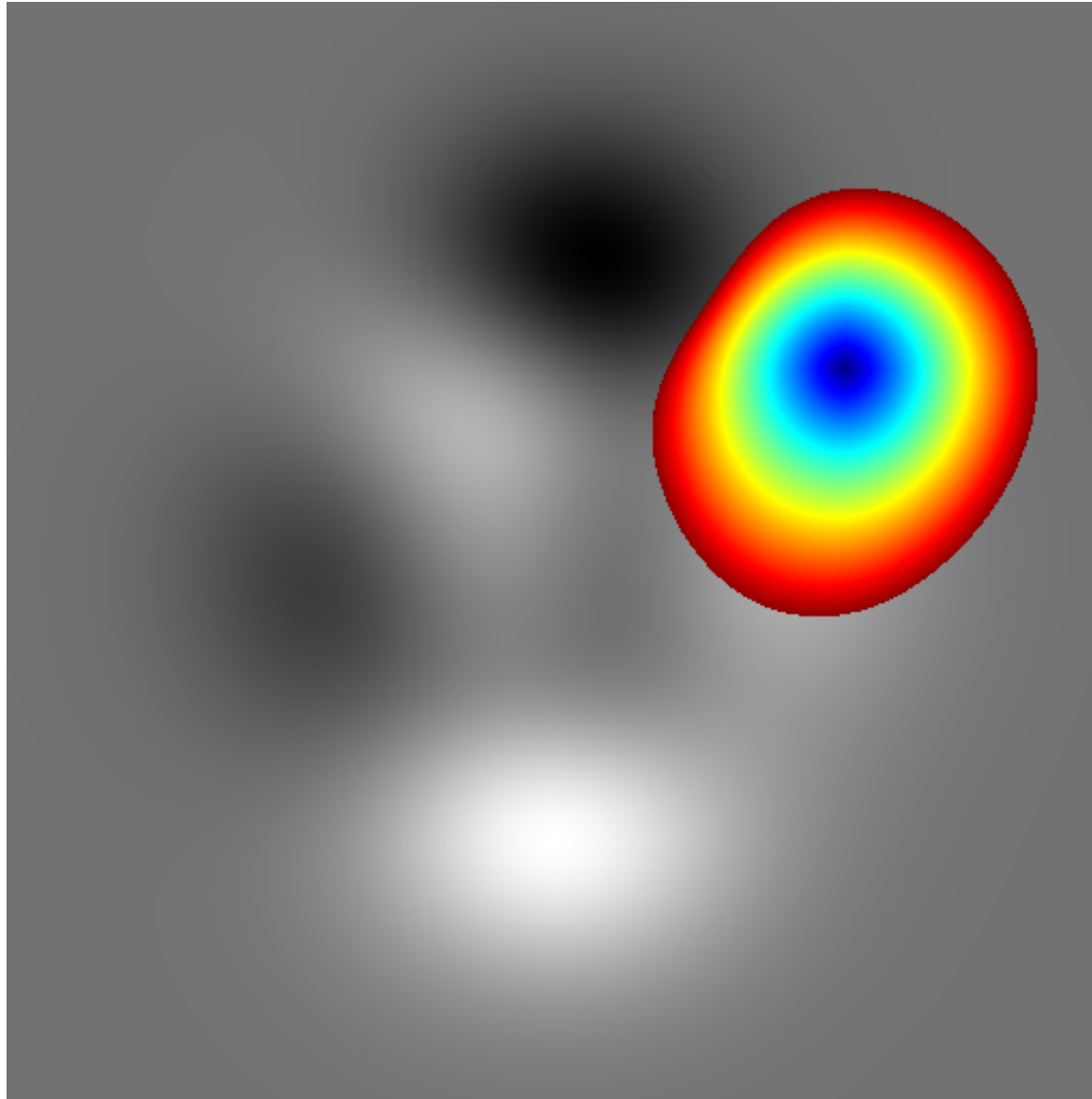




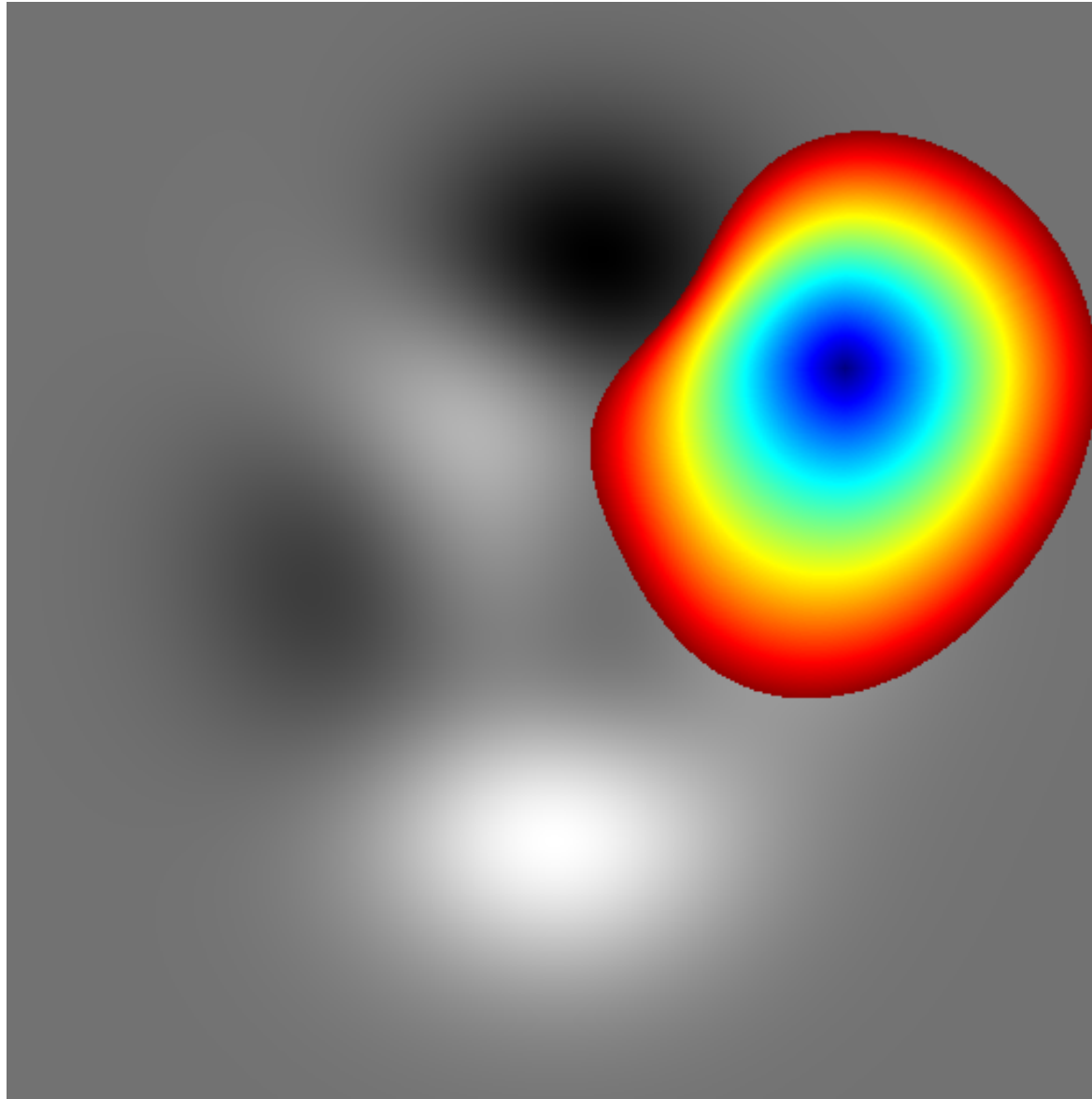


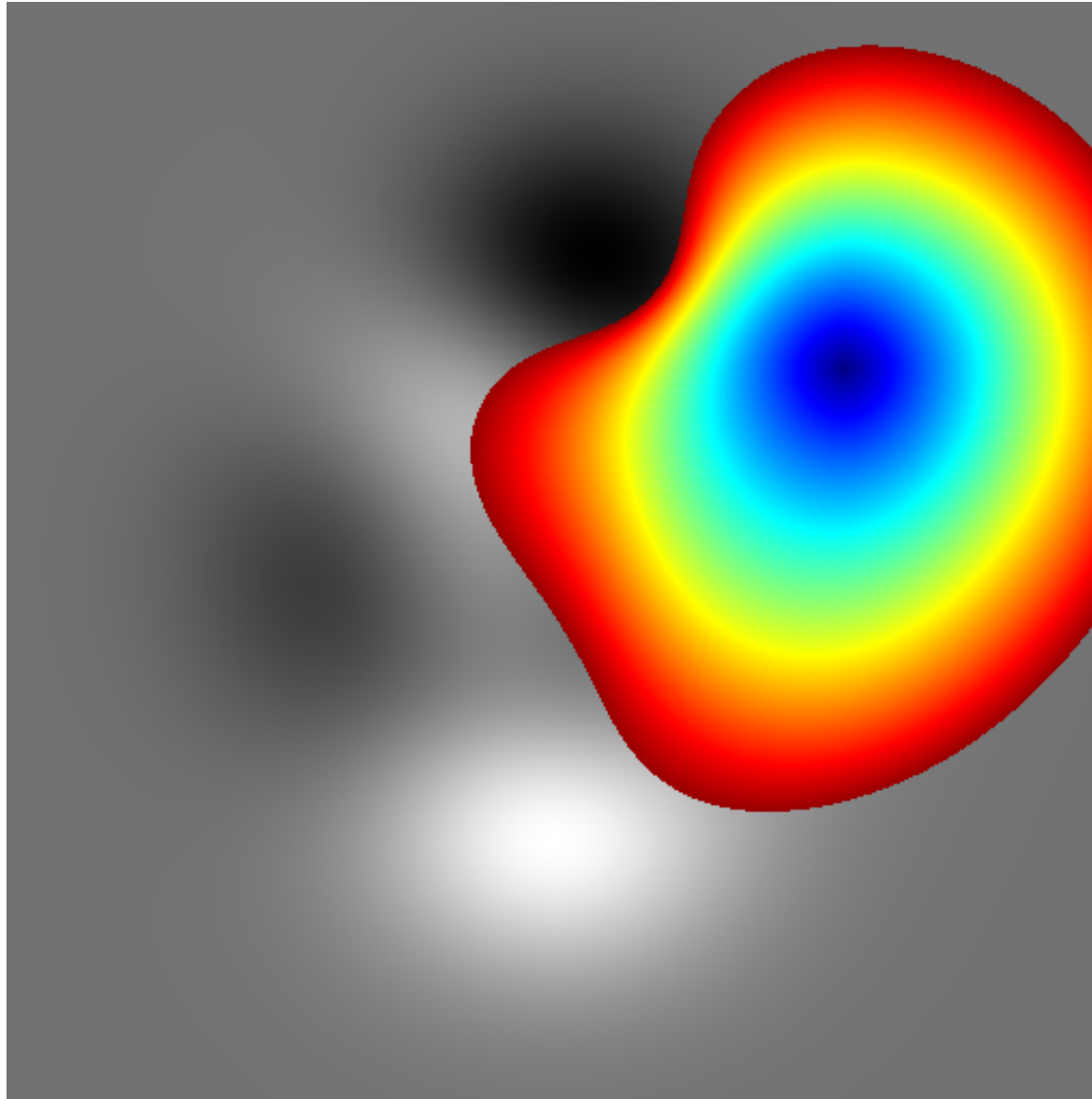


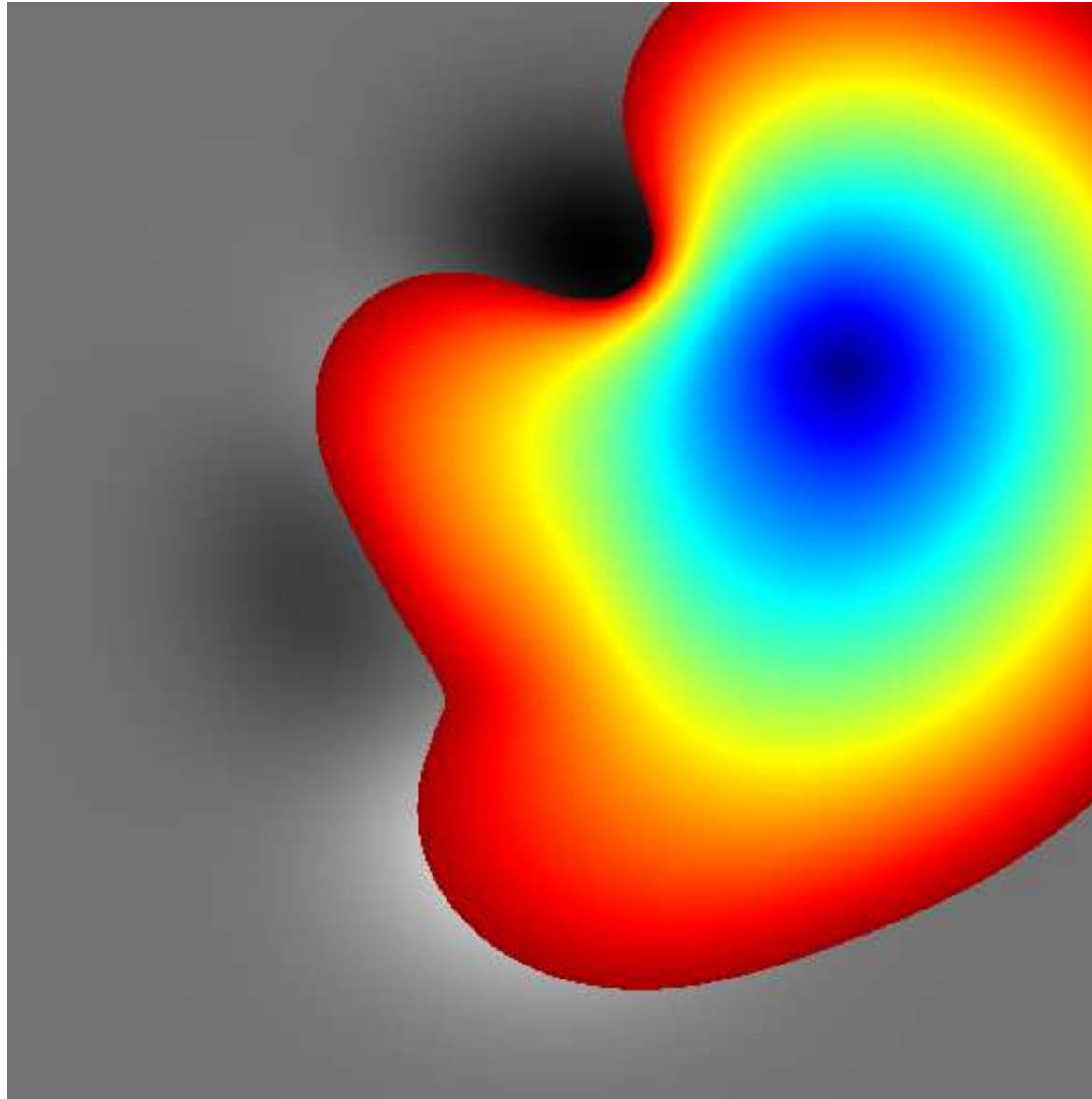


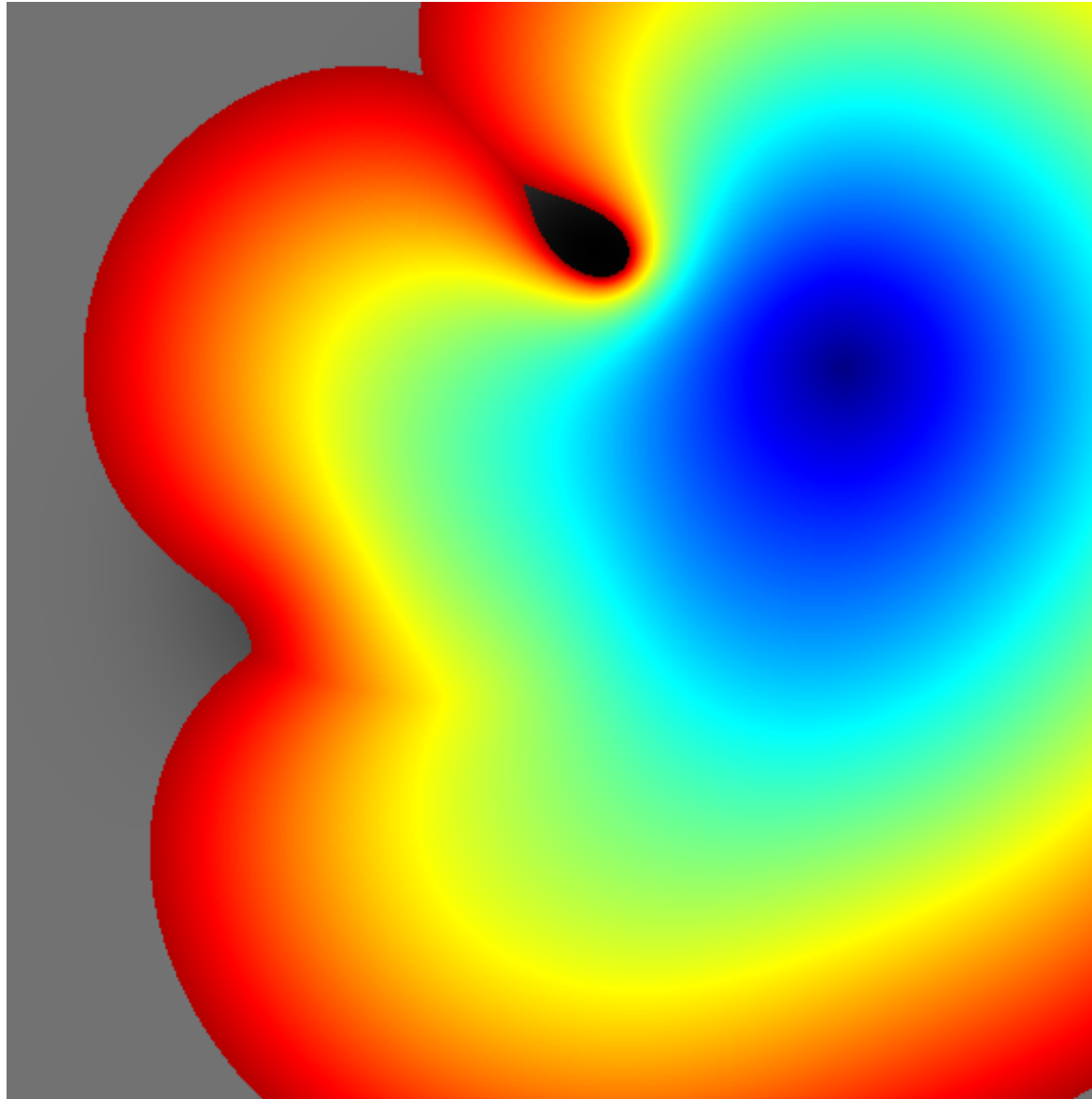


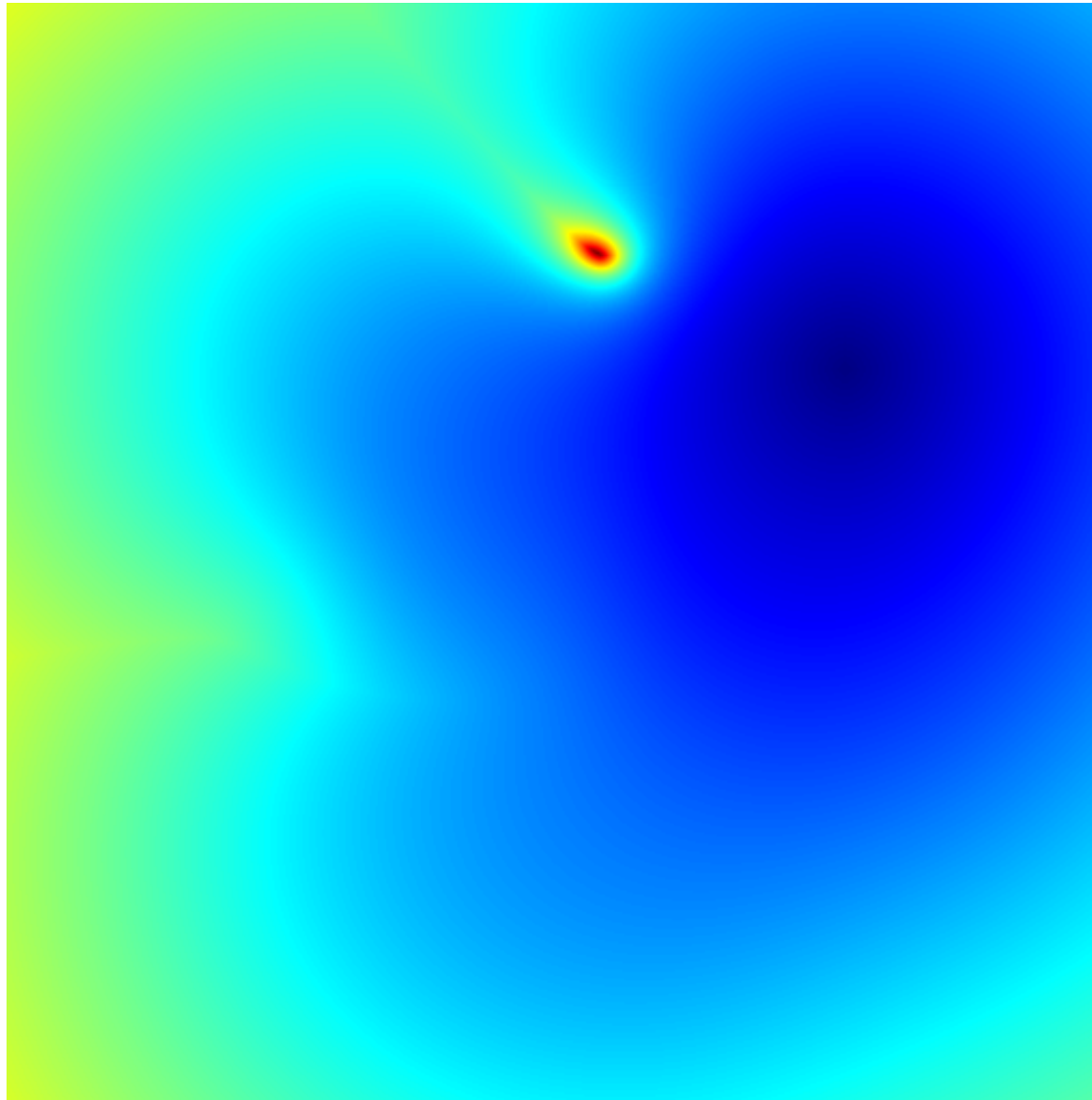








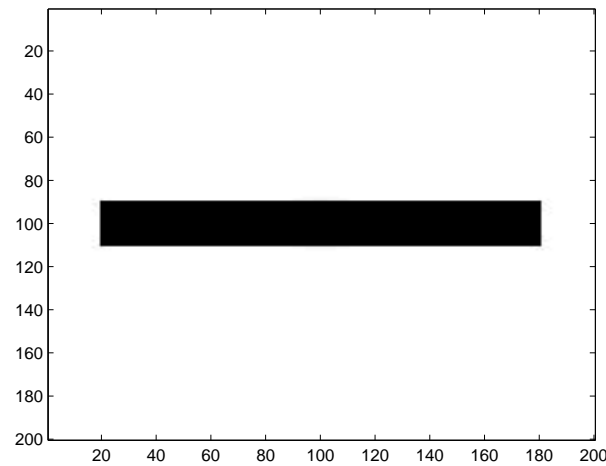
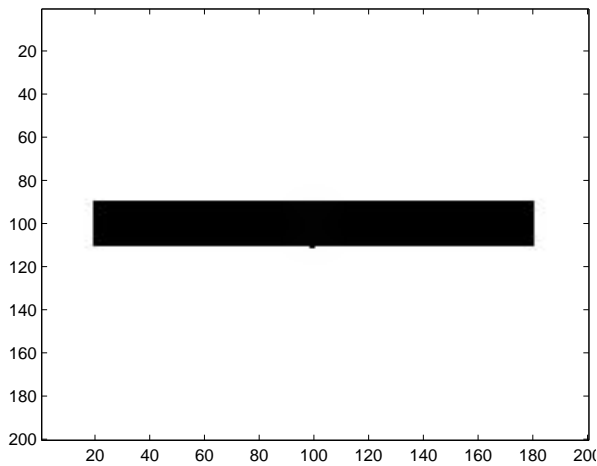
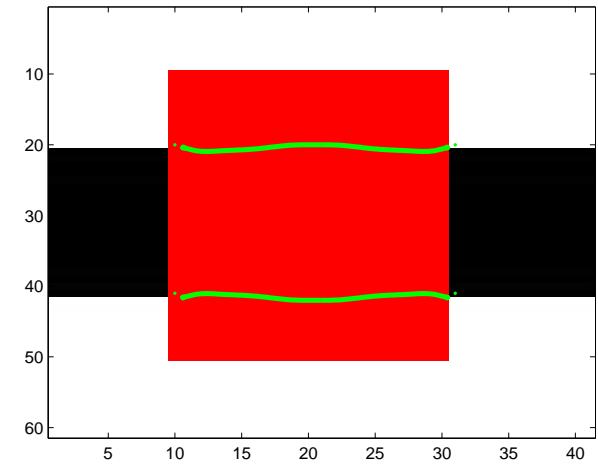
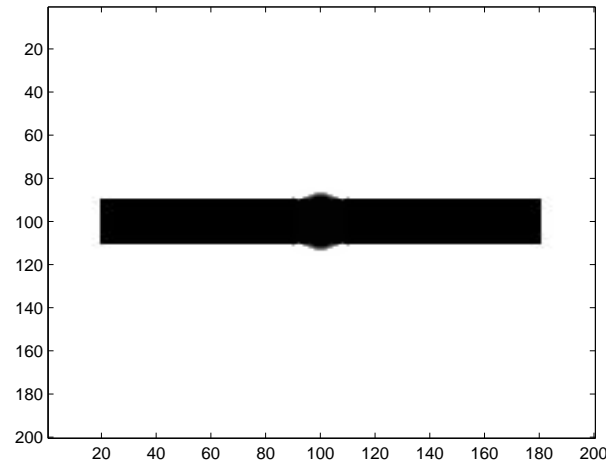
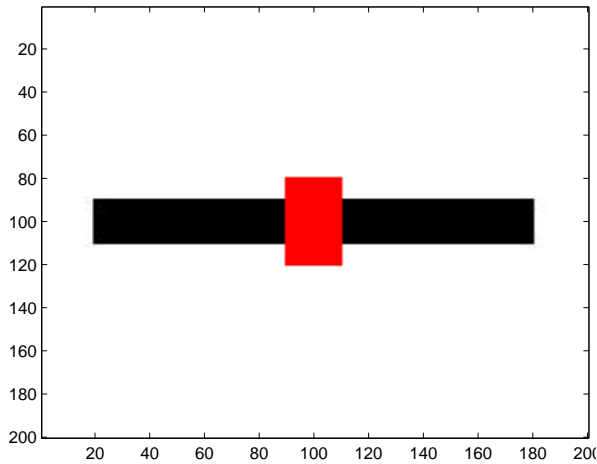




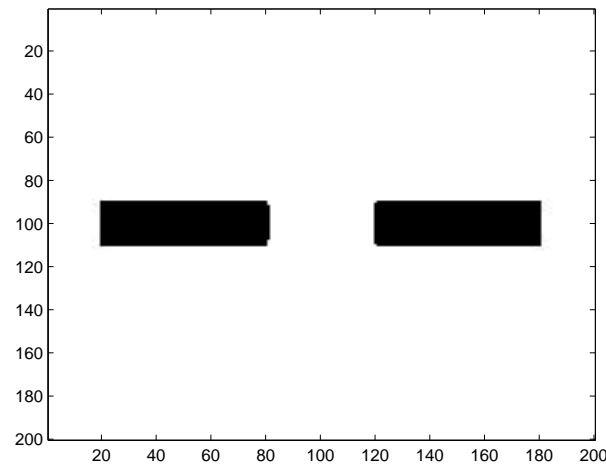
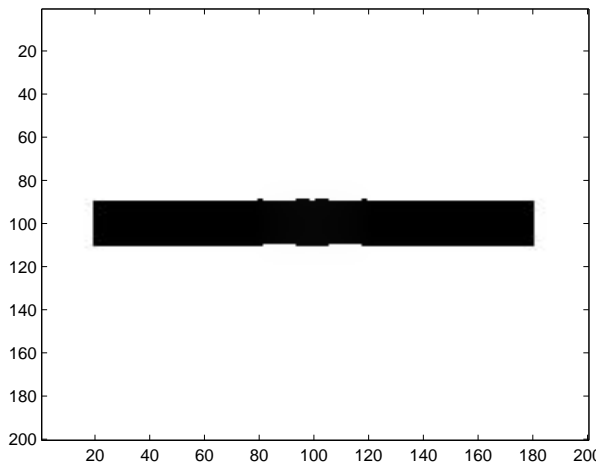
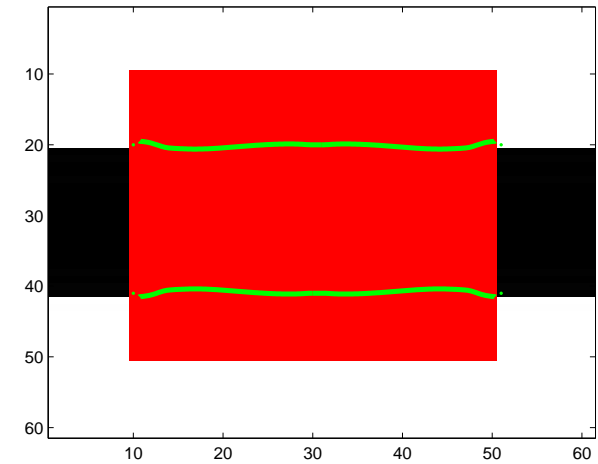
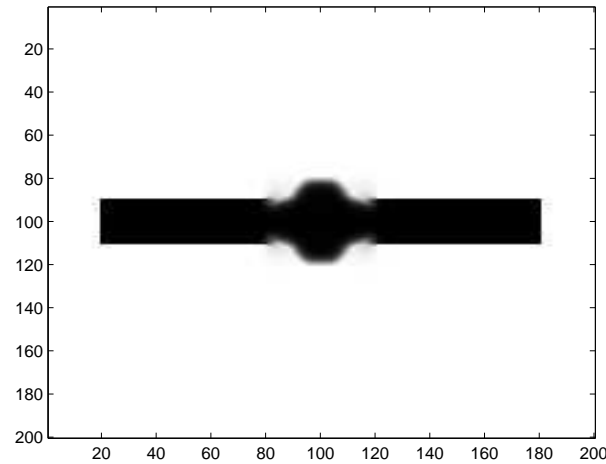
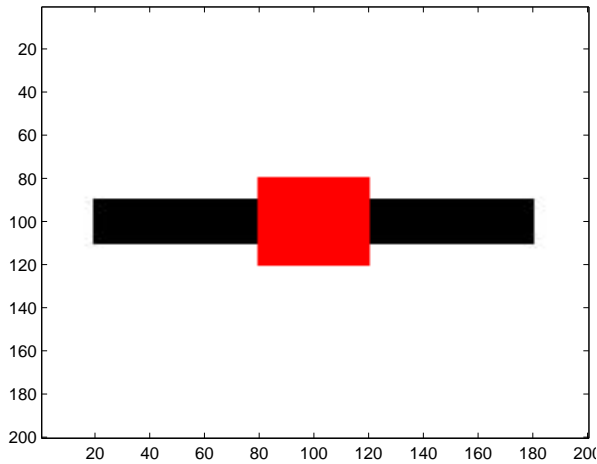
## Fast marching algorithm applied to the topological gradient:

- Compute the **topological gradient** of the image (for edge detection or inpainting).
- Set  $N$  the number of **keypoints**, and choose the  $N$  keypoints: the main one is the **global minimum** of the topological gradient, the other ones being the **most negative local minima** of the topological gradient.
- Compute the **distance function** with all these keypoints, and the corresponding **Voronoi diagram**.
- Compute the set of **saddle-points**: on each edge of the Voronoi diagram, determine the point of minimal distance.
- Sort all these points of minimal distance, from smaller to larger distance.
- Perform the **back-propagation** from these points: use the saddle-point as an initialization for a descent type algorithm in order to **connect the two corresponding keypoints**.

[A., Cohen and Masmoudi; 2011]

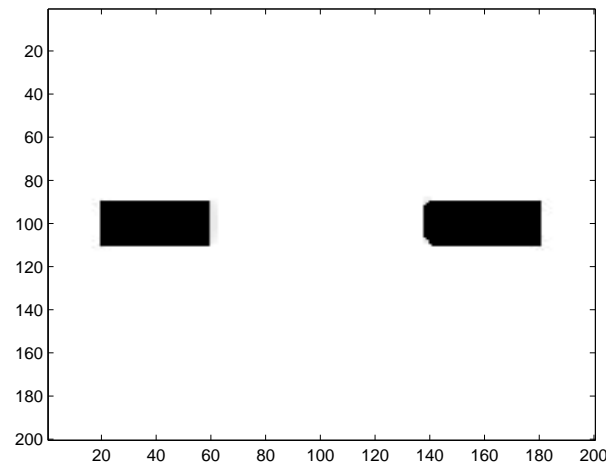
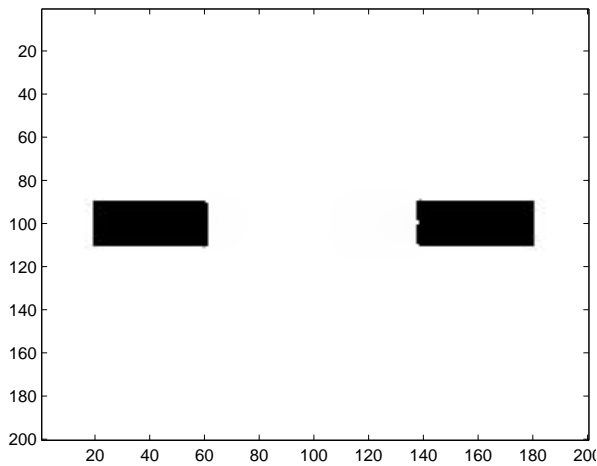
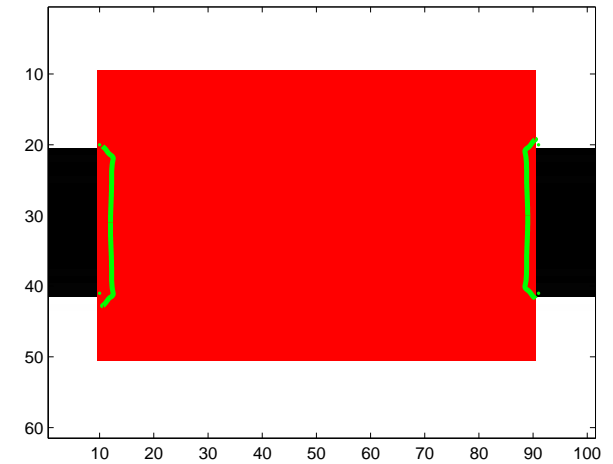
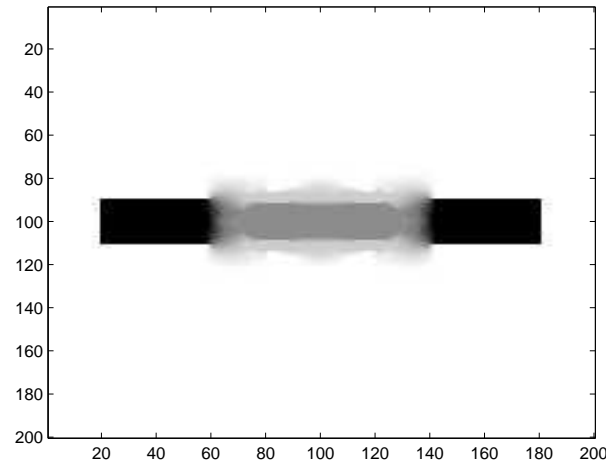
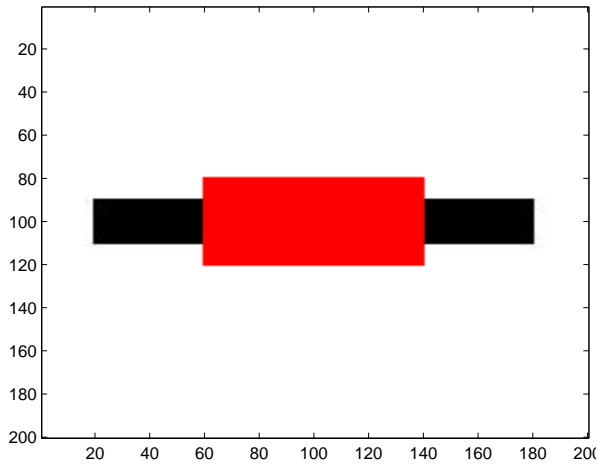


Original image; Inpainted image by the topological gradient; Minimal path on the topological gradient; Inpainted image by the hybrid scheme; Inpainted image using a TV method.

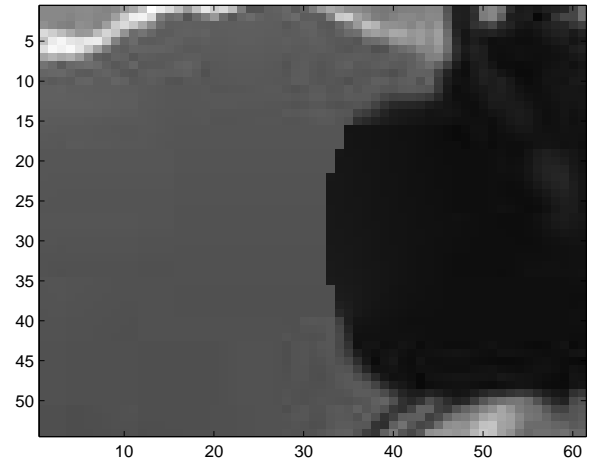
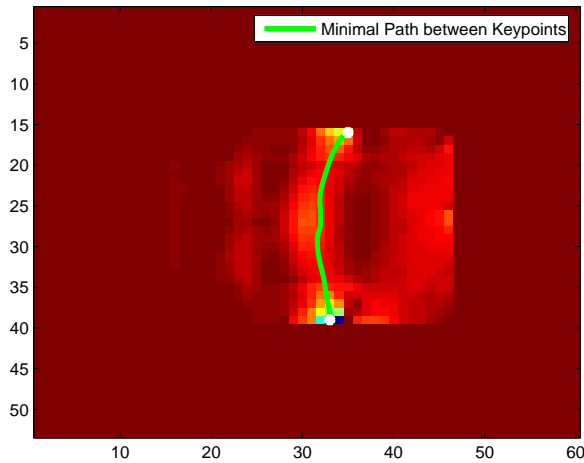
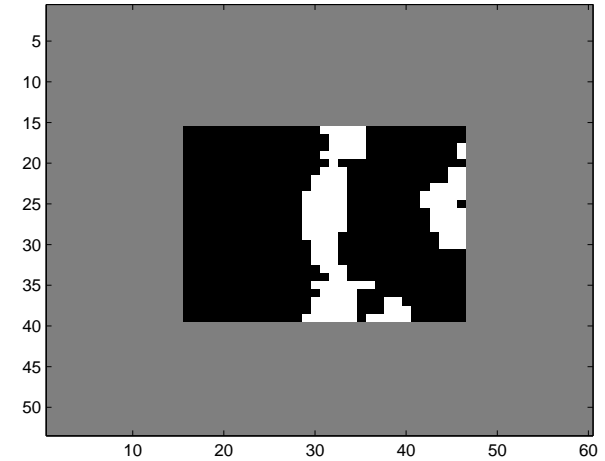
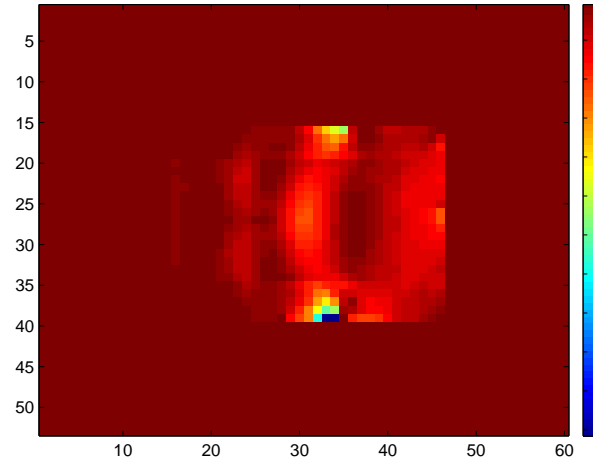
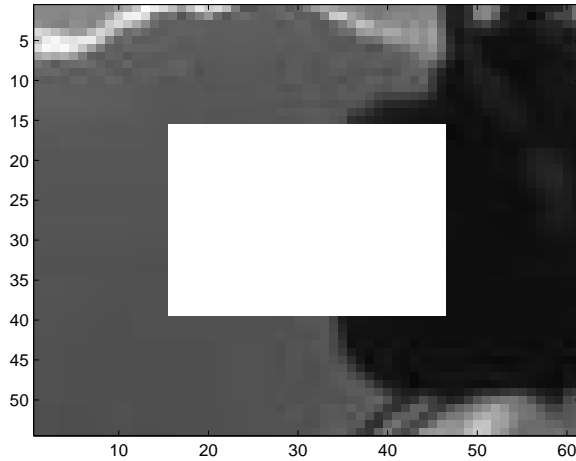


Original image; Inpainted image by the topological gradient; Minimal path on the topological gradient; Inpainted image by the hybrid scheme; Inpainted image using a TV method.





Original image; Inpainted image by the topological gradient; Minimal path on the topological gradient; Inpainted image by the hybrid scheme; Inpainted image using a TV method.



Inpainting problem: hidden part on the original image; topological gradient; edges identified by thresholding; identified minimal path on the topological gradient; inpainted image.

# CONCLUSIONS AND PERSPECTIVES

- Hybrid scheme, combining an efficient edge detection by topological asymptotic expansion, and minimal paths for contour completion.
- Extraction/identification of connected edges in 2D.
- Same complexity:  $\mathcal{O}(n \cdot \log(n))$  operations  $\rightsquigarrow$  real time processing of images and movies.
- Improvement for both segmentation and inpainting applications.
- Extension to 3D problems: almost straightforward.

---

**THANK YOU !**



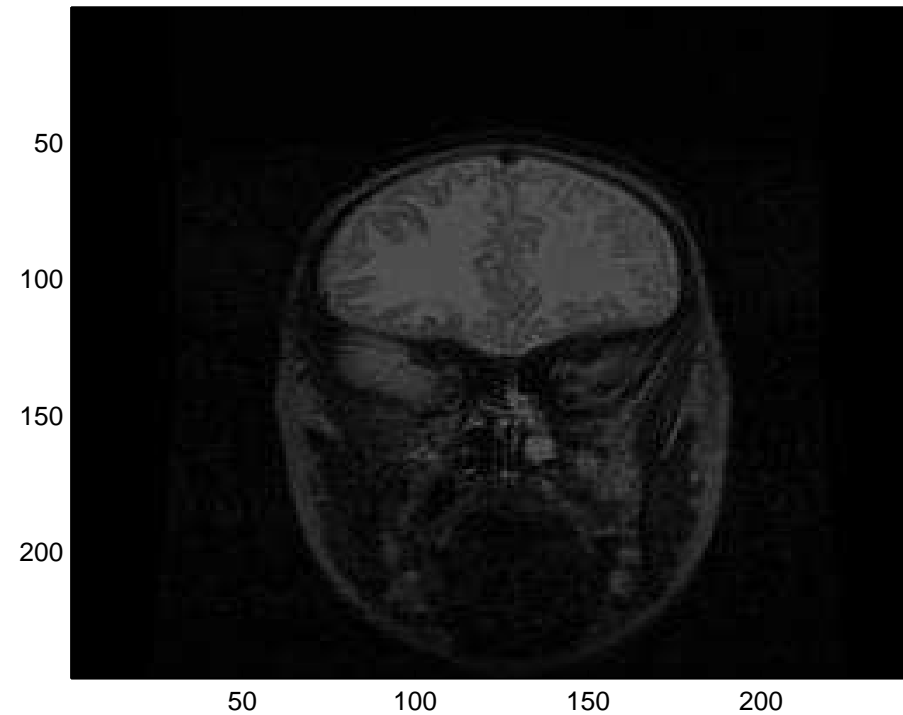
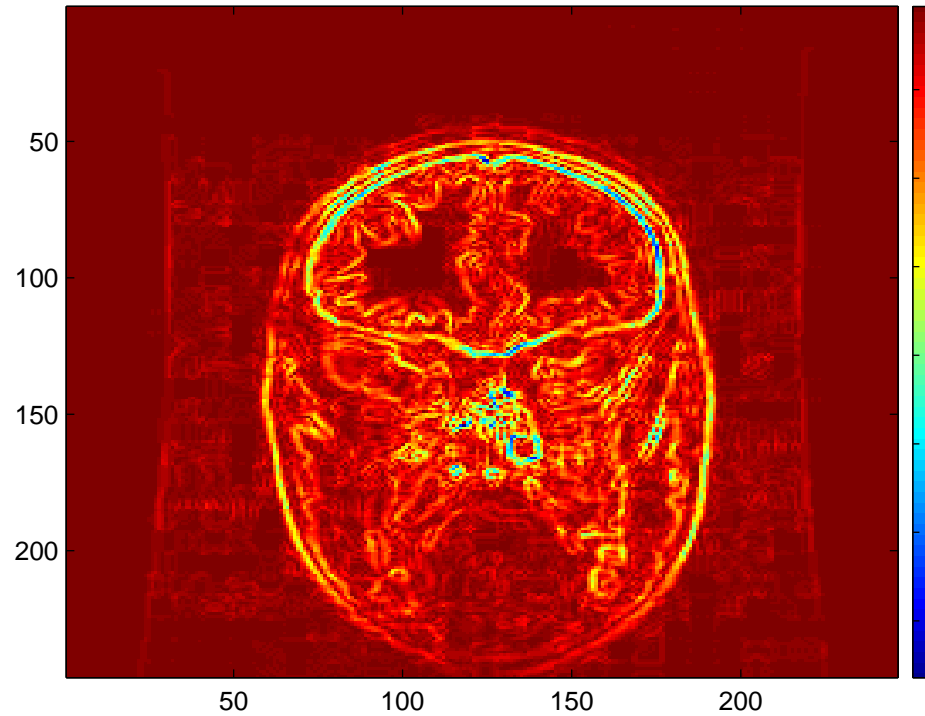
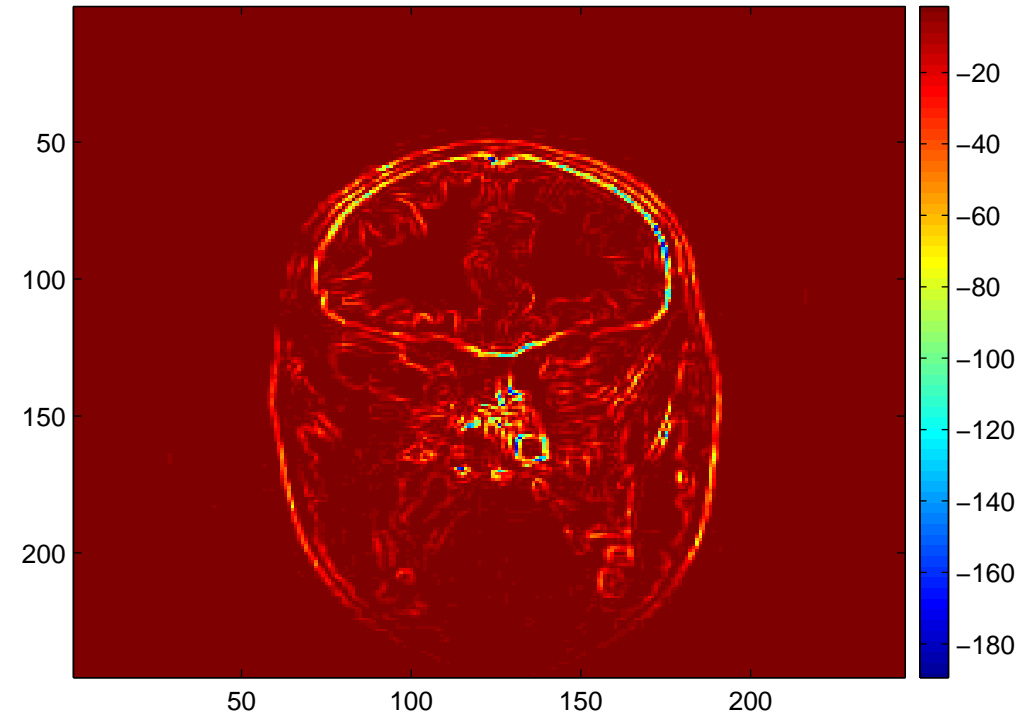


Figure 7: Original image.



(a)



(b)

Figure 8: Standard gradient (a); Topological gradient (b).



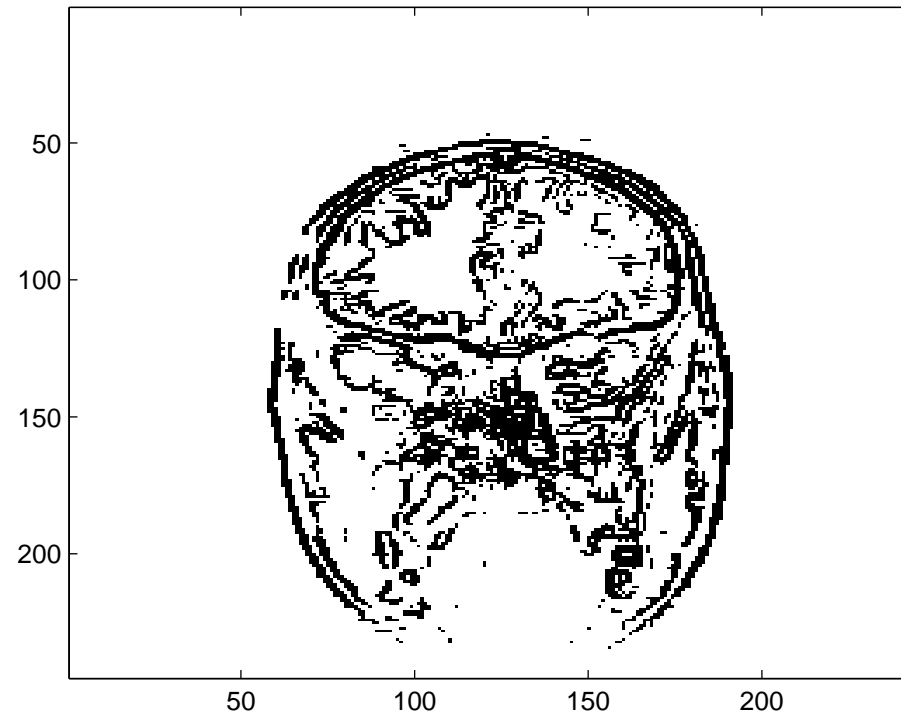
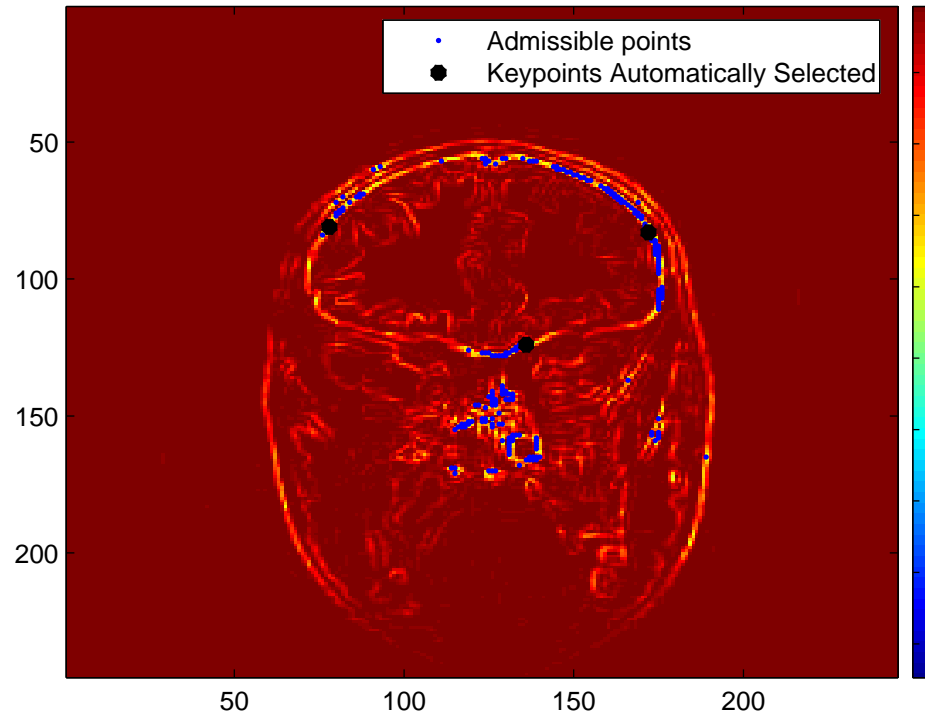
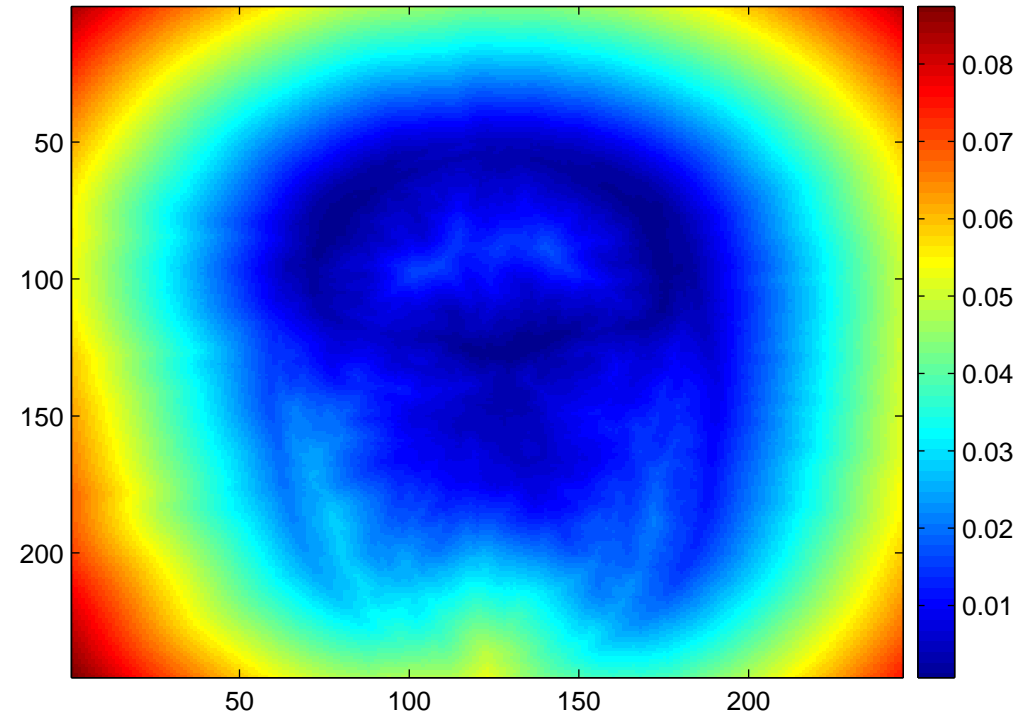


Figure 9: Edges identified by thresholding the topological gradient.

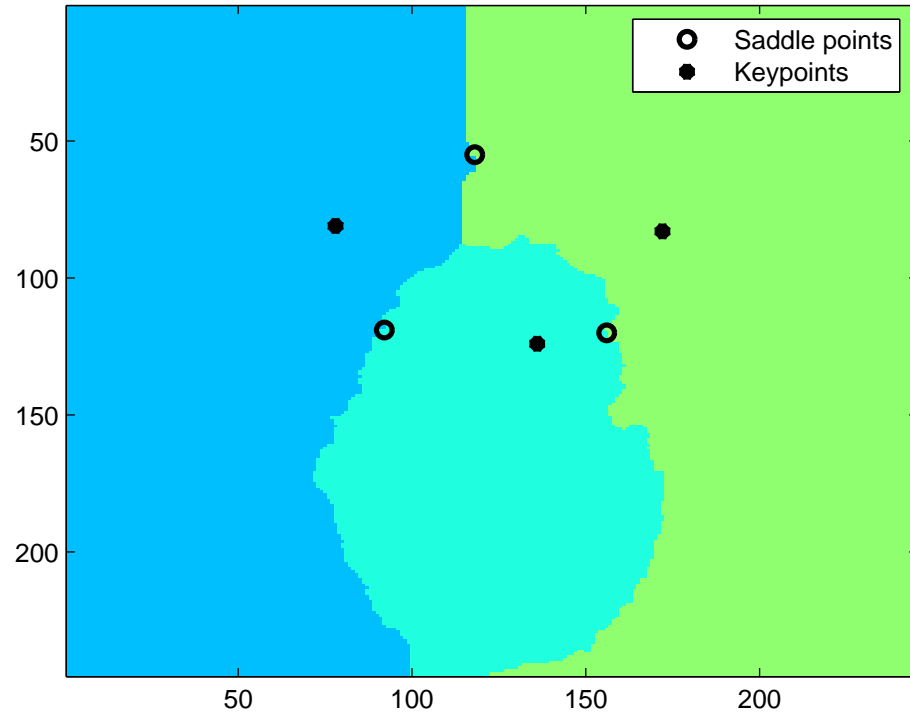


(a)

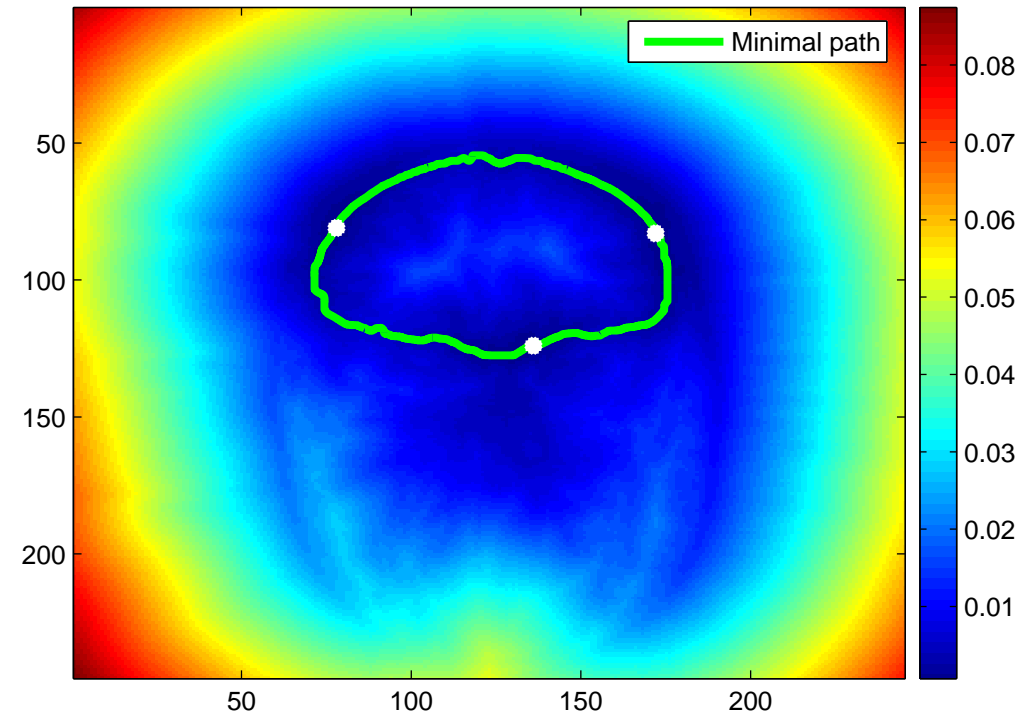


(b)

Figure 10: Keypoints (a); Distance function (b).

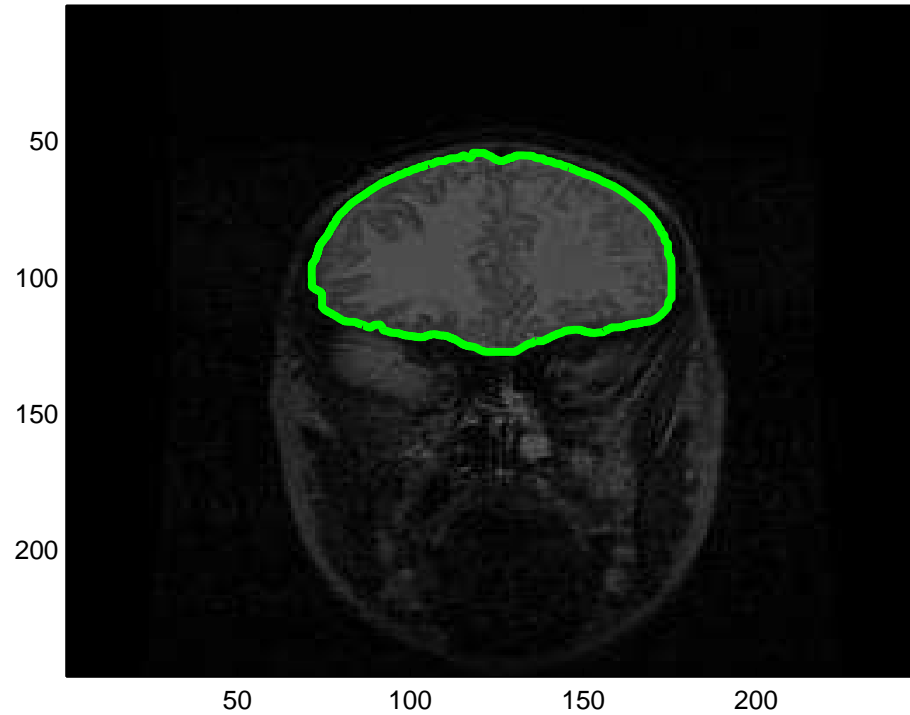


(a)

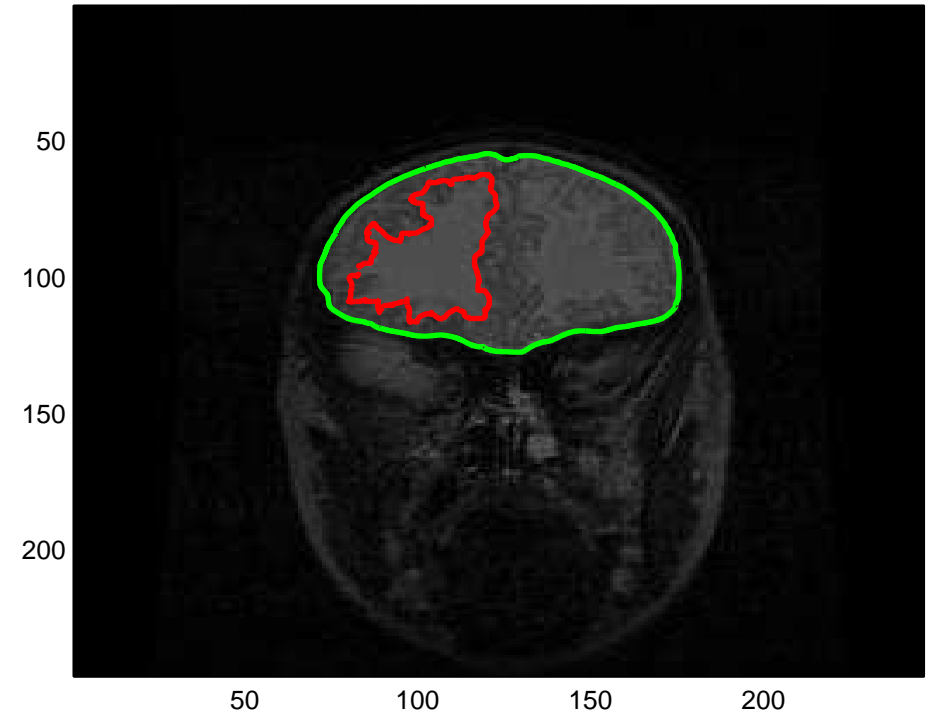


(b)

Figure 11: Voronoi diagram (a); Identified minimal path (b).

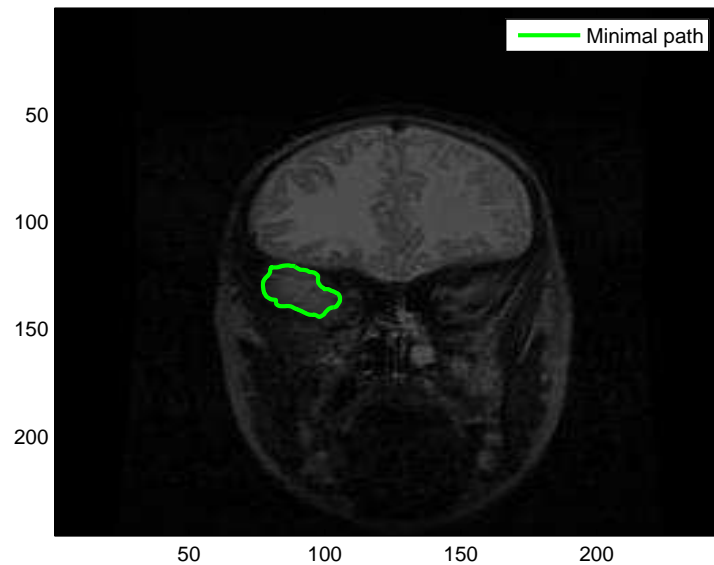
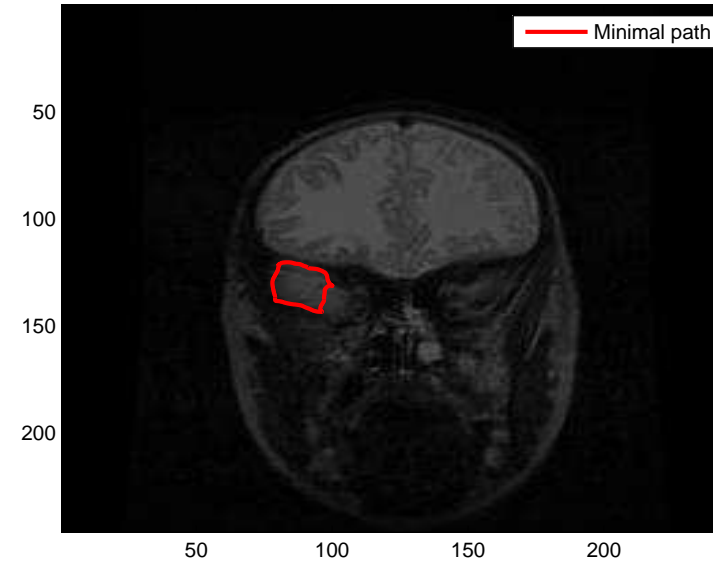
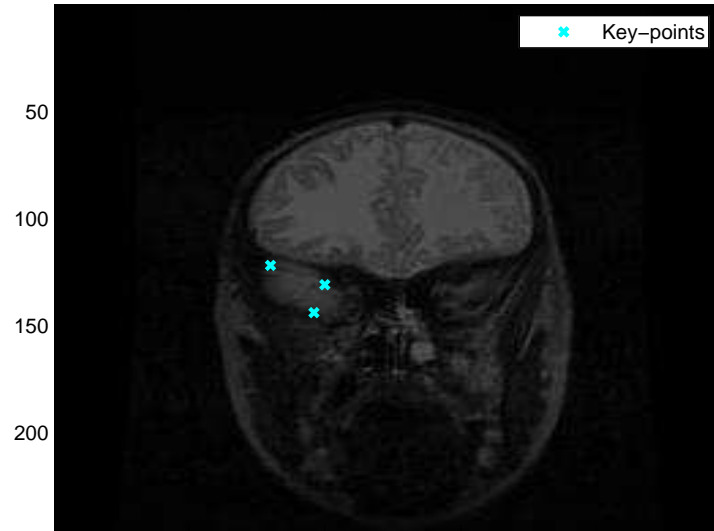


(a)



(b)

Figure 12: Minimal paths on the original image.



Three selected keypoints on the original image; Fast marching applied to the standard gradient, and topological gradient respectively.