Fundamentals of Image Processing

▶ Lecture 2: Basic operations and image enhancement ◀

Master of Computer Science
Sorbonne University

September 2022

Outline

Basic image transformations

Operations between images

Image thresholding

Image enhancement

Operations on an image

How to transform an image?

- A pixel in an image is defined by its coordinates (i, j) and its intensity k.
- Two types of image transformations:
  - geometric spatial transformations that modify pixels position,
  - radiometric transformations that modify pixels intensity.
- Obviously, we can combine the two types of transformation.

Geometric spatial transformations

Affine transformation

- Transformation applying on spatial coordinates of a pixel:

$$\begin{pmatrix} i' \\ j' \end{pmatrix} = T \begin{pmatrix} i \\ j \end{pmatrix} + V$$

  with T a matrix, V a vector

- Inverse transformation (if it exists), with $V = \vec{0}$:

$$\begin{pmatrix} i' \\ j' \end{pmatrix} = T^{-1} \begin{pmatrix} i \\ j \end{pmatrix}$$

- The transformed image J is then defined from I by: $J(i', j') = I(i, j)$

Translation

- Basic translation $(t_i, t_j)^t$, applied on pixel (i, j):

$$\begin{pmatrix} i' \\ j' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} + \begin{pmatrix} t_i \\ t_j \end{pmatrix}$$

Change of scale

- Change of scale of a pixel (i, j) with coefficients α_i et α_j:

$$\begin{pmatrix} i' \\ j' \end{pmatrix} = \begin{pmatrix} \alpha_i & 0 \\ 0 & \alpha_j \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix}$$

Rotation

- Rotation of a pixel (i, j) with angle θ (the origin is the image center)

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

Linear transformation

- Linear transformation of pixel (i, j) with coefficients β_{i_1}, β_{i_2}, β_{j_1} and β_{j_2} is expressed by:

$$\begin{pmatrix} i' \\ j' \end{pmatrix} = \begin{pmatrix} \beta_{i_1} & \beta_{i_2} \\ \beta_{j_1} & \beta_{j_2} \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix}$$

Geometric spatial transformations

Homogeneous coordinates

- System of coordinates associated with projective spaces:
  - Euclidean spaces $\subset$ affine spaces $\subset$ projective spaces
- Practically: we add a supplementary coordinate
  (x, y) affine $\rightarrow (x, y, 1) \sim (x \cdot w, y \cdot w, w)$
- All geometric transformations are now expressed as a matrix operation:
  $\Rightarrow$ translations of $\mathbb{R}^2$ are now expressed as linear transformations of $\mathbb{R}^3$
  $\Rightarrow$ same property for projections

Homogeneous coordinates

- Linear transformations of $\mathbb{R}^2$

$$\begin{pmatrix} i' \\ j' \\ 1 \end{pmatrix} = \begin{pmatrix} \beta_{i_1} & \beta_{i_2} & 0 \\ \beta_{j_1} & \beta_{j_2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}$$

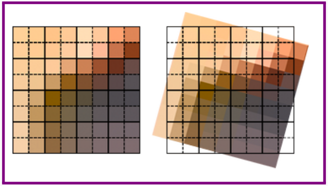
- Affine transformations of $\mathbb{R}^2$ (linear + translation):

$$\begin{pmatrix} i' \\ j' \\ 1 \end{pmatrix} = \begin{pmatrix} \beta_{i_1} & \beta_{i_2} & T_x \\ \beta_{j_1} & \beta_{j_2} & T_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}$$

Geometric spatial transformations

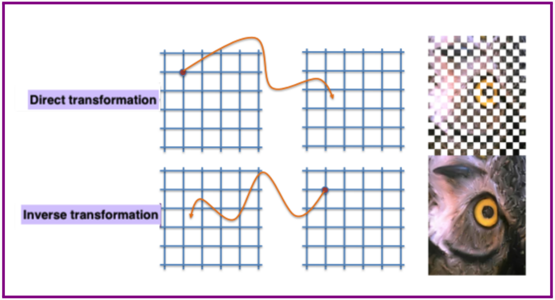
Issue

- Affine transformations are bijective in $\mathbb{R}^2$ but not in the discrete space of pixels! Some of pixels may be unassigned in the resulting image...
  $\hookrightarrow$ missing data, "holes" in the image
- Two solutions:
  - Use the inverse transformation (if possible). This approach also may also be inconvenient.
  - "Filling" the holes
- In both cases, we have to use interpolation techniques



Direct and inverse transformations

- Direct transformation: pixels coordinates in the output image are determined from pixels coordinates in the input image
  $\hookrightarrow$ can generate missing data or superposition.
- Inverse transformation: pixels coordinates in the input image are determined from pixels coordinates in the output image
  $\hookrightarrow$ can generate superposition and also missing data (due to bounded spatial domain).



Direct and inverse transformations

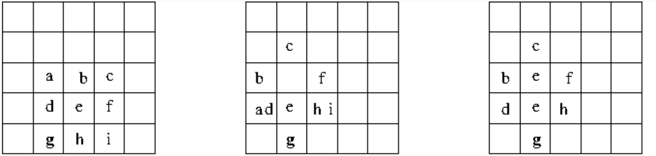
A simple example

$$(x', y') = t(x, y)$$

Issues:

- (x, y) = integer coordinates $\Rightarrow (x', y')$?
- Computation?
- Properties?

$\pi/4$ rotation: $x' = (x - y)\frac{\sqrt{2}}{2} \quad y' = (x + y)\frac{\sqrt{2}}{2}$

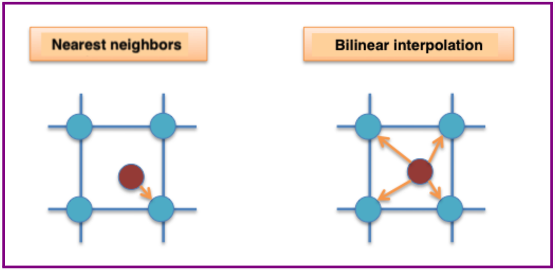


Direct transformation

Inverse transformation
(closest point interpolation)

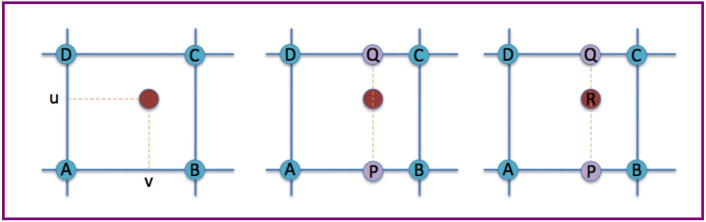
Interpolation

- Two examples of basic interpolation methods:
  - Nearest neighbor: pixel value is given by the value of the nearest neighboring pixel
  - Bilinear interpolation: pixel value is determined from the 4 nearest neighboring pixels using a bilinear interpolation



- Many other interpolation methods: B-splines, Hermite interpolation polynomials, ...

Bilinear interpolation



$$
\begin{aligned}
P &= (1-v)A + vB, v \in [0,1] \\
Q &= (1-v)D + vC \\
R &= (1-u)P + uQ, u \in [0,1] \\
&= (1-v)(1-u)A + (1-u)vB + uvC + u(1-v)D
\end{aligned}
$$

Geometric transformations: some applications

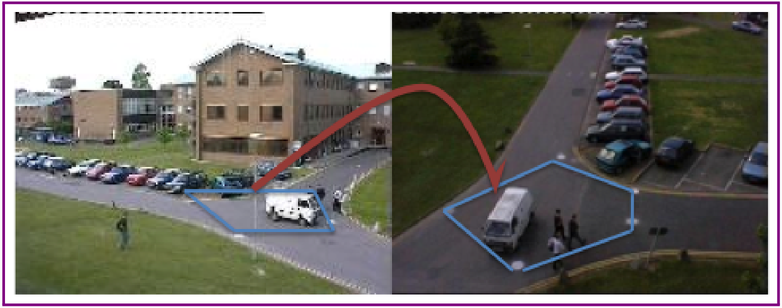
Tracking with several cameras

- Object tracking using several view points of a same scene

Geometric transformations: some applications

Tracking with several cameras

- A solution: homographic transformation (linear for homogeneous coordinates, encoding translation, rotation, change of scale, perspective projection: 3×3 matrix $\Rightarrow$ 9 parameters)

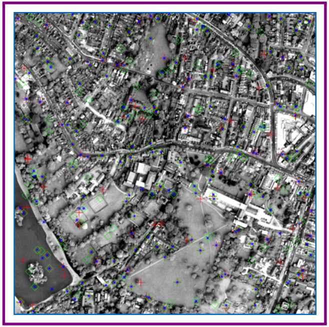


Rigid registration for cartography

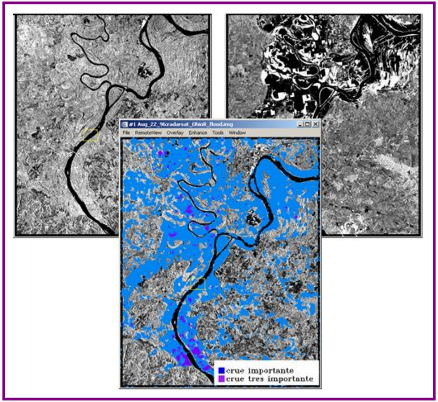
- Data: a map and a satellite image (IKONOS)

Rigid registration for cartography

- A solution: detection of points of interest (and invariant), association between points in each modality, and homographic transformation



Flood monitoring with satellite images



Geometric transformations: some applications

Image registration for medical multimodal imaging (fusion)

- Data: various *modalities* (CT, ultrasound, IRM...)
- Goal: data fusion for best diagnosis, a registration is required before fusion

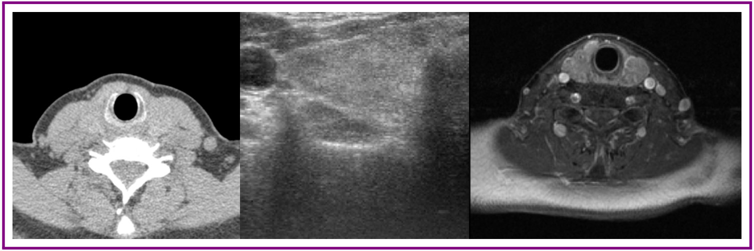
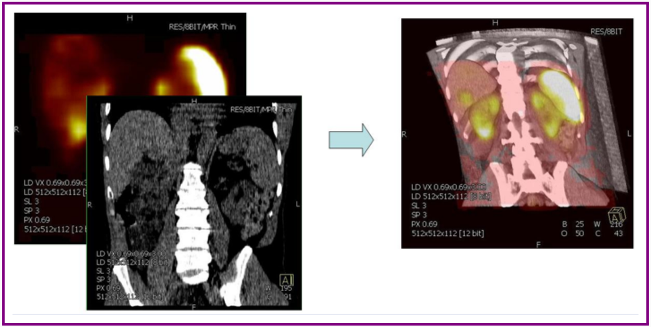


Image registration for medical multimodal imaging (fusion)

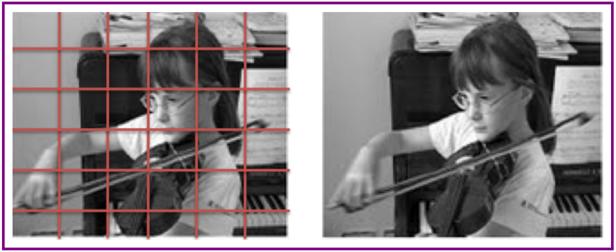


Temporal predictor and motion compensation for video coding

- Data: two successive images in a video sequence
- Goal: video compression using motion compensation

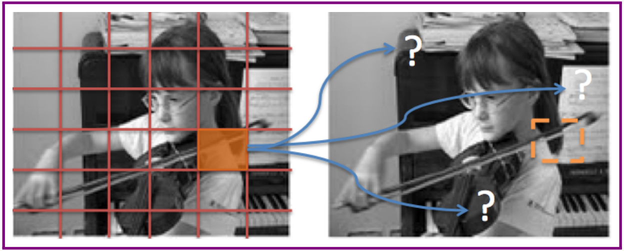
Temporal predictor and motion compensation for video coding

- Partition of image 1 (blocks)



Temporal predictor and motion compensation for video

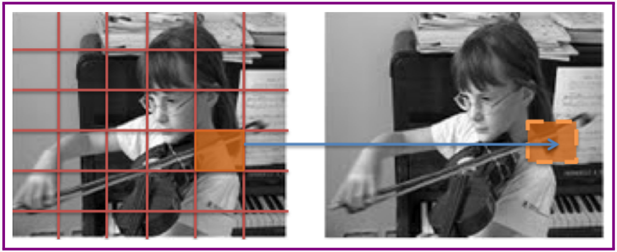
- Determine the block position in image 2 (motion estimation)



Geometric transformations: some applications

Motion compensated prediction for video compression

- Build the prediction of image 2 using translated blocks, **encode the prediction error**: a **good prediction** leads to a **low prediction error** (and **high compression rate**)

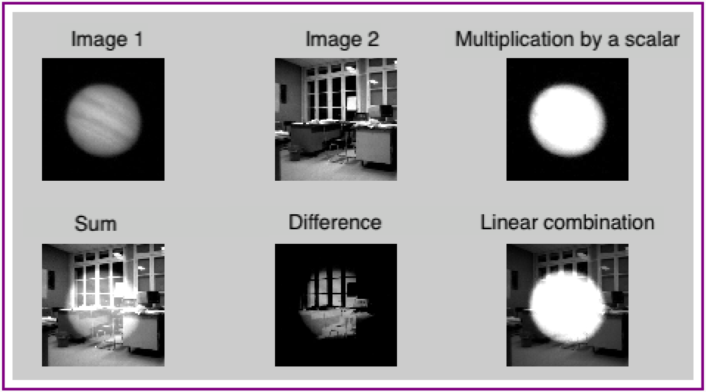


Outline

Operations between images

- Arrays are not matrices (matrix = linear application). Images are represented by 2D arrays or vectors by arranging them row by row or column by column
- Operations between matrices (linear algebra) $\neq$ Element wise operations (addition, subtraction, multiplication...)

Operations between images: some examples



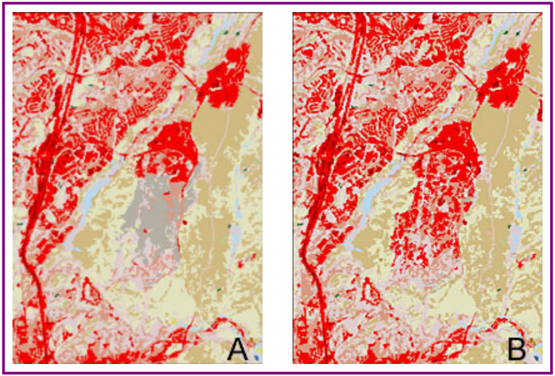
Operations between images: some applications

Subtraction and noise reduction

- Given 2 identical images, I, I^n, (with the exception of noise) that have been registered
- The difference image, $I - I^n$, allows for the detection of noise
- Black pixels: no difference
- Other values: noise amplitude
- Given two images of a same scene acquired at two different times: variation in time of pixels values (approximation of the time partial derivative)

Operations between images: some applications

Difference image for change detection: which differences are not due to noise?



Difference image for motion detection

Operations between images: some applications

Difference image for motion detection

⇒ the difference image is not sufficient for motion detection

- Lighting conditions, occultations, acquisition noise may induce a significant temporal gradient response
- It is not possible to estimate velocity (intensity and direction) from the difference image

Outline

Thresholding

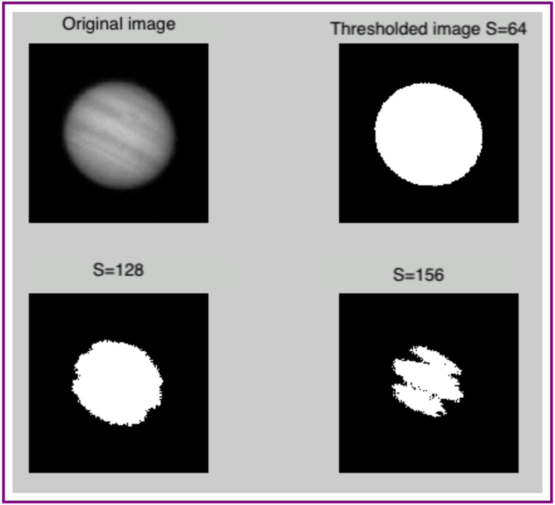
Definitions and principle

- **Thresholding**: reduction of image values to few levels of intensity
- **Binarization**: image values are reduced to two intensity levels
- Binary thresholding, defined by:

$$k' = \begin{cases} k_1 & \text{if} \quad k \leq S \\ k_2 & \text{if} \quad k > S \end{cases}$$

  with k_1, k_2 and S (threshold) are levels of intensity
- Highlights regions but does not enhance the image

Example of thresholding ($k_1 = 0$ and $k_2 = 255$)



Outline

Definition

- Making images more suitable for human or machine interpretation
- No general theory
- Filtering in the spatial domain: direct use of pixel values
- Filtering in the frequency domain: modification of the Fourier transform of the image

Image enhancement

Three categories

- Pixel-level enhancement: $f'(i,j) = T(f(i,j))$
  $\hookrightarrow$ the image brightness or contrast is modified
  $\hookrightarrow$ no spatial information, only radiometric value of the visited pixel is considered
  $\hookrightarrow$ histogram changes are in this category
- Local enhancement: $f'(i,j) = T(f(V))$, with V the neighborhood of pixel (i,j)
  $\hookrightarrow$ Use of **spatial filters** (see Lecture 5).
- Enhancement in the frequency domain: $\hat{f}' = T(\hat{f})$
  $\hookrightarrow$ Use of **Fourier transform** (see next lecture).

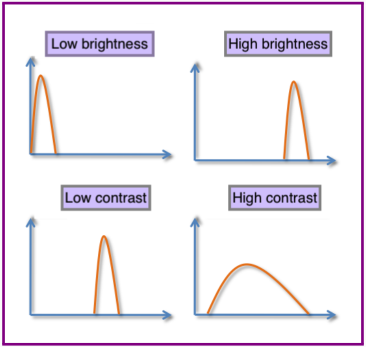
Applications of image enhancement

- Regions to highlight
- Images that are too bright or too dark
- Intensity levels should be changed in order to make some details in the image more visible

Image enhancement

Applications of image enhancement

- Modify image brightness
- Increase contrast (see Lecture 1)
- Pixel-level enhancement is closely related to **histogram** transformation



Histogram

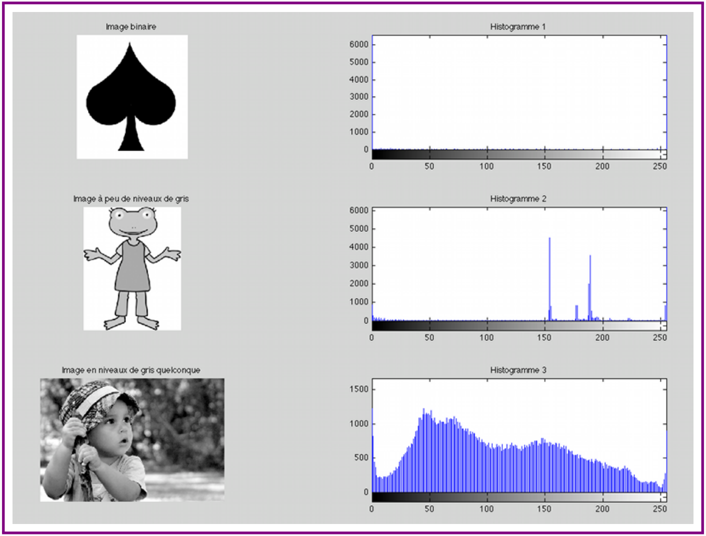
Definition

- Histogram is an array / function describing the image values (intensities / gray values / colors) distribution
- Provides image-specific information, such as:
  - The statistical distribution of image values
  - Minimal and maximal image values
- But no spatial information!
- For every image f of size $N \times M$, we can calculate the histogram H describing image values distribution:

$$H(k) = |\{(i,j) \mid 0 \leq i \leq N-1, 0 \leq j \leq M-1, f(i,j) = k\}| = n_k$$

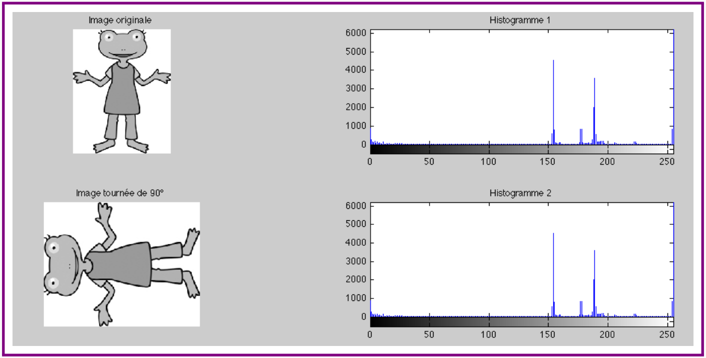
where $|.|$ denotes the cardinality

Some examples of image histogram



Some remarks about histogram

Histogram does not contain spatial information!



- Two semantically different images may have the same histogram

Normalized histogram

Definition

- Function H_n representing the probability (occurrence frequency here) for a pixel to have a given value k:

$$H_n(k) = \frac{H(k)}{N \times M}$$

  with N and M are the image dimensions
- Values of H_n are normalized ($\in [0, 1]$)
- Discrete approximation of **probability density function** (pdf) of the random variable X "value of a pixel": the image is seen as a realization of this random variable:
  $\hookrightarrow H_n(k) = P(X = k)$

Cumulative histogram

Definitions

- Cumulative histogram:

$$H_c(k) = \sum_{i \leq k} H(i)$$

  with $H(.)$ histogram

- Normalized cumulative histogram:

$$H_{nc}(k) = \sum_{i \leq k} H_n(i)$$

  with $H_n(.)$ the normalized histogram

- $H_{nc}(k)$ is the discrete approximation of the **cumulative distribution function** (cdf): $H_{nc}(k) = P(X \leq k)$

Cumulative histogram

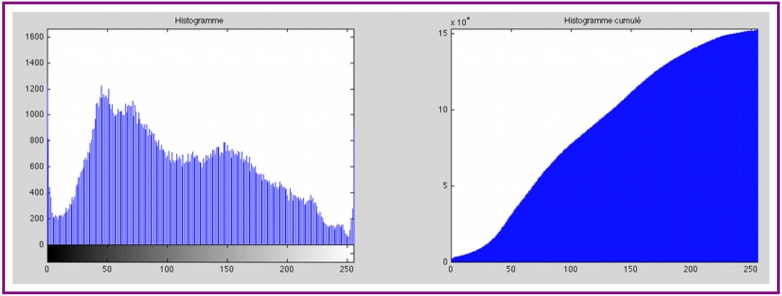


Image enhancement

Histogram transformations

- Change the intensity value k
  $T : k \mapsto k' = T(k)$.
- Various choices of T and their impact in the resulting image

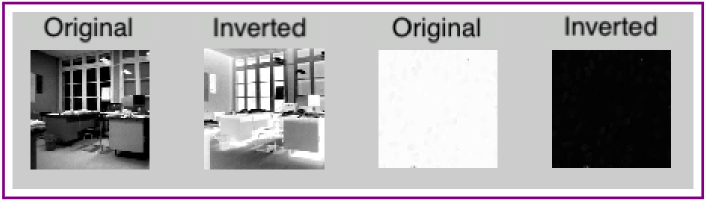
Image negative

Definition

- negative of the image obtained by the negative transformation in the range of $[0, L-1]$:

$$k' \;=\; L - 1 - k$$

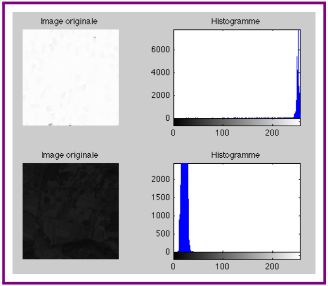
with L the dynamic range of the image (number of intensity levels)



- Dynamic range is unchanged

Back to the initial problem

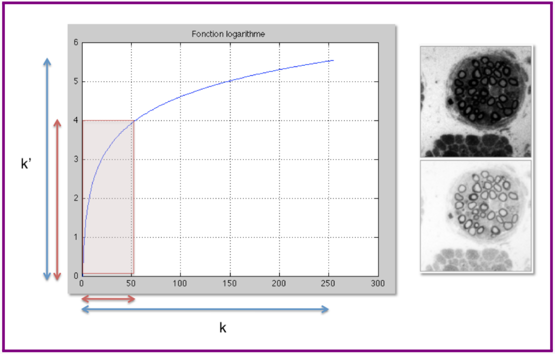
- Too bright or too dark images



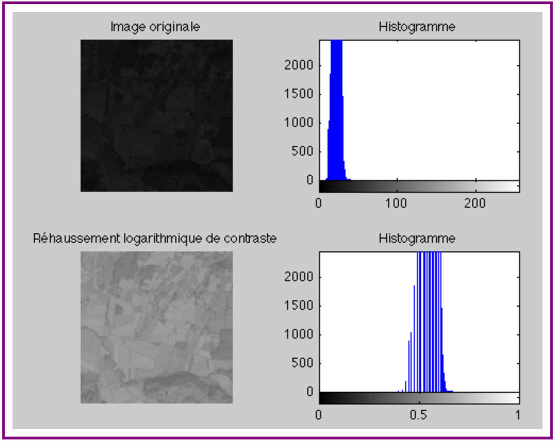
- Analysis of histograms: most of the values are distributed on the right (high values, too bright), or on the left (low values, too dark)

Definition

- $k' = \log(k)$
- Low values increase, high values decrease: allows increasing the contrast in dark parts of the image

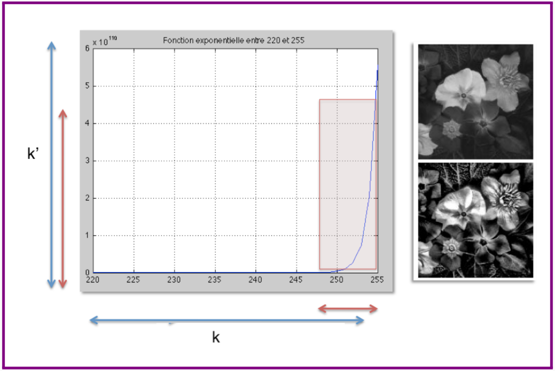


Logarithmic transformation

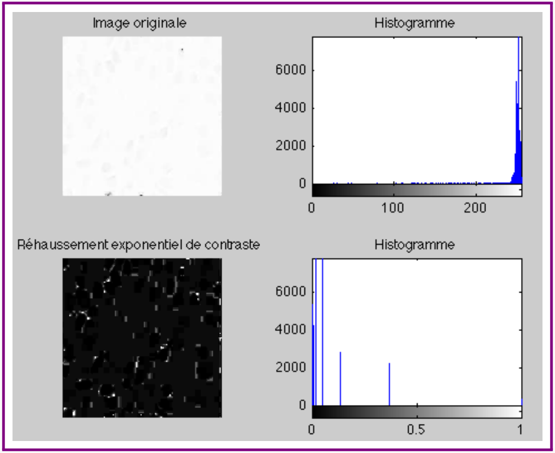


Definition

- $k' = e^k$
- Low values decrease, high values increase: allows increasing the contrast in bright parts of the image



Exponential transformation

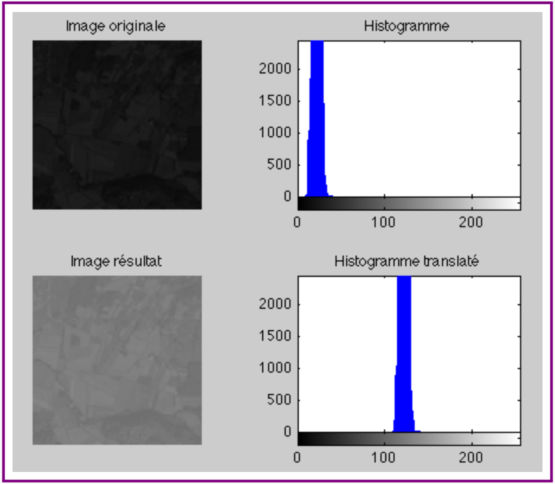


Histogram translation

Definition

- $k' = k + t$, with $t \in \mathbb{R}$ (or $\mathbb{N}$)
- Changes the brightness of an image, leaving the contrast unchanged
- The new image is brighter or darker
- Useful for images having a low dynamic range

Histogram translation



Affine transformation

Definition

- An affine transformation is applied on intensity levels:

$$k' = ak + b \quad \text{with} \quad a, b \in \mathbb{R}$$

- Contrast reduction: $a < 1$ and $b > 0$
- Contrast enhancement: $a > 1$ and $b < 0$
- Example: histogram stretching

Histogram stretching

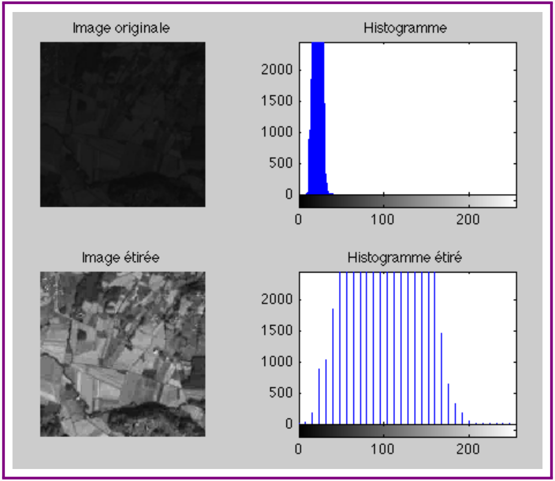
Definition: image normalization

- Let k_{min} and k_{max} be the minimal and maximal intensity levels of an image, respectively:
- Transformation:

$$k' = \frac{L-1}{k_{\text{max}} - k_{\text{min}}}(k - k_{\text{min}}), \quad k \in [k_{\text{min}}, k_{\text{max}}]$$

- After transformation, $k' \in [0, L-1]$, contrast is maximal
- No loss of information (same number of intensity levels)
- Before visualization, an image is often normalized (but not necessarily)

Histogram stretching



Histogram stretching

Linear transformation with saturation

- Let S_{min} et S_{max} be two threshold values such that:

$$k_{\text{min}} \leq S_{\text{min}} < S_{\text{max}} \leq k_{\text{max}}$$

- Transformation: $k' = \frac{L-1}{S_{\text{max}} - S_{\text{min}}}(k - S_{\text{min}})$
- Loss of information: values in $[k_{\text{min}}, S_{\text{min}}[$ and $]k_{\text{max}}, S_{\text{max}}]$ are lost
- Example: 1-byte image coding (256 intensity levels):

$$\begin{aligned} k' < 0 \quad &\rightarrow \quad k' = 0 \\ k' > 255 \quad &\rightarrow \quad k' = 255 \end{aligned}$$

Histogram stretching

General case?

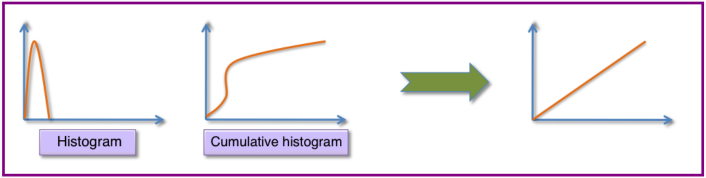
- We can choose a final interval $[l_{\min}, l_{\max}]$ different from $[0, L-1]$
- Affine transformation from $[k_{\min}, k_{\max}]$ to $[l_{\min}, l_{\max}]$:

$$k' = l_{\min} + \frac{l_{\max} - l_{\min}}{k_{\max} - k_{\min}}(k - k_{\min})$$

Histogram equalization

Definition

- a transformation on intensity levels leading to a flat histogram:
    - each intensity level is represented in the same proportion
    - regions of lower local contrast gain a higher contrast
    - global contrast increases
- Practically, the transformation is applied on the cumulative histogram



Histogram equalization

Definition

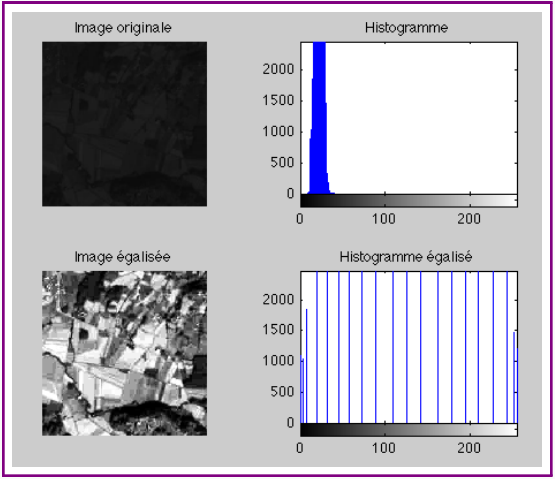
- The transformation is (details given in practical work):

$$k' = \texttt{Int}\left(\frac{L-1}{N \times M} H_c(k)\right)$$

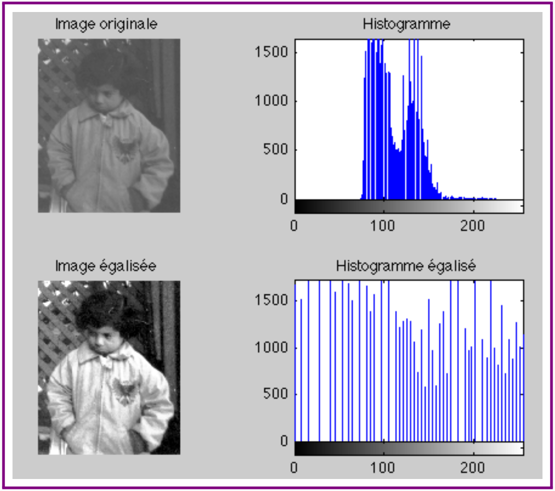
with:

- L the image dynamic range
- N and M the image size
- $H_c(k)$ the cumulative histogram
- `Int` rounding to the nearest integer

Histogram equalization: example 1



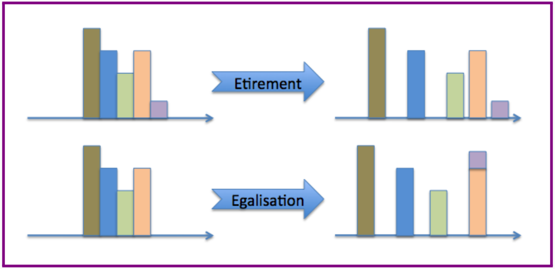
Histogram equalization: example 2



Histogram stretching versus histogram equalization: same operation?

No!

- Stretching: changes the bins distribution in the histogram, but not their size
- Equalization: changes the bins distribution in the histogram, and their size



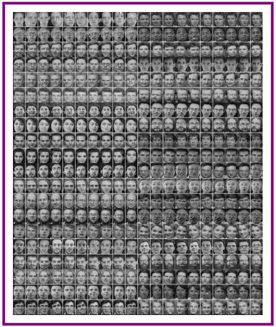
Histogram modification: some applications

Image mosaic

- Data: a target image and a base of small images

Machine learning on an image database

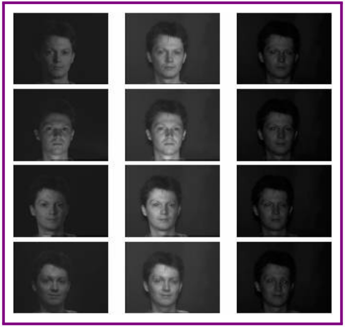
- Data: base of face images



Histogram modification: some applications

Machine learning on an image database

- Issue: brightness variation within the database
  $\hookrightarrow$ image normalization for a similar dynamic range and intensity level mean



Histogram modification: some applications

Segmentation

- Histogram of color image (huge table!)
- Clustering-based method: k-means (generalization of Otsu method with k classes)