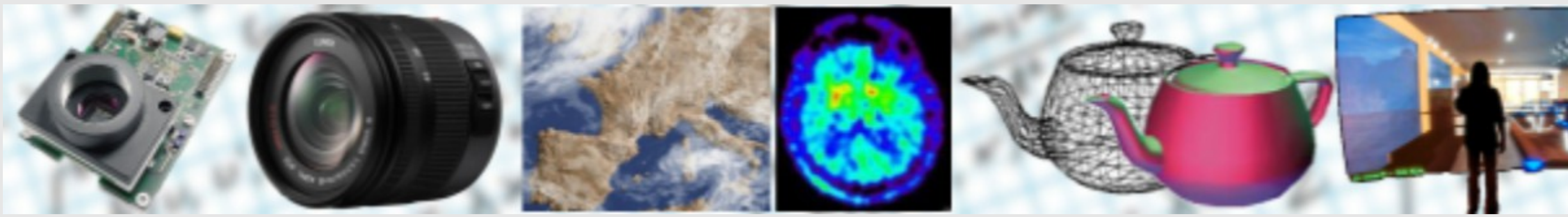


# TI – Traitement d'Images
## Semaine 9 : Détection de contours (1)
*Olivier Losson*

**Master ASE : http://master-ase.univ-lille1.fr**
**Master Informatique : http://www.fil.univ-lille1.fr**
**Spécialité IVI : http://master-ivi.univ-lille1.fr**

# Pourquoi étudier les contours ?

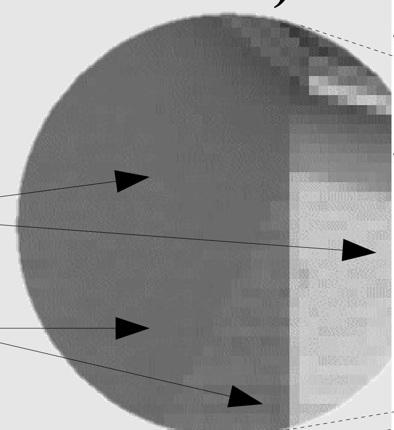
- **Rôle en traitement d'images**
  - **Réduction d'information**
    - Information de toute l'image résumée dans le contours des objets
    - Contours : parties les plus informatives d'une image
  - **Étape souvent nécessaire à l'extraction d'autres primitives**
    - notamment géométriques : droites, segments, cercles
  - **Importance des contours dans les systèmes de vision naturels**
    - Les caméras essaient généralement de reproduire la vision naturelle, notamment humaine.
    - Des études biologiques ont montré que les premières couches de la vision des mammifères réalisent des traitements de détection de contours.
- **Applications en traitement d'images**
  - **Reconnaissance d'objets, de formes, classification de scènes**
  - **Mise en correspondance stéréoscopique, reconstruction 3D**
  - **Compression d'images, ...**

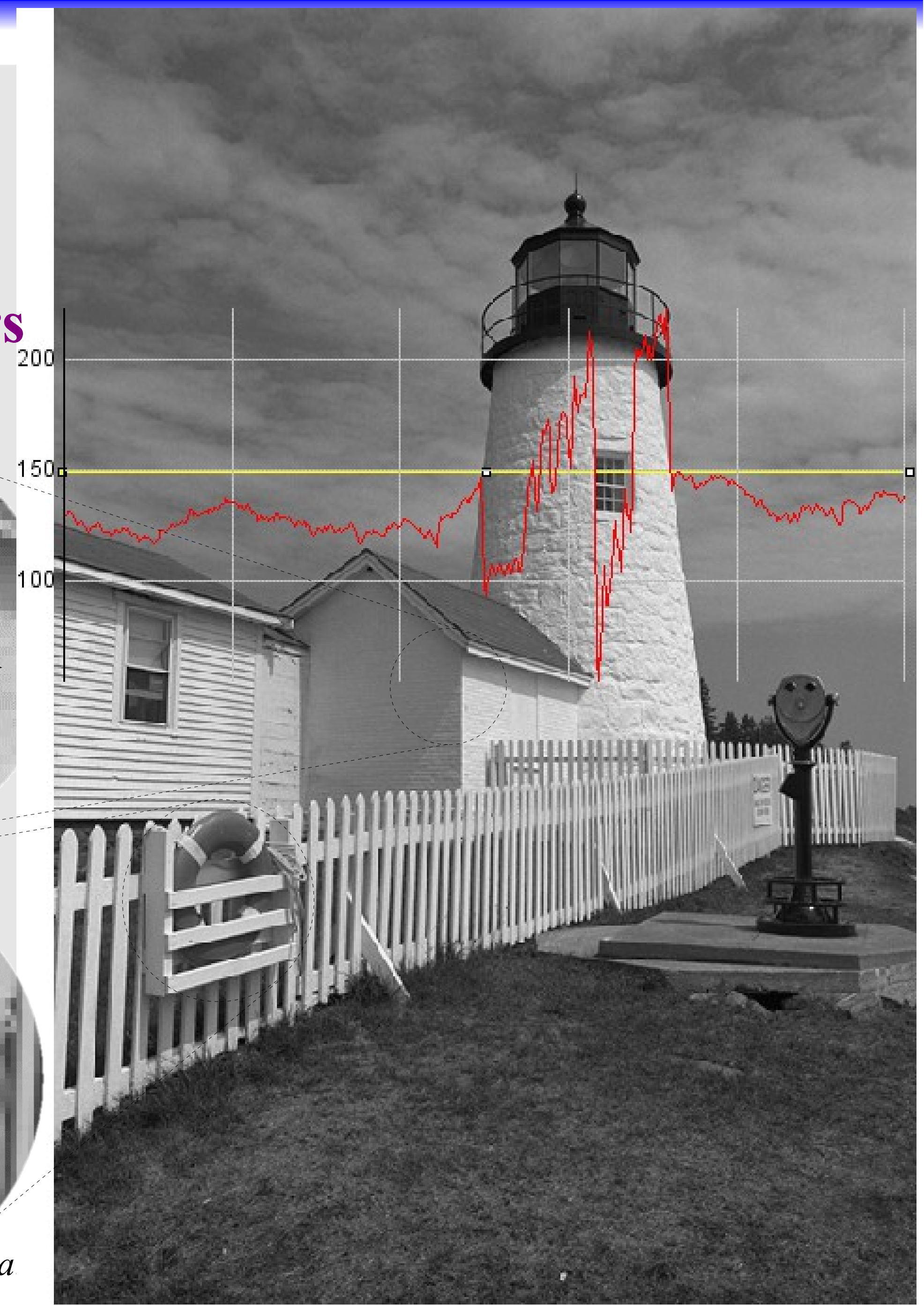
# Plan du cours

- **1 – Des dérivées aux points contours**
  - **Notion de contour**
  - **Caractérisation des points contours**
  - **Rappels sur le gradient (dérivées premières)**
  - **Utilisation des dérivées secondes**
- **2 – Approches par dérivées premières**
  - **Recherche des maxima locaux**
  - **Seuillage local par hystérésis**
- **Sélection de références**

# Notion de contour (1/2)

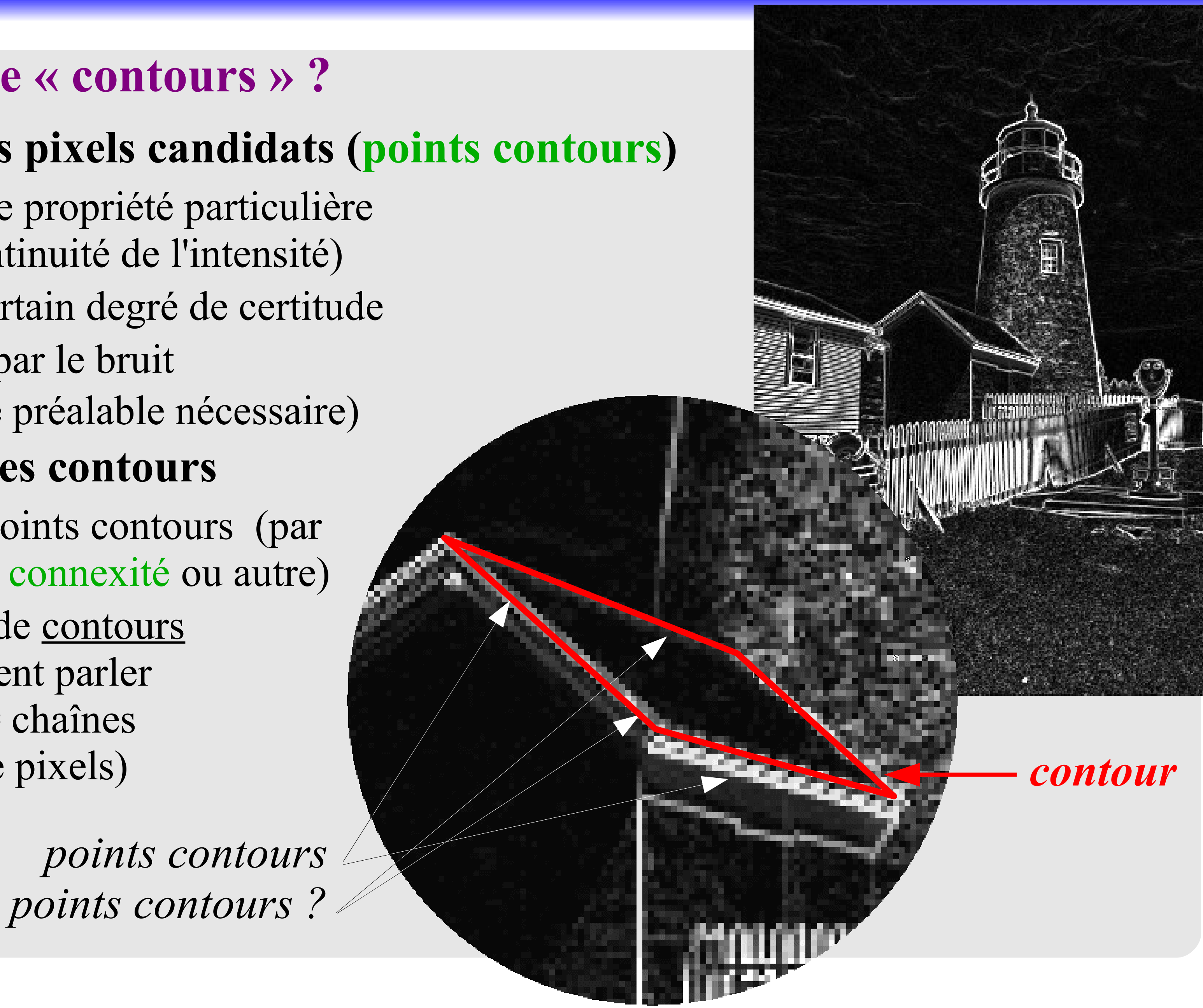
- **Définition**
  - **Frontière qui sépare 2 objets (ou 1 objet du fond) dans une image.**
- **Caractérisation des zones de contours**
  - **Variation brusque (discontinuité) de l'intensité .**
  - **Ex. de discontinuités**
    - d'orientation de surface et d'illumination
    - de réflectance
  - *Remarque* : toute zone de discontinuité ne caractérise pas forcément un contour.





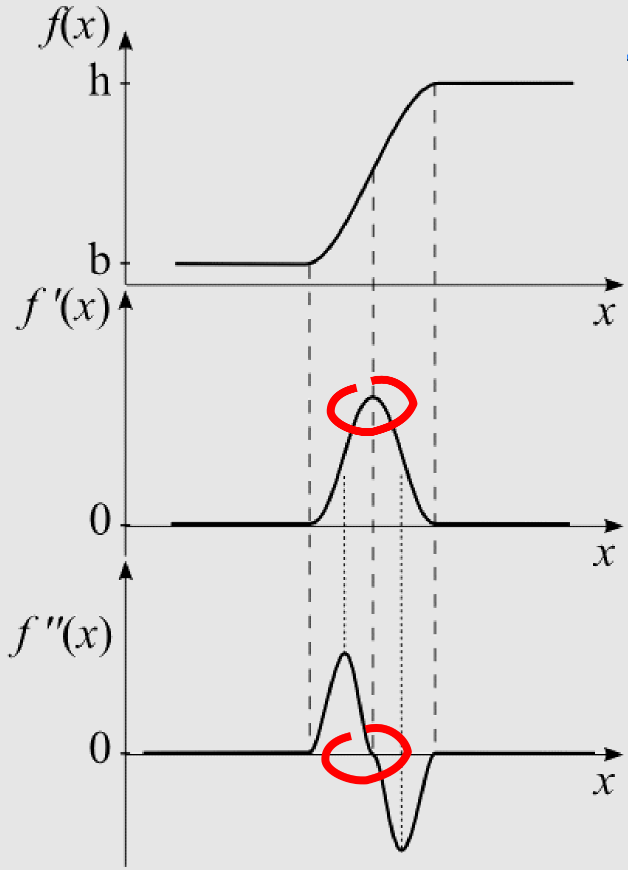
# Notion de contour (2/2)

- « Détection » de « contours » ?
  - **Détection des pixels candidats (points contours)**
    - grâce à une propriété particulière (*ex*. discontinuité de l'intensité)
    - avec un certain degré de certitude
    - perturbée par le bruit (⇒ lissage préalable nécessaire)
  - **Formation des contours**
    - relier les points contours (par analyse de connexité ou autre)
    - obtention de <u>contours</u> à proprement parler (courbes = chaînes fermées de pixels)

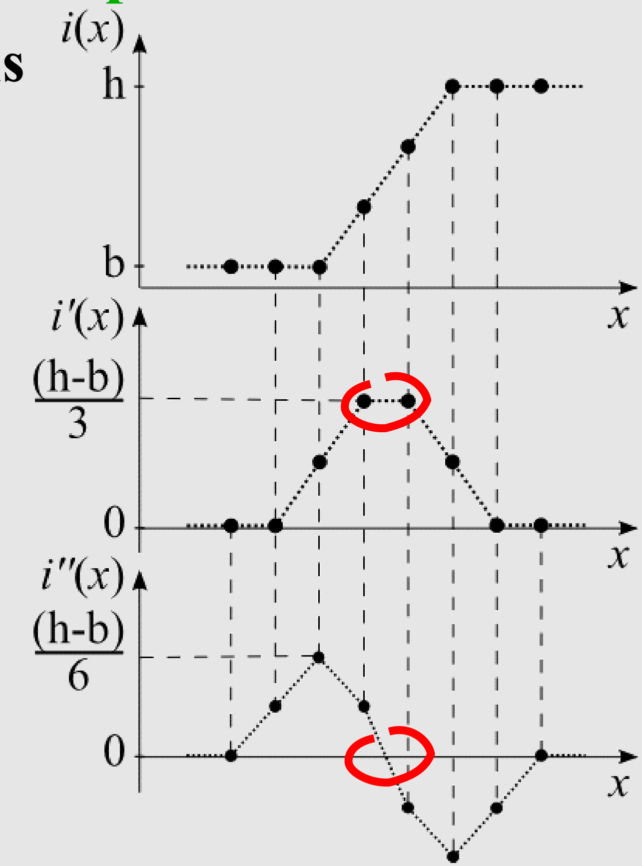


# Caractérisation des points contours

- **Mise en évidence des zones de contours : dérivées première et seconde**
  - **Fonctions continues**




  - **Fonctions discrètes**




- **Détection des points contours : utilisation d'un critère de décision**
  - **Dérivée première : maxima locaux**
  - **Dérivée seconde : passages par zéro**

# Rappels sur le gradient (1/2)

## Dérivées premières en 2D et vecteur gradient

- **On calcule une dérivée (partielle) de la fonction image *f* dans chaque direction principale.**
- **Le vecteur gradient est alors :**

$$\vec{\nabla} f(x,y) = \begin{pmatrix} \frac{\partial f}{\partial x}(x,y) \\ \frac{\partial f}{\partial y}(x,y) \end{pmatrix}$$

- **En chaque point (*x*,*y*), le vecteur gradient est caractérisé par :**
  - sa **norme** (ou module)

$$\|\vec{\nabla} f(x,y)\| = \sqrt{\left(\frac{\partial f}{\partial x}(x,y)\right)^2 + \left(\frac{\partial f}{\partial y}(x,y)\right)^2}$$

  - sa **direction**

$$\theta(x,y) = \arctan\left(\frac{\partial f}{\partial y}(x,y) / \frac{\partial f}{\partial x}(x,y)\right)$$

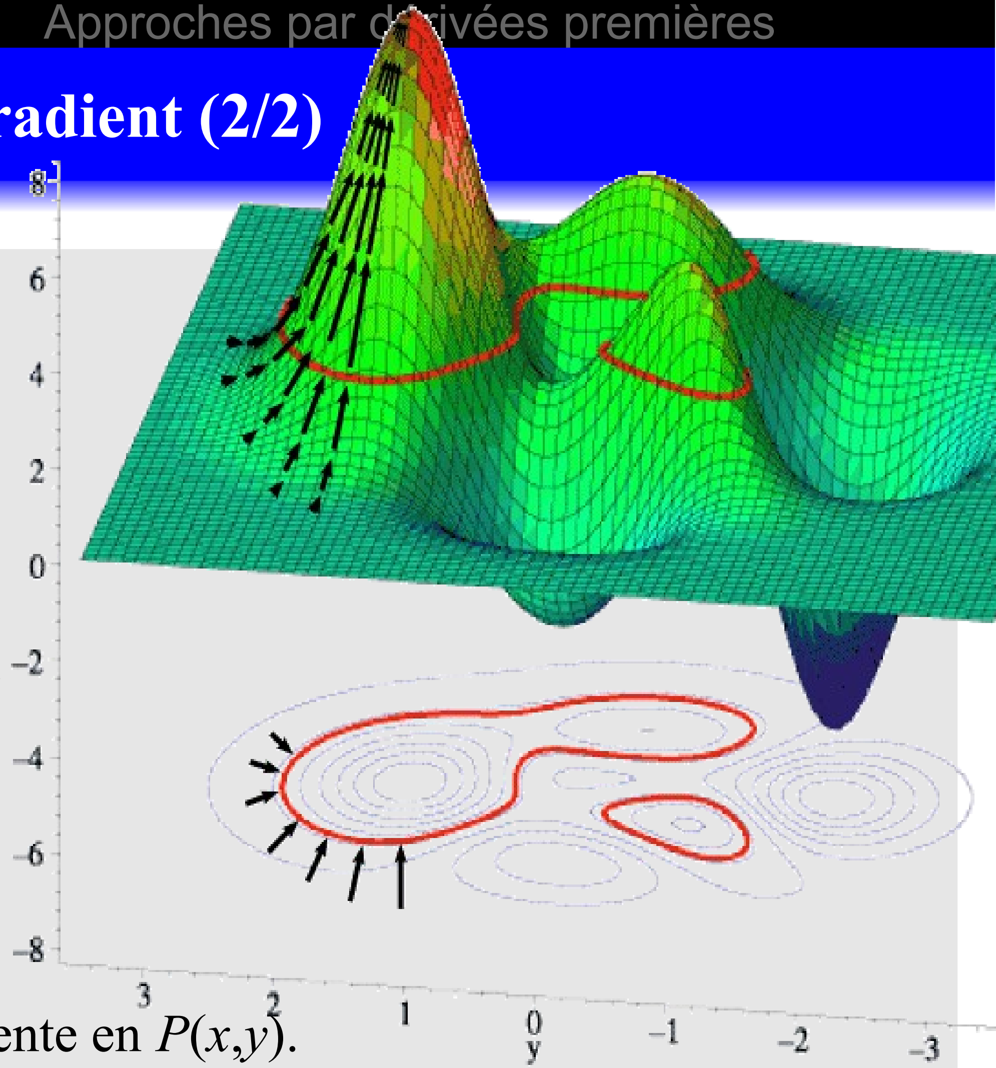
- **La dérivée première directionnelle selon α**

$$f'_\alpha := \frac{df}{d\alpha} = \frac{\partial f}{\partial x}\cos\alpha + \frac{\partial f}{\partial y}\sin\alpha = \vec{\nabla} f . \vec{\alpha}, \text{ avec } \vec{\alpha} \text{ vecteur unitaire dans dir. } \alpha,$$

**est maximale lorsque $\vec{\nabla} f$ et $\vec{\alpha}$ sont colinéaires. Le vecteur gradient est donc dans la direction de plus grande pente de *f* :** $\theta = \max_i \frac{df}{d\alpha_i}$

# Rappels sur le gradient (2/2)

- **Relation entre gradient et contour**
  - **Analogie avec un relief**
    - Courbe de niveau : relie les points d'égale altitude
    - Le **gradient d'altitude** est
      - orthogonal aux courbes de niveau
      - orienté dans le sens de la montée




  - **Propriétés du vecteur gradient**
    - $\|\vec{\nabla} f(x,y)\| \equiv$ pente de la surface image en $P(x,y)$.
    - $\theta(x,y) \equiv$ direction de la plus grande pente en $P(x,y)$.
    - orienté dans le sens de la montée (*i.e.* niveaux de gris croissants).
  - **Relation entre gradient et contour**
    - Contour $\equiv$ forte variation locale des niveaux de gris $\equiv \|\vec{\nabla} f(x,y)\|$ élevée.
    - $\vec{\nabla} f(x,y)$ est **perpendiculaire au contour** en $P(x,y)$.

# Détection des points contours : résumé (1/2)

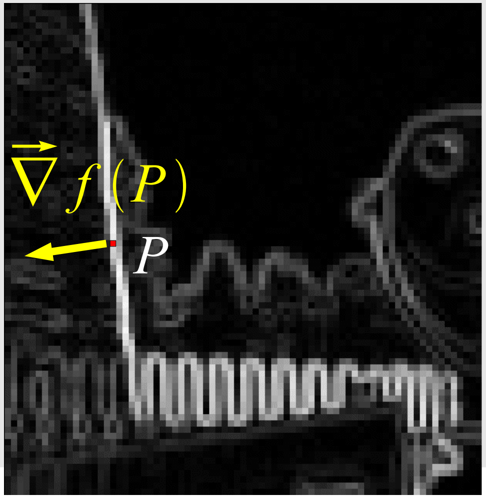
- **Résumé.**
  - **La détection des points contours est basée sur les dérivées premières (gradient) ou secondes de la fonction sous-jacente à l'image.**
  - **Le calcul de ces dérivées est approché au moyen de filtres de convolution.**
    - Avantages : grande rapidité de calcul, aspect local.
    - Inconvénients : ces filtres sont très sensibles au bruit. Ils nécessitent donc l'emploi de filtres de lissage débruiteurs, souvent intégrés aux filtres de dérivation.
  - **Les filtres de lissage/dérivation sont moins précis que le filtre de dérivation « pur », mais plus robustes. Ils privilégient donc la détection des points contours par rapport à leur localisation.**

# Détection des points contours : résumé (2/2)

- **Vers la détection des contours.**
  - **Ces filtres permettent seulement d'estimer la « probabilité » qu'un pixel soit un point contour candidat. Il reste donc à :**
    - décider si un pixel est *effectivement* un point contour, par exemple au moyen d'un seuillage :
      - si $\|\vec{\nabla} f(x,y)\| > S$, le pixel $P(x,y)$ est un point contour candidat ;
      - si $\|\vec{\nabla} f(x,y)\| \leqslant S$, le pixel $P(x,y)$ n'est pas point contour candidat.
    - utiliser les points contours pour former les contours proprement dits. Cela nécessite des étapes supplémentaires, car les contours formés par ces points sont :
      - épais
      - « bruités »
      - interrompus (non fermés)

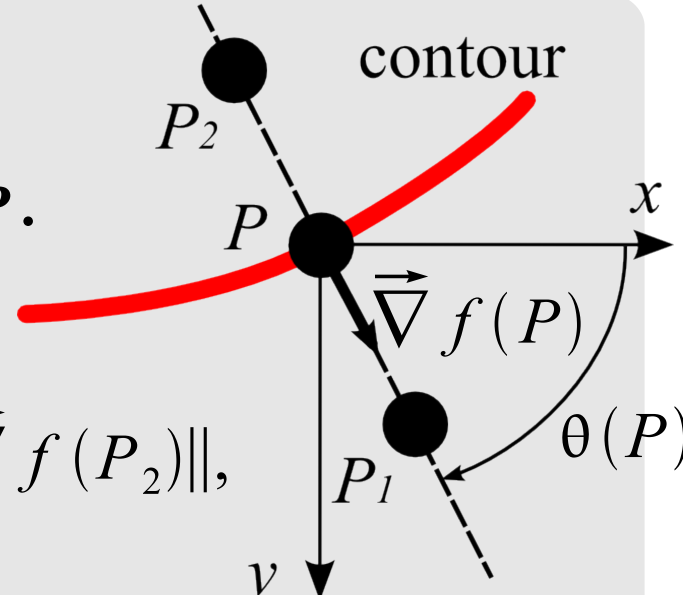
# Recherche des maxima locaux (1/3) : principe

- **Utilisation d'un seuil sur la norme du gradient.**
  - **Problème avec un seuil de décision $S$ unique pour toute l'image : on risque**
    - de ne pas détecter les points contours en zones de faible contraste ;
    - de sélectionner à tort des points contours dans les zones bruitées.
  - **Solution : on cherche les maxima locaux de la norme du gradient.**
- **Maxima locaux.**
  - **Principe : on cherche les points $P$ auxquels la norme du gradient est maximale dans la direction locale du gradient.**
  - **Calcul pratique : *cf.* diapos suivantes.**
- **Suppression des non-maxima locaux.**
  - **Mettre à 0 la norme du gradient pour les pixels non maxima locaux.**
  - **Permet d'obtenir ensuite des contours d'épaisseur 1 pixel.**

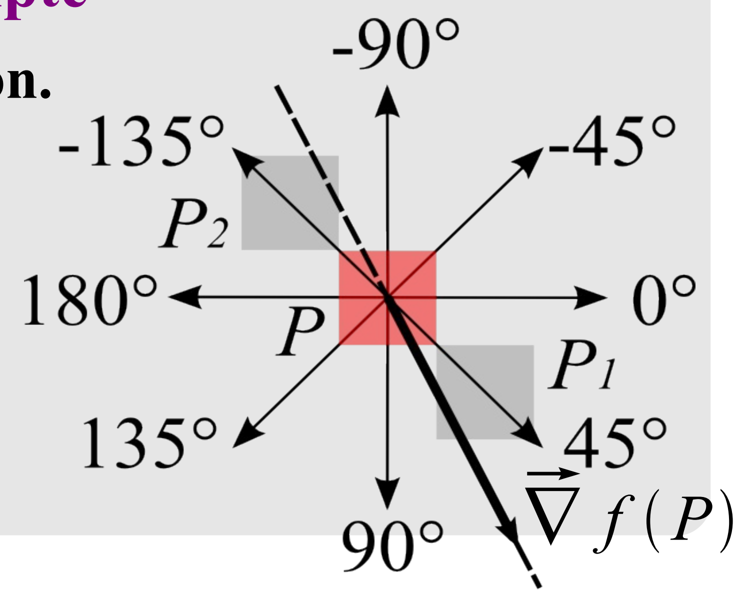



## Recherche des maxima locaux (2/3) : calcul

- **Principe détaillé :**
  - **soient $P_1$ et $P_2$ les pixels situés de part et d'autre de $P$ dans la direction $\theta$ du gradient en $P$.**
  - **on compare la norme du gradient en $P$ avec celles du gradient en $P_1$ et en $P_2$.**
  - Si $\|\vec{\nabla} f(P)\| \geqslant \|\vec{\nabla} f(P_1)\|$ et $\|\vec{\nabla} f(P)\| > \|\vec{\nabla} f(P_2)\|$, **alors $P$ est un maximum local.**




- **Détermination des 2 voisins à prendre en compte**
  - **Détermination par discrétisation de la direction.**
    - On arrondit la direction du gradient au multiple de 45° le plus proche.
    - Le point $P_1$ est le pixel voisin de $P$ situé dans cette direction, et $P_2$ est le pixel opposé par rapport à $P$.



# Recherche des maxima locaux (3/3) : calcul

- **Détermination des 2 voisins à prendre en compte (*suite*)**
  - **Calcul par interpolation.**
    - Soient $P_1$ et $P_2$ les points (cette fois virtuels) situés à distance unité de $P$ dans la direction du gradient en $P$.
    - On détermine une approximation de la norme du gradient en $P_1$ et $P_2$ par interpolation de la norme calculée en leurs 2 pixels voisins.

Soit $g_x := \frac{\partial f}{\partial x}$ et $g_y := \frac{\partial f}{\partial y}$, alors $k \approx \frac{g_x}{g_y}$

et si, par *ex.* $45° < \theta < 90°$, alors :

$$G(P_1) = \left(1 - \frac{g_x}{g_y}\right) G(x, y+1) + \frac{g_x}{g_y} G(x+1, y+1)$$

$$G(P_2) = \left(1 - \frac{g_x}{g_y}\right) G(x, y-1) + \frac{g_x}{g_y} G(x-1, y-1)$$

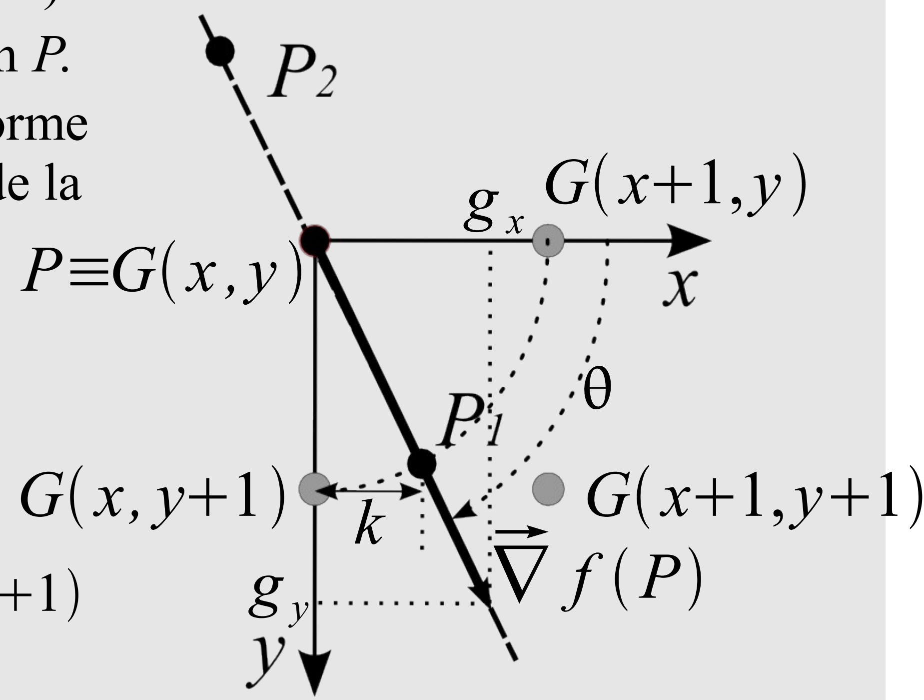
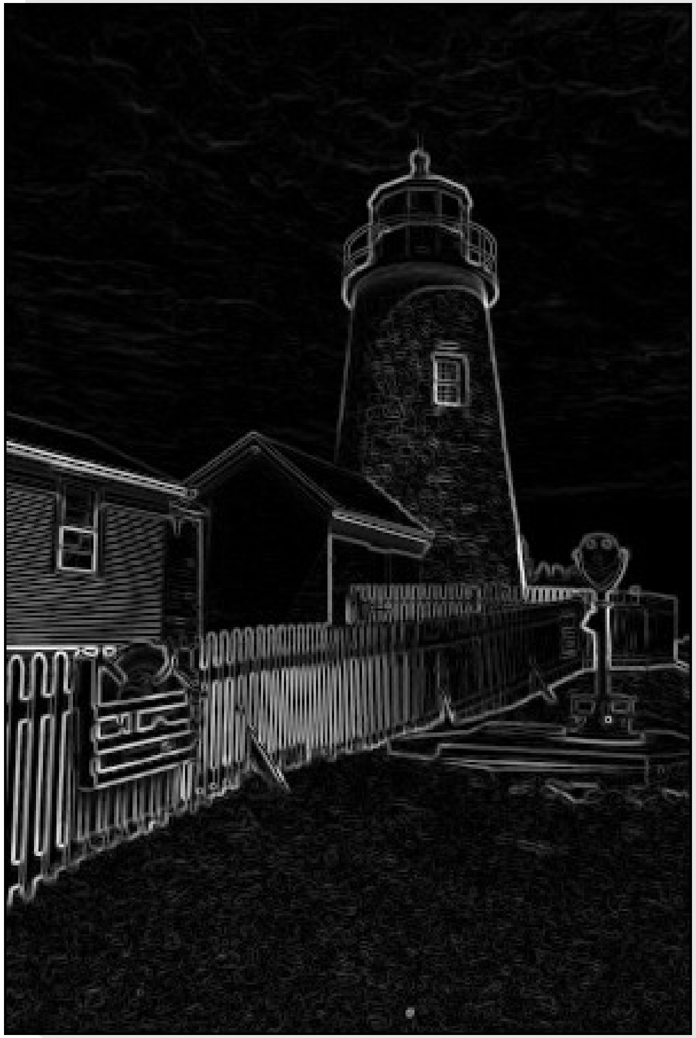
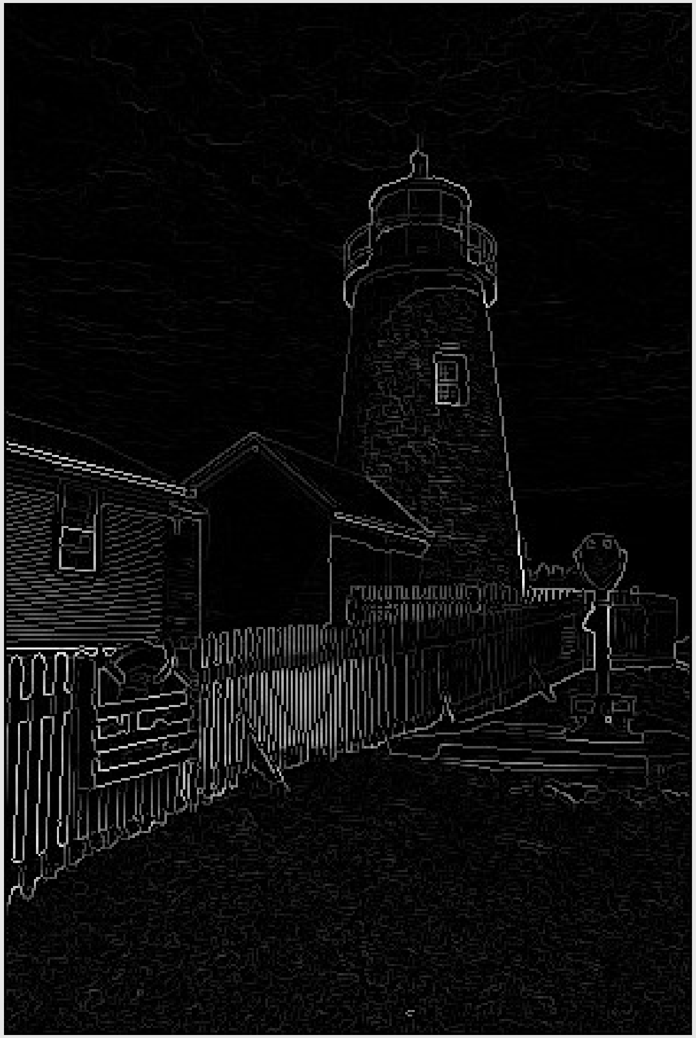
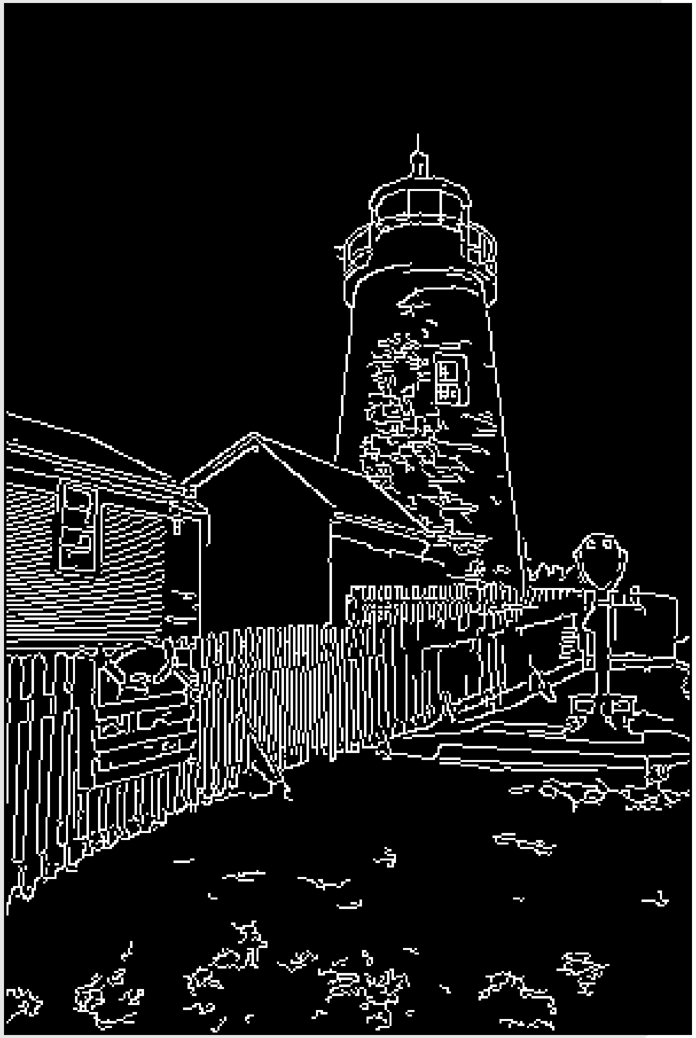



*Image G de la norme du gradient*

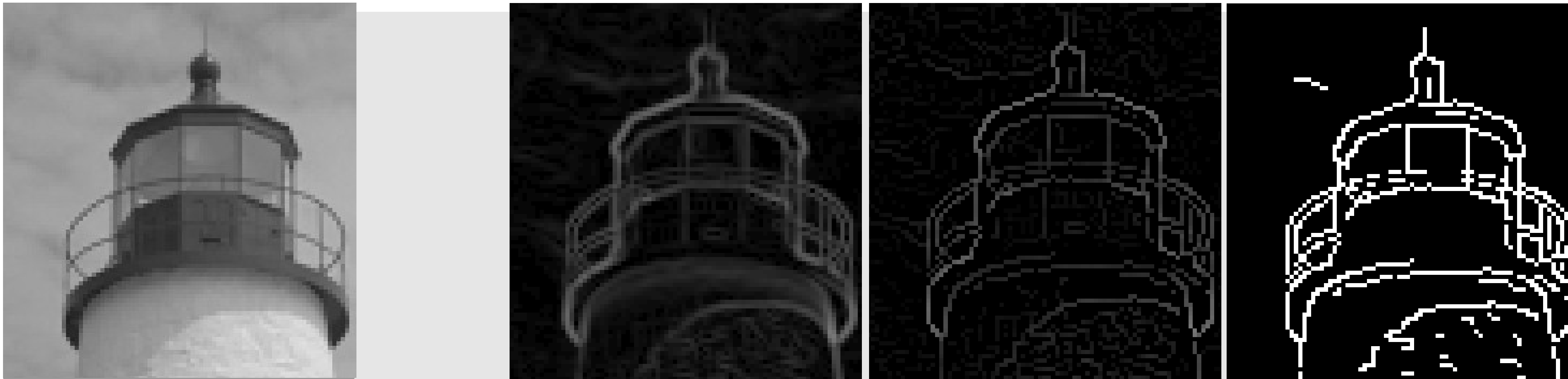
# Seuillage local par hystérésis (1/2)

- **But et principe**
  - **Obtenir une image binaire des pixels contours (0=non contour, 1=contour)**
  - **On réalise pour cela un seuillage local basé sur l'hystérésis (« mémoire »)**
- **Algorithme**
  - **Paramètres :**
    - Image $G_M$ des maxima locaux de la norme du gradient (codée sur 8 bits)
    - Un seuil bas ($S_b$) et un seuil haut ($S_h$), tous deux dans [0, 255]
  - **Résultat : image binaire $C$ de même taille que $G_M$**
  - **Principe : à partir des pixels pour lesquels $G_M(x,y) > S_h$ , on « propage » ces contours par connexité tant que $G_M(x,y) > S_b$**
  - **Détail : pour chaque pixel ($x$,$y$),**
    - Si $G_M(x,y) < S_b$, $C(x,y) = 0$ (le pixel n'est pas contour)
    - Si $G_M(x,y) > S_h$, $C(x,y) = 1$ (le pixel est un contour)
    - Si $S_b \leqslant G_M(x,y) \leqslant S_h$, $C(x,y) = 1$ s'il est connecté à un autre pixel déjà contour.

# Seuillage local par hystérésis (2/2)

| **Norme du gradient ‖G‖** (*Sobel*) normalisée [0..255] | **Maxima locaux de ‖G‖** normalisée [0..255] | **Contours après seuillage** seuil bas=17, seuil haut=46 |
|---|---|---|
|  |  |  |

# Approche gradient : résumé



$f(x,y)$ — (Lissage+) Dérivation (Sobel, ...)

Image en niveaux de gris

$$\frac{\partial f}{\partial x} \qquad \frac{\partial f}{\partial y}$$

Composantes

$$\|\vec{\nabla} f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

$$\theta = \arctan\left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}}\right)$$

Norme et direction

Vecteur gradient $\vec{\nabla} f$

$\xrightarrow{SNM}$ Maxima locaux de $\|\vec{\nabla} f\|$ $\xrightarrow{Seuillage}$ Points contours

*SNM* : Suppression des Non-Maxima locaux

# Sélection de références

- **Livre**
  - **W. Burger, M.J. Burge, *Digital Image Processing – An Algorithmic Introduction Using Java*, Springer 2008.**
- **Sites web**
  - **Cours de J.-H. Thomas (Université du Maine)**

    http://www.optique-ingenieur.org/fr/cours/OPI_fr_M04_C05/co/OPI_fr_M04_C05_web_1.html
  - **Cours d'A. Boucher (Institut de la Francophonie pour l'Informatique)**

    http://www2.ifi.auf.org/personnel/Alain.Boucher/cours/traitement_images/05-Contours.pdf
  - **Sharpening and Edge Detection (R. Wang)**

    http://fourier.eng.hmc.edu/e161/lectures/gradient/gradient.html
  - **Plugin ImageJ pour la détection des points contours (C. Pulvirenti)**

    http://svg.dmi.unict.it/iplab/imagej/Plugins/Edge%20Detectors/Canny/EdgeDetection.htm

# Rappels sur le gradient (2/3)

- **Dérivée première en 1D**
  - **Approximation discrète de la dérivée d'une fonction 1D :**

$$f'(x)=\frac{df}{dx}(x)\approx\frac{f(x+1)-f(x-1)}{2}$$

  - **Masque de convolution 1D correspondant :** $\frac{1}{2}$

| | | |
|---|---|---|
| +1 | 0 | -1 |

- **Dérivées premières en 2D et vecteur gradient**
  - **On calcule une dérivée (partielle) de *f* dans chaque direction principale**
  - **Le vecteur gradient est alors :**

$$\vec{\nabla} f(x,y)=\begin{pmatrix}\frac{\partial f}{\partial x}(x,y)\\ \frac{\partial f}{\partial y}(x,y)\end{pmatrix}$$

  - **En chaque point (*x*,*y*), le vecteur gradient est caractérisé par :**
    - sa **norme** (ou module)

$$\|\vec{\nabla} f(x,y)\|=\sqrt{\left(\frac{\partial f}{\partial x}(x,y)\right)^2+\left(\frac{\partial f}{\partial y}(x,y)\right)^2}$$

    - sa **direction**

$$\theta(x,y)=\arctan\left(\frac{\partial f}{\partial y}(x,y)/\frac{\partial f}{\partial x}(x,y)\right)$$