

# Des maths pour restaurer les images

Cécile Louchet

IDP, Université d'Orléans

Centre Galois 2019

- 1 Formation des images
- 2 Couleur
- 3 Occlusion - contours - gradient
- 4 Côté variationnel
- 5 Côté patches

- 1 Formation des images
- 2 Couleur
- 3 Occlusion - contours - gradient
- 4 Côté variationnel
- 5 Côté patches

# Appareil photo numérique

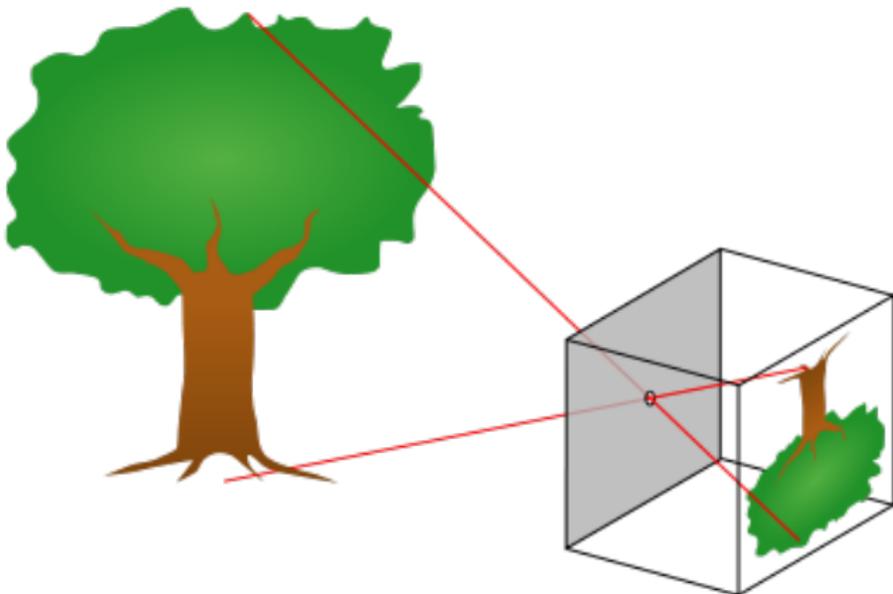


# Sténopé (*pinhole camera*)



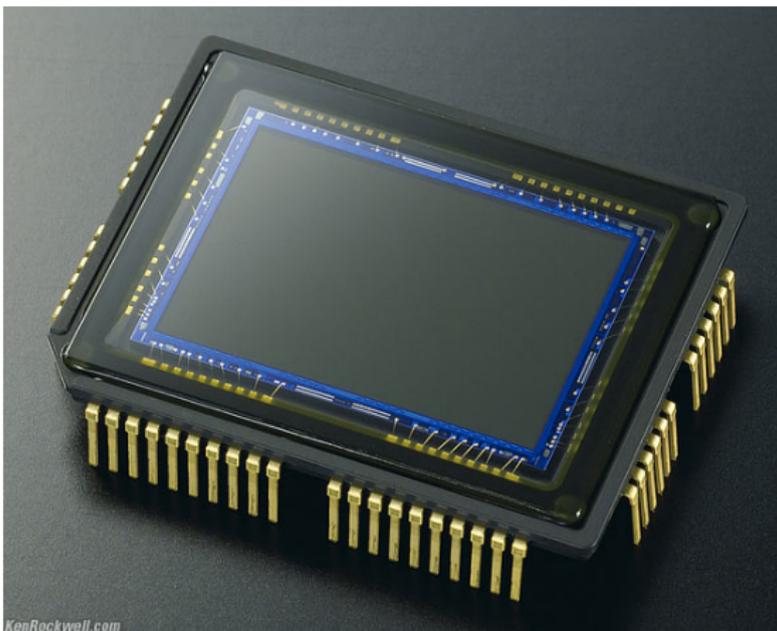
Sources : <http://www.diyphotography.net/23-pinhole-cameras-that-you-can-build-at-home/>  
[http://en.wikipedia.org/wiki/Pinhole\\_camera](http://en.wikipedia.org/wiki/Pinhole_camera)

# Sténopé (*pinhole camera*)



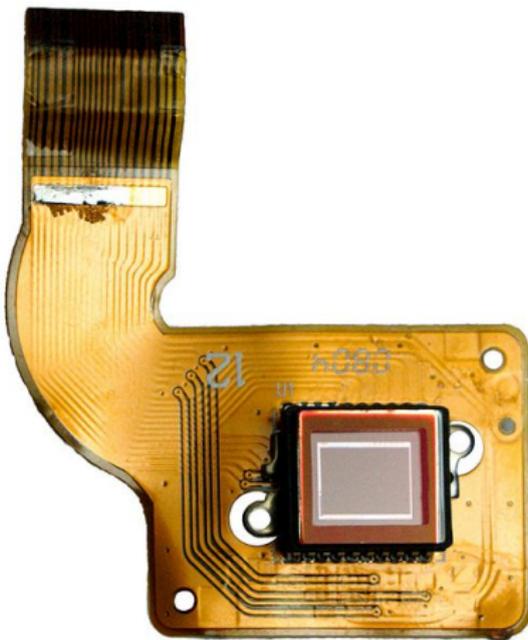
Sources : <http://www.diyphotography.net/23-pinhole-cameras-that-you-can-build-at-home/>  
[http://en.wikipedia.org/wiki/Pinhole\\_camera](http://en.wikipedia.org/wiki/Pinhole_camera)

# Capteur CCD



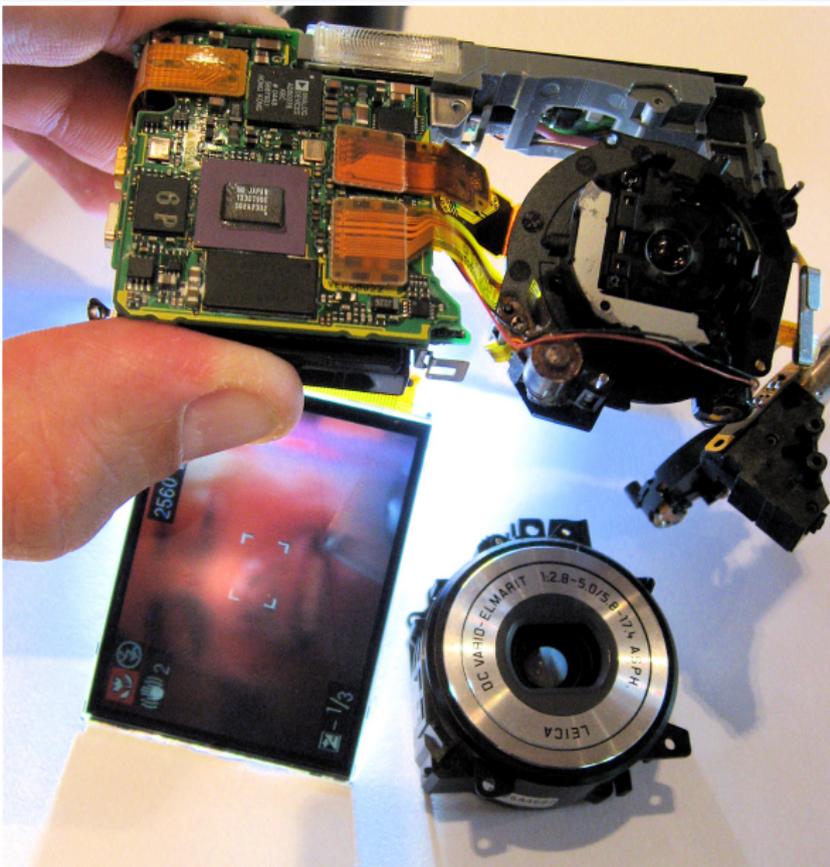
Nikon D60. Sensor: 10 MP CCD (same sensor as D80/D200).  
ISO: 100 - 1,600 in full stops, also a trick ISO 3,200 mode.  
Format:  $23.6 \times 15.8$  mm. Source : <http://www.kenrockwell.com/nikon/d60.htm>

# Capteur CCD



Source : <http://giorgioscappaticcio.wordpress.com/2010/09/>

# On démonte un autre appareil photo. . .



# Matrice des niveaux de gris

	Column															
	223	208	194	182	172	164	159	158	159	164	172	182	194	208	223	240
	208	191	176	163	151	143	137	135	137	143	151	163	176	191	208	225
	194	176	159	144	131	121	115	113	115	121	131	144	159	176	194	213
	182	163	144	128	113	101	93	90	93	101	113	128	144	163	182	202
	172	151	131	113	96	81	71	68	71	81	96	113	131	151	172	193
	164	143	121	101	81	64	50	45	50	64	81	101	121	143	164	186
	159	137	115	93	71	50	32	23	32	50	71	93	115	137	159	182
	158	135	113	90	68	45	23	0	23	45	68	90	113	135	158	180
Row	159	137	115	93	71	50	32	23	32	50	71	93	115	137	159	182
	164	143	121	101	81	64	50	45	50	64	81	101	121	143	164	186
	172	151	131	113	96	81	71	68	71	81	96	113	131	151	172	193
	182	163	144	128	113	101	93	90	93	101	113	128	144	163	182	202
	194	176	159	144	131	121	115	113	115	121	131	144	159	176	194	213
	208	191	176	163	151	143	137	135	137	143	151	163	176	191	208	225
	223	208	194	182	172	164	159	158	159	164	172	182	194	208	223	240
	240	225	213	202	193	186	182	180	182	186	193	202	213	225	240	255

- niveau de gris entier, de 0 (noir) à 255 (blanc)



Source : <http://www.ie.wh.sdu.edu.cn>

# Matrice des niveaux de gris

	Column															
	223	208	194	182	172	164	159	158	159	164	172	182	194	208	223	240
	208	191	176	163	151	143	137	135	137	143	151	163	176	191	208	225
	194	176	159	144	131	121	115	113	115	121	131	144	159	176	194	213
	182	163	144	128	113	101	93	90	93	101	113	128	144	163	182	202
	172	151	131	113	96	81	71	68	71	81	96	113	131	151	172	193
	164	143	121	101	81	64	50	45	50	64	81	101	121	143	164	186
	159	137	115	93	71	50	32	23	32	50	71	93	115	137	159	182
	158	135	113	90	68	45	23	0	23	45	68	90	113	135	158	180
Row	159	137	115	93	71	50	32	23	32	50	71	93	115	137	159	182
	164	143	121	101	81	64	50	45	50	64	81	101	121	143	164	186
	172	151	131	113	96	81	71	68	71	81	96	113	131	151	172	193
	182	163	144	128	113	101	93	90	93	101	113	128	144	163	182	202
	194	176	159	144	131	121	115	113	115	121	131	144	159	176	194	213
	208	191	176	163	151	143	137	135	137	143	151	163	176	191	208	225
	223	208	194	182	172	164	159	158	159	164	172	182	194	208	223	240
	240	225	213	202	193	186	182	180	182	186	193	202	213	225	240	255

- niveau de gris entier, de 0 (noir) à 255 (blanc) (codé sur 8 bits)

Source : <http://www.ie.wh.sdu.edu.cn>

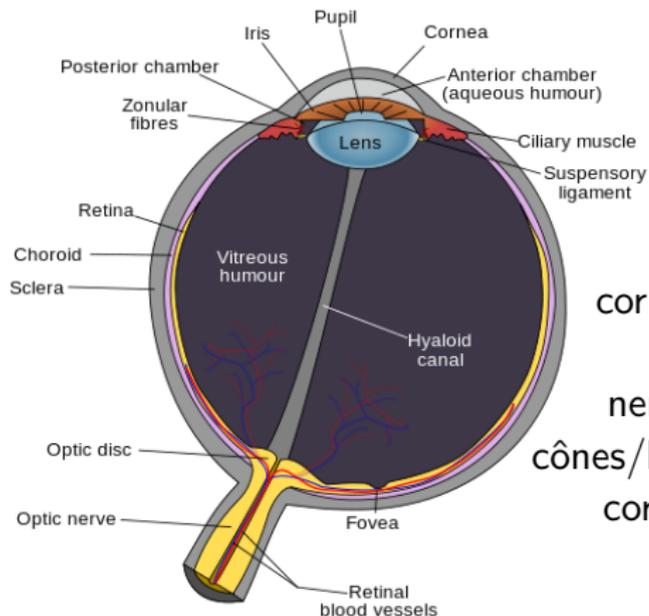
# Matrice des niveaux de gris

	Column																															
	223	208	194	182	172	164	159	158	159	164	172	182	194	208	223	240	208	191	176	163	151	143	137	135	137	143	151	163	176	191	208	225
	194	176	159	144	131	121	115	113	115	121	131	144	159	176	194	213	182	163	144	128	113	101	93	90	93	101	113	128	144	163	182	202
	172	151	131	113	96	81	71	68	71	81	96	113	131	151	172	193	164	143	121	101	81	64	50	45	50	64	81	101	121	143	164	186
	159	137	115	93	71	50	32	23	32	50	71	93	115	137	159	182	158	135	113	90	68	45	23	0	23	45	68	90	113	135	158	180
Row	159	137	115	93	71	50	32	23	32	50	71	93	115	137	159	182	164	143	121	101	81	64	50	45	50	64	81	101	121	143	164	186
	172	151	131	113	96	81	71	68	71	81	96	113	131	151	172	193	182	163	144	128	113	101	93	90	93	101	113	128	144	163	182	202
	194	176	159	144	131	121	115	113	115	121	131	144	159	176	194	213	208	191	176	163	151	143	137	135	137	143	151	163	176	191	208	225
	223	208	194	182	172	164	159	158	159	164	172	182	194	208	223	240	240	225	213	202	193	186	182	180	182	186	193	202	213	225	240	255

- niveau de gris entier, de 0 (noir) à 255 (blanc) (codé sur 8 bits)
- notation matricielle :  $I(i, j)$  est le niveau de gris à la  $i$ -ème ligne et  $j$ -ème colonne.

Source : <http://www.ie.wh.sdu.edu.cn>

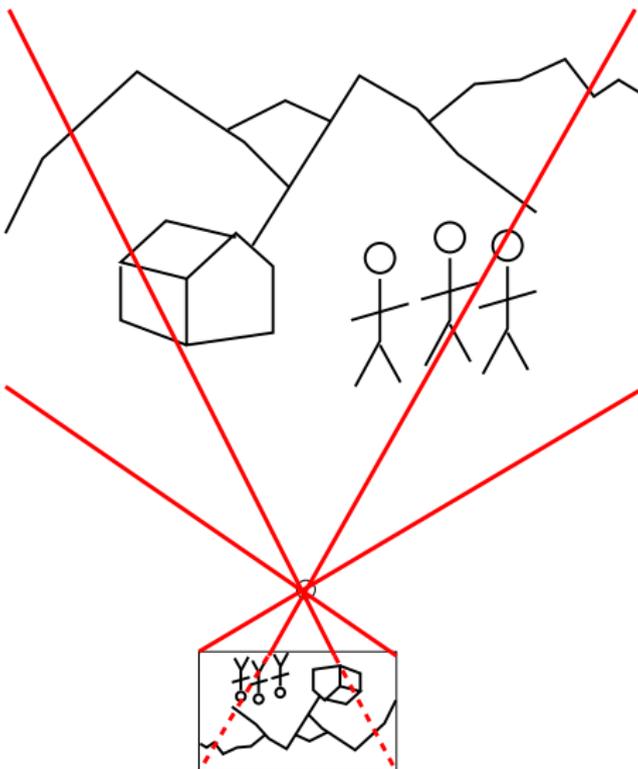
# Système visuel humain vs. appareil photo numérique



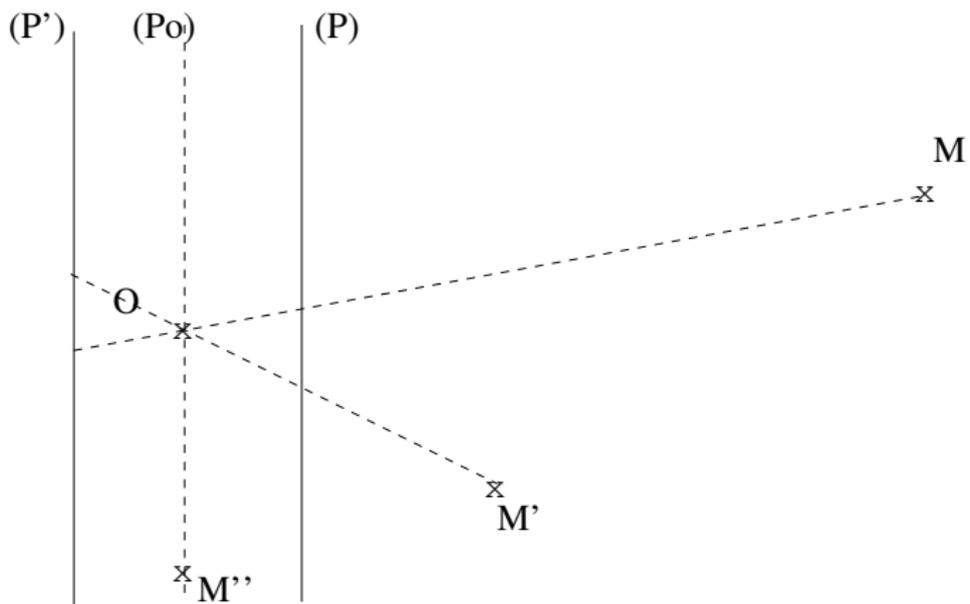
œil	appareil photo
cristallin	lentille
iris	diaphragme
rétine	grille de capteurs
corps vitreux	chambre noire
fovea	centre optique
nerf optique	canal de transmission
cônes/bâtonnets	sensibilité (ISO)
cortex visuel	circuit électronique

Source : [http://commons.wikimedia.org/wiki/File:Schematic\\_diagram\\_of\\_the\\_human\\_eye\\_en-edit.png](http://commons.wikimedia.org/wiki/File:Schematic_diagram_of_the_human_eye_en-edit.png)

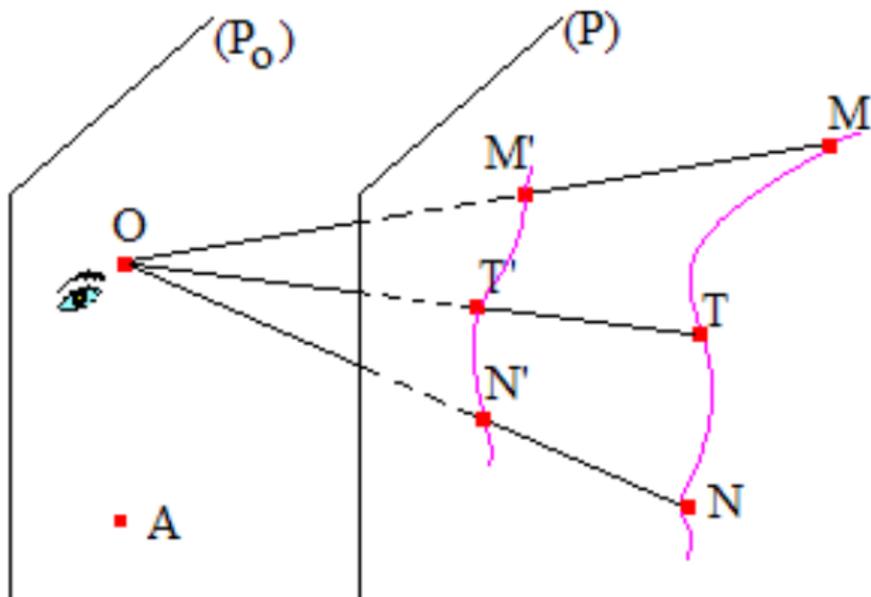
# Perspective



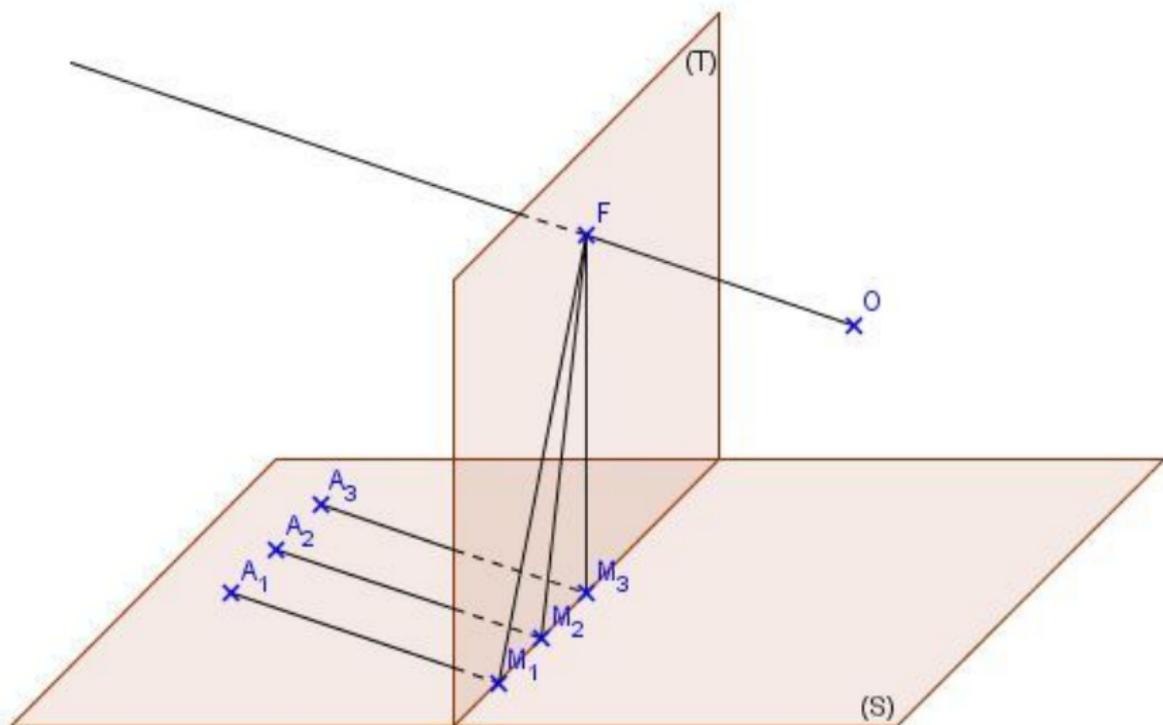
# Perspective



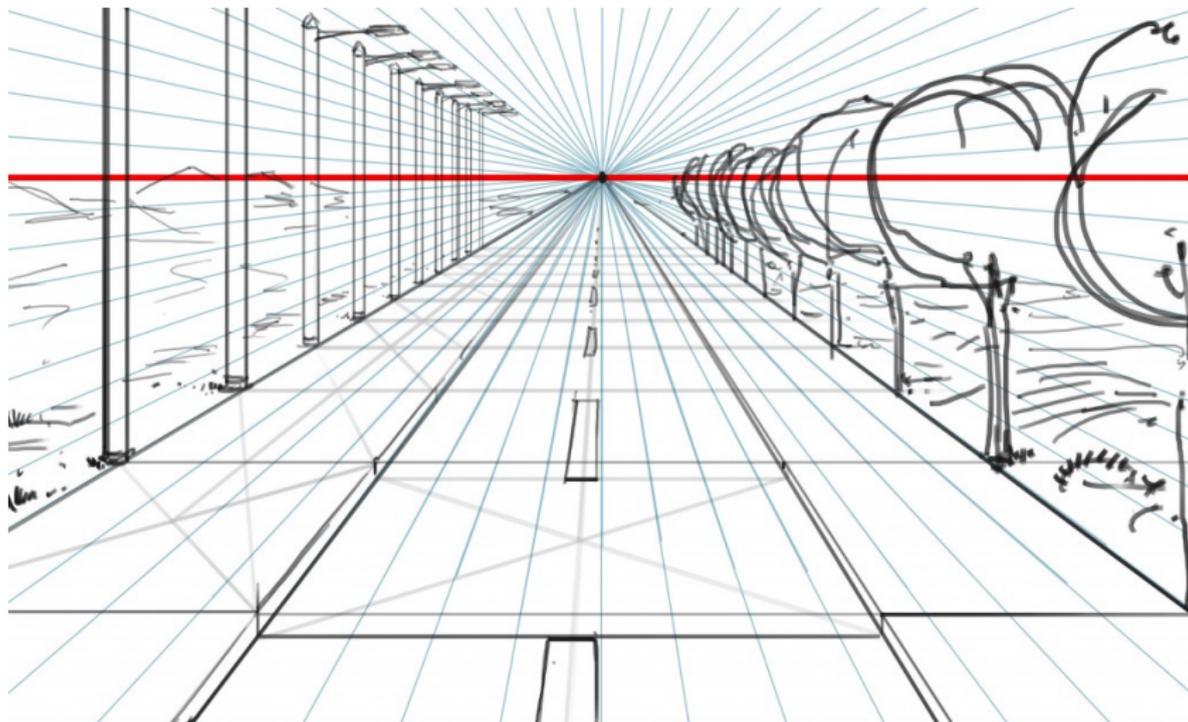
# Perspective



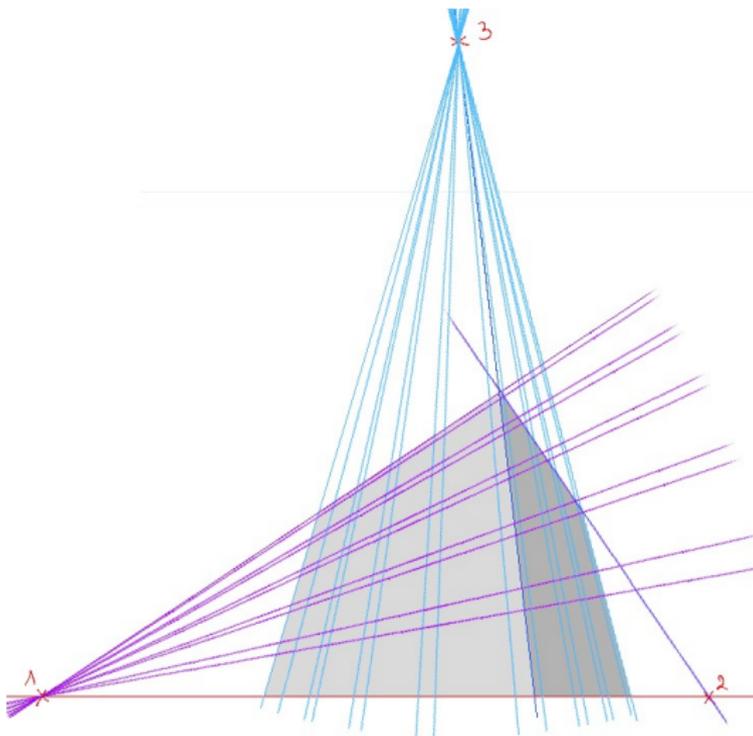
# Perspective



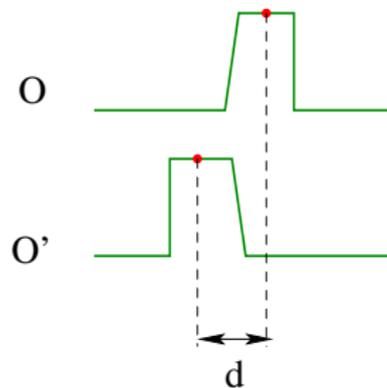
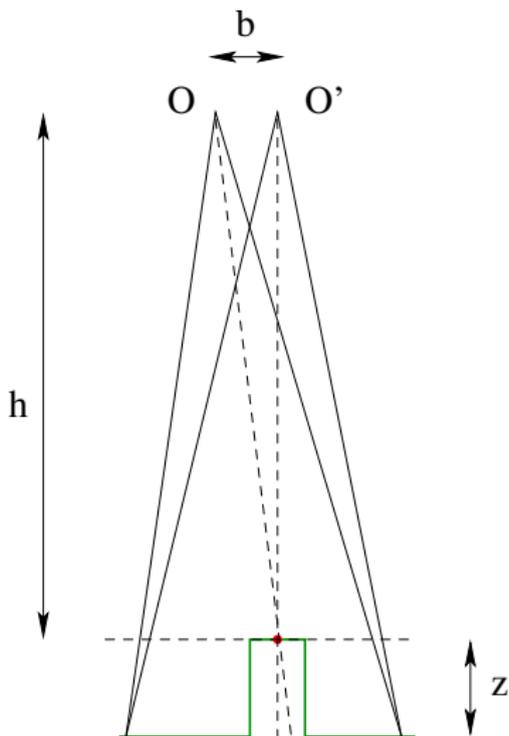
# Perspective



# Perspective



# Vision binoculaire



$z = ???$

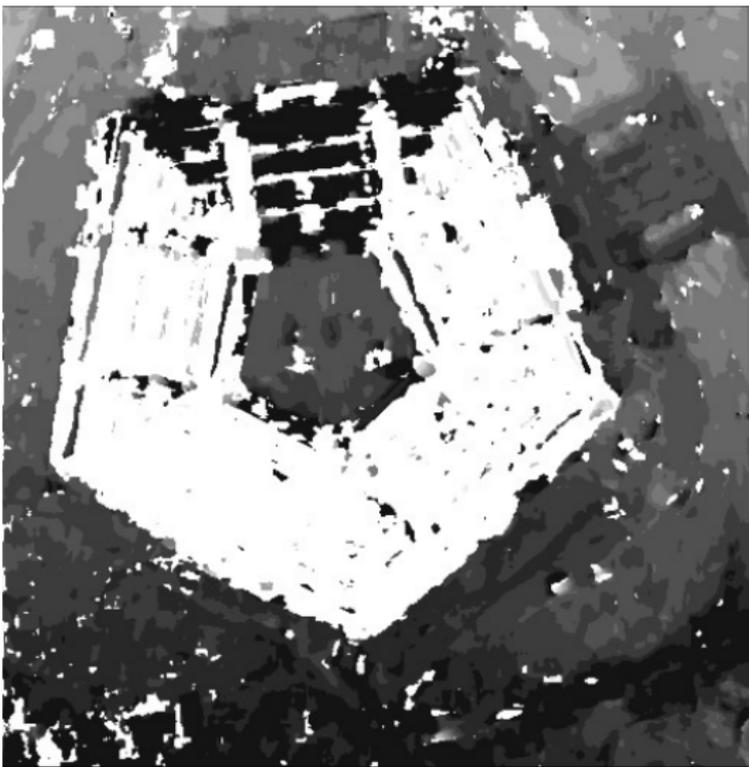
# Reconstruction 3D



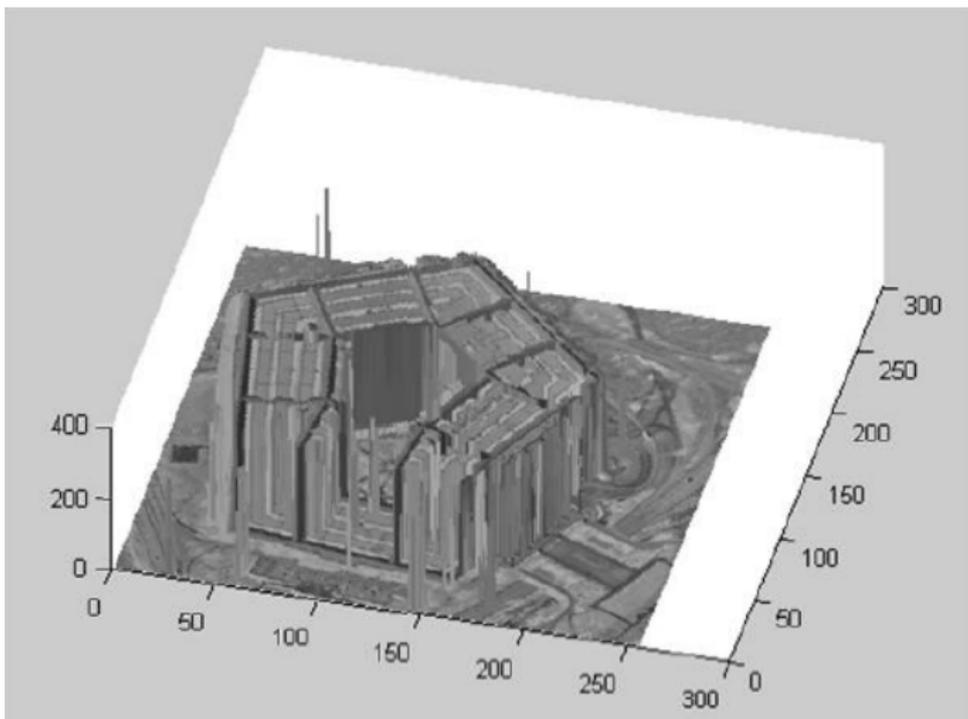
# Reconstruction 3D



# Reconstruction 3D



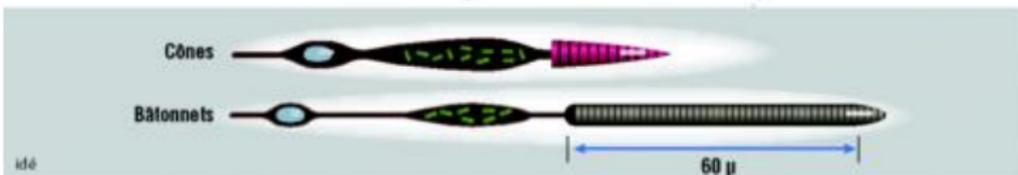
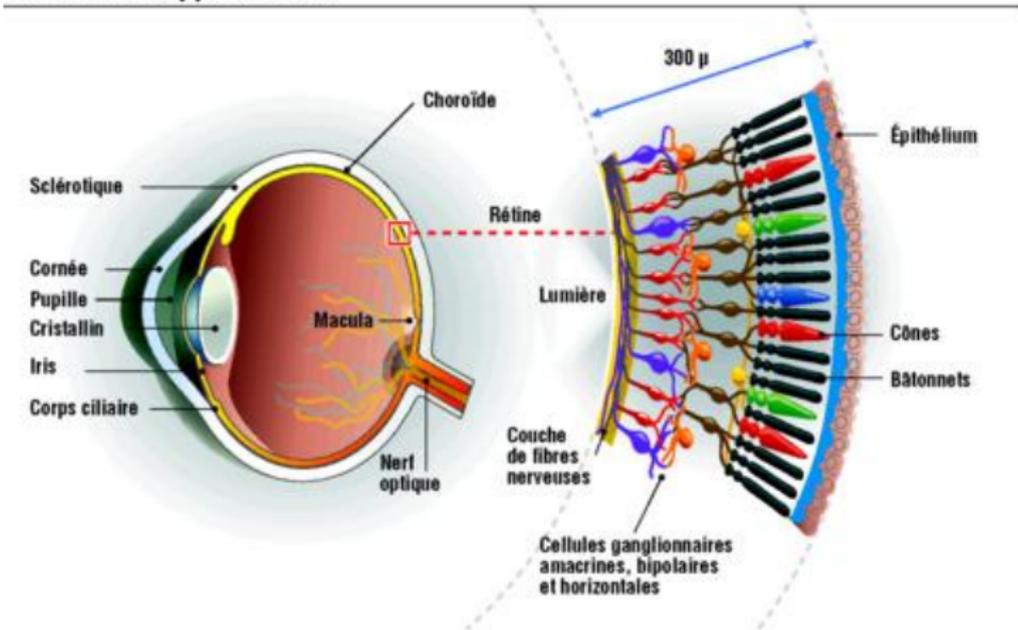
# Reconstruction 3D



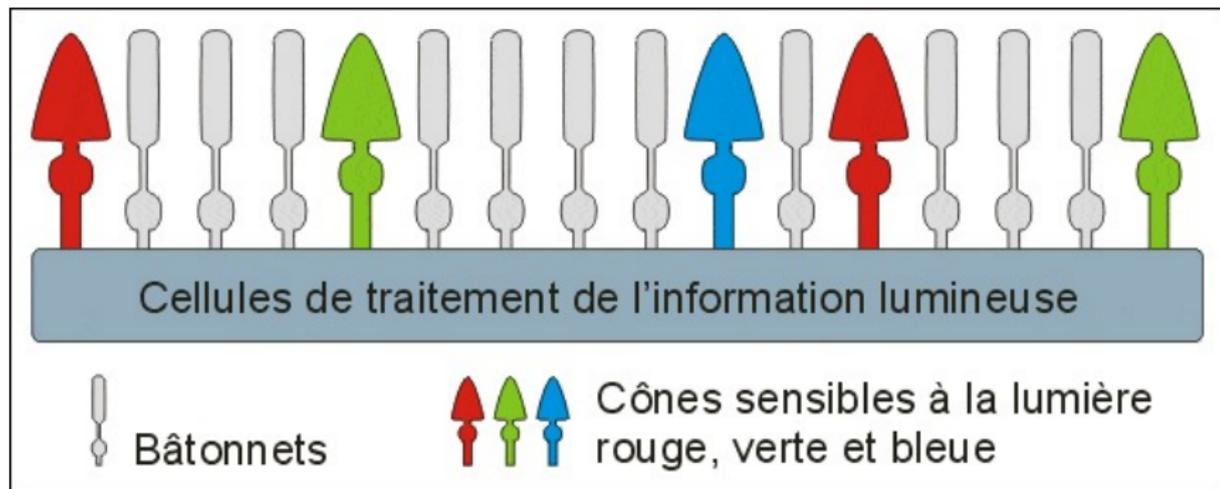
- 1 Formation des images
- 2 Couleur**
- 3 Occlusion - contours - gradient
- 4 Côté variationnel
- 5 Côté patches

# Cônes et batonnets

## Schéma de l'appareil visuel



# Cônes et batonnets



## Cônes et batonnets<sup>2</sup>

<sup>2</sup>Source : <http://www.happyview.fr>

# Vision trichromatique

- La rétine humaine comporte des “cônes” et des “bâtonnets”.
- Les **bâtonnets** des humains sont **monochromatiques** (*Dans la nuit, tous les chats sont gris*).
- Les **cônes** humains sont majoritairement **trichromatiques** (certains individus, surtout des femmes, seraient tétrachromatiques, avec une composante entre le rouge et le vert<sup>3</sup>).
- La plupart des animaux semblent être trichromatiques. Néanmoins, certains insectes et oiseaux ont une vision tétrachromatique (avec une composante UV). D'autres animaux mono-, ou di-chromatiques.

Le chat a une vision dichromatique (pas de rouge)

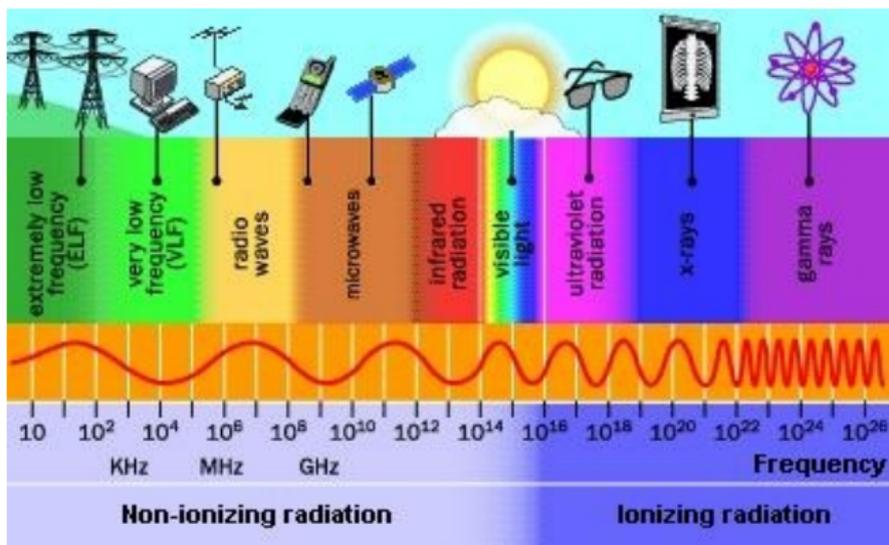
Le lapin est dichromatique (pas de rouge)

Le dauphin a une vision monochromatique (vers le bleu)

---

<sup>3</sup><http://en.wikipedia.org/wiki/Tetrachromacy>

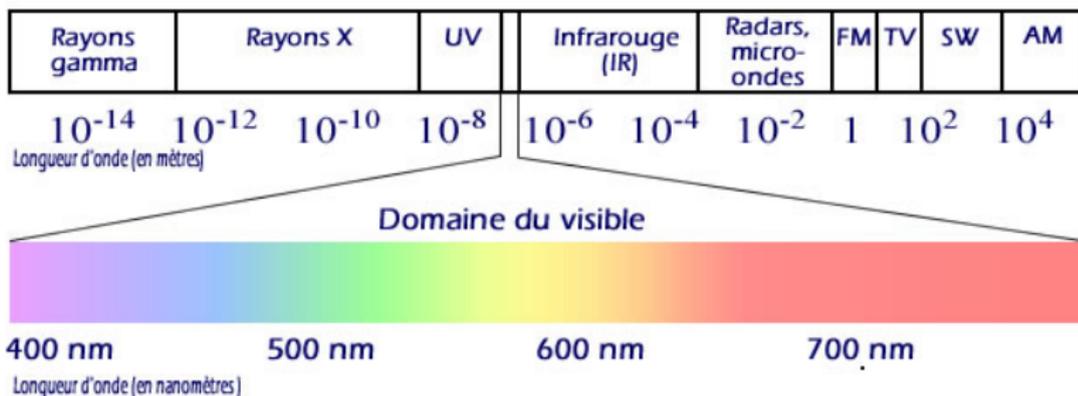
# Spectre de la lumière



## Spectre de la lumière<sup>4</sup>

<sup>4</sup>Source : [http://lesantennespourlesnuls.pagesperso-orange.fr/page\\_notion\\_1.htm](http://lesantennespourlesnuls.pagesperso-orange.fr/page_notion_1.htm)

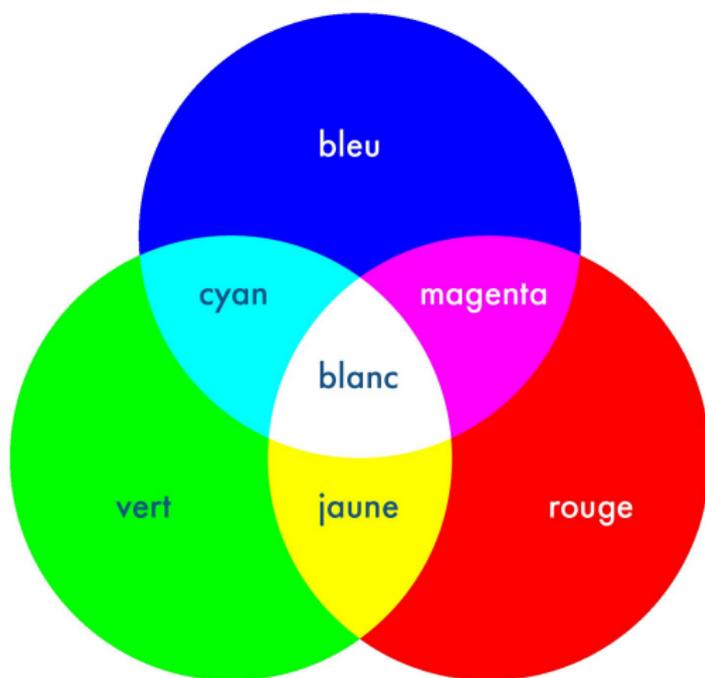
# Spectre de la lumière



## Spectre de la lumière<sup>5</sup>

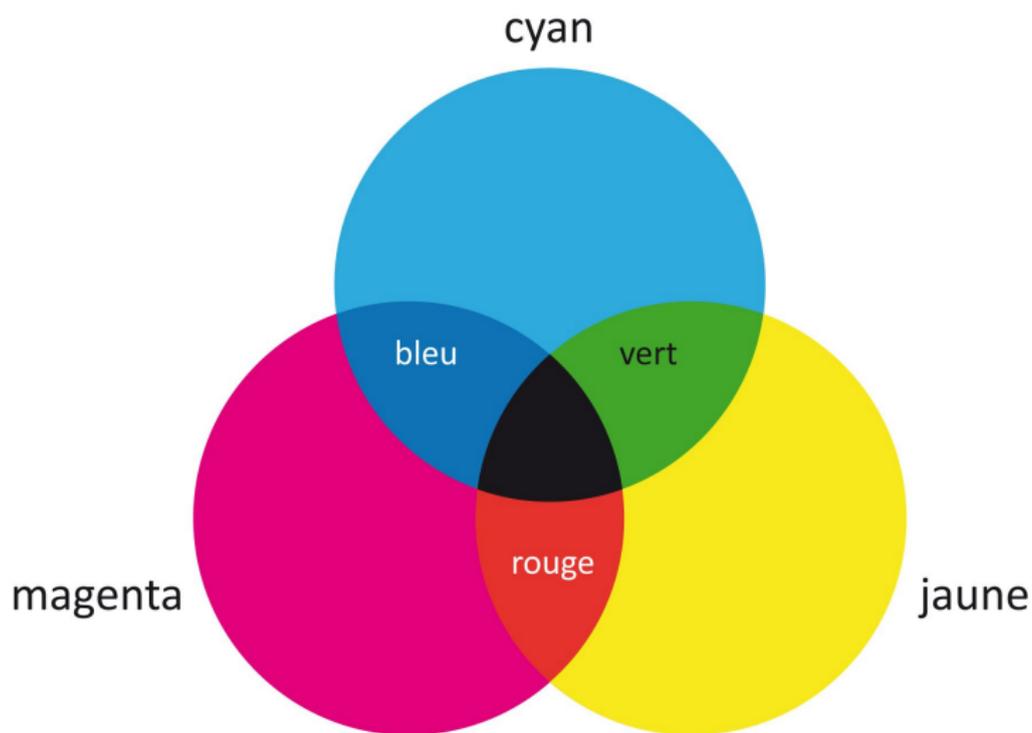
<sup>5</sup>Source :

# Cercle chromatique (synthèse additive)

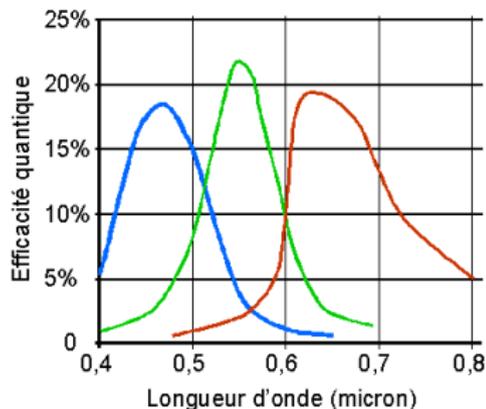
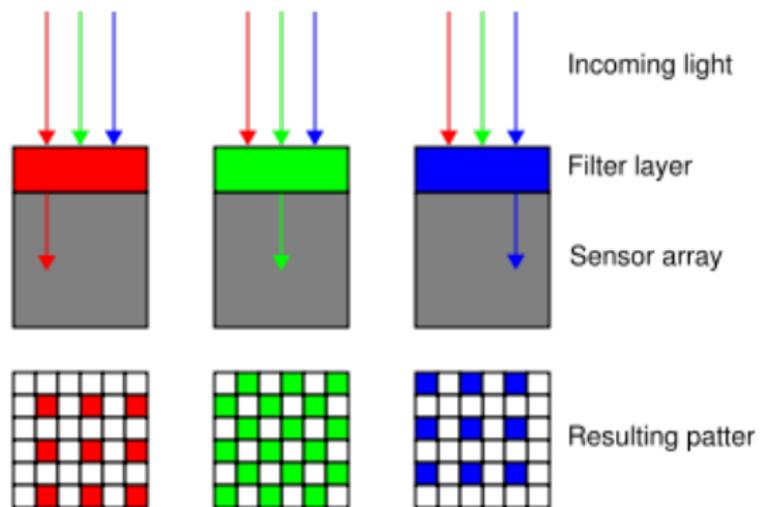


[coloringmate.co](http://coloringmate.co)

# Cercle chromatique (synthèse soustractive)



# Filtres couleur

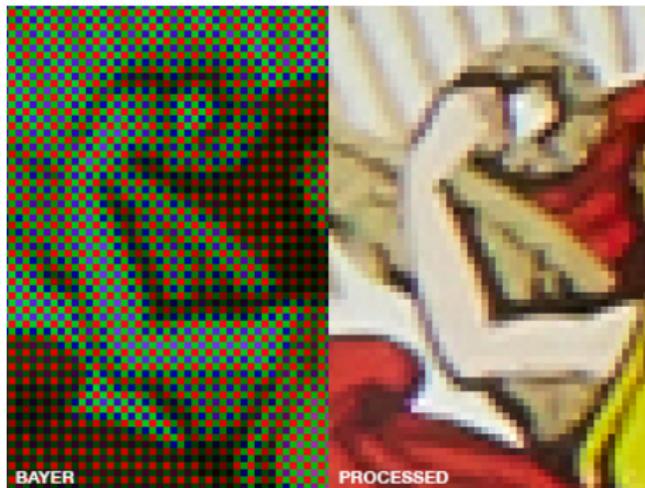
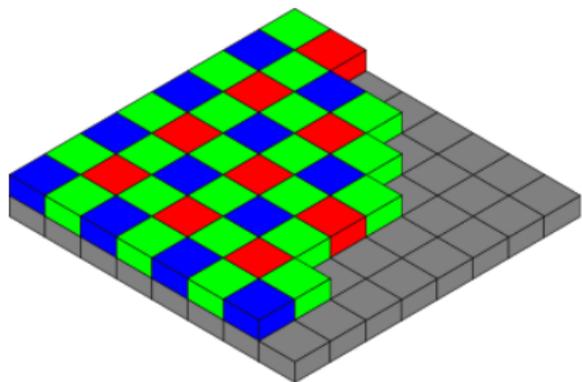


Récepteurs filtrés rouge, vert, bleu<sup>6</sup> – sensibilité en fréquence<sup>7</sup>

<sup>6</sup>Source : [http://commons.wikimedia.org/wiki/File:Bayer\\_pattern\\_on\\_sensor\\_profile.svg](http://commons.wikimedia.org/wiki/File:Bayer_pattern_on_sensor_profile.svg)

<sup>7</sup>Source : [http://commons.wikimedia.org/wiki/File:CCD\\_Pleine-Trame\\_Spectrale.png](http://commons.wikimedia.org/wiki/File:CCD_Pleine-Trame_Spectrale.png)

# Matrice de Bayer

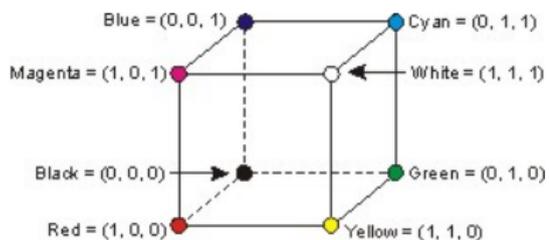


Matrice de Bayer<sup>8</sup> – Image mosaïquée – Image démosaïquée<sup>9</sup>

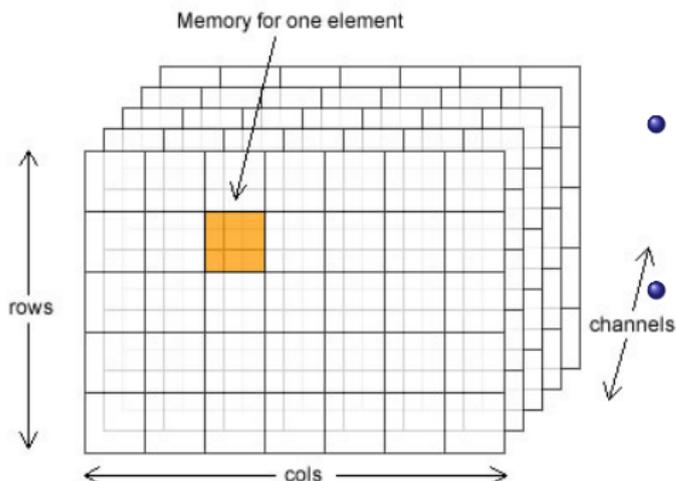
<sup>8</sup> [http://en.wikipedia.org/wiki/Bayer\\_filter](http://en.wikipedia.org/wiki/Bayer_filter)

<sup>9</sup> <http://www.phaseone.com/en/Imaging-Software/Capture-One-Express-7/Features.aspx>

# Représentation matricielle d'une image couleur



- Standard = trichromatique RGB (rouge vert bleu).
- Image représentée par matrice à 3 entrées (3 dimensions)
- $I(i, j, k)$  est l'intensité de l'image à la ligne  $i$ , colonne  $j$  et canal  $k$ .
- $k = 1 \Rightarrow$  canal rouge
- $k = 2 \Rightarrow$  canal vert
- $k = 3 \Rightarrow$  canal bleu



# Décomposition RGB



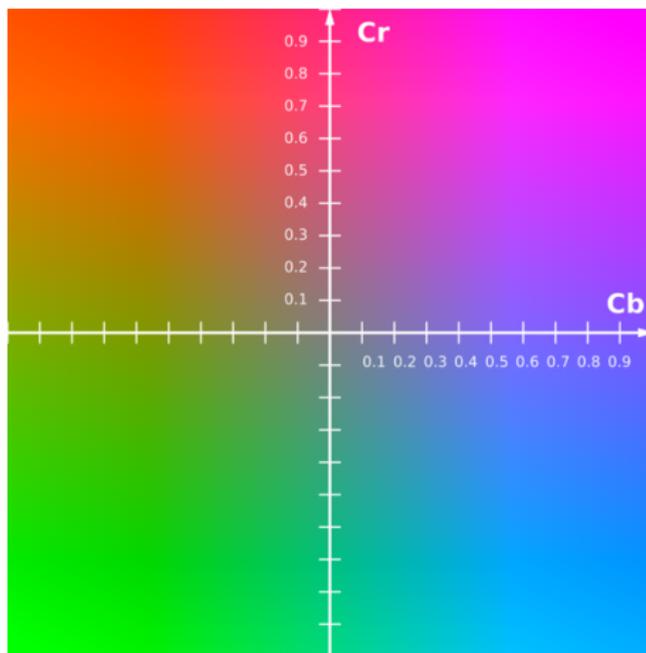
Couleur – Canaux Rouge - Vert - Bleu<sup>10</sup>

---

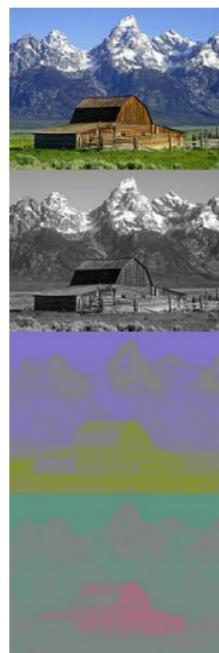
<sup>10</sup> <http://digital.pho.to/>

# Décomposition YCbCr

Y : luminance ; Cb : chrominance bleue ; Cr : chrominance rouge.



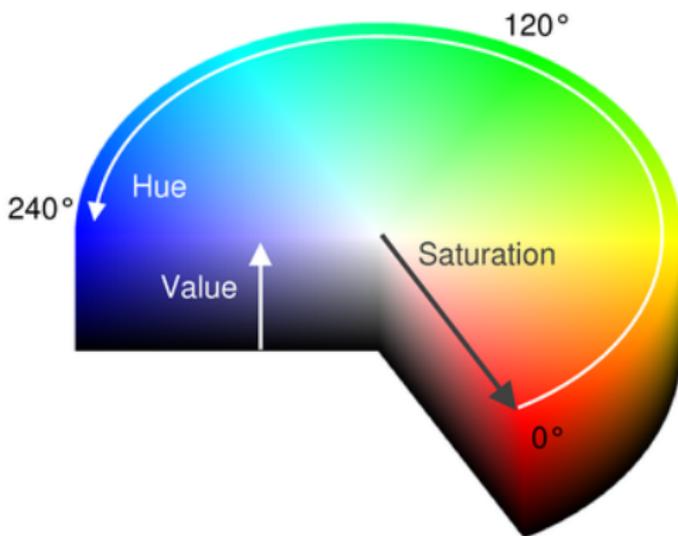
Plan Chrominance



Canaux Y-Cb-Cr<sup>11</sup>

<sup>11</sup> <http://en.wikipedia.org/wiki/YCbCr>

# Décomposition HSV



Espace HSV. H : *hue* (teinte). S : saturation. V : valeur.

Source :

<http://stackoverflow.com/questions/11598043/get-slightly-lighter-and-darker-color-from-uicolor>

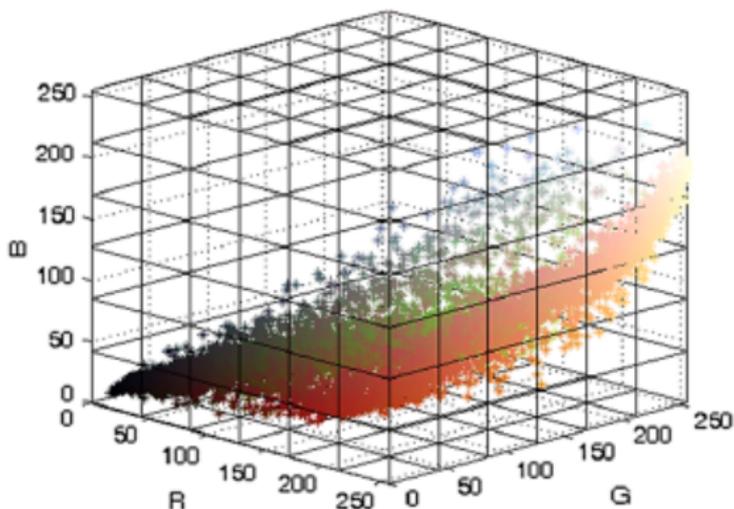
# Décomposition HSV



couleur – teinte – saturation – valeur

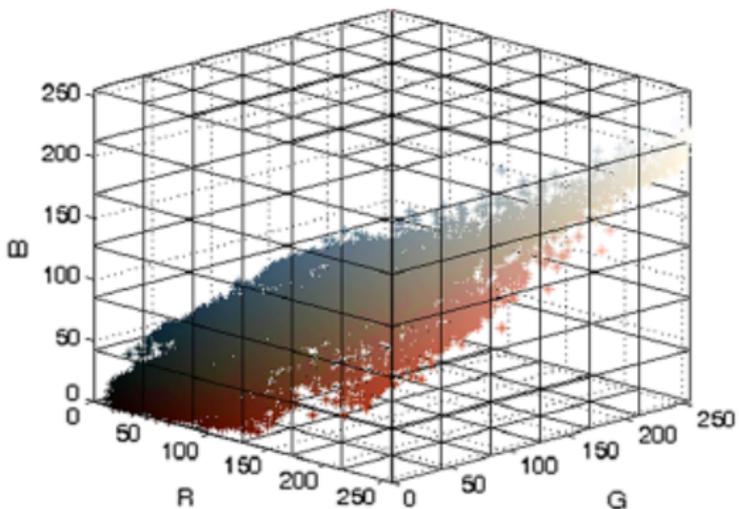
Source : <http://blog.patdavid.net>

# Segmentation d'image couleur



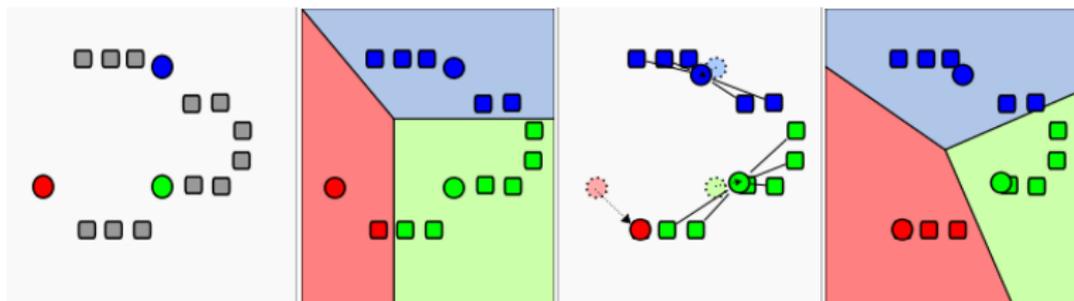
tiré de Vantaram, Saber, "Survey of contemporary trends in color image segmentation", *Journal of Electronic Imaging*

# Segmentation d'image couleur

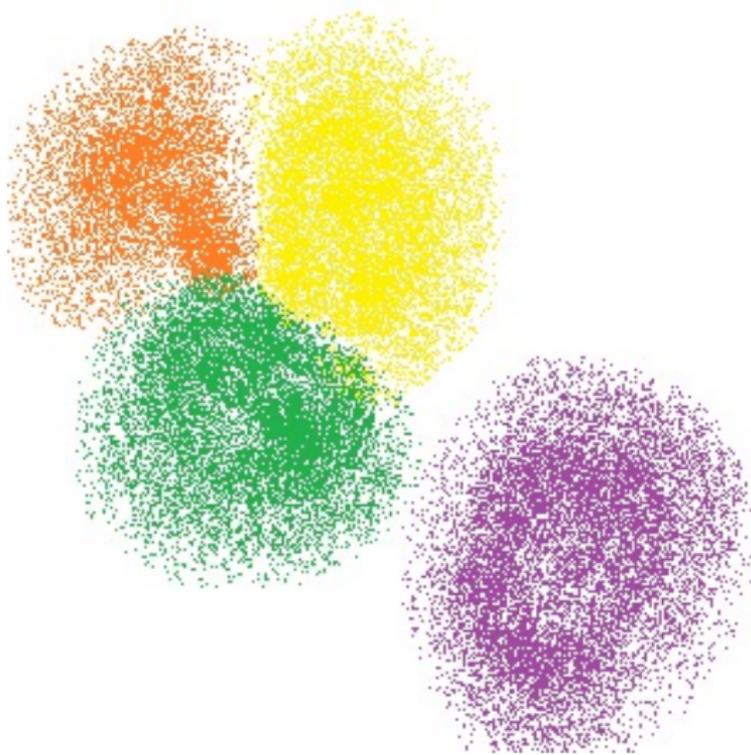


tiré de Vantaram, Saber, "Survey of contemporary trends in color image segmentation", *Journal of Electronic Imaging*

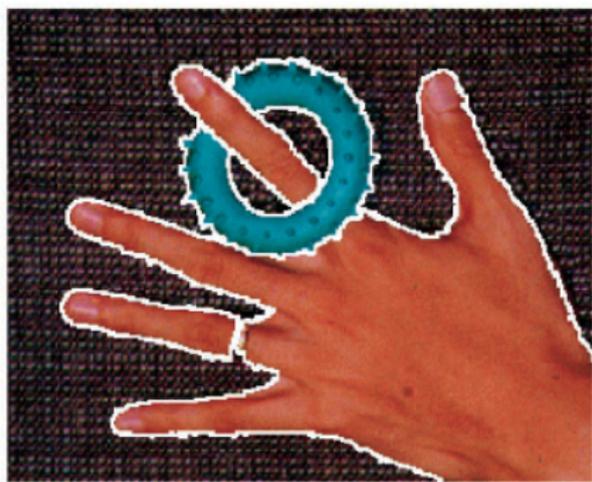
# Algorithme des $k$ -moyennes



# Algorithme des $k$ -moyennes



# Version un peu améliorée



tiré de Comaniciu, Meer, "Mean shift: a robust approach toward feature space analysis", PAMI 2002

# Version un peu améliorée



tiré de Comaniciu, Meer, "Mean shift: a robust approach toward feature space analysis", PAMI 2002

# Version un peu améliorée



- 1 Formation des images
- 2 Couleur
- 3 Occlusion - contours - gradient**
- 4 Côté variationnel
- 5 Côté patches

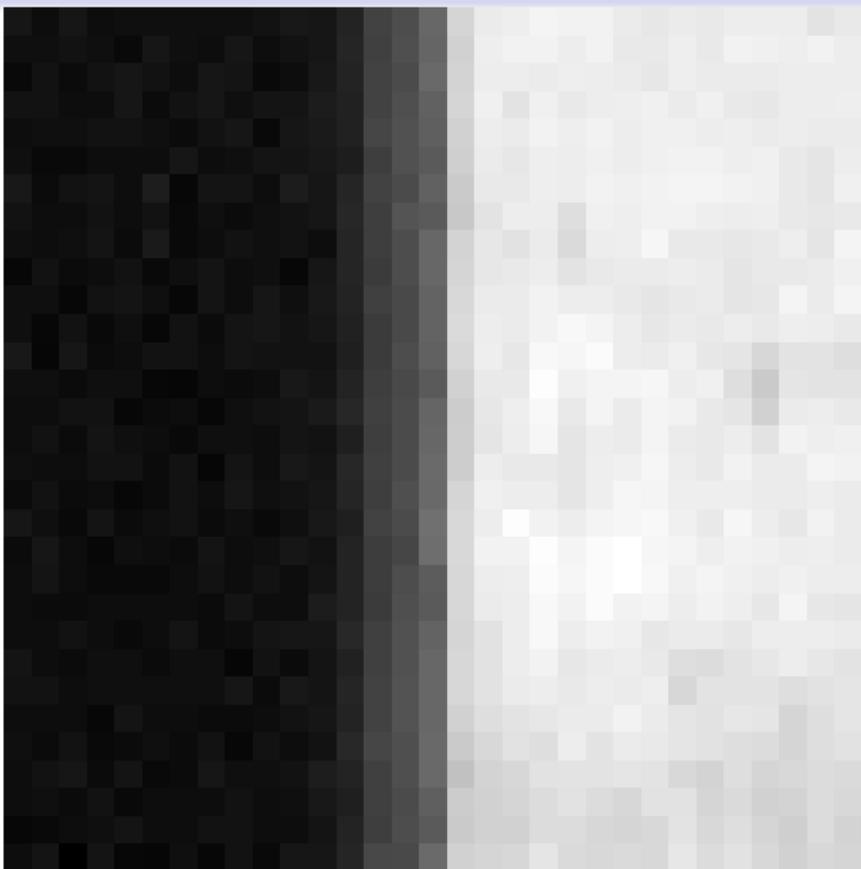
# Gradient



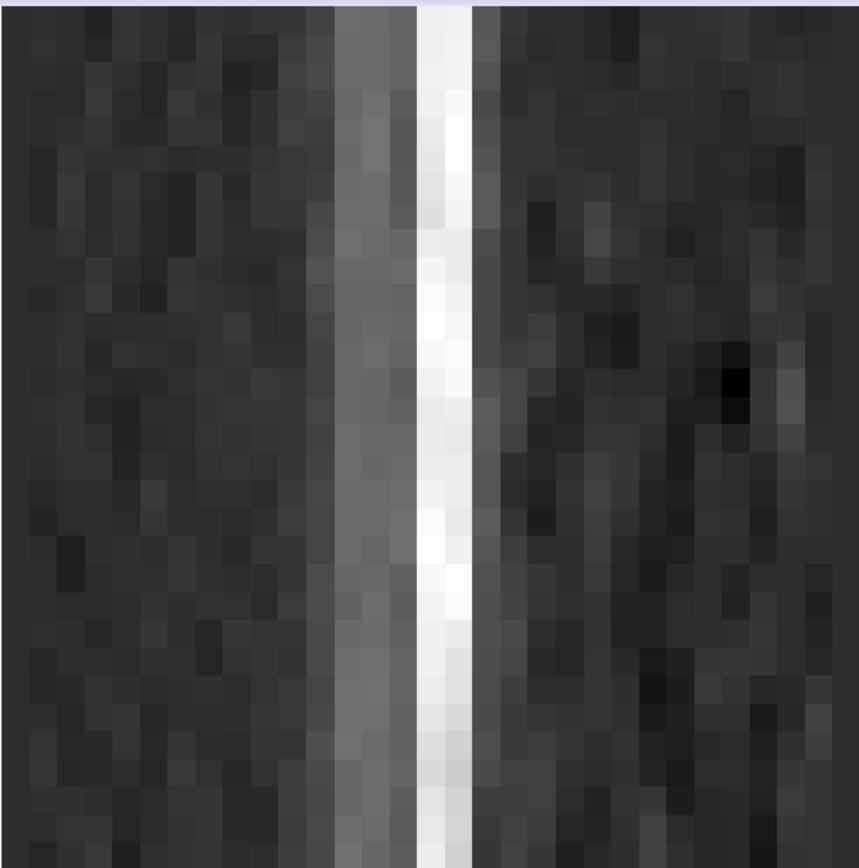
# Gradient



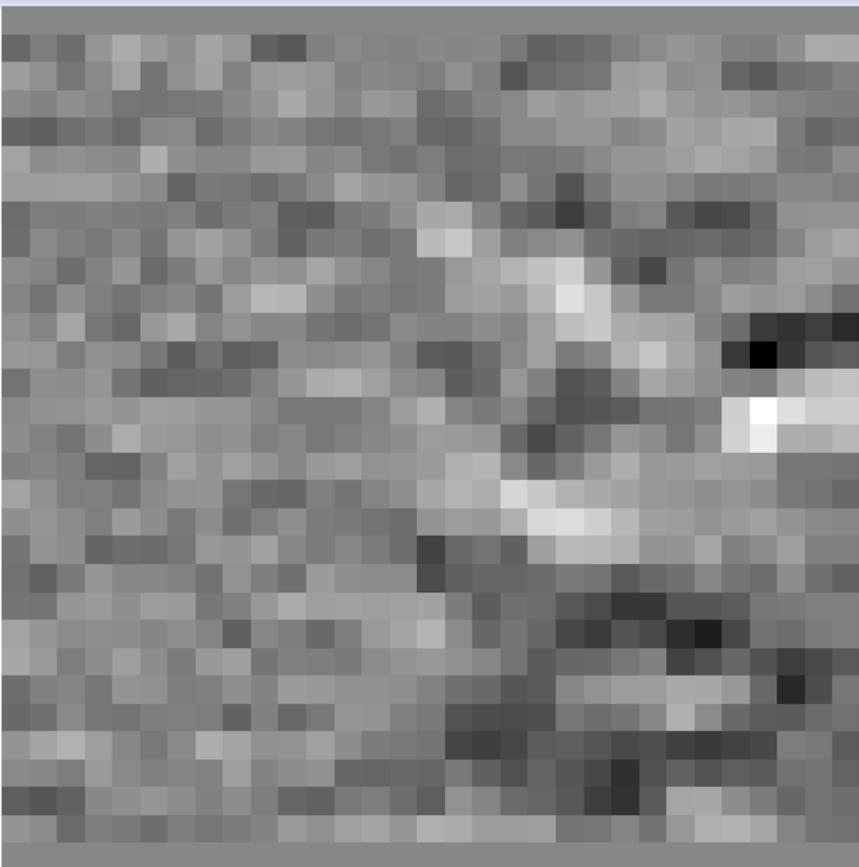
# Gradient



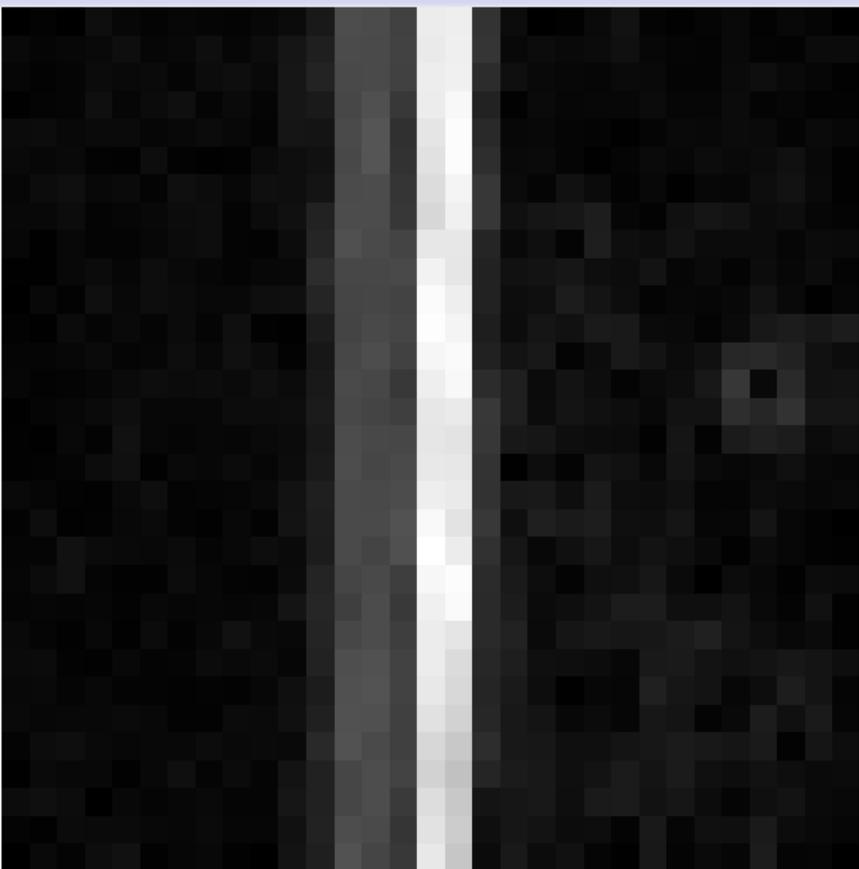
# Gradient



# Gradient



# Gradient



# Gradient



# Gradient



# Gradient



# Gradient

$$\begin{cases} \nabla_x u(i, j) = u(i + 1, j) - u(i, j) \\ \nabla_y u(i, j) = u(i, j + 1) - u(i, j) \\ |\nabla u|(i, j) = |\nabla_x u(i, j)| + |\nabla_y u(i, j)| \end{cases}$$



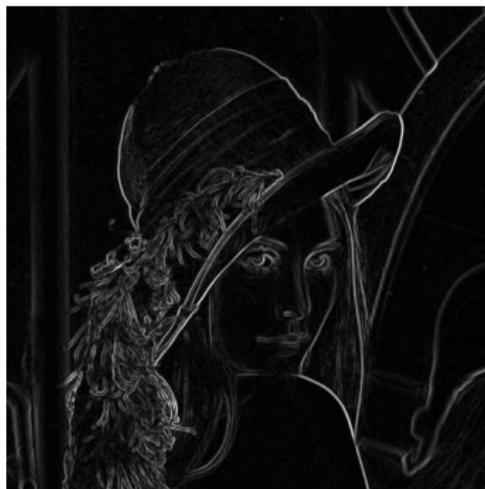
# Gradient

$$\begin{cases} \nabla_x u(i, j) = u(i + 1, j) - u(i, j) \\ \nabla_y u(i, j) = u(i, j + 1) - u(i, j) \\ |\nabla u|(i, j) = |\nabla_x u(i, j)| + |\nabla_y u(i, j)| \end{cases}$$



# Gradient

$$\begin{cases} \nabla_x u(i, j) = u(i + 1, j) - u(i, j) \\ \nabla_y u(i, j) = u(i, j + 1) - u(i, j) \\ |\nabla u|(i, j) = |\nabla_x u(i, j)| + |\nabla_y u(i, j)| \end{cases}$$



# Gradient + seuillage



# Gradient + seuillage



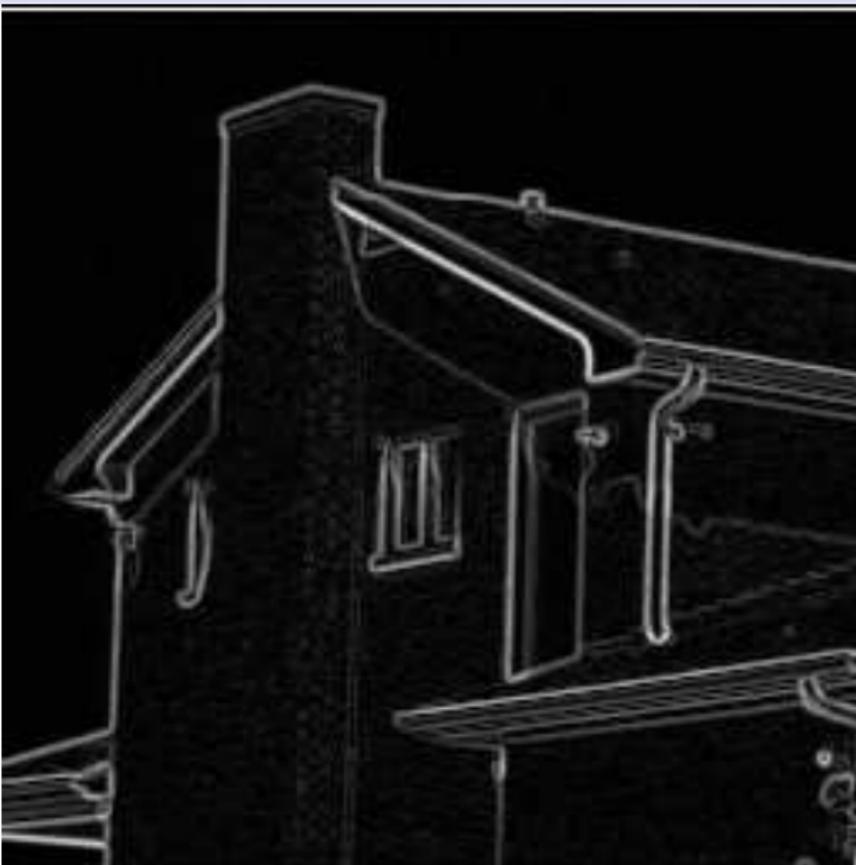
# Gradient + seuillage



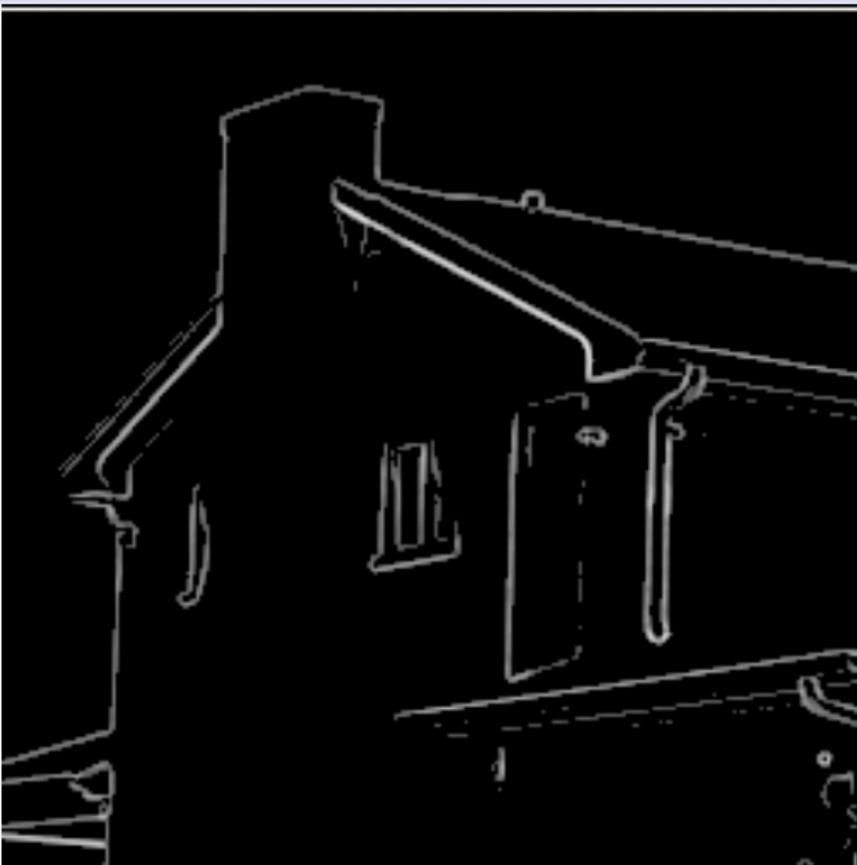
# Gradient + seuillage



# Gradient + seuillage



# Gradient + seuillage



# Gradient + seuillage



# Gradient + seuillage



# Gradient + seuillage



# Gradient + seuillage

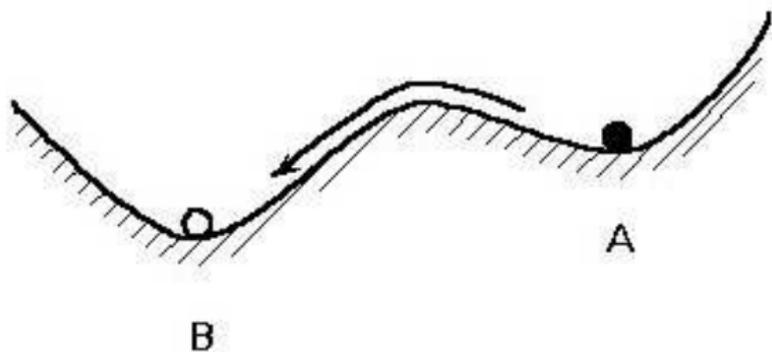
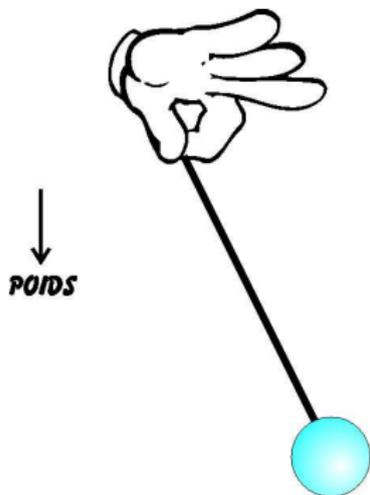
A quoi peut servir de connaître les contours d'objets?

- médical : volume d'une tumeur / d'un organe
- surveillance d'un hall de gare : bagage abandonné
- suivi d'objet dans un film (ballon de foot, balle de tennis)
- reconnaissance de formes
- comptage d'animaux, d'arbres, pour la préservation des espèces
- ...

- 1 Formation des images
- 2 Couleur
- 3 Occlusion - contours - gradient
- 4 Côté variationnel**
- 5 Côté patches

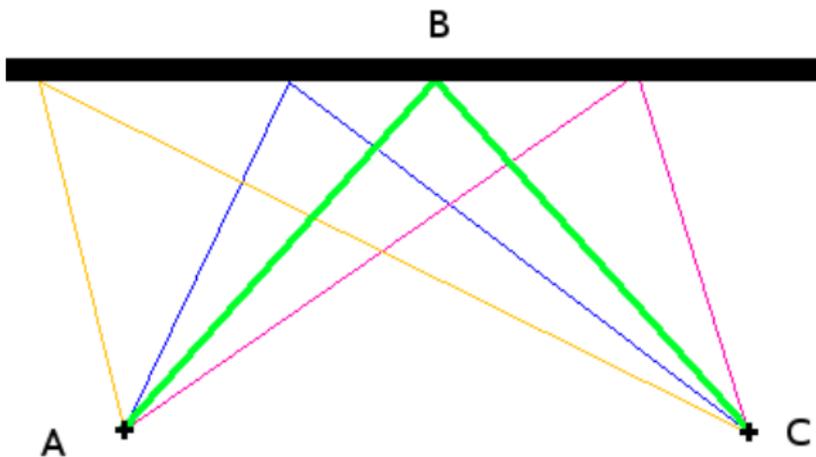
# Minimisation d'une énergie

- pendule rigide : équilibre stable et instable.
- énergie : minimum local, maximum local.



# Exemple de problème d'optimisation (1)

Quelle est la trajectoire la plus courte entre  $A$  et  $C$  d'un rayon lumineux réfléchi sur un miroir?

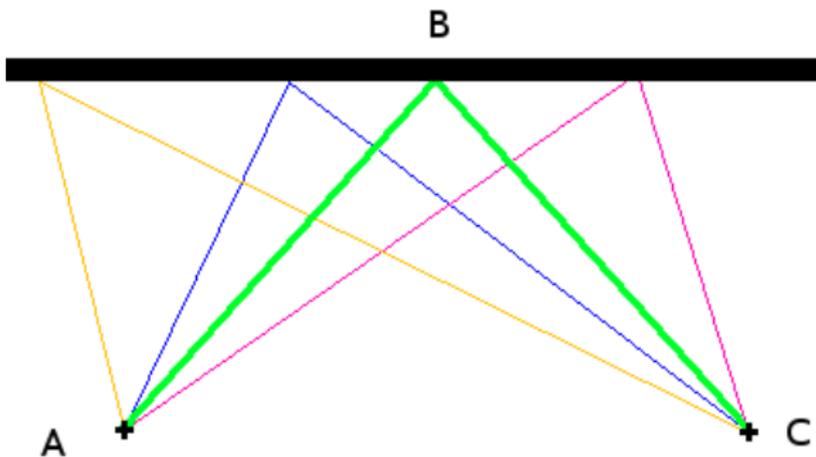


On peut définir et minimiser une énergie qui dépend de l'angle :

$$E(\theta) = \|\overrightarrow{AB(\theta)}\| + \|\overrightarrow{B(\theta)C}\|.$$

# Exemple de problème d'optimisation (1)

Quelle est la trajectoire la plus courte entre  $A$  et  $C$  d'un rayon lumineux réfléchi sur un miroir?

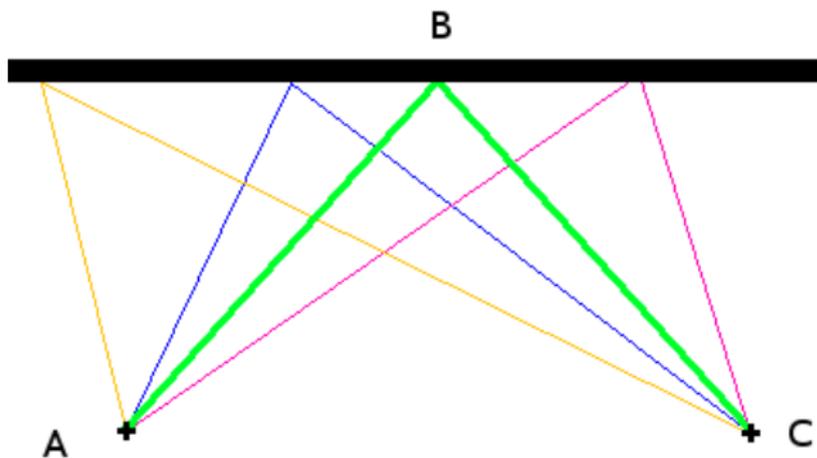


On peut définir et minimiser une énergie qui dépend de l'angle :

$$E(\theta) = \|\overrightarrow{AB(\theta)}\| + \|\overrightarrow{B(\theta)C}\|.$$

# Exemple de problème d'optimisation (1)

Quelle est la trajectoire la plus courte entre  $A$  et  $C$  d'un rayon lumineux réfléchi sur un miroir?



On peut définir et minimiser une énergie qui dépend de l'angle :

$$E(\theta) = \|\overrightarrow{AB(\theta)}\| + \|\overrightarrow{B(\theta)C}\|.$$

## Exemple de problème d'optimisation (2)

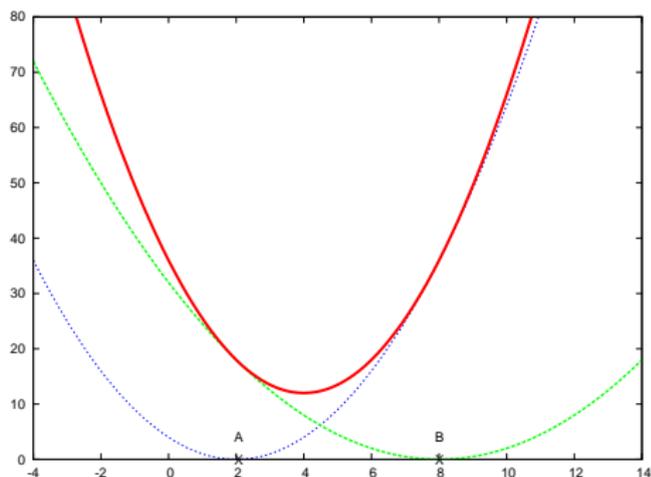
Je souhaite acheter une maison.

- Mon meilleur ami habite en un lieu A et je voudrais habiter le plus près possible de chez lui.
- Je travaille dans un lieu B et je voudrais aussi habiter pas trop loin de mon lieu de travail (mais c'est 2 fois moins important).

Où dois-je m'installer pour être au mieux?

Formulation variationnelle:

$$E(x) = (x - a)^2 + \frac{1}{2}(x - b)^2.$$



## Exemple de problème d'optimisation (2)

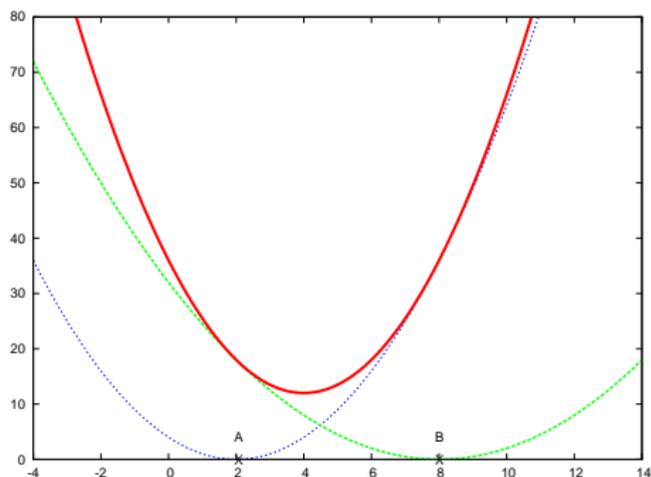
Je souhaite acheter une maison.

- Mon meilleur ami habite en un lieu A et je voudrais habiter le plus près possible de chez lui.
- Je travaille dans un lieu B et je voudrais aussi habiter pas trop loin de mon lieu de travail (mais c'est 2 fois moins important).

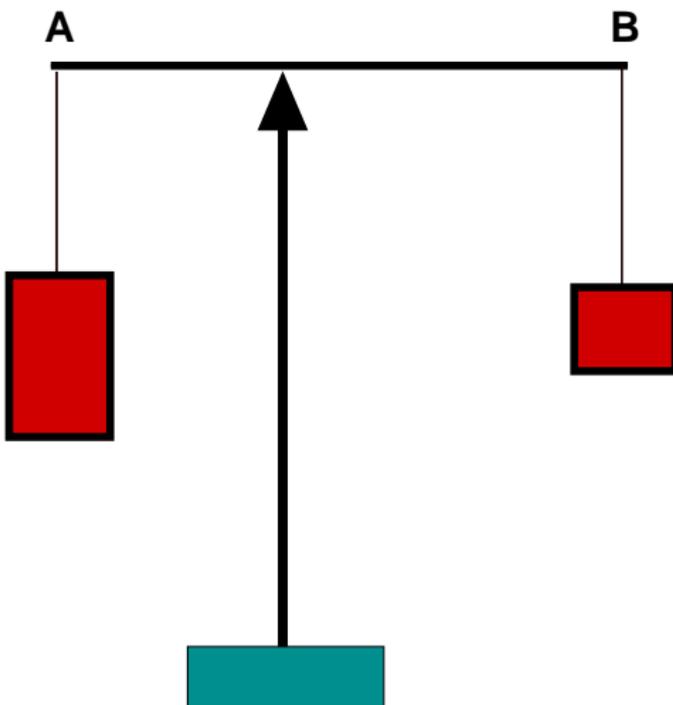
Où dois-je m'installer pour être au mieux?

Formulation variationnelle:

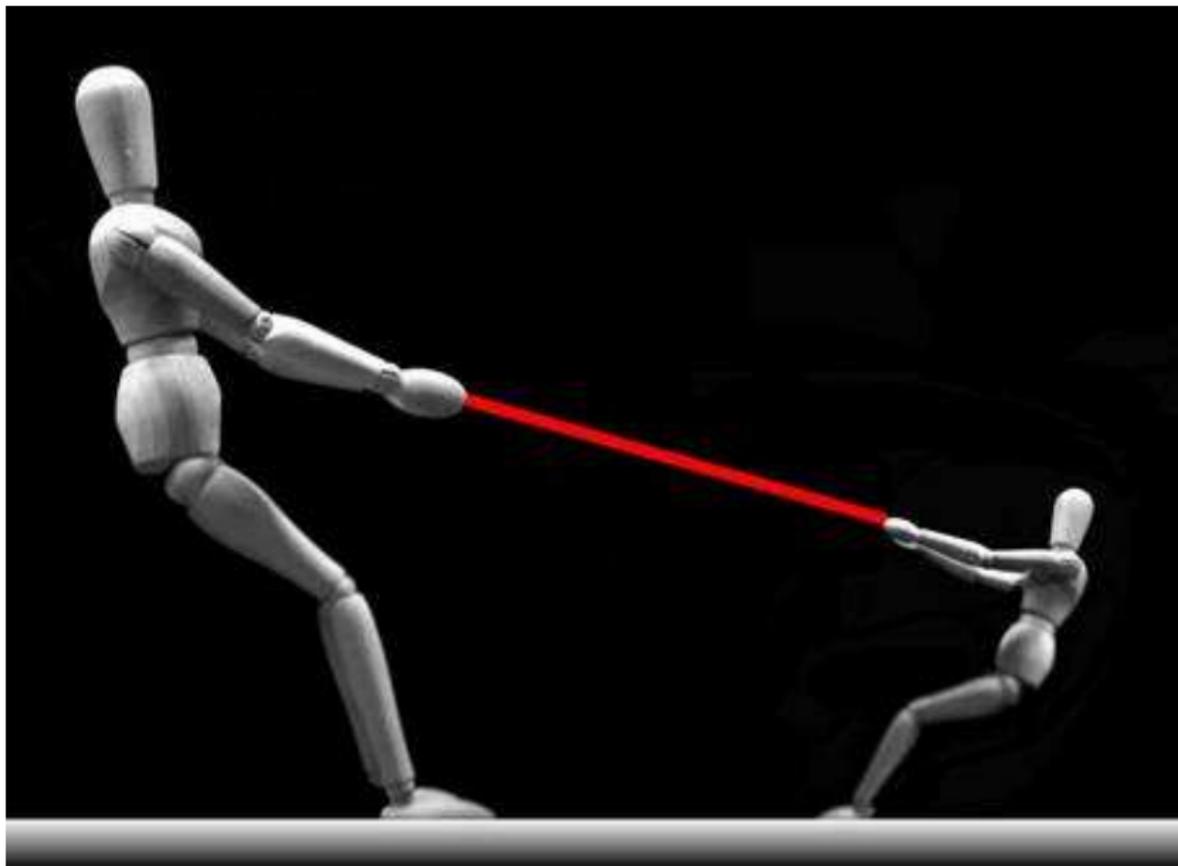
$$E(x) = (x - a)^2 + \frac{1}{2}(x - b)^2.$$



## Exemple de problème d'optimisation (2)



Solution:  $x = \text{barycentre} ((a, 2), (b, 1))$ .



## Exemple de problème d'optimisation (3)

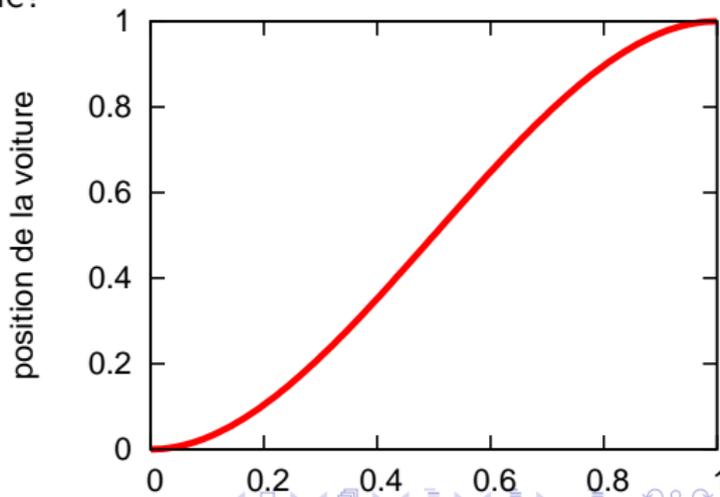
Je souhaite aller d'un point A à un point B en voiture (trajectoire rectiligne, départ et arrivée à vitesse nulle).

- je consomme une dose d'essence proportionnelle au carré de l'accélération (positive ou négative)
- je veux mettre T minutes au plus, mais je veux économiser le plus possible d'essence.

Quelle est la trajectoire optimale?

Ici la particule est une trajectoire. On cherche  $x(t)$  qui minimise la quantité totale d'essence utilisée

$E(x) = \int_0^T x''(t)^2 dt$ , avec  $x(0) = a$ ,  $x(T) = b$ . Ici la particule est une trajectoire.



## Exemple de problème d'optimisation (3)

Je souhaite aller d'un point A à un point B en voiture (trajectoire rectiligne, départ et arrivée à vitesse nulle).

- je consomme une dose d'essence proportionnelle au carré de l'accélération (positive ou négative)
- je veux mettre T minutes au plus, mais je veux économiser le plus possible d'essence.

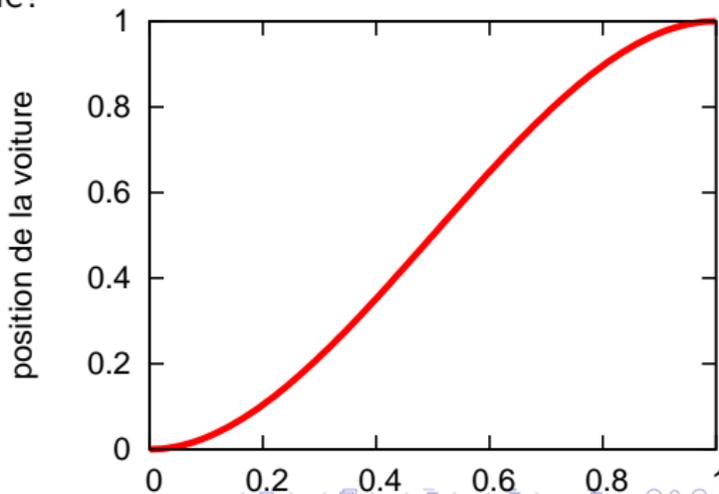
Quelle est la trajectoire optimale?

Ici la particule est une trajectoire. On cherche  $x(t)$  qui minimise la quantité totale d'essence utilisée

$$E(x) = \int_0^T x''(t)^2 dt, \text{ avec}$$

$$x(0) = a, x(T) = b. \text{ Ici la}$$

particule est une trajectoire.



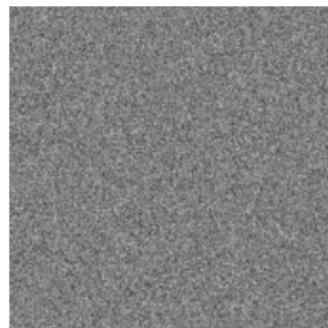
# Optimisation sur une image - débruitage

 $v$ 

=

 $u$ 

+

 $\varepsilon$ 

=

+

Critère de régularité : la variation totale

$$TV(u) = \sum_{\text{pixels } x} |\nabla u|(x).$$

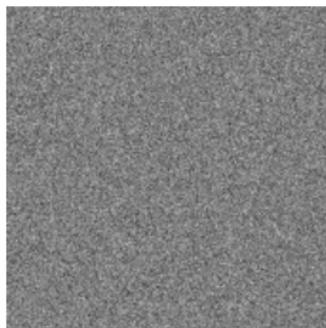
# Optimisation sur une image - débruitage

 $v$ 

=

 $u$ 

+

 $\varepsilon$ 

Critère de régularité : la variation totale

$$TV(u) = \sum_{\text{pixels } x} |\nabla u|(x).$$



$TV = 10.0156$



$TV = 15.4342$



$TV = 22.7455$



$TV = 38.8978$



$TV = 72.5113$

# Optimisation sur une image - débruitage

- Fidélité : distance à l'image bruitée
- Régularité : variation totale.

On minimise  $E(u) = \|u - v\|^2 + \lambda TV(u)$ .









# Optimisation sur une image - déflouage

$$E(u) = \|A * u - v\|^2 + \lambda TV(u)$$



# Optimisation sur une image - déflouage

$$E(u) = \|A * u - v\|^2 + \lambda TV(u)$$



# Optimisation sur une image - déflouage

$$E(u) = \|A * u - v\|^2 + \lambda TV(u)$$

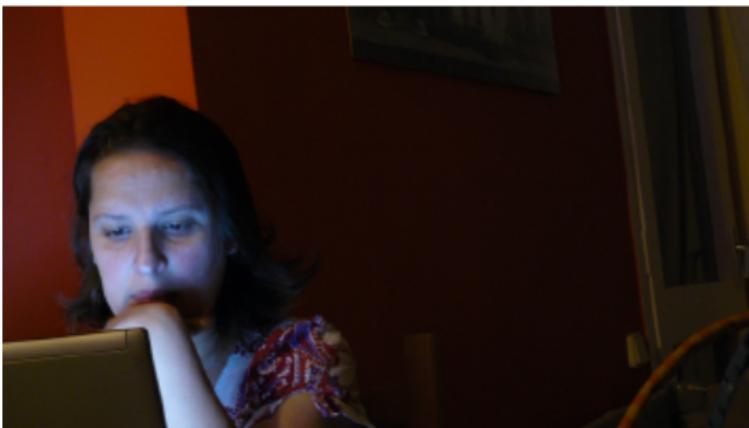


# Optimisation sur une image - déflouage

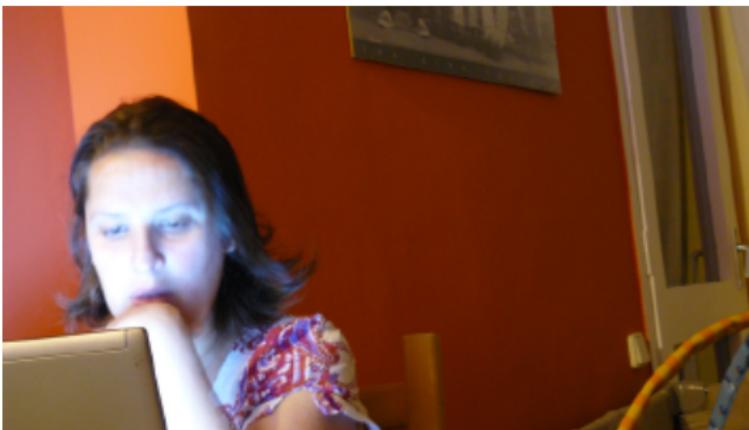
$$E(u) = \|A * u - v\|^2 + \lambda TV(u)$$



# Construire une bonne image à partir de 2 mauvaises



# Construire une bonne image à partir de 2 mauvaises



# Construire une bonne image à partir de 2 mauvaises

On va minimiser

$$E(u) = E_{\text{contours}}(u) + \lambda E_{\text{couleurs}}(u)$$

avec

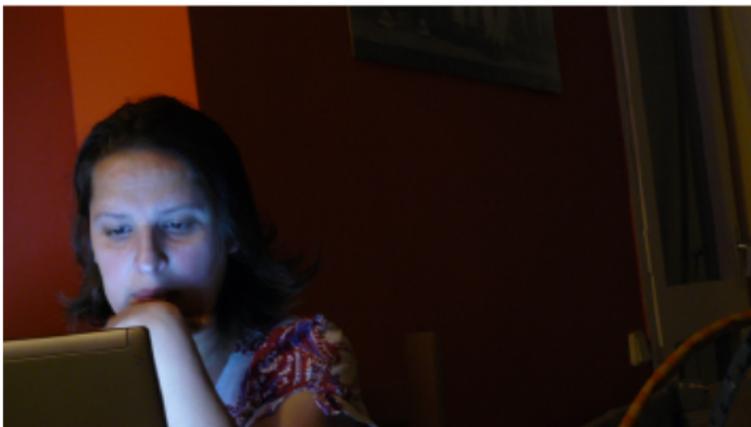
$$E_{\text{contours}}(u) = \sum_x \left( \left| |\nabla u|(x) - \nabla u(x) \cdot \frac{\nabla u_{\text{sombre}}(x)}{|\nabla u_{\text{sombre}}(x)|} \right| \right)$$

qui force les contours de  $u$  à bien suivre ceux de  $u_{\text{sombre}}$ , et

$$E_{\text{couleurs}}(u) = 2 \sum_x \sum_y |u(x) - u_{\text{floue}}(y)| + \sum_x \sum_y |u(x) - u(y)|,$$

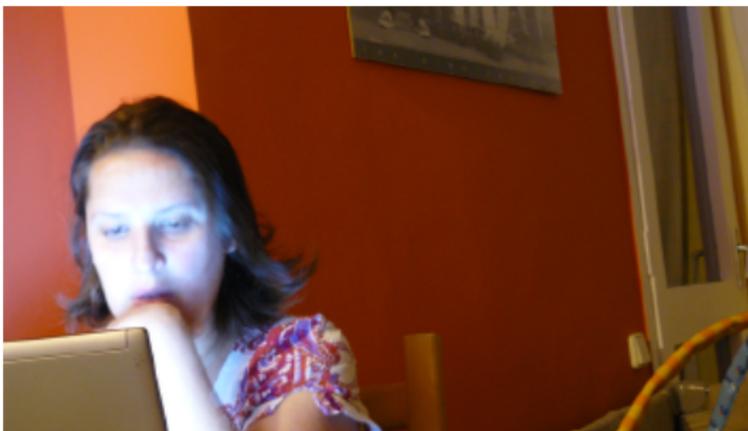
qui force les couleurs de  $u$  à ressembler à celles de  $u_{\text{floue}}$ .

# Construire une bonne image à partir de 2 mauvaises



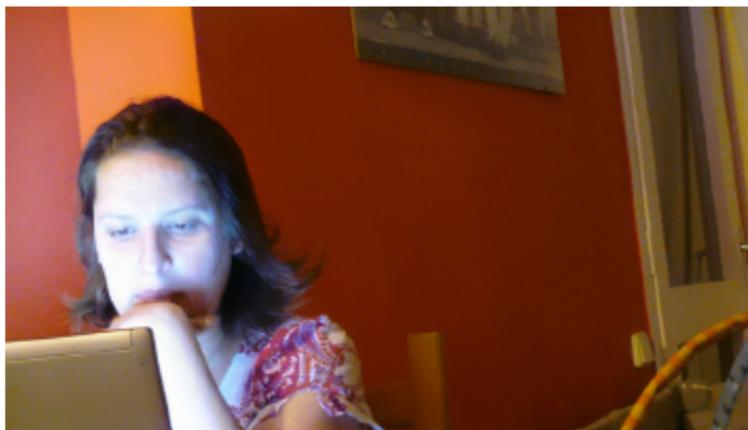
tiré de Bertalmio, Levine, "A Variational Approach for the Fusion of Bracketing Images", 2011

# Construire une bonne image à partir de 2 mauvaises



tiré de Bertalmio, Levine, "A Variational Approach for the Fusion of Bracketing Images", 2011

# Construire une bonne image à partir de 2 mauvaises



tiré de Bertalmio, Levine, "A Variational Approach for the Fusion of Bracketing Images", 2011

# Construire une bonne image à partir de 2 mauvaises



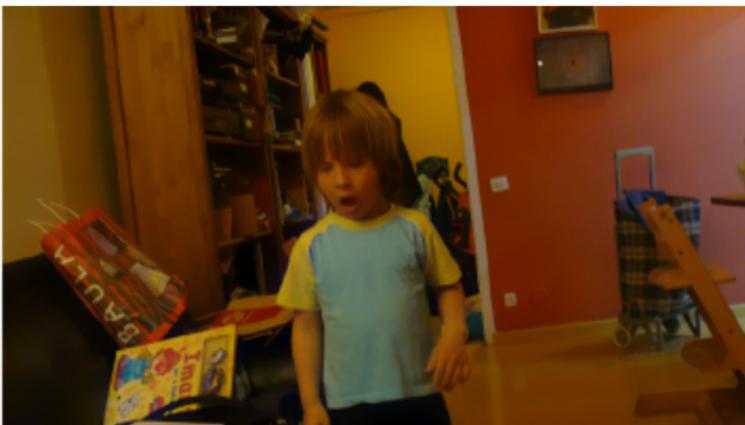
tiré de Bertalmio, Levine, "A Variational Approach for the Fusion of Bracketing Images", 2011

# Construire une bonne image à partir de 2 mauvaises



tiré de Bertalmio, Levine, "A Variational Approach for the Fusion of Bracketing Images", 2011

# Construire une bonne image à partir de 2 mauvaises



tiré de Bertalmio, Levine, "A Variational Approach for the Fusion of Bracketing Images", 2011

# Construire une bonne image à partir de 2 mauvaises



tiré de Bertalmio, Levine, "A Variational Approach for the Fusion of Bracketing Images", 2011

# Construire une bonne image à partir de 2 mauvaises



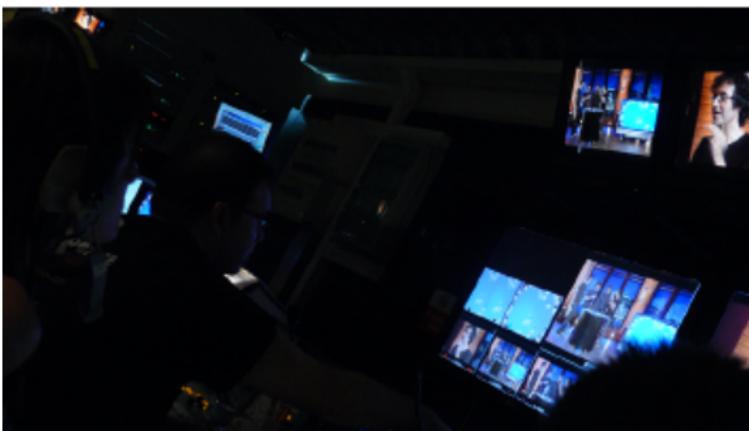
tiré de Bertalmio, Levine, "A Variational Approach for the Fusion of Bracketing Images", 2011

# Construire une bonne image à partir de 2 mauvaises



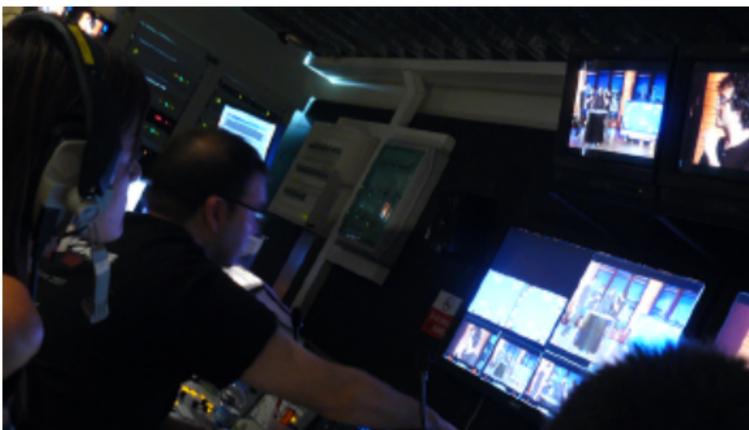
tiré de Bertalmio, Levine, "A Variational Approach for the Fusion of Bracketing Images", 2011

# Construire une bonne image à partir de 2 mauvaises



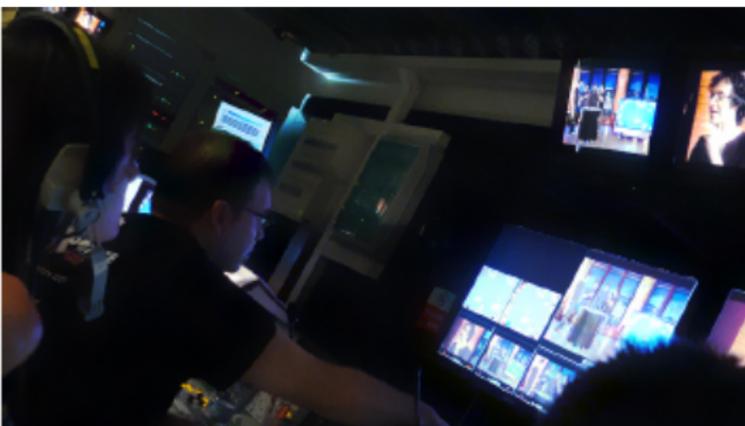
tiré de Bertalmio, Levine, "A Variational Approach for the Fusion of Bracketing Images", 2011

# Construire une bonne image à partir de 2 mauvaises



tiré de Bertalmio, Levine, "A Variational Approach for the Fusion of Bracketing Images", 2011

# Construire une bonne image à partir de 2 mauvaises



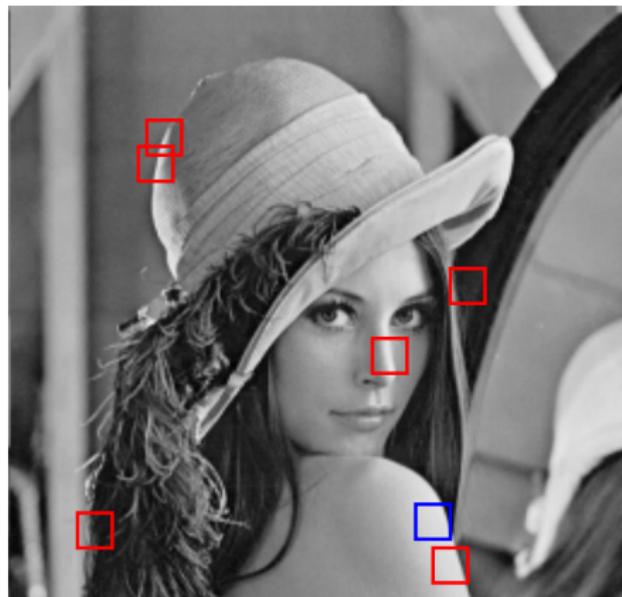
tiré de Bertalmio, Levine, "A Variational Approach for the Fusion of Bracketing Images", 2011

- 1 Formation des images
- 2 Couleur
- 3 Occlusion - contours - gradient
- 4 Côté variationnel
- 5 Côté patches**

# Optimisation de la redondance des patches d'une image



un patch



redondance des patches

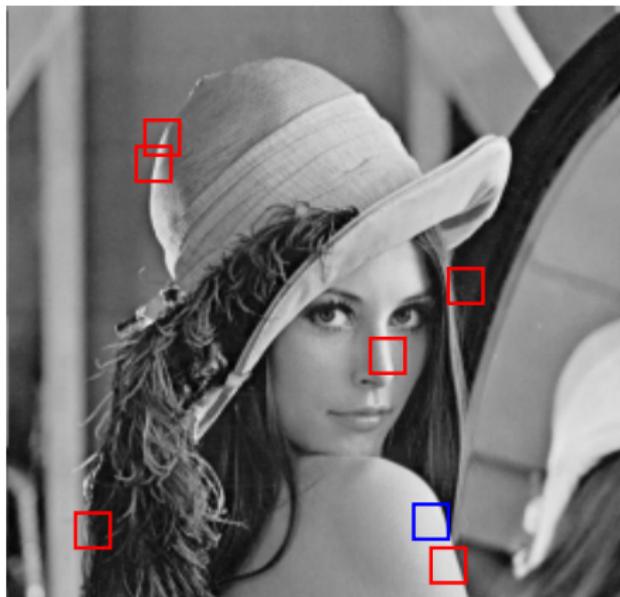
2 patches  $p_x$  et  $p_y$  se "ressemblent" ssi

$$\|p_x - p_y\|^2 = \sum_t |p_x(t) - p_y(t)|^2 < \varepsilon.$$

# Optimisation de la redondance des patches d'une image



un patch



redondance des patches

2 patches  $p_x$  et  $p_y$  se "ressemblent" ssi

$$\|p_x - p_y\|^2 = \sum_t |p_x(t) - p_y(t)|^2 < \varepsilon.$$

# Optimisation de la redondance des patches d'une image

- On définit [Kervrann 2006]

$$E(u) = \sum_x \sum_y \varphi (\|p_x(u) - \tilde{p}_y(u)\|)$$

avec  $\varphi$  croissante.

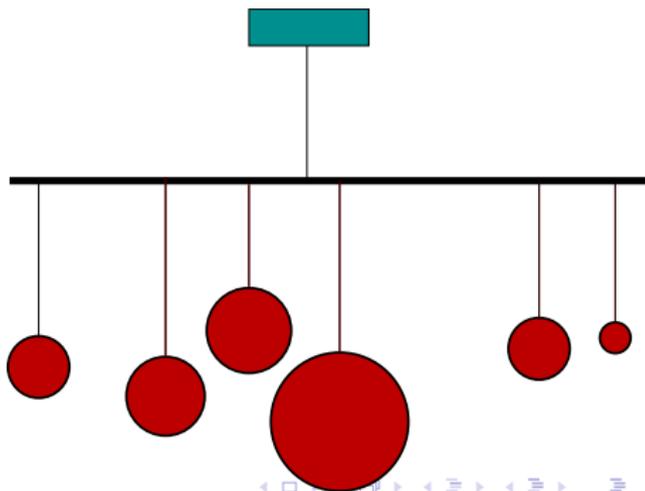
- Minimiser  $E$  revient à choisir comme image débruitée  $\hat{u}$ , où

$$\hat{u}(x) = \frac{\sum_{y \in \Omega} w(x, y) v(y)}{\sum_{y \in \Omega} w(x, y)}$$

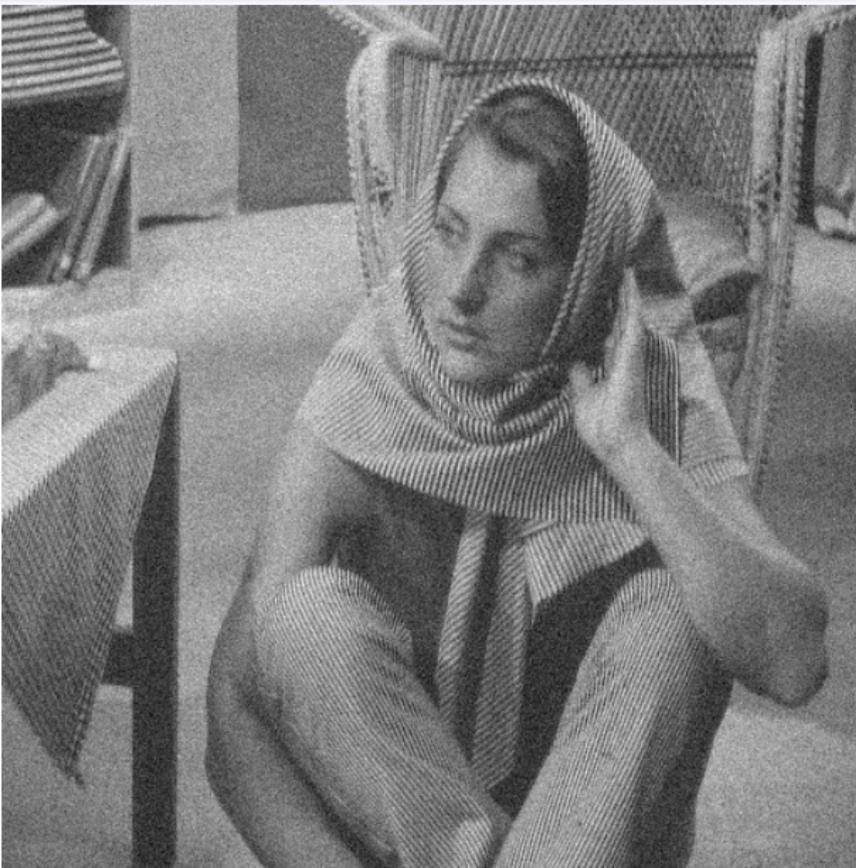
avec

$$w(x, y) = \psi (\|p_x(v) - p_y(v)\|)$$

où  $\psi$  est décroissante  
(définie à partir de  $\varphi$ ).



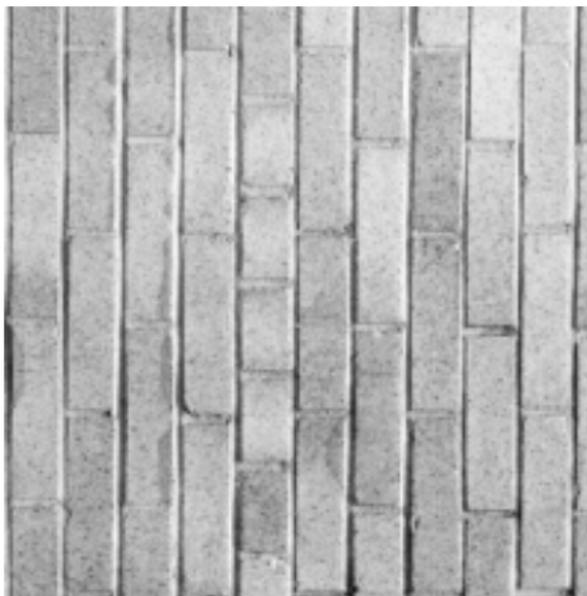
# NLmeans: expériences



# NLmeans: expériences

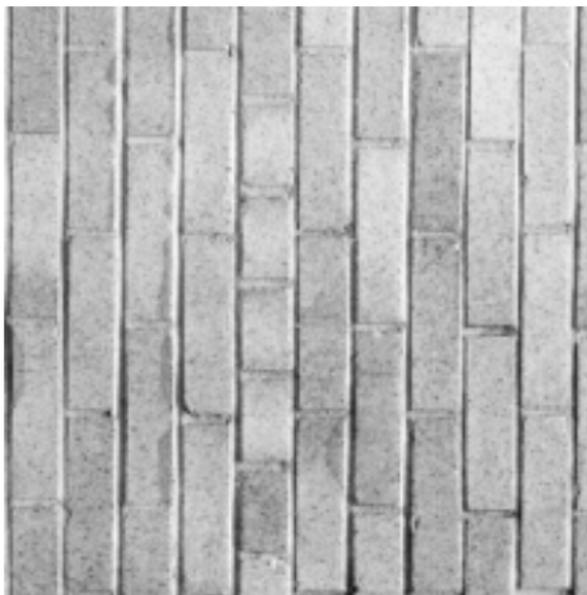
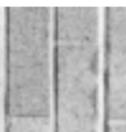


# Synthèse de texture



- jeux vidéo
- compression d'image
- analyse d'image

# Synthèse de texture



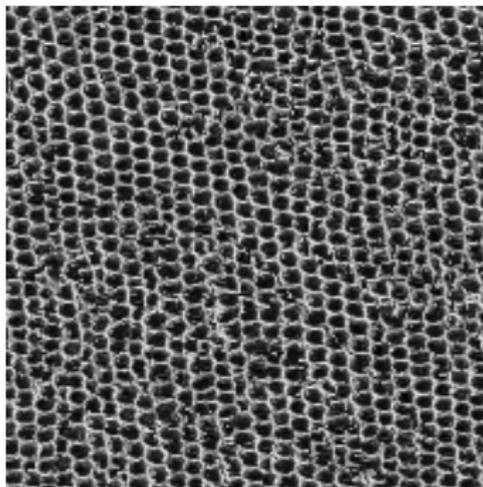
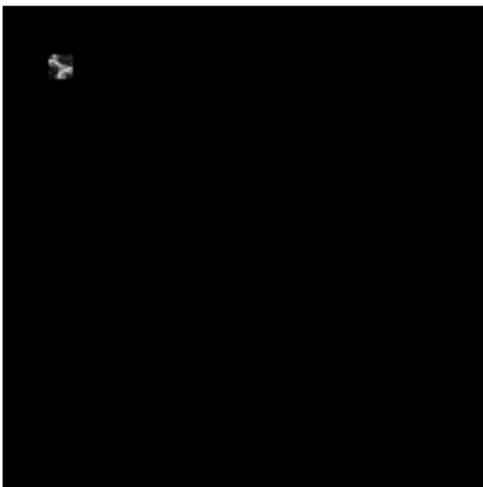
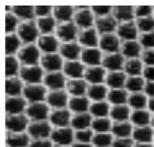
- jeux vidéo
- compression d'image
- analyse d'image

# Synthèse de texture

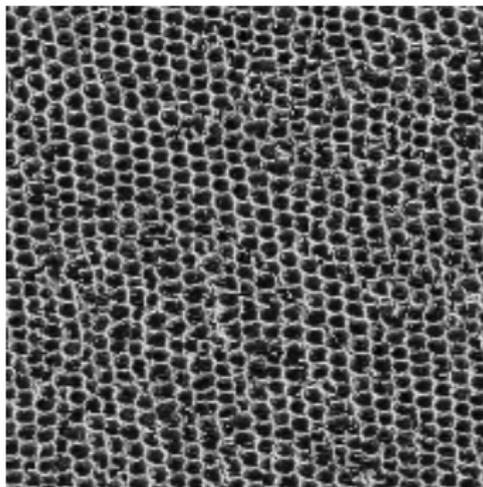
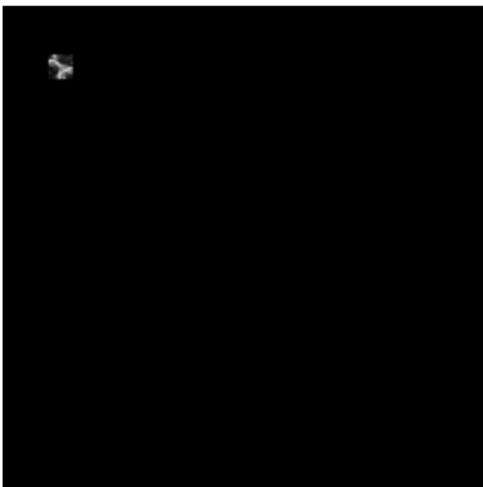
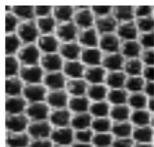
Principe [Efros, Leung 1999]:

- sélectionner un patch qui couvre à la fois la zone connue et la zone manquante.
- trouver dans le germe tous les patches qui lui ressemblent (la similarité est basée sur les pixels connus)
- parmi tous ces patches, en choisir un au hasard, et remplir la zone manquante du patch avec les valeurs de ce patch.
- et continuer...

# Synthèse de texture



# Synthèse de texture



# Inpainting

“Inpainting” : restaurer des parties d’images manquantes



# Conclusion

On trouve du traitement d'image dans

- retouche, restauration d'image
- retrouver des informations (couleur, altitude, inpainting)
- cacher des informations (stéganographie)

Mais on fait aussi traitement d'image "haut-niveau" avec

- découpage en objets (segmentation)
- suivi d'objets (voiture sans conducteur, surveillance)
- détection de certaines caractéristiques (cellules au microscope, dénombrement, chaîne de production)