Hiver 2018

Analyse d'images
IMN 259

Opérations ponctuelles

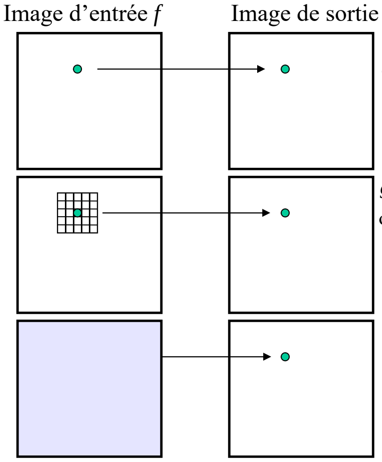
Par
Pierre-Marc Jodoin

Avant de commencer

Qu'est-ce qu'une opération ponctuelle?

Soit un opérateur T() qui transforme une (ou plusieurs) intensités de gris/couleurs en une nouvelle intensité de gris/couleur.

Image d'entrée f — Image de sortie g



Opération ponctuelle $g(x,y) = T(f(x,y))$

Opération locale $g(x,y) = T(\{f(x_i,y_i)\})$ où (x_i,y_i) est un pixel voisin de (x,y)

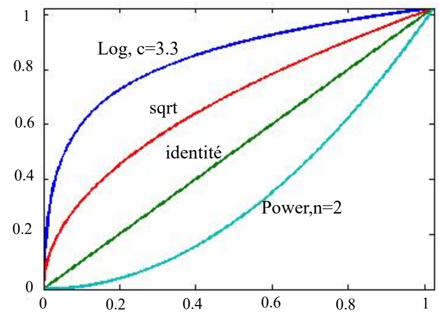
Opération globale $g(x,y) = T(\{f(x_i,y_i)\})$ où (x_i,y_i) désigne tous les pixels dans l'image.

Transformations simples

$$g(x,y) = c\log_{10}(1+f(x,y))$$
$$g(x,y) = \sqrt{f(x,y)}$$
$$g(x,y) = (f(x,y))^n$$

*T(**f(x,y)**)*

Niveaux de gris des pixels de l'image de sortie ***g(x,y)***



Niveaux de gris des pixels de l'image d'entrée ***f(x,y)***

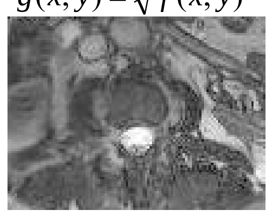
Dans ce contexte, T est parfois appelée « fonction de transfert ».

Note : les niveaux de gris ont été **<u>normalisés entre 0 et 1</u>**.

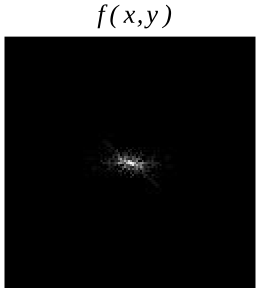
Transformations simples

$f(x,y)$

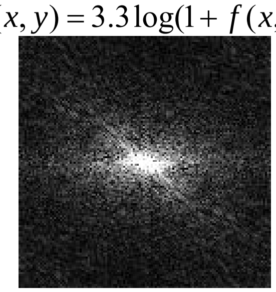
$g(x,y) = \sqrt{f(x,y)}$



$f(x,y)$

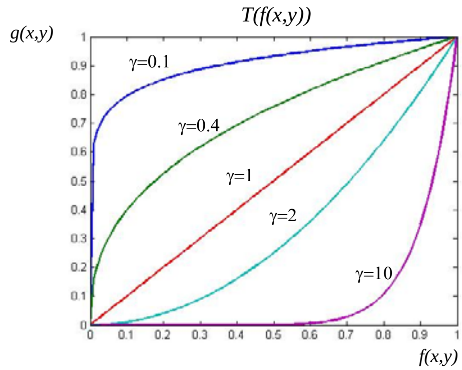


$g(x,y) = 3.3\log(1+f(x,y))$



Correction gamma

$$g(x,y)=\left(f(x,y)\right)^{\gamma}$$



Note : les niveaux de gris ont été normalisés entre 0 et 1.

5

Correction gamma

$$\boxed{g(x,y)=\left(f(x,y)\right)^{\gamma}}$$

γ=0.1

γ=0.4

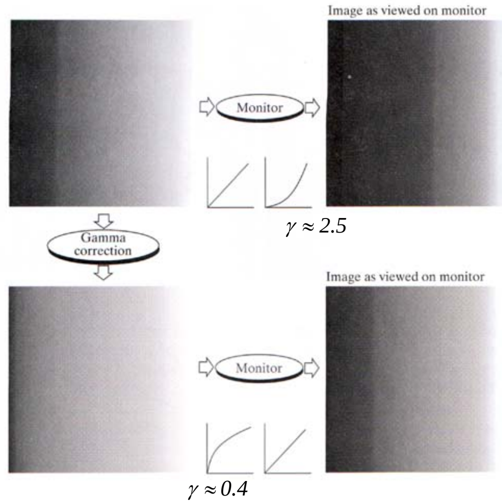
γ=1

γ=2

6

Correction gamma

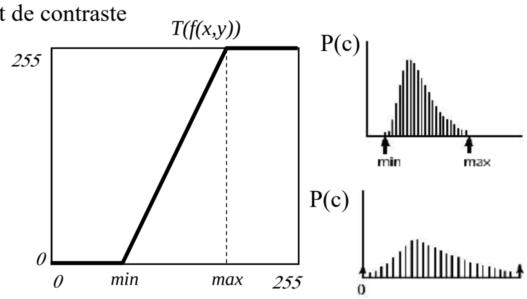
La correction gamma est une opération fort utile pour compenser des distortions de matériel (imprimante, écran, …)



Gonzalez-Woods

Transformation linéaire

Rehaussement de contraste



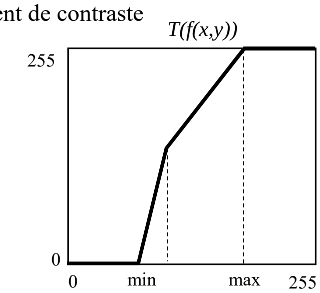
$$g(x,y) = \frac{255}{max - min}(f(x,y) - min)$$

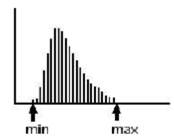
f(x,y)

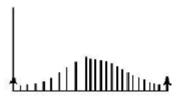
g(x,y)

Transformation linéaire par morceaux

Rehaussement de contraste





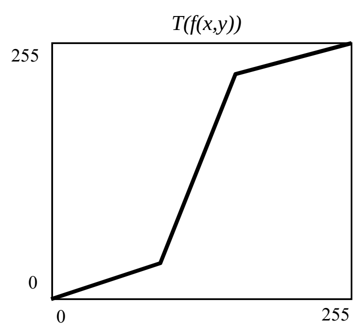


f(x,y)

g(x,y)

Transformations linéaires par morceaux

Rehaussement de contraste



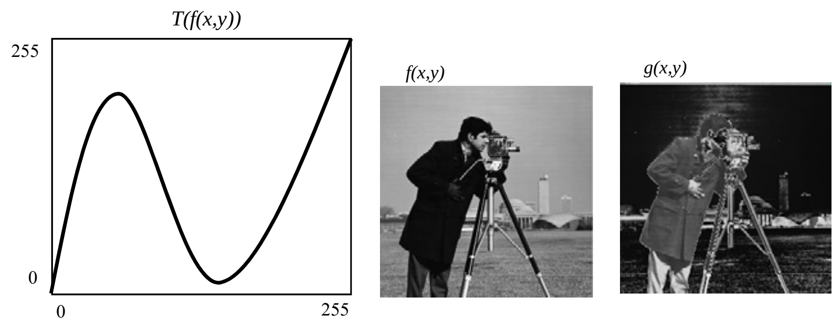
f(x,y)

g(x,y)

Pour accélérer (et simplifier) les calculs, on peut utiliser une *Look-up table (LUT):*

f(x,y)	*T(f(x,y))*
0	0
1	0.8
2	1.7
3	2.5
(…)	(…)
254	254.8
255	255

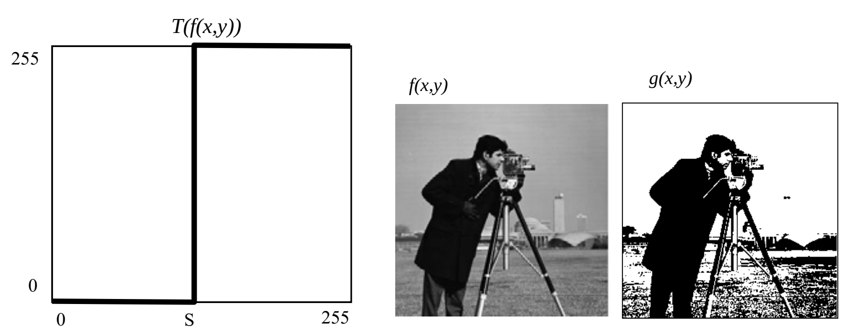
Transformations non linéaires



Remarque: Pente négative = inversion locale de contraste

Transformations linéaires par morceaux

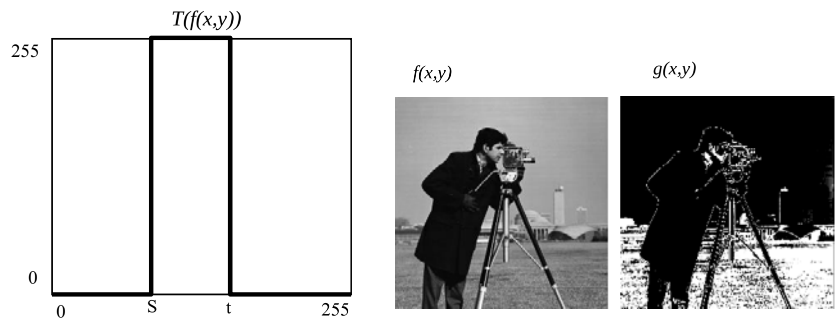
Seuil



Remarque: les sauts verticaux entraînent des discontinuités

Transformations linéaires par morceaux

Slicing



13

Traitement de l’histogramme

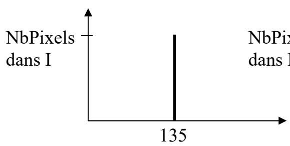
14

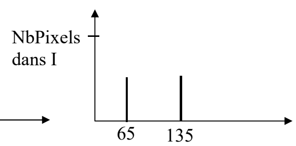
Traitement de l’histogramme

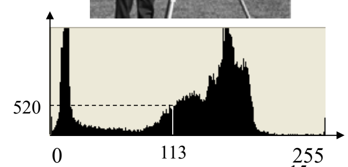
Définitions

Un **histogramme** représente le nombre de pixels appartenant à chaque niveaux de gris (ou couleur) pouvant être représenté dans l’image.

$$H(c) = \text{Nb pixels d'intensité "}c\text{"}$$







15

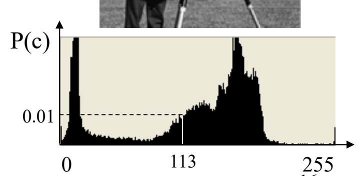
Traitement de l’histogramme

Parfois l’histogramme est normalisé par le nombre de pixels dans l’image:

$$H'(c) = P(c) = \frac{\text{Nb pixels d'intensité c}}{\text{Nb total de pixels dans l'image}}$$

Ainsi défini, *P(c)* donne une idée de la probabilité d’occurrence d’un pixel de niveau de gris “c”.

$$\sum_{c=0}^{255} P(c) = 1$$



Si je tire un pixel au hasard dans l’image, j’ai 1% de chance qu’il soit d’intensité 113.

16

Traitement de l’histogramme

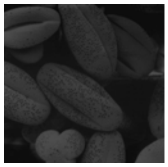


Image foncée

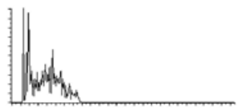
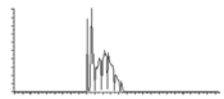


Image peu contrastée



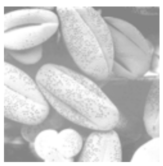


Image claire

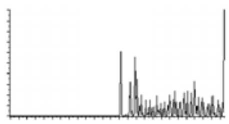
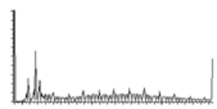


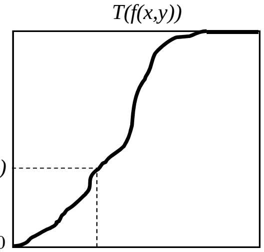
Image fortement contrastée

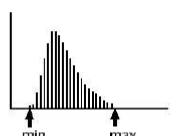


Gonzalez-Woods 17

Traitement de l’histogramme

Égalisation d’histogramme



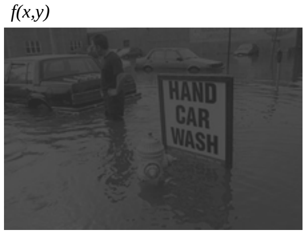


Soit *P(i),* l’histogramme normalisée (ou si vous préférez la *densité de probabilité*) de l’image *f(x,y)*. Pour effectuer une égalisation d’histogramme, la fonction de transfert prend la forme d’une *densité de probabilité cumulative*

$$T(a) = \int_0^a P(r)dr = \sum_{r=0}^{a} P(r)$$

$$\Rightarrow g(x,y) = T(f(x,y)) * 255$$

f(x,y)



g(x,y)

Note : s = niveaux de gris de g(x,y), r = niveaux de gris de f(x,y).

18

Traitement de l'histogramme

Égalisation d'histogramme

Note : voici la façon simple et efficace de calculer la densité de probabilité cumulative T:

$$T(0) = P(0)$$

POUR a allant de 1 à 255

$$T(a) = T(a-1) + P(a)$$

19

Traitement de l'histogramme

(Preuve)

Si une fonction de transfert T(r) définie entre 0 et 1 est **monotone croissante**, alors on peut affirmer que

$$p(s)ds = p(r)dr \qquad (1)$$

Étant donnée que

$$s = T(r)$$

Si T(r) est une densité de probabilité cumulative alors

$$s = T(r) = \int_0^r p(r')dr'$$

Étant donnée que la dérivée de T par rapport à r donne

$$\frac{dT}{dr} = \frac{ds}{dr} = \frac{d}{dr}\left(\int_0^r p(r')dr'\right) = p(r) \qquad (2)$$

En combinant (1) et (2) on obtient que

$$\frac{ds}{dr} = p(r)$$

$$= p(s)\frac{ds}{dr}$$

$$\Rightarrow p(s) = 1$$

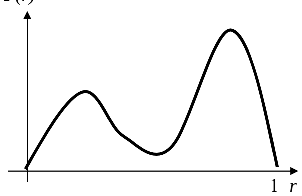
20

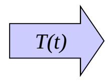
Traitement de l'histogramme

En conclusion, si *T(r)* est un densité de probabilité cumulative alors *P(s)=1*

Histogramme normalisé
des niveaux de gris de *f(x,y)*

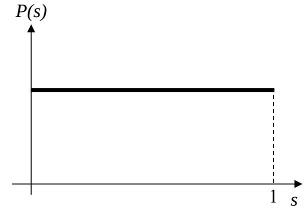
P(r)





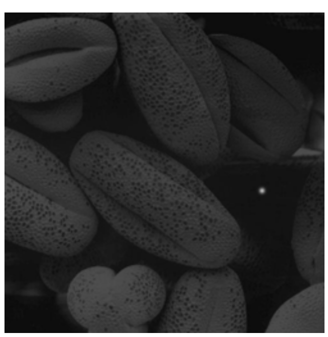
Histogramme normalisé
des niveaux de gris de *g(x,y)*

P(s)



Traitement de l'histogramme

Différence entre un changement de dynamique ordinaire et l'égalisation d'histogramme



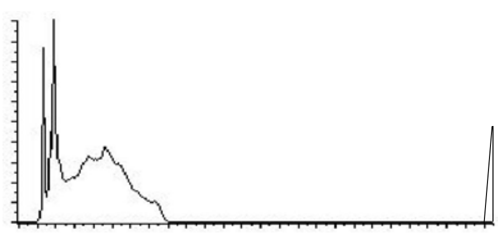
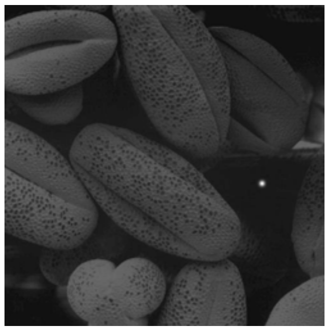


Image foncée avec une tache claire

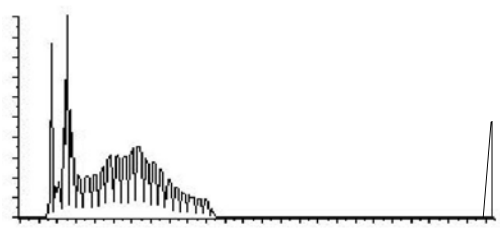
Gonzalez-Woods

Traitement de l'histogramme

Différence entre un changement de dynamique ordinaire et l'égalisation d'histogramme



$$g(x,y)=\frac{255}{max-min}(f(x,y)-min)$$

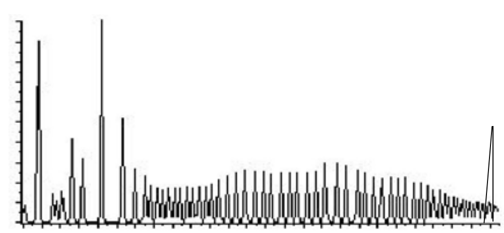


Résultat après un changement de dynamique linéaire (la fameuse fonction "*rescale*").

23
Gonzalez-Woods

Traitement de l'histogramme

Différence entre un changement de dynamique ordinaire et l'égalisation d'histogramme

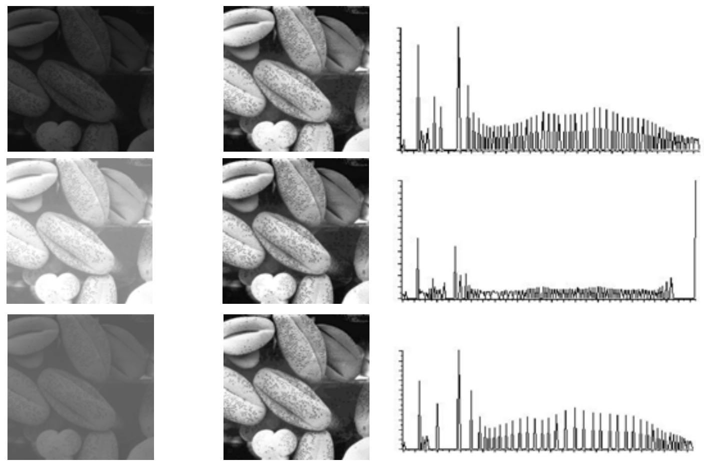


Résultat après une égalisation d'histogramme

24
Gonzalez-Woods

Traitement de l’histogramme

L’égalisation d’histogramme a l’avantage de s’adapter au contenu de l’image



Gonzalez-Woods

Gonzalez-Woods

Traitement de l’histogramme

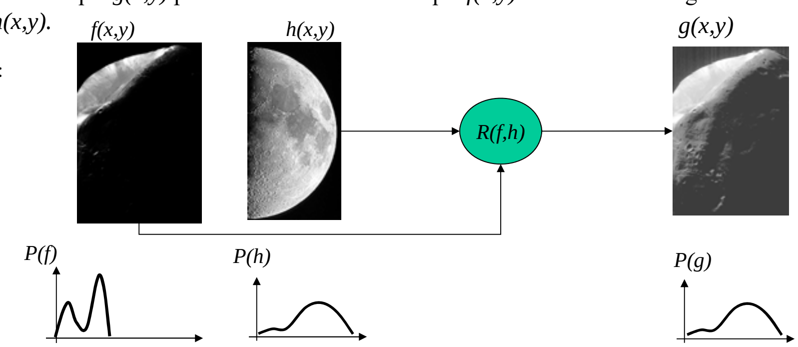
Transfert d’histogramme (*histogram matching*)

Parfois, on souhaite que l’histogramme d’une image $f(x,y)$ ait la même forme que l’histogramme d’une seconde image $h(x,y)$. Pour ce faire, on a besoin d’une fonction de transfert « R » de la forme :

$$g(x,y) = R(f(x,y),h(x,y))$$

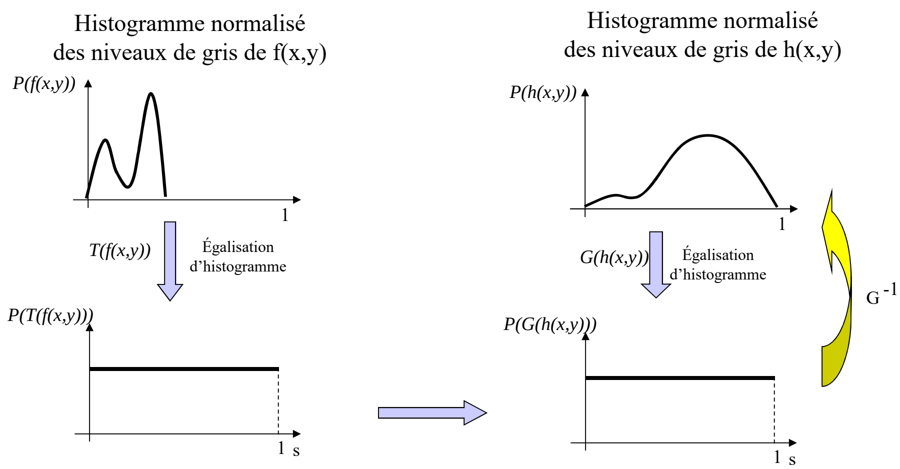
de sorte que $g(x,y)$ possède le même contenu que $f(x,y)$ mais avec l’histogramme de $h(x,y)$.

Exemple:



Traitement de l'histogramme

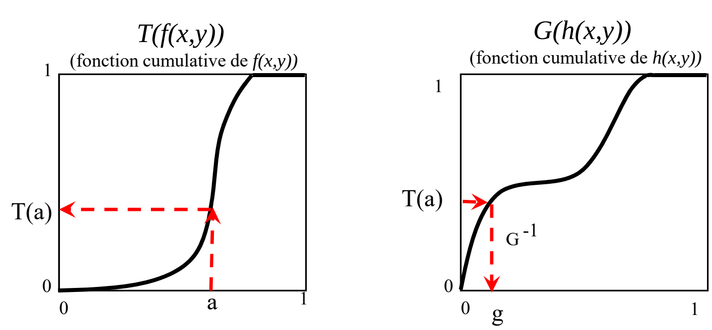
Transfert d'histogramme (*histogram matching)*



27

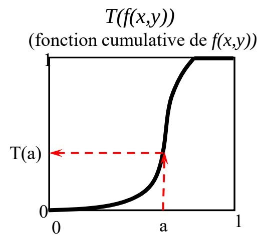
Traitement de l'histogramme

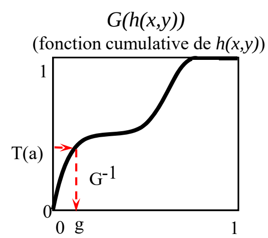
Transfert d'histogramme (*histogram matching)*



28

Algorithme du transfert d'histogramme



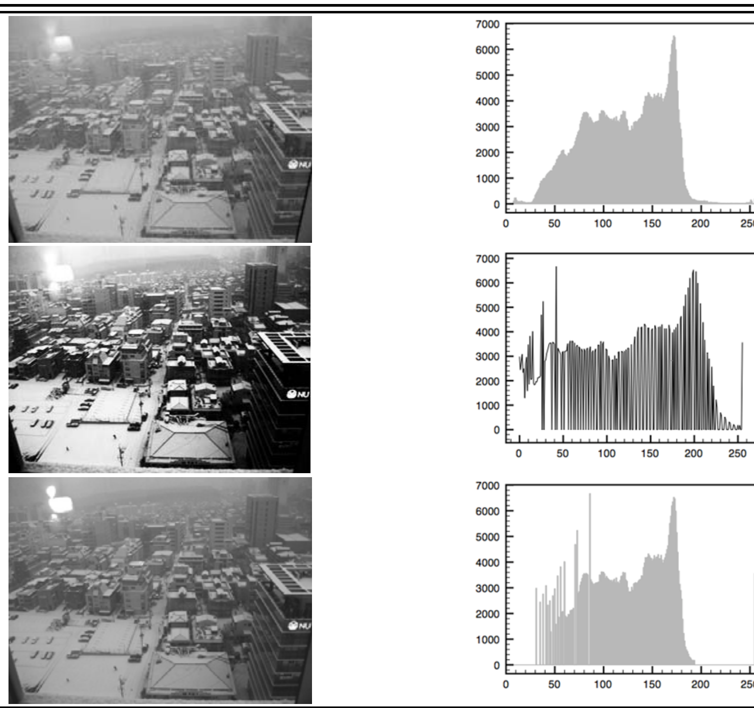


Algorithme D1

1. Calculer la fonction cumulative **T** de l'histogramme de f(x,y)
2. Calculer la fonction cumulative **G** de l'histogramme de h(x,y)

3. POUR chaque pixel *(x,y)* de l'image f
    3a. Calculer A = *T(f(x,y))*
    3b. Calculer g(x,y) = G^{-1}(A)

29



30

Credit : http://paulbourke.net/miscellaneous/equalisation/

Transformations géométriques

31

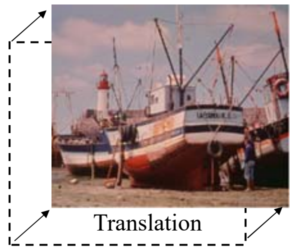
Transformations géométriques

Jusqu’à présent, nous nous sommes concentré sur des procédés ayant pour but de modifier la distribution des niveaux de gris/couleurs de l’image. Dans cette sous-section, nous porterons notre attention sur des procédés ayant pour but de modifier la **distribution spatiale des pixels**.

32

Transformations géométriques

Transformation de base:



Translation

Cisaillement (*Shear*)

Rotation

Changement d'échelle

33

Transformations géométriques

Transformations « artistiques » (le « *warping* »):

Ripples

Twirl

Zig Zag

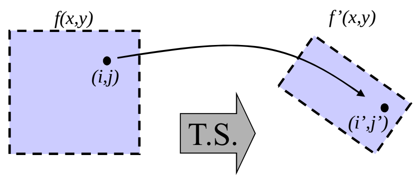
Lens

34

Transformations géométriques

Toute transformation géométrique est constituée de deux opérations de base:

1- **transformation spatiale**;
2- interpolation de niveaux de gris.



$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \Delta x + i \\ \Delta y + j \end{bmatrix}$$

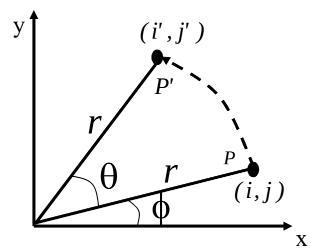
$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix}$$

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \Delta xi \\ \Delta yj \end{bmatrix}$$

35

Transformations géométriques

Rotation



θ positif = sens anti-horaire

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix}$$

$P' = RP$

$$P : \begin{cases} i = r\cos\phi \\ j = r\sin\phi \end{cases}$$

$$P' : \begin{cases} i' = r\cos(\phi + \theta) \\ \quad = \underline{r\cos\phi}\cos\theta - \underline{r\sin\phi}\sin\theta \\ \quad = i\cos\theta - j\sin\theta \\ j' = r\sin(\phi + \theta) \\ \quad = \underline{r\cos\phi}\sin\theta + \underline{r\sin\phi}\cos\theta \\ \quad = i\sin\theta + j\cos\theta \end{cases}$$

36

Transformations géométriques

Toute transformation géométrique est constituée de deux opérations de base:

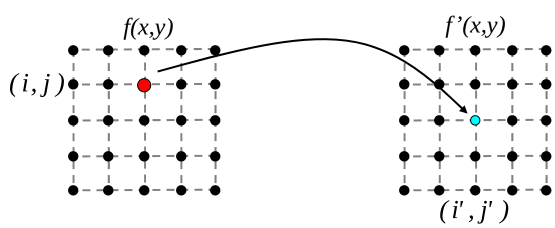
1- **transformation spatiale**;
2- interpolation de niveaux de gris/couleur.

Wave

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} i \\ j - 50\sin\left(\frac{2\pi i}{100}\right) \end{bmatrix}$$

Transformations géométriques

Transformation spatiale **<u>directe</u>**



Contrairement à ce que l'intuition commande, la transformation spatiale directe est **<u>À ÉVITER</u>.** Pourquoi? Car elle peut laisser des trous béants dans l'image de sortie *f'*.

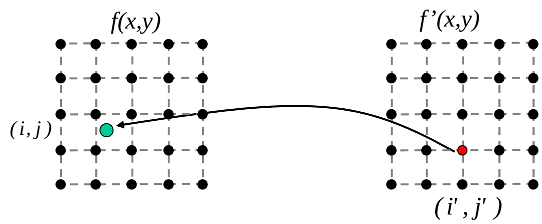
Exemple: changement d'échelle d'un facteur 5.

$$f(1,1) \to f'(5,1)$$
$$f(2,1) \to f'(10,1)$$

Et qu'y a-t-il entre *f'(5,1)* et *f'(10,1)*? En mode **directe**, RIEN!!!!!

Transformations géométriques

Transformation spatiale **inverse**



De cette façon, on pose la question suivante :

« Quelle est la couleur du pixel *(i',j')*? Réponse: c'est *f(i,j)*! »

Grâce à cette méthode, on évite les trous. Toutefois, le pixel *(i,j)* possède rarement des coordonnées entières. Par exemple, que faire lorsque *(i',j')* pointe vers

f(20.2,44.9)?

Je connais *f(20,44), f(21,44), f(20,45)* et *f(21,45)* mais pas *f(20.2, 44.9).*

39

Transformations géométriques

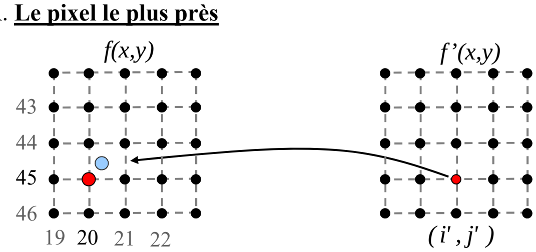
Transformation spatiale inverse

1- transformation spatiale;

2- **interpolation de niveaux de gris.**

Parmi les [très] nombreuses méthodes d'interpolation, trois sont fréquemment sollicitées en imagerie.

A. **Le pixel le plus près**



$$(i,j)=(20.2,44.9)\rightarrow(20,45)$$

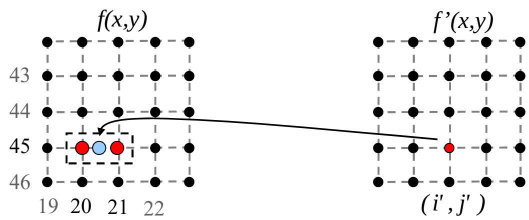
Solution simple mais qui induit des imprécisions.

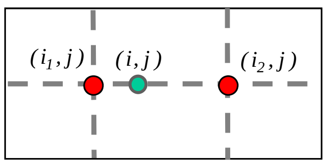
40

Transformations géométriques

Transformation spatiale inverse
1- transformation spatiale;
2- **interpolation de niveaux de gris.**

B. **<u>Interpolation linéaire</u>** (si *(i,j)* tombe entre deux pixels)





$$f(i,j) \approx f(i_1,j)(i_2 - i) + f(i_2,j)(i - i_1)$$

si $(i,j) = (20.2, 45)$

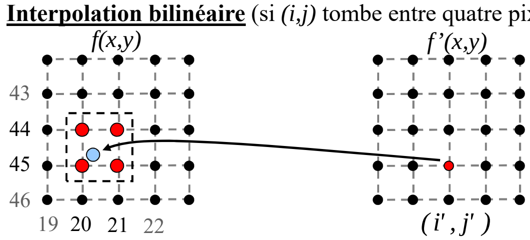
$$f(20.2,45) \approx f(20,45) \times 0.8 + f(21,45) \times 0.2$$

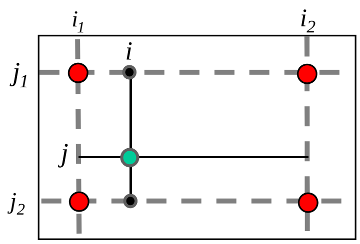
41

Transformations géométriques

Transformation spatiale inverse
1- transformation spatiale;
2- **interpolation de niveaux de gris.**

C. **<u>Interpolation bilinéaire</u>** (si *(i,j)* tombe entre quatre pixels)





$$f(i,j_1) \approx f(i_1,j_1)(i_2 - i) + f(i_2,j_1)(i - i_1)$$

$$f(i,j_2) \approx f(i_1,j_2)(i_2 - i) + f(i_2,j_2)(i - i_1)$$

$$f(i,j) \approx f(i,j_1)(j_2 - j) + f(i,j_2)(j - j_1)$$

42

Transformations géométriques

Transformation spatiale inverse
1- transformation spatiale;
2- **interpolation de niveaux de gris.**

f’(x,y)

f(x,y)

Plus proche voisin

Interpolation bilinéaire

43

Une autre transformation géométrique :

le changement d’échelle

44

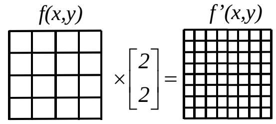
Transformations géométriques

Changement d'échelle (*image resizing*) $\begin{bmatrix} M' \\ N' \end{bmatrix} = \begin{bmatrix} \Delta x M \\ \Delta y N \end{bmatrix}$ où M,N est la dimension de l'image.

Dans le cas d'un changement d'échelle, le pixel destination *(i,j)* ne possède pas la même taille que le pixel d'origine *(i',j')*.

Lorsque $\Delta x, \Delta y \geq 1$, une interpolation bilinéaire sied bien.

f(x,y) *f'(x,y)*



(i',j')	*(i,j)*
(0,0)	*(0,0)*
(1,0)	*(0.5,0)*
(2,0)	*(1,0)*
(1,1)	*(0.5,0.5)*
…	…

45

Transformations géométriques

Changement d'échelle (*image resizing*) $\begin{bmatrix} M \\ N \end{bmatrix} = \begin{bmatrix} 0.5M' \\ 0.5N' \end{bmatrix}$ où M,N est la dimension de l'image.

f(x,y)

	0	1	2
0	*a*	*b*	*c*
1	*d*	*e*	*f*
2	*g*	*h*	*i*

f'(x,y)

	0	1	2	3	4	5
0	a	$\frac{a+b}{2}$	b			
1	$\frac{a+d}{2}$	A	$\frac{b+e}{2}$			
2	d	$\frac{d+e}{2}$	e			
3						
4						
5						

$$A = \frac{a+b+d+e}{4}$$

46

Transformations géométriques

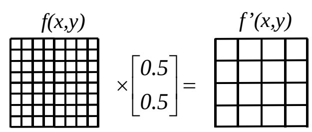
Changement d'échelle
(*image resizing*)

$$\begin{bmatrix} M' \\ N' \end{bmatrix} = \begin{bmatrix} \Delta x M \\ \Delta y N \end{bmatrix}$$

où M,N est la dimension de l'image.

Dans le cas d'un changement de résolution, le pixel destination *(i,j)* ne possède pas la même taille que le pixel d'origine *(i',j')*.

Par contre, lorsque $0 < \Delta x, \Delta y < 1$, il nous faut trouver une autre méthode.



Pourquoi une autre méthode? Voici un exemple.

Transformations géométriques

Changement d'échelle
(*image resizing*)

$$\begin{bmatrix} M \\ N \end{bmatrix} = \begin{bmatrix} 2M' \\ 2N' \end{bmatrix}$$

où M,N est la dimension de l'image.

La façon la plus simple est de prendre le voisin le plus proche.

(i',j')	*(i,j)*
(0,0)	*(0,0)*
(1,0)	*(2,0)*
(2,0)	*(4,0)*
…	…

	0	1	2	3	4	5
0	a	b	c	d	e	f
1	g	g	i	j	k	l
2	m	n	o	p	q	r
3	s	t	u	v	w	x
4	y	z	α	β	χ	δ
5	ε	ϕ	γ	η	ι	φ

	0	1	2
0	a	c	e
1	m	o	q
2	y	α	χ

Transformations géométriques

Changement d'échelle
(*image resizing*)

$$\begin{bmatrix} M' \\ N' \end{bmatrix} = \begin{bmatrix} 0.5M \\ 0.5N \end{bmatrix}$$ où M,N est la dimension de l'image.

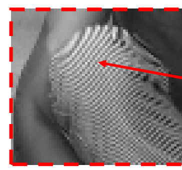
$$\begin{bmatrix} M \\ N \end{bmatrix} = \begin{bmatrix} 2M' \\ 2N' \end{bmatrix}$$

La façon la plus simple est de prendre le voisin le plus proche.

(i',j')	*(i,j)*
(0,0)	*(0,0)*
(1,0)	*(2,0)*
(2,0)	*(4,0)*
…	…

f(x,y)

f'(x,y)



Aliassing
(Moiré)

49

Transformations géométriques

Changement d'échelle
(*image resizing*)
$\Delta x, \Delta y = 0.5$

$$\begin{bmatrix} M' \\ N' \end{bmatrix} = \begin{bmatrix} 0.5M \\ 0.5N \end{bmatrix}$$ où M,N est la dimension de l'image.

$$\begin{bmatrix} M \\ N \end{bmatrix} = \begin{bmatrix} 2M' \\ 2N' \end{bmatrix}$$

La façon la plus simple est de prendre le voisin le plus proche.

(i',j')	*(i,j)*
(0,0)	*(0,0)*
(1,0)	*(2,0)*
(2,0)	*(4,0)*
…	…

Les pixels aux coordonnées impaires dans *f(x,y)* ne sont pas sollicités. Une telle approche porte le nom de « **décimation** » lorsque $\Delta x = \frac{1}{P}$ et $\Delta y = \frac{1}{R}$ où $P, R \in N^{+}$

$$f(x,y) \longrightarrow \boxed{2\downarrow} \longrightarrow f'(x,y)$$

50

Transformations géométriques

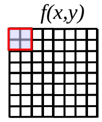
Changement d'échelle
(*image resizing*)
$\Delta x, \Delta y = 0.5$

$$\begin{bmatrix} M' \\ N' \end{bmatrix} = \begin{bmatrix} 0.5M \\ 0.5N \end{bmatrix}$$ où M,N est la dimension de l'image.

$$\begin{bmatrix} M \\ N \end{bmatrix} = \begin{bmatrix} 2M' \\ 2N' \end{bmatrix}$$

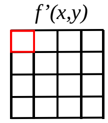
En fait, lorsque $\Delta x, \Delta y = 0.5$ un pixel *(i',j')* couvre quatre pixels dans *f(x,y)*

f(x,y)



$\times \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} =$

f'(x,y)



Par conséquent, lorsque $0 < \Delta x, \Delta y < 1$:

f'(x,y) = Moyenne des pixels couverts par *(x,y)* dans *f(x,y)*

51

Transformations géométriques

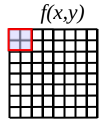
Changement d'échelle
(*image resizing*)

$$\begin{bmatrix} M' \\ N' \end{bmatrix} = \begin{bmatrix} 0.5M \\ 0.5N \end{bmatrix}$$ où M,N est la dimension de l'image.

$$\begin{bmatrix} M \\ N \end{bmatrix} = \begin{bmatrix} 2M' \\ 2N' \end{bmatrix}$$

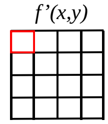
En fait, lorsque $\Delta x, \Delta y = 0.5$ un pixel *(i',j')* couvre quatre pixels dans *f(x,y)*

f(x,y)

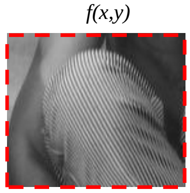


$\times \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} =$

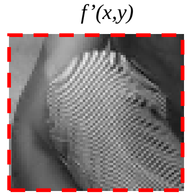
f'(x,y)



f(x,y)

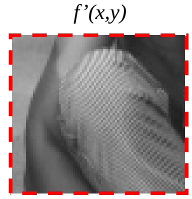


f'(x,y)



plus proche voisin

f'(x,y)



moyenne

52

Transformations géométriques

Changement d'échelle

Lorsque $\Delta x = \frac{1}{P}$ et $\Delta y = \frac{1}{R}$ où $P,R \in N^{+}$, une façon simple de calculer « la moyenne des pixels couverts par (x,y) » est de filtrer *f(x,y)* par un filtre passe-bas et de décimer l'image filtrée.

$$f(x,y) \longrightarrow \boxed{h} \xrightarrow{(f*h)(x,y)} \boxed{P,R\downarrow} \longrightarrow f'(x,y)$$

Les deux filtres les plus couramment utilisés sont les filtres:

- Moyenneur de taille : $\frac{1}{\Delta x} \times \frac{1}{\Delta y}$
- Gaussien d'écart type est : $\sigma_x = \frac{1}{4\Delta x}$, $\sigma_y = \frac{1}{4\Delta y}$

Transformations géométriques

Changement de résolution

Lorsque $\Delta x = \frac{1}{P}$ et $\Delta y = \frac{1}{R}$ où $P,R \in N^{+}$, une façon simple de calculer « la moyenne des pixels couverts par (x,y) » est de filtrer *f(x,y)* par un filtre passe-bas et de décimer l'image filtrée.

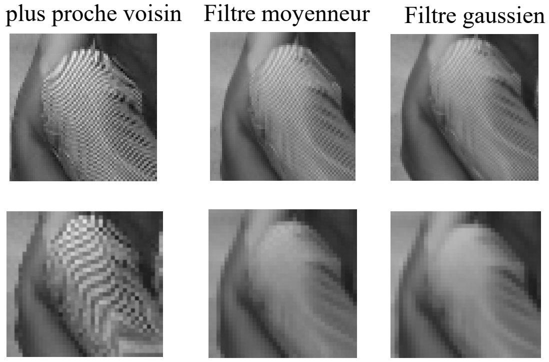
f(x,y)

f'(x,y)

plus proche voisin | Filtre moyenneur | Filtre gaussien

$\Delta x = \Delta y = 0.5$

$\Delta x = \Delta y = 0.25$



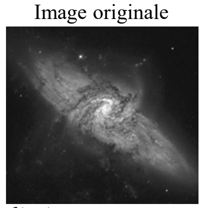
Opérations inter-images

55

Opérations inter-images.

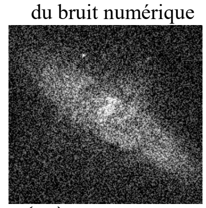
Réduction du bruit par moyennage

Image originale



f(x,y)

Image corrompue par du bruit numérique



g(x,y)

$$g(x,y) = f(x,y) + \eta(x,y)$$

g(x,y) a été obtenue en additionnant un bruit $\eta(x,y)$ **non corrélé** et de **moyenne nulle**.

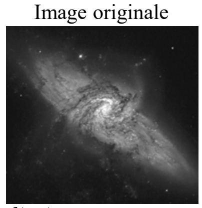
56

Gonzalez-Woods

Opérations inter-images.

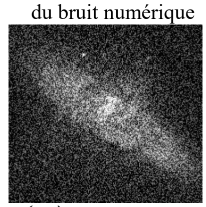
Réduction du bruit par moyennage

Image originale



$f(x,y)$

Image corrompue par du bruit numérique



$g(x,y)$

$$\boxed{g(x,y) = f(x,y) + \eta(x,y)}$$

Si la lentille peut prendre plusieurs images (toujours avec un bruit non corrélé et de moyenne nulle) alors on peut réduire le bruit en additionnant ces images entre elles :

$$\boxed{\bar{g}(x,y) = \frac{1}{N}\sum_{i=1}^{N} g_i(x,y)}$$

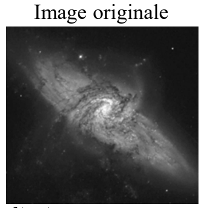
57

Gonzalez-Woods

Opérations inter-images.

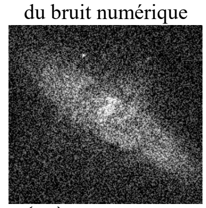
Réduction du bruit par moyennage

Image originale

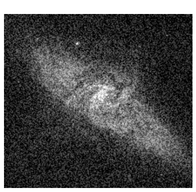


$f(x,y)$

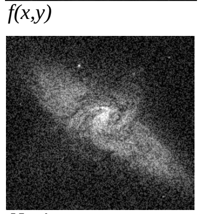
Image corrompue par du bruit numérique



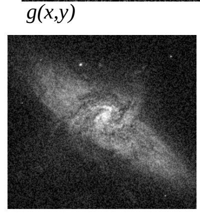
$g(x,y)$



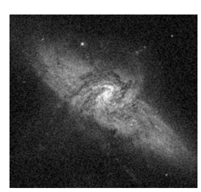
$N=2$



$N=4$



$N=8$



$N=16$

Ratio signal sur bruit : $\left(\frac{\bar{g}}{\eta}\right) = \sqrt{N}\left(\frac{g}{\eta}\right)$ Donc, pour $N=64$, le bruit dans $\bar{g}$ est réduit d'un facteur 8.

58

Gonzalez-Woods

Opérations inter-images.

Détection de mouvement

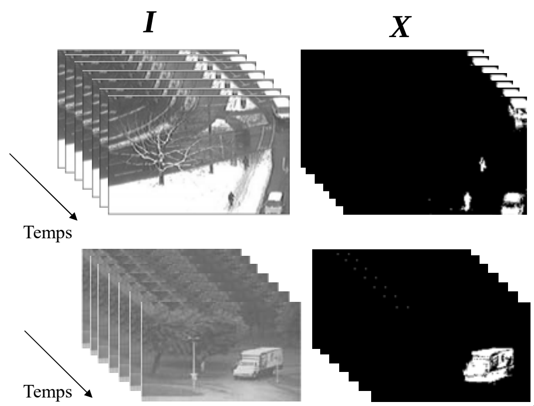
Le but? différencier (ou *segmenter*) les zones mobiles des zones immobiles dans une séquence vidéo.

Ce dont nous disposons:

une séquence vidéo ***I*** constituée de *N* images, une à chaque temps *t*: I_t.
On appelle souvent « *frame* » une image dans une séquence vidéo.

Ce qu'il nous faut estimer:

un champ d'étiquettes X_t pour chaque frame I_t. En général, X_t est une image binaire de la même taille que I_t.



Opérations inter-images.

Détection de mouvement

Le but? différencier (ou *segmenter*) les zones mobiles des zones immobiles dans une séquence vidéo.

Deux grandes familles de méthodes :
(1) celles avec caméra **fixe**;
(2) celles avec caméra **mobile**. ⟶ Sujet de cours avancés

Les méthodes avec caméra **fixe** sont très utiles pour les applications de surveillance et de contrôle de la qualité (sur une ligne de montage par exemple). Pour ces méthodes, on considère souvent que:

$\boxed{I_t = B + \text{Objets en mouvement}}$ où B est un image du fond (backgroun d)

On émet souvent deux hypothèses:
(1) les objets en mouvement ont une couleur différente des pixels du fond.
(2) l'image du fond est toujours la même, du début à la fin.

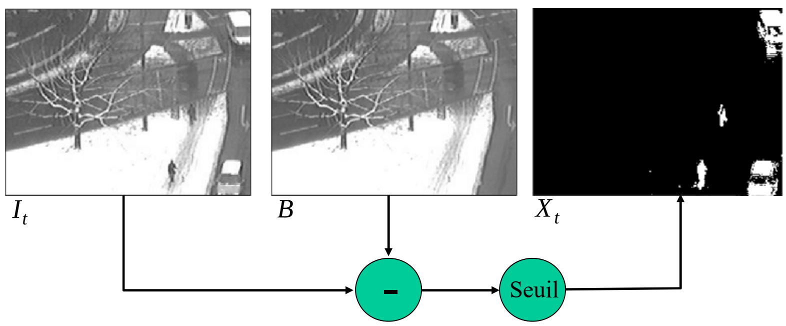
60

Opérations inter-images.

Détection de mouvement

En admettant pour vraies ces hypothèses, on peut estimer X par une simple opération de **soustraction de fond** :

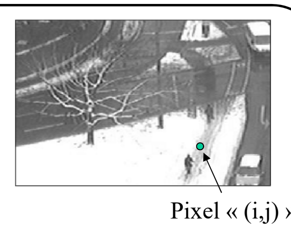
$$\boxed{X_t(x,y) = \mathrm{Seuil}(|I_t(x,y) - B(x,y)|)}$$

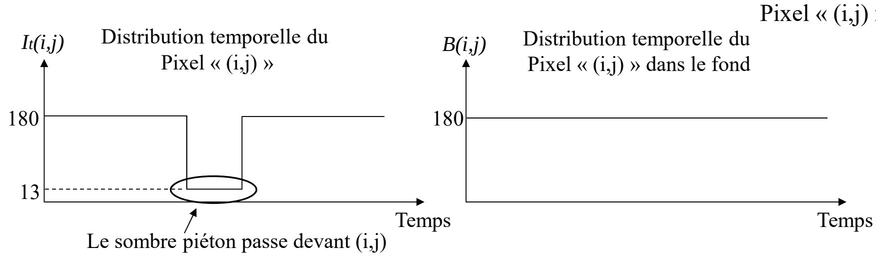


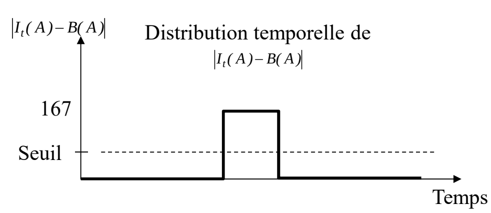
61

Opérations inter-images.

Détection de mouvement



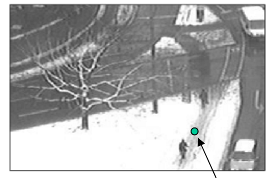




62

Opérations inter-images.

Détection de mouvement



Pixel « (i,j) »

$$\boxed{X_t(x,y) = \mathrm{Seuil}(|I_t(x,y) - B(x,y)|)}$$

Bien sûr, la méthode de soustraction de fond fonctionne bien *ssi* :

1. B est connu;
2. B est constant dans le temps;
3. I_t n'est pas une séquence « trop bruitée »;
4. les objets du fond sont immobiles;
5. la caméra est parfaitement fixe;
6. les objets en mouvement ont une couleur différente du fond.

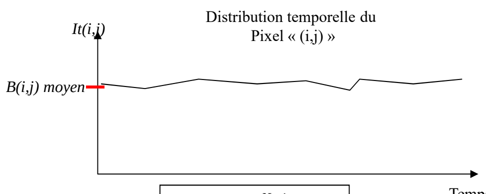
63

Opérations inter-images.

Détection de mouvement

Pixel « (i,j) »

1. Que faire si B est inconnu? Il faut l'estimer!



$$\boxed{B(i,j) = \frac{1}{N}\sum_{i=0}^{N-1} I_i(i,j)}$$

Pour estimer B à partir d'un séquence vidéo, il faut, pour chaque pixel de la séquence, distinguer la couleur/l'intensité du fond de la couleur des objets en mouvement et des erreurs de transmission. Pour ce faire, on peut

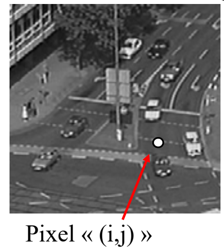
1. Prendre la moyenne
2. Prendre la médiane.

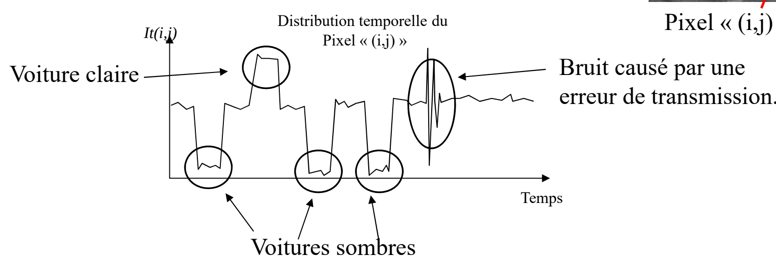
64

Opérations inter-images.

Détection de mouvement

1. Que faire si B est inconnu? Il faut l'estimer!

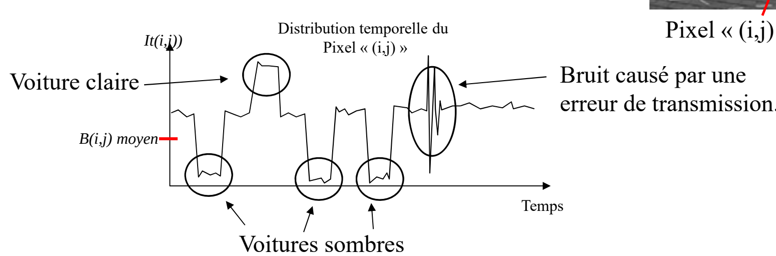




Opérations inter-images.

Détection de mouvement

1. Que faire si B est inconnu? Il faut l'estimer!



Moyenne: $\boxed{B(i,j) = \frac{1}{N}\sum_{i=0}^{N-1} I_i(i,j)}$

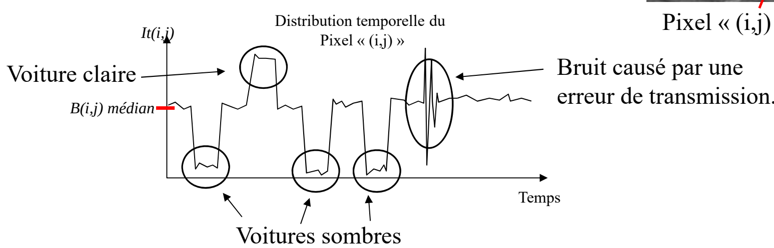
<u>ATTENTION</u> : la moyenne de fonctionne que si les observations $I_i(i,j)$ contiennent des valeurs se distribuant de façon **gaussienne** autour d'une valeur moyenne. Cette méthode est donc très sensible aux valeurs extrêmes (les *outliers*).

Opérations inter-images.

Détection de mouvement

Pixel « (i,j) »

1. Que faire si B est inconnu? Il faut l'estimer!



Médiane: $\boxed{B(i,j) = \text{médiane}\big(I_0(i,j), I_1(i,j), I_2(i,j), \ldots, I_{N-1}(i,j)\big)}$

$\boxed{B(i,j) = \text{médiane}\big(I_0(i,j), I_{50}(i,j), I_{100}(i,j), \ldots, I_{N-101}(i,j), I_{N-51}(i,j), I_{N-1}(i,j)\big)}$

<u>ATTENTION</u> : la médiane est robuste aux *outliers* à condition qu'ils ne correspondent pas à plus de 50% des observations.

67

Opérations inter-images.

Détection de mouvement

Médiane temporelle

Médiane temporelle

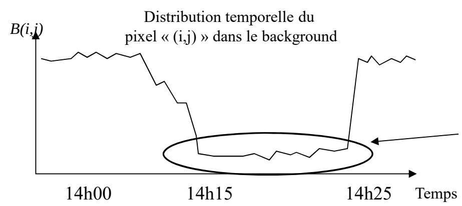
Opérations inter-images.

Détection de mouvement

2. Que faire si B n'est pas constant dans le temps?

À 14h00 il faisait beau.

À 14h15 un gros nuage a obstrué le ciel.



Un nuage a momentanément obstrué le ciel.

Dans ce cas, il faut mettre à jour B à chaque temps t

$$\boxed{B_t(i,j) = (1-\alpha)B_{t-1}(i,j) + \alpha I_t(i,j)}$$

où α est la « *forgetting constant* », généralement beaucoup plus petite que 1. Afin d'éviter qu'un objet lent ne soit associé au fond, on peut faire ceci

$$\boxed{B_t(i,j) = (1-\alpha)B_{t-1}(i,j) + \alpha I_t(i,j) \quad \forall (i,j) \text{ t.q. } X_{t-1}(i,j) = 0}$$

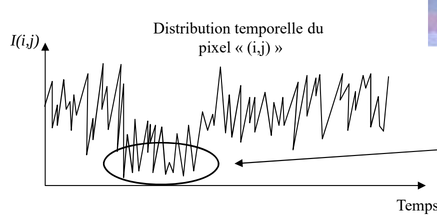
69

Opérations inter-images.

Détection de mouvement

3. Que faire si I est bruitée?

Pixel « (i,j) »



Le camion passe devant (i,j)

Le bruit peut causer de nombreux faux positifs et faux négatifs. Que pouvons-nous faire?

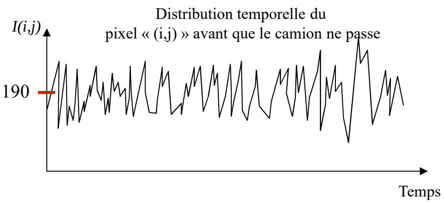
1- Puisque les faux positifs/négatifs forment un bruit impulsionnel, on peut utiliser un **filtre médian**.

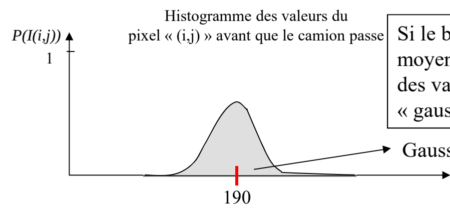
2- On peut modéliser le fond de façon plus intelligente, c-à-d. de façon **probabiliste**.

Opérations inter-images.

Détection de mouvement

3. Que faire si *I* est bruitée?





Si le bruit est non corrélé dans le temps et de moyenne nulle, généralement l'histogramme des valeurs se distribuera suivant une « gaussienne ».

Gaussienne centrée sur 190 et d'écart type σ.

Opérations inter-images.

Détection de mouvement

3. Que faire si *I* est bruitée?

Au lieu de modéliser le pixel (i,j) par la valeur 190, on le modélisera par une gaussienne centrée sur 190 et d'écart type σ. Pour estimer la moyenne et l'écart-type de la gaussienne associée au pixel (i,j), on a besoin d'une série de N frames sans mouvement, c'est-à-dire ne contenant que le fond. Dans ce cas,

$$\mu_{(i,j)} = \frac{1}{N}\sum_{t=1}^{N-1} I_t(i,j)$$

$$\sigma^2_{(i,j)} = \frac{1}{N-1}\sum_{t=1}^{N-1}\left(I_t(i,j) - \mu_{(i,j)}\right)^2$$

Une fois $\mu_{(i,j)}$ et $\sigma_{(i,j)}$ estimés, la détection se fait à l'aide de l'opération suivante:

$$X_t(i,j) = \text{Seuil}(P_{\mu,\sigma}(I_t(i,j))$$
$$= Seuil(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}})$$

Opérations inter-images.

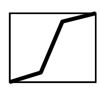
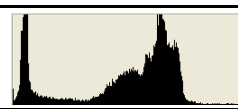
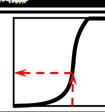
Détection de mouvement

6. les objets en mouvement ont la même couleur que le fond

C’est ce qu’on appelle un problème de « camouflage ». Malheureusement, il y a très peu de choses que l’on puisse faire avec une méthode par soustraction de fond. C’est une des limites inhérentes à cette approche. Toutefois, dans certains cas, on jouit d’une connaissance *a priori* du problème que l’on peut exploiter *a posteriori*.

Par exemple: Il me faut détecter des camions en mouvement. Puisque je sais qu’un camion est un objet plein, je peux remplir les trous (c-à-d éliminer les faux négatifs) à l’intérieur du véhicule une fois la détection faite.

Les faits saillants

1. Correction gamma	$g(x,y)=(f(x,y))^{\gamma}$
2. Transformations linéaires/non linéaires	
3. Histogramme	
4. Égalisation d’histogramme	
5. Interpolation	$f(i,j)\approx f(i,j_1)(j_2-j)+f(i,j_2)(j-j_1)$
6. Détection de mouvement