

Yohann Tendero  
yohann.tendero@telecom-paristech.fr  
<http://perso.telecom-paristech.fr/~ytendero/>

# Plan

- ▶ Couleur : definition, propriétés et mesure
- ▶ Synthèse des couleurs
- ▶ Espaces couleurs
- ▶ Changement de contraste - inversion de contraste
- ▶ Lire si une image est bien contrastée : histogramme, histogramme cumulé
- ▶ Spécification du contraste, égalisation et algorithme midway
- ▶ Dithering
- ▶ Crop d'histogramme
- ▶ Manipulation de contraste et image couleurs

# Couleur

La couleur n'existe pas par elle-même, seul des objets colorés existent<sup>1</sup>.

Trois éléments sont nécessaires pour l'existence d'une couleur :

1. Une source lumineuse, pour éclairer la scène ;
2. Un objet, qui reflète, diffuse, absorbe or diffracte la lumière ;
3. Un capteur ou récepteur qui capture le spectre lumineux réfléchi ou émis par l'objet.

---

1. P. Bonton, C. Fernandez-Maloigne, and A. Trémeau. IMAGE NUMERIQUE COULEUR : de l'acquisition au traitement, volume 2. 2004.

# Couleur

La couleur n'existe pas par elle-même, seul des objets colorés existent<sup>1</sup>.

Trois éléments sont nécessaires pour l'existence d'une couleur :

1. Une source lumineuse, pour éclairer la scène ;
2. Un objet, qui reflète, diffuse, absorbe or diffracte la lumière ;
3. Un capteur ou récepteur qui capture le spectre lumineux réfléchi ou émis par l'objet.

La couleur dépend de ces trois éléments et si, l'un est absent alors nous ne sommes pas capables de percevoir la couleur.

On remarque :

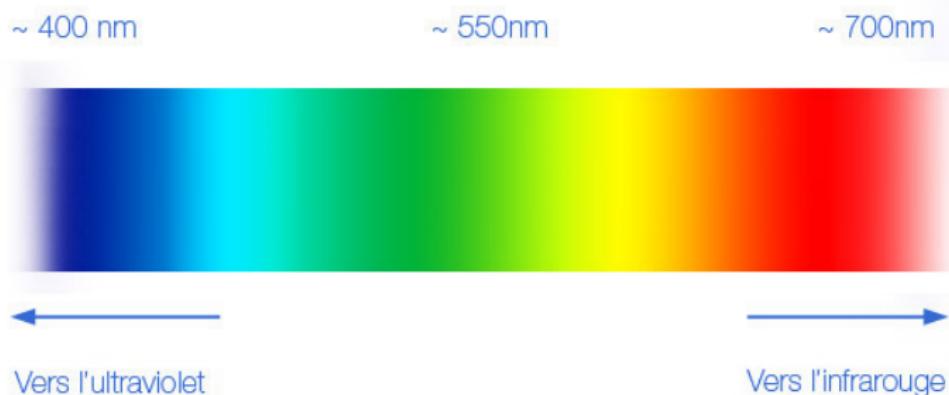
- ▶ la couleur est un attribut physique, à cause de sa dépendance à une source lumineuse et aux caractéristiques physiques des objets ;
- ▶ la couleur dépend de notre perception, et donc, de caractéristiques psychophysiques et physiologiques.

---

1. P. Bonton, C. Fernandez-Maloigne, and A. Trémeau. IMAGE NUMERIQUE COULEUR : de l'acquisition au traitement, volume 2. 2004.

# La lumière

La lumière dépend d'un flux de particules appelées photons qui peuvent aussi être vus comme de petites ondes électromagnétiques<sup>2 3</sup>. Ces ondes doivent avoir une longueur d'onde entre (grossièrement) 400 et 700 nm pour être perçues par notre système visuel.



2. voir aussi expérience des fentes d'Young [https://fr.wikipedia.org/wiki/Fentes\\_d%27Young](https://fr.wikipedia.org/wiki/Fentes_d%27Young)

3. G Wyzecki and WS Stiles. Color science : concepts and methods, quantitative data and formulae. New York, London, Sidney, 1982.

# Définition et mesure de la couleur

La longueur d'onde d'un faisceau lumineux peut être estimée en regardant combien d'énergie est contenue dans une petit intervalle de fréquences et la lumière peut être décrite par la distribution spectrale (ou fréquentielle) de son énergie.

La couleur d'un objet est définie et peut être mesurée par le spectre de la lumière réfléchie.

Lorsque la lumière est réfléchie, elle peut (simultanément) :

- ▶ être absorbée et l'énergie est converties en chaleur (comme lorsque le soleil réchauffe quelque chose) ;
- ▶ passer à travers l'objet (une vitre par exemple) ;
- ▶ réfléchie, comme dans le cas d'un miroir ou de n'importe quel objet coloré.

De plus, un même objet peut avoir différentes couleurs, dépendant de la source lumineuse, de sa géométrie ou de changements de ses caractéristiques physiques<sup>4</sup>.

---

4. [fr.wikipedia.org/wiki/Source\\_lumineuse\\_orthotrope](https://fr.wikipedia.org/wiki/Source_lumineuse_orthotrope)

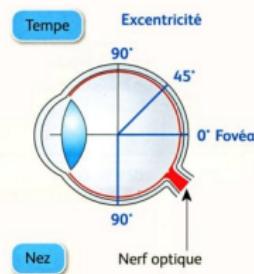
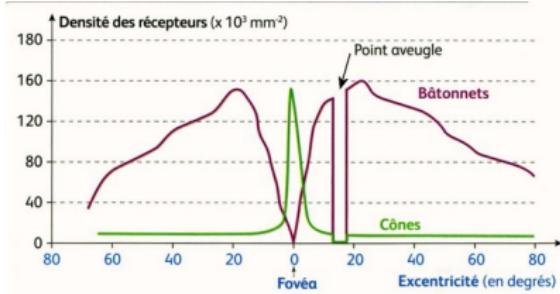
# Capteurs de couleurs

Les capteurs qui permettent de mesurer la couleur peuvent être de différentes natures : un capteur CCD ou CMOS, les yeux, ...

Dans le système visuel humain, l'information relative à la perception de la couleur est encodée selon deux niveaux. Le premier niveau est situé directement sur les récepteurs de la rétine. La rétine comporte deux types de récepteur : des cônes et des bâtonnets. Les bâtonnets ne sont pas sensibles à la couleur : ils se apportent seulement une information sur le niveau d'éclairage un peu à la manière d'une image en niveaux de gris. L'autre type de récepteur présent sur la rétine est appelé "cône" et il en existe trois variantes.

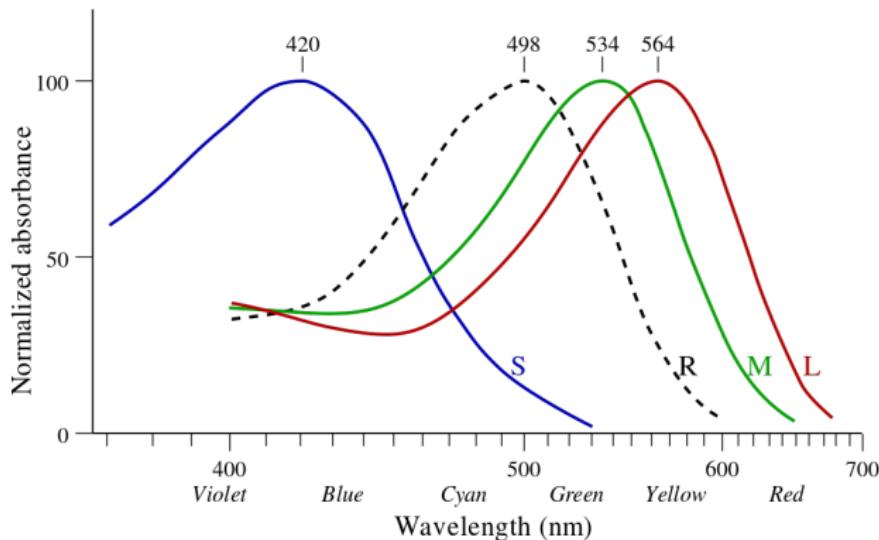
Ce sont les cônes qui sont responsable de notre perception de la couleur.

# Répartition des cones et des batonnets dans l'oeil



# Sensibilité spectrale des trois types de cones

Trois types de cônes (L,M,S) s'occupent des longueurs d'ondes Longues, Moyennes et Courtes ("S" stands for short) qui sont traités lors du deuxième étage du processus.



# Metamerisme, reproduction digitale des couleurs

On notera que notre vision des couleurs n'est pas parfaite : deux longueurs d'ondes (et donc couleurs du point de vue physique) peuvent être perceptuellement indifférentiables : on parle alors de couleurs métamère ou homochromes.

Le métamérisme est ce qui rend l'encodage des couleurs possible : il n'est pas nécessaire de reproduire le spectre exact d'un stimulus coloré : il est suffisant de produire un stimulus qui est visuellement équivalent au stimulus original.

## Vision des couleurs et sensibilité à l'éclairage

Notre système visuel a la capacité de s'adapter à l'intensité ambiante moyenne : si l'intensité moyenne de l'éclairage change (ombre ou en plein soleil) les couleurs ne semblent pas changer pour l'observateur.

Ce phénomène est particulièrement visible pour les blancs, et fait l'objet d'une correction numérique dans toutes les caméras (la balance des blancs).

Une feuille de papier éclairée par différents illuminants (la lumière du jour, une lampe incandescente, un lampe au tungstène, des néons, etc.) "change de couleur" du point de vue physique...

# Vision des couleurs et sensibilité à l'éclairage

Notre système visuel a la capacité de s'adapter à l'intensité ambiante moyenne : si l'intensité moyenne de l'éclaire change (ombre ou en plein soleil) les couleurs ne semblent pas changer pour l'observateur.

Ce phénomène est particulièrement visible pour les blancs, et fait l'objet d'une correction numérique dans toutes les caméras (la balance des blancs).

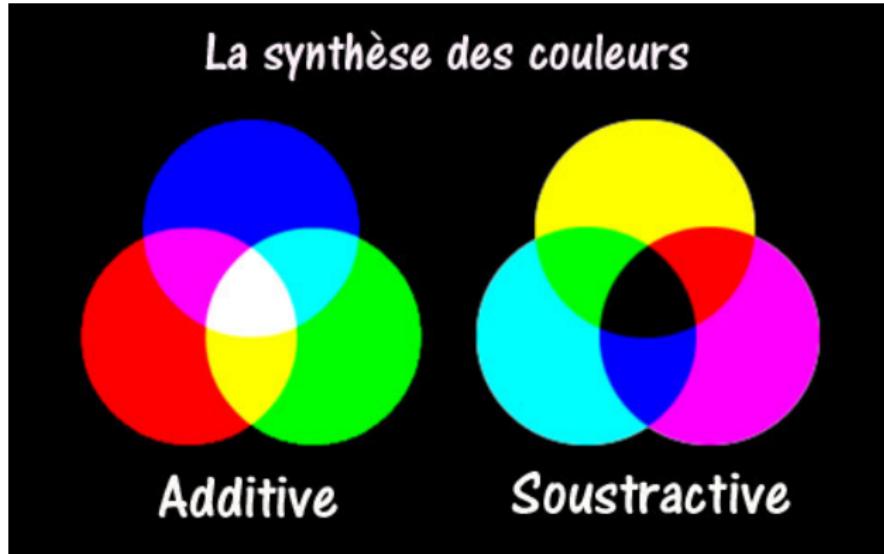
Une feuille de papier éclairée par différents illuminants (la lumière du jour, une lampe incandescente, un lampe au tungstène, des néons, etc.) "change de couleur" du point de vue physique...

mais, nous voyions le papier toujours "blanc". En effet, les cones adaptent leurs réponse au niveau de réflectance de l'environnement. On peut donc en conclure que le mécanisme de perception des contrastes entre couleurs dépend de la proportion relative entre les différentes illuminations dans l'image.

# Plan

- ▶ Couleur : définition, propriétés et mesure
- ▶ Synthèse des couleurs
- ▶ Espaces couleurs
- ▶ Changement de contraste - inversion de contraste
- ▶ Lire si une image est bien contrastée : histogramme, histogramme cumulé
- ▶ Spécification du contraste, égalisation et algorithme midway
- ▶ Dithering
- ▶ Crop d'histogramme
- ▶ Manipulation de contraste et image couleurs

# La synthèse des couleurs

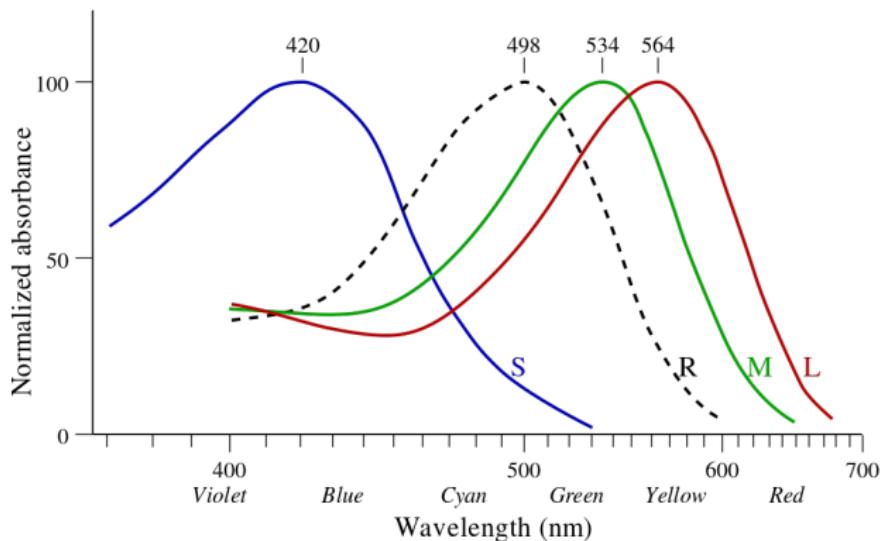


Grosso modo : à gauche (écran ordinateur, télévision) on part du noir et on ajoute des couleurs pour arriver au blanc.

A droite (feuille de papier) : on part du blanc et on soustrait pour arriver au noir.

# Sensibilité spectrale des trois types de cones

Trois types de cônes (L,M,S) s'occupent des longueurs d'ondes Longues, Moyennes et Courtes ("S" stands for short) qui sont traités lors du deuxième étage du processus.



# Plan

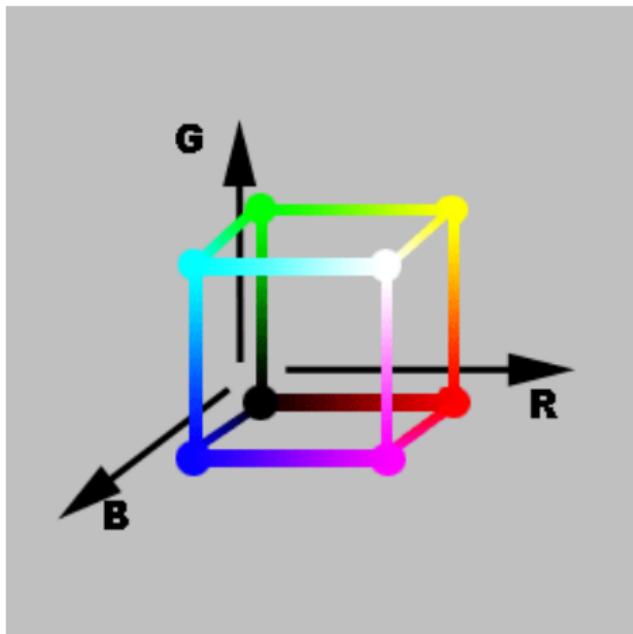
- ▶ Couleur : définition, propriétés et mesure
- ▶ Synthèse des couleurs
- ▶ Espaces couleurs
- ▶ Changement de contraste - inversion de contraste
- ▶ Lire si une image est bien contrastée : histogramme, histogramme cumulé
- ▶ Spécification du contraste, égalisation et algorithme midway
- ▶ Dithering
- ▶ Crop d'histogramme
- ▶ Manipulation de contraste et image couleurs

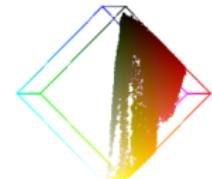
# Encodage (ou représentation numérique) des couleurs, espaces couleurs

Dans la littérature on peut distinguer 4 grandes familles d'espaces couleurs :

- Les modèles inspirés de la physiologie dont l'espace RGB est le plus connu.
- Les modèles colorimétriques, qui sont basés sur la mesure du spectre de réflectance. L'espace CIE en est un exemple.
- Les modèles psycho-physiques qui sont basés sur notre perception des couleurs. Ces modèles sont soit basés sur des critères subjectifs ou sont construits en tenant compte de la vision humaine.
- Les modèles basés sur des couleurs opposées qui sont construits en utilisant des résultats d'expériences qui reposent sur l'observation de deux couleurs opposées.

# Espace couleur RGB

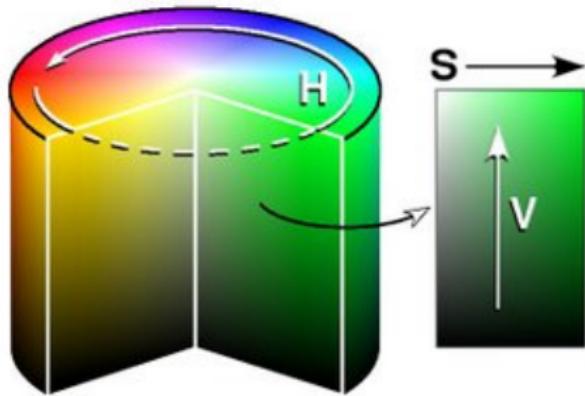




Défauts de l'espace RGB :

- ▶ les couleurs d'une image naturelle sont corrélées et semblent n'occuper qu'une petite partie de cet espace tri-dimensionnel. En fait, les expériences de la figure suggèrent que les couleurs sont réparties suivant un variété (un plan déformé) de dimension 2.
- ▶ la distance (Euclidienne donc) associée à cet espace n'est pas relié à notre perception des couleurs.
- ▶ (difficulté à choisir une couleur).

## Espace HSV (ou HSI)



# Changement d'espace

$$H = \begin{cases} \text{undefined}, & \text{si } MAX = MIN \\ 60 \times \frac{G-B}{MAX-MIN} + 0, & \text{si } MAX = R \\ & \text{et } G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & \text{si } MAX = R \\ & \text{et } G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{si } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{si } MAX = B \end{cases}$$

$H \in [0, 360]$

$$S = \begin{cases} 0, & \text{si } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{sinon} \end{cases}$$

$S, V, R, G, B \in [0, 1]$

$MAX = MAX(R, G, B)$

$MIN = MIN(R, G, B)$

$$V = MAX$$

## Exemple d'utilisation HSV : specification, réhaussement des couleurs



**Gauche** : image originale

**Centre** : diminution de 20% de la saturation

**Droite** : augmentation de 20% de la saturation

# Plan

- ▶ Couleur : définition, propriétés et mesure
- ▶ Synthèse des couleurs
- ▶ Espaces couleurs
- ▶ Changement de contraste - inversion de contraste
- ▶ Lire si une image est bien contrastée : histogramme, histogramme cumulé
- ▶ Spécification du contraste, égalisation et algorithme midway
- ▶ Dithering
- ▶ Crop d'histogramme
- ▶ Manipulation de contraste et image couleurs

# Changements de contraste

## Definition

Soit  $u$  une image (discrète ou non) à valeur dans  $[0, 1]$ . On appelle *changement de contraste* toute fonction : continue, croissante, surjective  $\varphi : [0, 1] \rightarrow [0, 1]$ .

Appliquer un changement de contraste à une image  $u$  c'est calculer  $\varphi \circ u$ , i.e.  $\varphi(u(i, j))$ .

# Exemples classiques

Exemples classiques :

- ▶  $\varphi(x) = x^\gamma$ ,  $\gamma \geq 0$  (correction gamma)<sup>5</sup> ;  
La sensibilité d'un écran CRT n'est pas linéaire, ce changement de  
contraste permettait de "linéariser" la réponse de l'écran.
- ▶ fonctions linéaires, affines
- ▶ log (déjà utilisé pour visualiser le module d'une TFD)
- ▶ défini de manière implicite (p. ex. dépendant du contenu)

---

5. voir aussi [https://fr.wikipedia.org/wiki/Correction\\_gamma](https://fr.wikipedia.org/wiki/Correction_gamma)

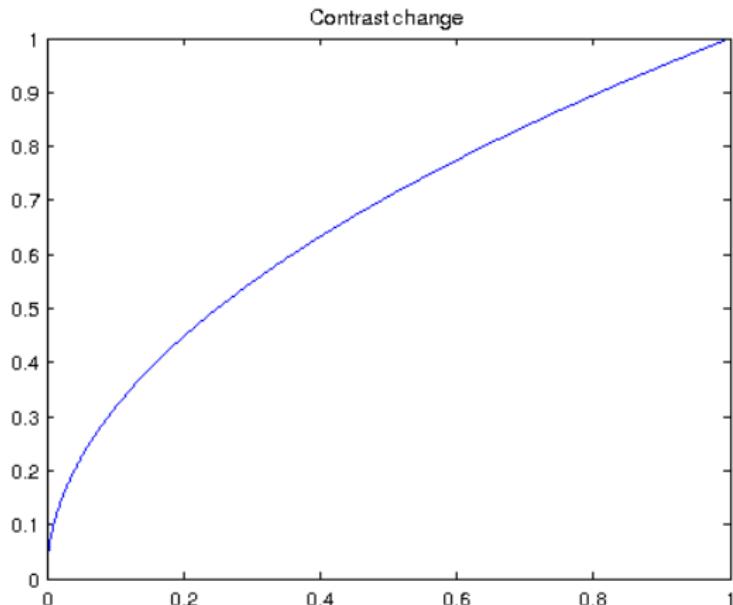
# Image originale



Gamma correction :  $\gamma = 0.6$



## Gamma correction : $\gamma = 0.6$ , fonction $\varphi$



# Image originale



# Changement affine : valeur minimale à 0 et maximale à 1



## Inversion de contraste

# Inversion de contraste

Can you tell who this is?



# Inversion de contraste



## Remarque

Notre système visuel est très robuste aux changements de contraste tels que nous les avons définis.

La lecture d'une image après une inversion de contraste est en revanche plus difficile.

# Plan

- ▶ Couleur : définition, propriétés et mesure
- ▶ Synthèse des couleurs
- ▶ Espaces couleurs
- ▶ Changement de contraste - inversion de contraste
- ▶ Lire si une image est bien contrastée : histogramme, histogramme cumulé
- ▶ Spécification du contraste, égalisation et algorithme midway
- ▶ Dithering
- ▶ Crop d'histogramme
- ▶ Manipulation de contraste et image couleurs

# Histogramme cumulé (Fonction de répartition)

## Definition

Soit  $u$  une image discrète définie sur une grille

$\Omega = \{0, \dots, N-1\} \times \{0, \dots, M-1\}$ . On appelle histogramme cumulé de  $u$  la fonction croissante

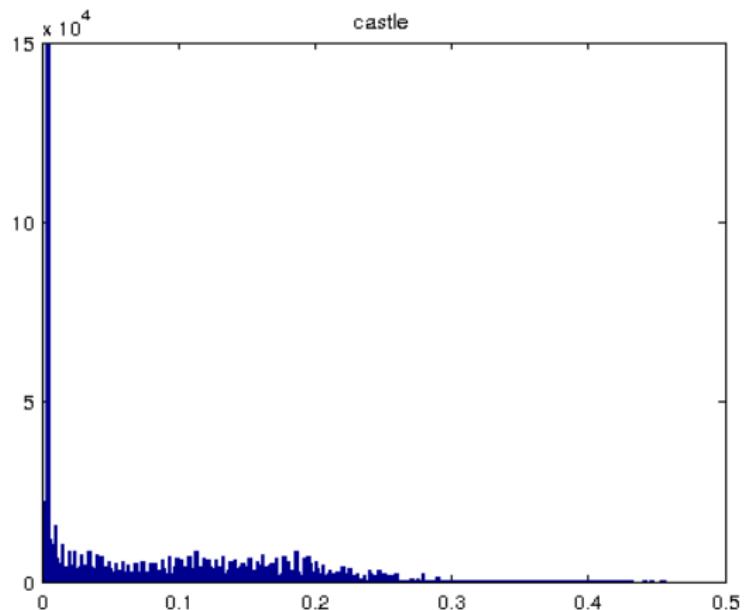
$$H_u(\lambda) = \frac{1}{|\Omega|} |\{(i, j) \in \Omega : u(i, j) \leq \lambda\}|,$$

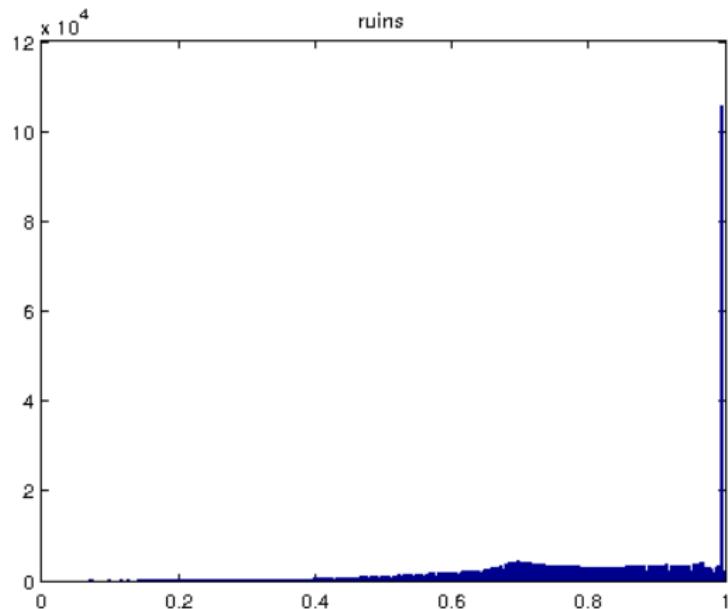
et  $|X|$  représente la cardinalité de l'ensemble discret fini  $X$ .

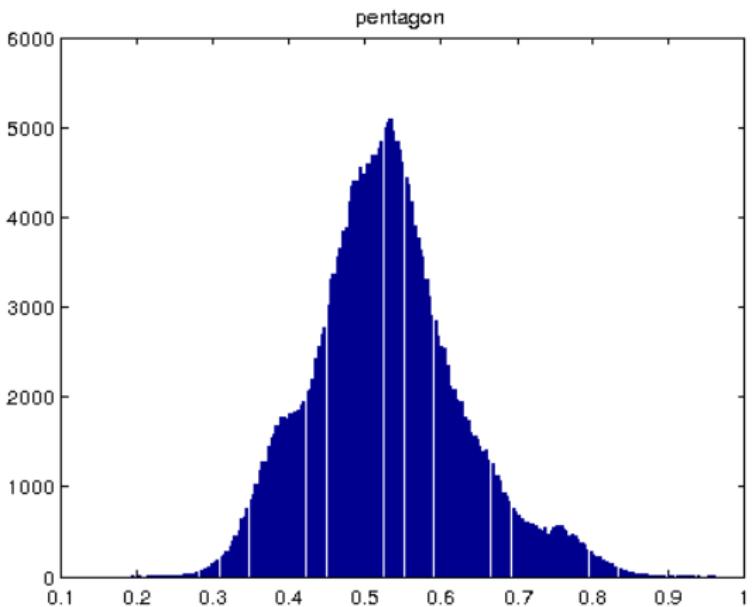
# Histogramme (Distribution)

Si  $u : \Omega \rightarrow \{0, \dots, L\}$  l'histogramme discret  $h_u$  de  $u$  est la dérivée discrète de  $H_u$  :

$$\forall I \in \{0, \dots, L\} \quad h_u(I) = \frac{1}{|\Omega|} |\{(i, j) \in \Omega : u(i, j) = I\}|.$$







castle



ruins



pentagon



# Plan

- ▶ Couleur : définition, propriétés et mesure
- ▶ Synthèse des couleurs
- ▶ Espaces couleurs
- ▶ Changement de contraste - inversion de contraste
- ▶ Lire si une image est bien contrastée : histogramme, histogramme cumulé
- ▶ Spécification du contraste, égalisation et algorithme midway
- ▶ Dithering
- ▶ Crop d'histogramme
- ▶ Manipulation de contraste et image couleurs

# Pourquoi a t-on besoin de manipuler le contraste des images ?

La comparaison de deux images est une tâche importante en vision par ordinateur et en traitement d'images.

La comparaison d'image est en général beaucoup plus robuste et efficace si on peut la rendre invariante aux changements de contraste car

- éclairage différents,
- capteurs différents (la réponse d'un capteur est en général non linéaire)

On a donc besoin de manipuler, de manière non-linéaire, le contraste des images.

# Prescrire (spécifier) le contraste

Comment donner à un image un histogramme que l'on a choisi ?

Exemples classiques :

- Donner à une image un histogramme le plus uniforme possible (égalisation) ;
- Donner à une image l'histogramme d'une autre ;
- Donner à deux (ou plus) image un histogramme commun, "moyenne de leur histogrammes".

# Inverse à gauche, inverse à droite

Si  $\varphi$  est strictement croissante, alors  $\varphi$  est inversible. Sinon, on utilise un pseudo-inverse

## Definition

Le pseudo-inverse (à gauche)  $\varphi_g^{(-1)}$  de tout changement de contraste  $\phi$  est défini par

$$\varphi_g^{(-1)}(\lambda) = \inf\{r \in \mathbb{R} : \varphi(r) \geqslant \lambda\}.$$

Le pseudo-inverse (à droite)  $\varphi_d^{(-1)}$  de tout changement de contraste  $\phi$  est défini par

$$\varphi_d^{(-1)}(\lambda) = \sup\{r \in \mathbb{R} : \varphi(r) \leqslant \lambda\}.$$

# Inverse à gauche, inverse à droite

Si  $\varphi$  est strictement croissante, alors  $\varphi$  est inversible. Sinon, on utilise un pseudo-inverse

## Definition

Le pseudo-inverse (à gauche)  $\varphi_g^{(-1)}$  de tout changement de contraste  $\phi$  est défini par

$$\varphi_g^{(-1)}(\lambda) = \inf\{r \in \mathbb{R} : \varphi(r) \geqslant \lambda\}.$$

Le pseudo-inverse (à droite)  $\varphi_d^{(-1)}$  de tout changement de contraste  $\phi$  est défini par

$$\varphi_d^{(-1)}(\lambda) = \sup\{r \in \mathbb{R} : \varphi(r) \leqslant \lambda\}.$$

En pratique, souvent  $\varphi$  est strictement croissante et  $\varphi_g^{(-1)} = \varphi_d^{(-1)}$ .

# Specification ou prescription d'histogramme

## Definition

Soit  $G$  un histogramme cumulatif, on appelle spécification de  $u$  sur  $G$  l'image

$$\tilde{u} := G_d^{(-1)} \circ H_u(u).$$

- ▶ Si on prend  $G = H_v$  où  $H_v$  représente l'histogramme cumulatif de l'image  $v$ , alors  $\tilde{u}$  est appelée spécification de  $u$  sur  $v$ .
- ▶ Si on prend  $G$  l'histogramme cumulatif de la distribution uniforme  $\tilde{u}$  est appelée égalisation d'histogramme.

# Remarques

- ▶ A partir de ces formules on peut calculer de combien les niveaux de gris sont "écartés" ou, au contraire, "rapprochés".
- ▶ Ces formules sont indentiques à celles utilisées pour générer des réalisations de variables aléatoire. Exemple : ordinateur génère des  $\mathcal{U}[0, 1]$  (on peut calculer l'histogramme cumule empirique) et on souhaite obtenir une variable gaussienne (histogramme cumulé spécifié).
- ▶ On en déduit aussi l'algorithme à appliquer

# Algorithme égalisation

1. Calculer histogramme cumulé  $H_u$
2.  $\varphi(x) = (\text{nb niveaux} - 1)H_u(x)$

# Algorithme égalisation

1. Calculer histogramme cumulé  $H_u$
2.  $\varphi(x) = (\text{nb niveaux} - 1)H_u(x)$   
En effet, pour  $H_{cible} = Id$  (histogramme cumulé d'une distribution uniforme) son inverse est... lui même.
3. (On verra comment spécifier une image sur une autre comme cas particulier plus loin.)

# Egalisation

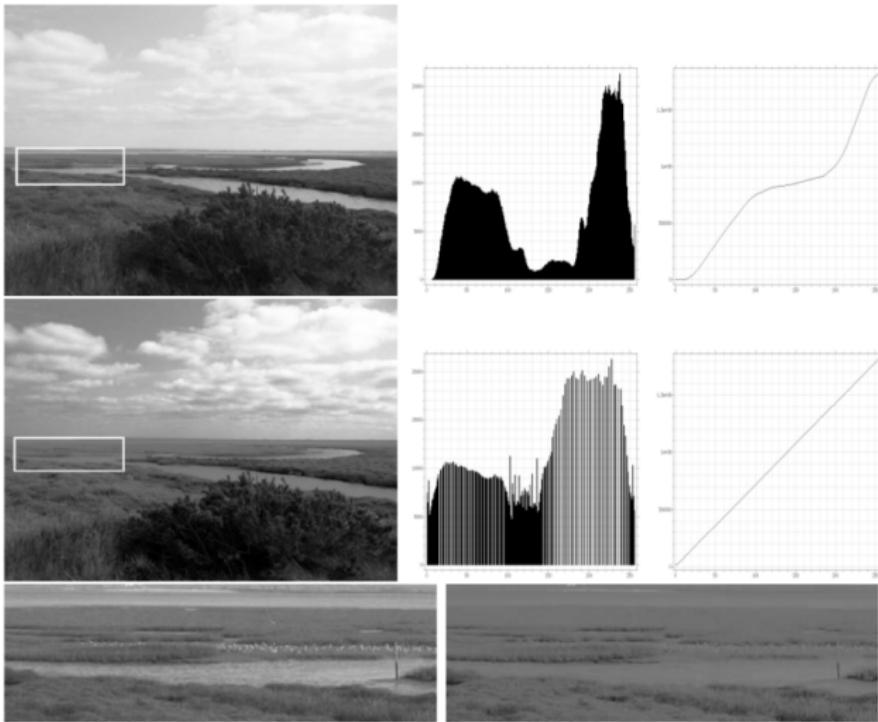
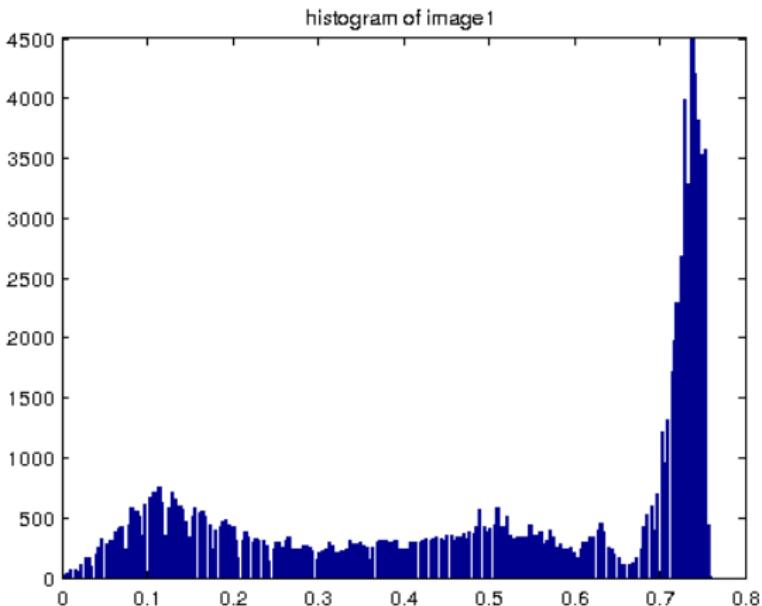


image 1





histogram of image1 after equalization

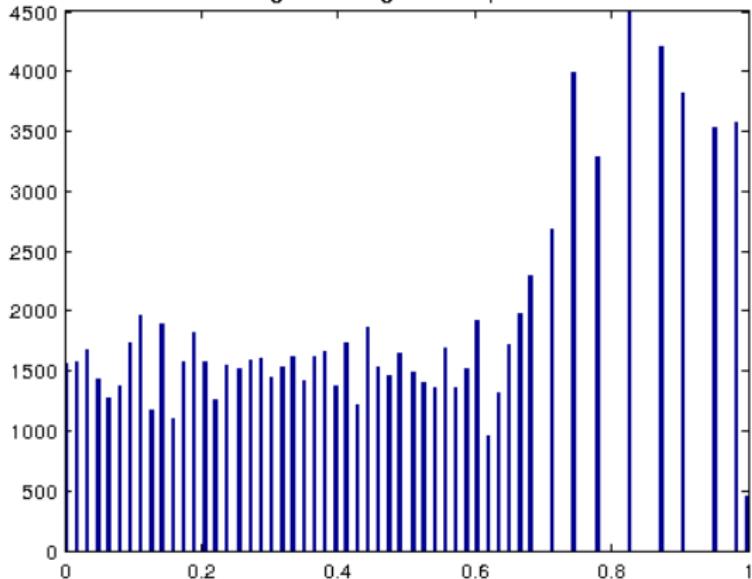


Image1 after histogram equalization





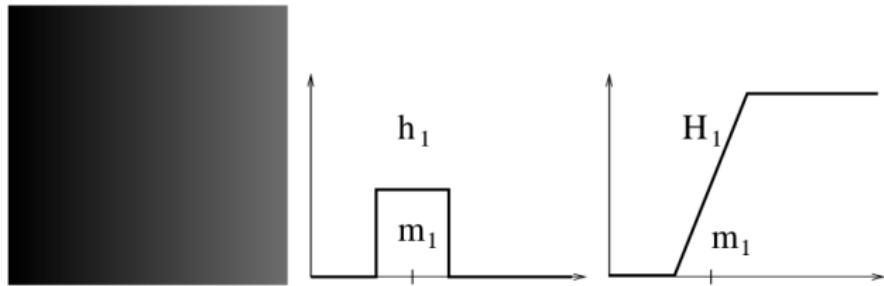


## Normaliser le contraste : "The midway method"

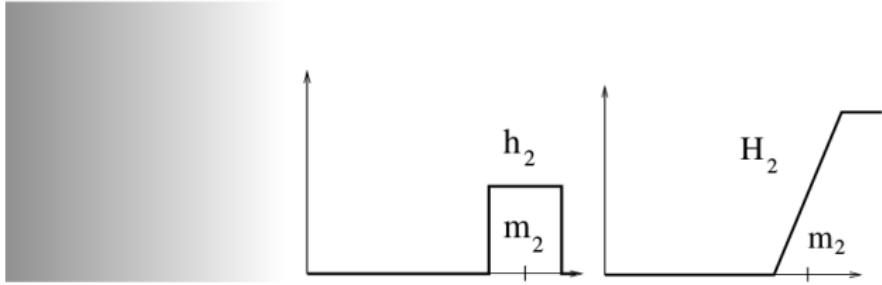
Soit deux images  $u_1$  et  $u_2$  et leurs histogrammes cumulatifs  $H_1$  et  $H_2$ .  
L'histogramme moyen est donné par (moyenne harmonique)

$$H_{moy} := \left( \frac{H_1^{-1} + H_2^{-1}}{2} \right)^{-1}$$

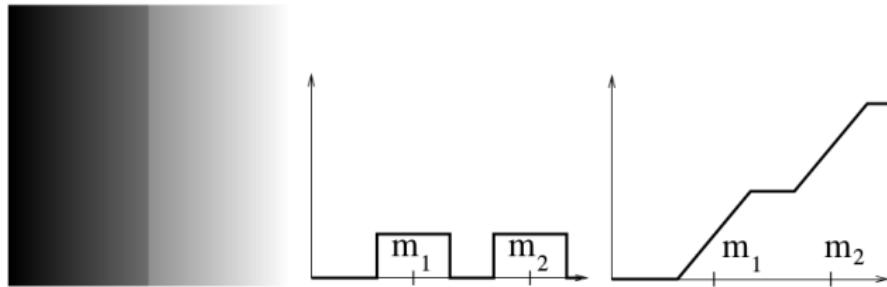
On spécifie  $u_1$  et  $u_2$  sur  $H_{moy}$ .



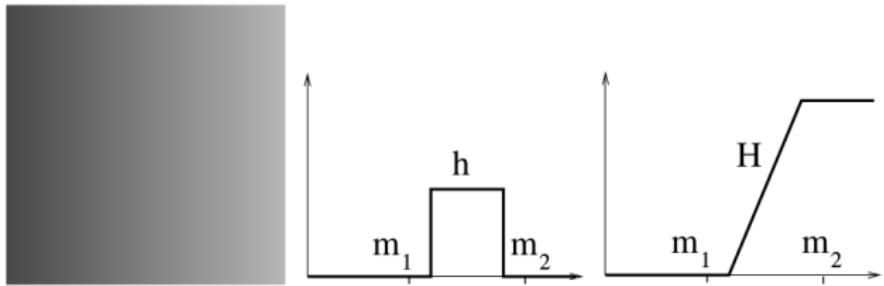
(a) Image  $u_1$ , its histogram  $h_1$  and its cumulative histogram  $H_1$ .



(b) Image  $u_2$ , its histogram  $h_2$  and its cumulative histogram  $H_2$ .



(c) Resulting image by averaging the histograms  $h_1$  and  $h_2$ .



(d) Resulting image by applying a *midway* equalization

image 1



image2

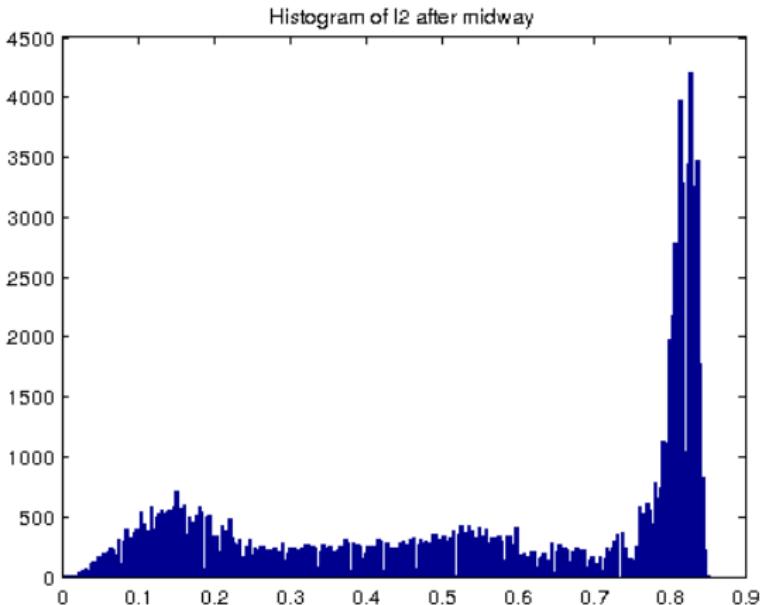


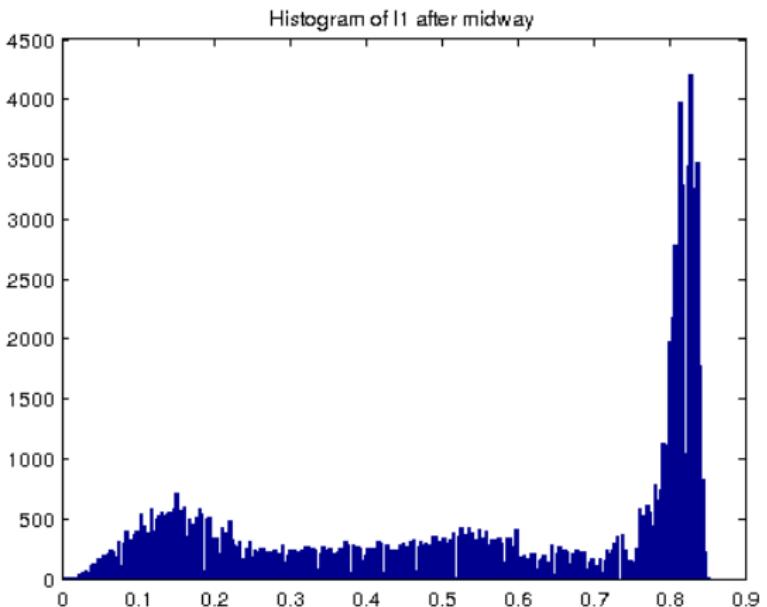
|1 after midway



12 after midway











# Implémentation matlab, cas deux image de même taille

```
u_midway_1 = zeros(size(u_1));
u_midway_2 = zeros(size(u_2));

[u_sort_1,index_u_1] = sort(u_1(:));
[u_sort_2,index_u_2] = sort(u_2(:));
u_midway_1(index_u_1) = (u_sort_1 + u_sort_2)/2;
u_midway_2(index_u_2) = (u_sort_1 + u_sort_2)/2;

u_midway_1 = reshape(u_midway_1 ,size(u_1));
u_midway_2 = reshape(u_midway_2 ,size(u_2));
```

# Implémentation, cas général

$H_p$  le histogrammes cumulés de  $u_1, u_2$ . Images à  $L$  niveaux de gris.

```
// Computation of the contrast function f12 using a lookup table
9 for each bin position k ∈ {0, ..., L} do
10    initialize bin position l = 0
11    while H1(k) > H2(l) do
12      | l = l + 1;           // Find the position l where the values H1(k) and H2(l) are equal
13    end
14    | f12(k) = ½(k + l)
15 end

// Computation of the contrast function f21 using a lookup table
16 for each bin position k ∈ {0, ..., L} do
17    initialize bin position l = 0
18    while H2(k) > H1(l) do
19      | l = l + 1;           // Find the position l where the values H2(k) and H1(l) are equal
20    end
21    | f21(k) = ½(k + l)
22 end

// Application of the contrast function f12 to u1 to compute ū1
23 for each pixel position (i, j) ∈ {1, ..., N1} × {1, ..., M1} do
24  | ū1(i, j) = f12(u1(i, j))
25 end

// Application of the contrast function f21 to u1 to compute ū1
26 for each pixel position (i, j) ∈ {1, ..., N2} × {1, ..., M2} do
27  | ū2(i, j) = f21(u2(i, j))
28 end
```



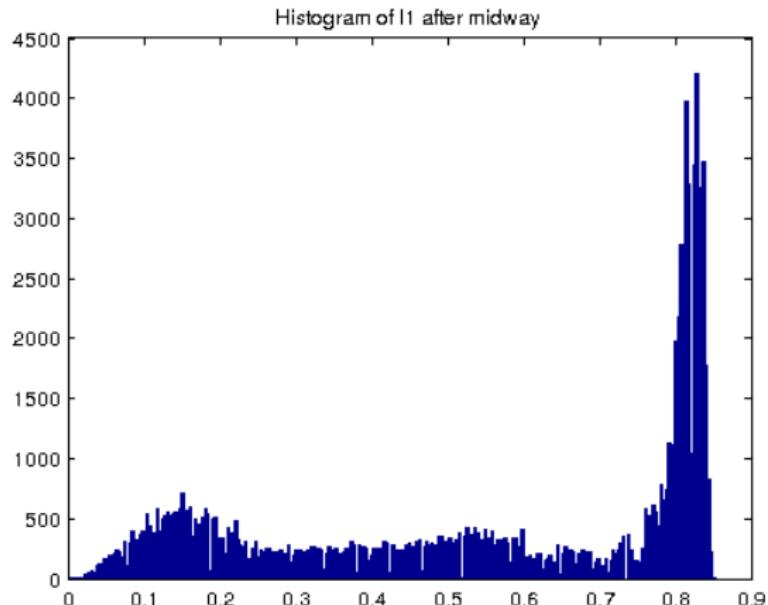
# Plan

- ▶ Couleur : définition, propriétés et mesure
- ▶ Synthèse des couleurs
- ▶ Espaces couleurs
- ▶ Changement de contraste - inversion de contraste
- ▶ Lire si une image est bien contrastée : histogramme, histogramme cumulé
- ▶ Spécification du contraste, égalisation et algorithme midway
- ▶ Dithering
- ▶ Crop d'histogramme
- ▶ Manipulation de contraste et image couleurs

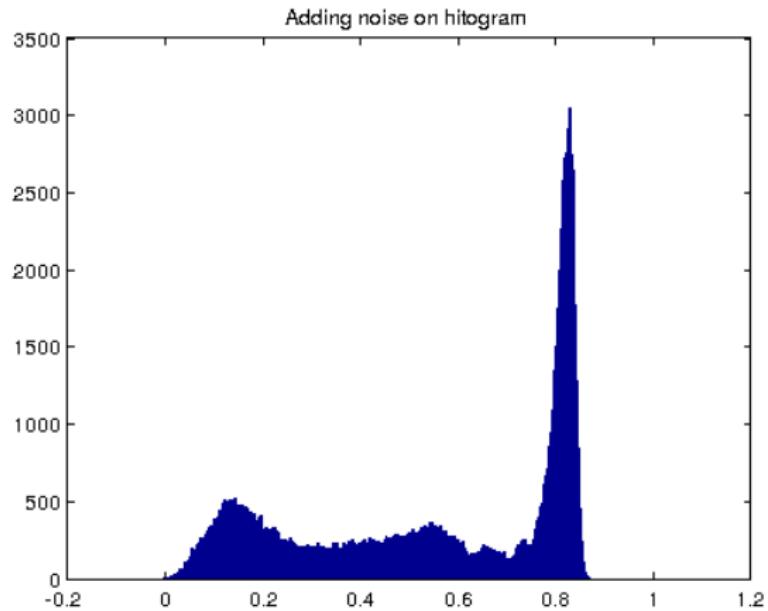
# Dithering



# Dithering



# Dithering



Le "dithering" propose de corriger cet effet en ajoutant du bruit avant de traiter l'image avant de modifier. Le bruit (demo matlab) permet de lisser l'histogramme. Le résultat est en général visuellement acceptable.

# Avant



Apres



# Plan

- ▶ Couleur : définition, propriétés et mesure
- ▶ Synthèse des couleurs
- ▶ Espaces couleurs
- ▶ Changement de contraste - inversion de contraste
- ▶ Lire si une image est bien contrastée : histogramme, histogramme cumulé
- ▶ Spécification du contraste, égalisation et algorithme midway
- ▶ Dithering
- ▶ Crop d'histogramme
- ▶ Manipulation de contraste et image couleurs

## Quand un changement affine ne suffit pas

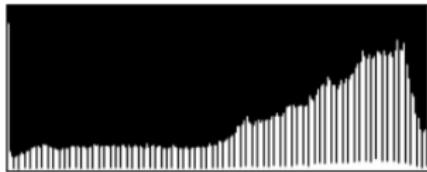
Pour ré-hausser le contraste d'une image on peut immédiatement penser à un changement affine, i.e.

$\varphi$  tel que le minimum de l'image soit 0 et le maximum 1.

Cette approche peut ne pas suffire : un seul pixel blanc et un seul pixel noir suffit à rendre l'approche inopérante.

Une alternative est de laisser saturer  $\varepsilon$  % des pixels les plus saturés.

## Exemple



# Plan

- Couleur : definition, propriétés et mesure
- Synthèse des couleurs
- Espaces couleurs
- Changement de contraste - inversion de contraste
- Lire si une image est bien contrastée : histogramme, histogramme cumulé
- Spécification du contraste, égalisation et algorithme midway
- Dithering
- Crop d'histogramme
- Manipulation de contraste et image couleurs

# Le problème des images couleurs

Pour normaliser le contraste de deux images couleurs proprement : il nous faut une théorie du transport optimal dont nous ne disposons pas encore.

- ▶ On peut faire des choses "canal par canal", mais ce n'est en général pas la solution optimale => on peut voir apparaître des artefacts (en général des changements de couleurs désagréables).
- ▶ On peut tenter utiliser l'espace HSV et n'appliquer le changement à  $V$ .

Choix : fortement dépendant de l'application.

# Image 1 (canal par canal)



## Image 2 (canal par canal)



# midway image 1 (canal par canal)



## midway image 2 (canal par canal)



# Image 1 (canal par canal)



## Image 2 (canal par canal)



# midway image 1 (canal par canal)



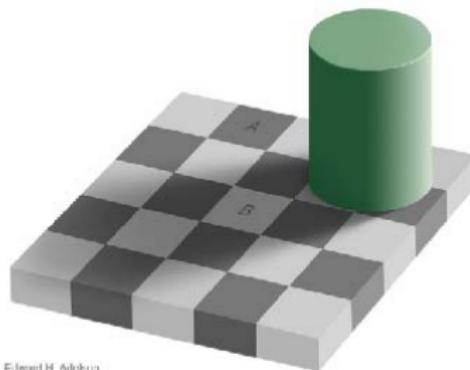
## midway image 2 (canal par canal)



## Slides supplémentaires

# Retinex

In 1964 Edwin H. Land formulated the Retinex theory. It is, to the best of my knowledge the first attempt to simulate and explain how the humans perceives color. This theory tries to explain why we perceive colors more or less independently of the shading as illustrated below



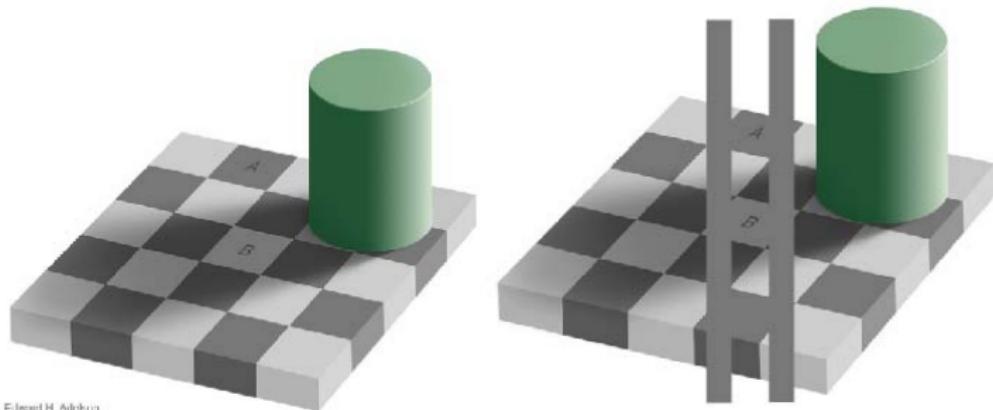
Edmund H. Land 1964

**The original image of the illusion.**

The squares marked A and B  
are the same shade of gray,  
yet they appear different.

# Retinex

In 1964 Edwin H. Land formulated the Retinex theory. It is, to the best of my knowledge the first attempt to simulate and explain how the humans perceives color. This theory tries to explain why we perceive colors more or less independently of the shading as illustrated below



Edmund H. Land's illusion

## The original image of the illusion.

The squares marked A and B are the same shade of gray, yet they appear different.

## The original image plus two stripes.

By joining the squares marked A and B with two vertical stripes of the same shade of gray, it becomes apparent that both squares are the same.

## Détection de contours : paradigme

Si un objet clair est photographié sur un fond noir on peut s'attendre à ce que la silhouette de l'objet corresponde aux points où le gradient de l'image est une réponse impulsionnelle 1D g. fort, i.e., les zones où l'intensité lumineuse varie rapidement. L'algorithme chercherait donc simplement à identifier les zones pour lesquelles l'image  $u$  satisfait  $|\nabla u|$  grand.

# Détection de contours

Cette conclusion n'est pas réaliste :

- a A cause du bruit le gradient peut être grand, sans qu'il y ait un edge à cet endroit ;
- b Les points où  $|\nabla u|$  est grand a de grandes chances de ne pas former une courbe mais des régions entières

Pour répondre à (a) il a été proposé de lisser ("smooth")  $u$  préalablement au calcul de  $|\nabla u|$ . C'est en général fait à l'aide d'une Gaussienne.

Pour répondre à (b), il a été proposé de redéfinir les edges. Au lieu de dire qu'un edge est un point où  $|\nabla u|$  dépasse un certain seuil, on demande aussi au gradient de satisfaire une propriété de maximalité qui conduit à traquer les passages par zéro du Laplacien.

# Détection de contours : premier ordre (Canny)

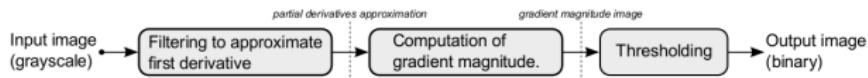


Figure 1: Block diagram of first derivative edge detection algorithms.

# Détection de contours : algorithmes du premier ordre

---

**Algorithm 1** First derivative edge detection algorithms.

Require:  $im$  input image, threshold  $th$ .

```
1: Define  $operator_x$  and  $operator_y$                                 ▷ Roberts, Prewitt or Sobel.  
2:  $g_x \leftarrow \text{convolution}(im, operator_x)$   
3:  $g_y \leftarrow \text{convolution}(im, operator_y)$   
4:  $max_M \leftarrow 0$   
5: for all pixel  $i$  in image do  
6:    $M[i] \leftarrow \sqrt{g_x^2 + g_y^2}$                                          ▷ Gradient magnitude.  
7:   if  $M[i] > max_M$  then  
8:      $max_M \leftarrow M[i]$   
9:   end if  
10:  end for  
11:  for all pixel  $i$  in image do  
12:    if  $M[i] \geq th \times max_M$  then  
13:       $edges[i] \leftarrow 1$   
14:    else  
15:       $edges[i] \leftarrow 0$   
16:    end if  
17:  end for  
18: return  $edges$ 
```

---

# Détection de contours : second ordre

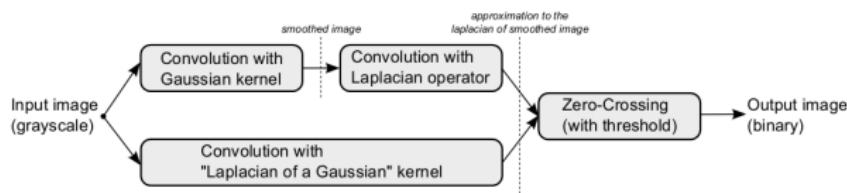


Figure 8: Block diagram of the Marr-Hildreth algorithm.

# Détection de contours : second ordre

---

**Algorithm 2** Marr-Hildreth edge detection algorithm.

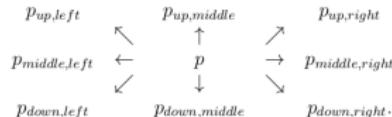
Require:  $im$  input image, standard deviation  $\sigma$ , kernel size  $n$  and zero-crossing threshold  $t_{ZC}$ .

```
1: if Gaussian kernel then
2:   kernel ← generate_gaussian.kernel( $n, \sigma$ )
3:    $im_{smooth}$  ← convolution( $im, kernel$ )
4:   Define Laplacian operator  $laplacian$                                 ▷ See mask definition in Equation (3.3)
5:    $im_{lap}$  ← convolution( $im_{smooth}, laplacian$ )
6: else
7:   log_kernel ← generate_log.kernel( $n, \sigma$ )
8:    $im_{lap}$  ← convolution( $im, log\_kernel$ )
9: end if
10:  $max_L \leftarrow 0$ 
11: for all pixel  $i$  in image  $im_{lap}$  do
12:   if  $im_{lap}[i] > max_L$  then
13:      $max_L \leftarrow im_{lap}[i]$ 
14:   end if
15: end for
16: for all pixel  $i$  in image  $im_{lap}$ , except borders do
17:   for all pair  $(p_1, p_2)$  of opposite neighbors of  $p$  in  $im_{lap}$  do
18:     if ( $\text{sign}(im_{lap}[p_1]) \neq \text{sign}(im_{lap}[p_2])$ ) and ( $|im_{lap}[p_1] - im_{lap}[p_2]| > th_{ZC}$ ) then
19:        $edges[i] \leftarrow 1$ 
20:     else
21:        $edges[i] \leftarrow 0$ 
22:     end if
23:   end for
24: end for
25: return  $edges$ 
```

---

# Détection de contours : second ordre

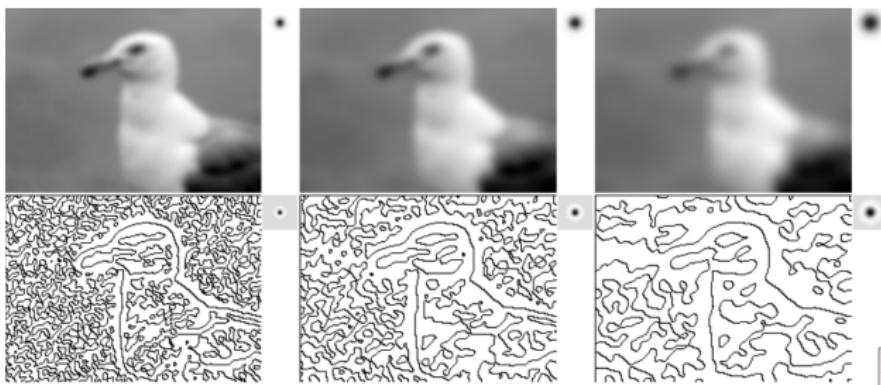
The zero-crossing threshold ( $th_{ZC}$ ) is given as a percentage of the maximum value  $max_L$  of the Laplacian image for both Gaussian and LoG kernels. Each pixel  $p$  has eight neighbors, named according to their position as follows:



Then a pixel  $p$  is considered an edge pixel if any of the following conditions is true (for simplicity the Laplacian image is denoted as  $\mathcal{L}$ ):

- $(\text{sign}(\mathcal{L}[p_{up, left}]) \neq \text{sign}(\mathcal{L}[p_{down, right}]) \& |\mathcal{L}[p_{up, left}] - \mathcal{L}[p_{down, right}]| > th_{ZC} * max_L$
- $(\text{sign}(\mathcal{L}[p_{up, middle}]) \neq \text{sign}(\mathcal{L}[p_{down, middle}]) \& |\mathcal{L}[p_{up, middle}] - \mathcal{L}[p_{down, middle}]| > th_{ZC} * max_L$
- $(\text{sign}(\mathcal{L}[p_{down, left}]) \neq \text{sign}(\mathcal{L}[p_{up, right}]) \& |\mathcal{L}[p_{down, left}] - \mathcal{L}[p_{up, right}]| > th_{ZC} * max_L$
- $(\text{sign}(\mathcal{L}[p_{middle, left}]) \neq \text{sign}(\mathcal{L}[p_{middle, right}]) \& |\mathcal{L}[p_{middle, left}] - \mathcal{L}[p_{middle, right}]| > th_{ZC} * max_L$ .

# Détection de contours : zero crossings of the Laplacian across different scales



Le terme "edge detection" semble être apparu en 1959, en relation avec un problème de transmission télé (B. Julesz, "A method of coding TV signals based on edge detection", 1959)

L'idée que le calcul des dérivées d'une image requiert un filtrage passe bas préalable est étudiée dans p.ex L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever, "Linear scale-space", 1994 et J. Blom, B. T. te Haar Romeny, A. Bel, and J. J. Koenderink, "Spatial derivatives and the propagation of noise in Gaussian scale-space", 1991 ou les livres "L. M. J. Florack, "Image Structure", 1997 ;

T. Lindeberg, Scale-Space Theory in Computer Vision, 1993.

Le détecteur d'Haralick (R. M. Haralick, Digital step edges from zero-crossing of second directional derivatives, 1984), implémenté par Canny (J. F. Canny, "A computational approach to edge detection", 1986) est probablement un des opérateur d'analyse d'image les plus connus.