Summary

I. Introduction
   - DIP, Examples, Fundamental steps, components

II. Digital Image Fundamentals
   - Visual perception, light
   - Image sensing, acquisition, sampling, quantization
   - Linear, and non linear operation

III. Discrete 2D Processing
   - Vector space, Convolution
   - Unitary Transformation

IV. Image Improvement
   - Enhancement, restoration, geometrical modifications
Image Improvement

- Image improvement denotes three types of image manipulation processes:
  - Image enhancement entails operations that improve the appearance to a human viewer, or operations to convert an image to a format better suited to machine processing.
  - Image restoration has commonly been defined as the modification of an observed image in order to compensate for defects in the imaging system that produced the observed image.
  - Geometrical image modification includes image magnification, minification, rotation and nonlinear spatial warping.

Image Improvement

- Image enhancement
  - Contrast and histogram
  - Noise cleaning
  - Edge enhancement
  - Color/multispectral image enhancement

- Image restoration

- Geometrical image modification

Image Enhancement

- Improve the visual appearance of an image or to convert the image to a form better suited for analysis by a human or a machine
- A lot of techniques exist
- There is no general unifying theory of image enhancement at present because there is no general standard of image quality that can serve as a design criterion for an image enhancement processor

Contrast improvement

- The most common defects of photographic or electronic images is poor contrast resulting from a reduced, and perhaps nonlinear, image amplitude range
- Image contrast can often be improved by amplitude rescaling of each pixel
  - Histogram
  - Transformation functions
Histogram

- Number of pixels that have a given intensity value

\[
\begin{array}{ccc}
1 & 2 & 1 \\
2 & 1 & 1 \\
1 & 2 & 3 \\
\end{array}
\]

image

- Similar to the probability density function

\[ p(i) = \frac{h(i)}{\text{nb of pixels}} \]

Examples

Mean = 100
Mean = 130
\[ \sigma = 20 \]
Histogram manipulation

- Use a transformation function
  - Try an existing (classical) one
  - Build your own!

\[ T(u) \]

Input Range

Output Range

\[ v = f(u) \]

\[ v = f(u) \]

\[ v = f(u) \]
Histogram manipulation

- Brightness and contrast

![Brightness and contrast diagram](image)

- Grey level histogram equalization

\[
\int_{g_{\text{min}}}^{g} p_g(g) \, dg = \int_{f_{\text{min}}}^{f} p_f(f) \, df
\]

\[
\int_{g_{\text{min}}}^{g} p_g(g) \, dg = P_f(f)
\]
Histogram manipulation

- Histogram equalization

\[ \int_{g_{\min}}^{g} p_g(g) \, dg = P_f(f) \]

- Examples

  - The output density is forced to be the uniform density

    \[ p_g(g) = \frac{1}{g_{\max} - g_{\min}} \quad g_{\min} \leq g \leq g_{\max} \]

  \[ g = (g_{\max} - g_{\min}) P_f(f) + g_{\min} \]

  - Other functions for the output density (exponential, logarithmic)

    \[ p_g(g) = \alpha \exp\{-\alpha(g - g_{\min})\} \quad g \leq g_{\min} \quad \Rightarrow \quad g = g_{\min} - \frac{1}{\alpha} \ln\{1 - P_f(f)\} \]

    \[ p_g(g) = \frac{1}{g[\ln\{g_{\max}\} - \ln\{g_{\min}\}]} \quad \Rightarrow \quad g = g_{\min}\left(\frac{g_{\max}}{g_{\min}}\right)^{P_f(f)} \]

- Histogram manipulation, example

  - X-ray projectile image and histogram

  - Transfer function \((P_f(f))\)

  - Histogram equalized image
Histogram manipulation

- Threshold

![Histogram manipulation example](image)

- Threshold

![Histogram manipulation example](image)
Histogram manipulation

- Limitations
  - Histogram equalization is not well adapted for good quality images
  - Histogram threshold is not a noise removing technique!
  - Histogram equalization should be adaptive!
    - Some methods exist (local equalization)

Local Histogram analysis

- features measured on the smallest neighborhood (1 pixel) grey-level (NG), color, quantitative value (Bq/cc, ...) ...

- features measured on a neighborhood → local histogram
Common computed values from the density probability function \( p(x) \) (based on histogram, local or not)

- **Moments**
  \[
  m_i = E[x^i] = \sum_{n=0}^{N-1} x^i p(x)
  \]
  \( N: \) number of grey levels

- **Centered Moments**
  \[
  \hat{m}_i = E[(x - E[x])^i] = \sum_{n=0}^{N-1} (x - m_1)^i p(x)
  \]

- **Entropy**
  \[
  H = -\sum_{x=0}^{N-1} p(x) \log_2(p(x))
  \]

- **Absolute moments, median value, max/min value, mode, percentiles, invariant moments (Legendre, …) …**

Computed on a 16x16 neighborhood

- Variance
- Mean

128x128 pixels
Noise cleaning

- Mean filter (linear filtering)
- Median filter (nonlinear filtering)
- Frequency domain filter (LP, HP, band-rejection (notch), …)
- …

Many types of noise…

Gaussian Noise (sd 25)  Salt and pepper noise
Noise cleaning

- Periodic and quasi periodic noise

![Quasi periodic noise](image1)

![Fourier amplitude spectrum](image2)

Linear filtering

- Mean filter
  \[ g(i, j) = \sum_{(k,l) \in W} h(k,l) f(i-k, j-l) \]
  
  W: 25 neighbors
  \[ H = \frac{1}{25} I \]

Result on Gaussian noise

Result on Salt & Pepper noise
Linear filtering

- Mean filter and DFT

\[ h(x,y) \]

\[ H(u,v) \]

\[ \rightarrow \text{Anisotropic low pass filter with poor selectivity} \]

Linear filtering

- Mean filter anisotropy, (mask 11x11)
**Linear filtering**

- **Gaussian Filter**

\[ h(x, y) = \frac{1}{2\pi \sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \]

Filter with infinite impulse response! \( \rightarrow \) Approximate the ideal filter by truncating and windowing the infinite impulse response to make a FIR

![h(x,y) vs. H(u,v)](image)

Isotropic low pass filter with poor selectivity

H(u,v) is a gaussian

\( \rightarrow \) Many other types of filter

\( \rightarrow \) high pass, low pass, band-stop, derivative…

\( \rightarrow \) Ideal, Butterworth,…

**Non linear filtering**

- **Median filter**

  - Replace the central value by the median value of the neighborhood

  ![Median filter example](image)
Non linear filtering

- Median filter
  - Advantage of median filtering over linear filtering: edges are preserved

  Mean Filter
  width = 3

  Median Filter
  neighborhood = 3

Non linear filtering

- Median filter

Result on Gaussian noise

W=25

W=9

Result on salt & pepper noise
Frequency domain filtering

- **Approach**
  - Region selection in the frequency space \((u,v)\)

\[
\begin{align*}
  f(i,j) & \xrightarrow{\text{DFT}} F(u,v) \times H(u,v) = F'(u,v) \xrightarrow{\text{DFT}}^{-1} f'(i,j)
\end{align*}
\]

- to keep \(f'(i,j)\) real, regions must be symmetric about the origin
- In frequency space \((u,v)\), region boundaries can be
  - Steep (but … oscillations can appear: Gibbs)
  - Smooth (but less selective)

---

Frequency domain filtering

- **Approach**
  - Many forms for regions

  - Choice of one direction
  - Choice of the direction and frequency bands
  - Choice of frequency bands
  - Mix!

- \(\rightarrow\) Low-pass filters
- \(\rightarrow\) High-pass filters
- \(\rightarrow\) Band-pass filters
- \(\rightarrow\) Band-reject filters…
Frequency domain filtering

- Notch filter
  Periodic noise

Clean dots and lines

Edge enhancement

**Edges**: Changes or discontinuities of amplitude in an image

Edges provide an indication of the physical extent of objects within the image ➔ **Contours**

- Edge detection
  - Differential detection
  - Model fitting

- Edge enhancement filters
Edge enhancement

- Edge detection

![Edge detection diagram]

What are a change or a discontinuity?

What about the direction (in image)?

What are changes or discontinuities in the context of images?

Contour

Contour in which direction?

What are changes or discontinuities in the context of images?

What about the direction (in image)?
Definition of continuous contour and gradient

\[ f(x, y) \]

Continuous one-dimensional gradient \( g(x, y) \) of \( f(x, y) \) along a line normal to the edge slope which is at an angle \( \theta \) with respect to the horizontal axis:

\[
\frac{\partial f(x, y)}{\partial r} \max \text{ for } \theta_g = \theta + \frac{\pi}{2} \quad \Rightarrow \quad \frac{\partial}{\partial \theta} \left( \frac{\partial f(x, y)}{\partial r} \right) = 0
\]

\[
g(x, y) = \nabla f(x, y) = \left\{ \frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right\}
\]

\[
\frac{\partial f}{\partial r} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial r} = \frac{\partial f}{\partial x} \cos(\theta) + \frac{\partial f}{\partial y} \sin(\theta)
\]

\[
\frac{\partial}{\partial \theta} \left( \frac{\partial f(x, y)}{\partial r} \right) = 0 \quad \Rightarrow \quad -\frac{\partial f}{\partial x} \sin(\theta) + \frac{\partial f}{\partial y} \cos(\theta) = 0
\]

Direction

\[
\theta_g = \arctan \left( \frac{\partial f}{\partial y} \right) \left( \frac{\partial f}{\partial x} \right)
\]

Amplitude

\[
\left( \frac{\partial f}{\partial r} \right)_{\max} = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2}
\]
Gradient in discrete domain

- For each pixel \((i,j)\): gradient computation in two orthogonal directions \(D_x, D_y\)

- Gradient amplitude \(M = \sqrt{D_x^2 + D_y^2}\)

- Gradient direction \(\theta = \arctan \left( \frac{D_y}{D_x} \right)\)

\[f(i,j)\]

\[D_x \quad \uparrow \quad M \quad \downarrow \quad \theta\]

\[D_y\]

Amplitude map

Direction map

For computational efficiency, the gradient amplitude is sometimes approximated by the magnitudes combination

\[M = \left| D_x \right| + \left| D_y \right|\]

If the gradient amplitude \(M\) is large enough (i.e., above some threshold value), an edge is deemed present.

The direction (angle) map is used to follow the contour.

Many \(H_1 H_2\) operators exist:

Pixel difference

\[D_x (i, j) = f(i, j) - f(i - 1, j)\]

\[D_y (i, j) = f(i, j) - f(i, j - 1)\]

Separated pixel difference

\[D_x (i, j) = f(i + 1, j) - f(i - 1, j)\]

\[D_y (i, j) = f(i, j + 1) - f(i, j - 1)\]

Convolution windows!
**H₁, H₂**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Row gradient H₁</th>
<th>Column gradient H₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel difference</td>
<td>0 0 0</td>
<td>0 -1 0</td>
</tr>
<tr>
<td></td>
<td>0 1 -1</td>
<td>0 1 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Separated pixel difference</td>
<td>0 0 0</td>
<td>0 -1 0</td>
</tr>
<tr>
<td></td>
<td>1 0 -1</td>
<td>0 0 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>Roberts</td>
<td>0 0 -1</td>
<td>-1 0 0</td>
</tr>
<tr>
<td></td>
<td>0 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Prewitt</td>
<td>1 0 -1</td>
<td>-1 -1 -1</td>
</tr>
<tr>
<td></td>
<td>1 0 -1</td>
<td>0 0 0</td>
</tr>
<tr>
<td></td>
<td>1 0 -1</td>
<td>1 1 1</td>
</tr>
<tr>
<td>Sobel</td>
<td>1 0 -1</td>
<td>-1 -2 -1</td>
</tr>
<tr>
<td></td>
<td>2 0 -2</td>
<td>0 0 0</td>
</tr>
<tr>
<td></td>
<td>1 0 -1</td>
<td>1 2 1</td>
</tr>
<tr>
<td>Frei–Chen</td>
<td>( \frac{1}{2+\sqrt{2}} ) ( \begin{bmatrix} 1 &amp; 0 &amp; -1 \ \sqrt{2} &amp; 0 &amp; -\sqrt{2} \ 1 &amp; 0 &amp; -1 \end{bmatrix} )</td>
<td>( \frac{1}{2+\sqrt{2}} ) ( \begin{bmatrix} -1 &amp; -\sqrt{2} &amp; -1 \ 0 &amp; 0 &amp; 0 \ 1 &amp; \sqrt{2} &amp; 1 \end{bmatrix} )</td>
</tr>
</tbody>
</table>

Département GE - DIP - Thomas Grenier

**Roberts**

Amplitude

- Zoom (inhomogeneities!)

- white=\( \pi/4 \)
- grey = \( \pi + \pi/4 \)
- black = \( 2\pi + \pi/4 \)

128x128
Increasing the contrast, many edges appear due to noise

→ Edge detectors are high-pass filters
**Compass operator**

- Computation of the gradient in N directions
- Selection of the maximum value

![Diagram showing the compass operator](image)

**Examples**

<table>
<thead>
<tr>
<th>Gradient direction</th>
<th>Prewitt compass gradient</th>
<th>Kirsch</th>
<th>Robinson 3-level</th>
<th>Robinson 5-level</th>
</tr>
</thead>
<tbody>
<tr>
<td>East</td>
<td>[1 1 -1; 1 -2 -1]</td>
<td>[5 -3 -3; 5 0 -3]</td>
<td>[1 0 -1]</td>
<td>[1 0 -1]</td>
</tr>
<tr>
<td>Northeast</td>
<td>[1 -1 -1; 1 -2 -1]</td>
<td>[-3 -3 -3; 5 0 -3]</td>
<td>[1 0 -1]</td>
<td>[0 -1 -2]</td>
</tr>
<tr>
<td>North</td>
<td>[1 1 1; 1 -2 1]</td>
<td>[-3 0 -3; 5 5 5]</td>
<td>[1 0 -1]</td>
<td>[1 0 -1]</td>
</tr>
<tr>
<td>Northwest</td>
<td>[-1 -1 1; 1 -2 1]</td>
<td>[-3 -3 -3; -3 0 5]</td>
<td>[1 0 -1]</td>
<td>[-1 -2 -1]</td>
</tr>
<tr>
<td>West</td>
<td>[-1 1 1; 1 -2 1]</td>
<td>[-3 -3 5; -3 0 5]</td>
<td>[-1 0 1]</td>
<td>[-2 -1 0]</td>
</tr>
<tr>
<td>Southwest</td>
<td>[-1 -1 1; 1 -2 1]</td>
<td>[-3 5 5; -3 0 5]</td>
<td>[1 0 1]</td>
<td>[-1 0 1]</td>
</tr>
<tr>
<td>South</td>
<td>[1 1 1; 1 -2 1]</td>
<td>[-3 3 -3; -3 0 5]</td>
<td>[1 1 1]</td>
<td>[0 1 2]</td>
</tr>
<tr>
<td>Southeast</td>
<td>[1 1 1; 1 -2 1]</td>
<td>[5 5 -3; -3 3 -3]</td>
<td>[1 1 0]</td>
<td>[2 1 0]</td>
</tr>
</tbody>
</table>

Scale factor: 1/5, 1/15, 1/3, 1/4
\[ \Delta f(x, y) = \nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \]

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & -1 \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & -1 \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0
\end{bmatrix}
\]

then absolute value

\[
\begin{bmatrix}
0 & -1 & 0 \\
-1 & 4 & -1 \\
0 & -1 & 0
\end{bmatrix}
\]

**Emphasis filter**

\[
\begin{bmatrix}
0 & -1 & 0 \\
-1 & 4 & -1 \\
0 & -1 & 0
\end{bmatrix} \rightarrow \begin{bmatrix}
0 & -1 & 0 \\
-1 & 5 & -1 \\
0 & -1 & 0
\end{bmatrix}
\]

= Input Image + Laplacian

\[ \Rightarrow \text{Enhancement of high frequencies} \]
Unsharp masking, Highboost Filtering

- Used by the printing and publishing industry

1. Blur the original image
2. Subtract the blurred image from the original (the result is called the mask)
3. Add the mask (multiplied by $k$) to the original

Edge fitting

- Image data $f$ can be fitted to an ideal edge model $s$

$\text{MSE} = \int_{x_0-L}^{x_0+L} [f(x) - s(x)]^2 dx$

An edge is assumed present if the Mean Square Error is below a threshold value.

Model+minimization… image restoration
Image Restoration

- Image restoration attempts to recover an image that has been degraded using a priori knowledge of degradation phenomenon
  - Modeling the degradation
  - Applying the inverse process (in order to recover the original image)

- Involves formulating a criterion of goodness that will yield an optimal estimate of the desired result

A model of the image degradation/restoration process

\[
g(x, y) = h(x, y) \otimes f(x, y) + \eta(x, y)
\]

\[
G(u, v) = H(u, v).F(u, v) + N(u, v)
\]

- Original Image
- Noise
- Degraded image (input)
- Estimate of the original image
- Degradation (Sensor, digitizer, display, movement)
- Restoration
Noise Models

- **Gaussian (normal) noise**
  \[ p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(z-z_0)^2}{2\sigma^2}} \]

- **Uniform noise**
  \[ p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases} \]

- **Impulse (salt and pepper) noise**
  \[ p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases} \]

- **Erlang (gamma) noise**

- **Rayleigh noise**
  \[ p(z) = \begin{cases} \frac{2}{b} (z-a) e^{-\frac{(z-a)^2}{b}} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases} \]

Restoration: presence of noise only

\[ g(x, y) = f(x, y) + \eta(x, y) \]

- **Mean filters**
  - Arithmetic, geometric, harmonic
  \[ f(x, y) = \frac{1}{m.n} \sum_{i,j \in S_{xy}} g(i, j) \quad \frac{1}{m.n} \prod_{i,j \in S_{xy}} g(i, j) \quad \frac{1}{m.n} \sum_{i,j \in S_{xy}} \frac{1}{g(i, j)} \]

- **Order statistic filters**
  - Median, min & max, midpoint
  \[ \hat{f}(x, y) = \text{median}_{s,t \in S_{xy}} (g(s,t)) \quad \hat{f}(x, y) = \max_{s,t \in S_{xy}} (g(s,t)) \quad \hat{f}(x, y) = \min_{s,t \in S_{xy}} (g(s,t)) \quad \frac{1}{2} \max + \frac{1}{2} \min \]

- **Adaptive filters**
  - Local noise reduction, adaptive median, …
Adaptive Median Filter

Notations
- \( z_{\text{min}} \) = minimum intensity value in \( S_{xy} \)
- \( z_{\text{max}} \) = maximum intensity value in \( S_{xy} \)
- \( z_{\text{med}} \) = median of intensity values in \( S_{xy} \)
- \( z_{xy} \) = intensity value at coordinates \((x,y)\)
- \( S_{\text{max}} \) = maximum allowed size of \( S_{xy} \)

Algorithm for a pixel \((x,y)\)

Stage 1:
- Compute \( z_{\text{min}}, z_{\text{max}}, z_{\text{med}} \)
- \( A1 = z_{\text{med}} - z_{\text{min}} \)
- \( A2 = z_{\text{med}} - z_{\text{max}} \)
- if \( A1 > 0 \) and \( A2 < 0 \) → stage 2
- increase the window size
- if window size < \( S_{\text{max}} \) → stage 1
- else output \( z_{\text{med}} \)

Stage 2:
- \( B1 = z_{xy} - z_{\text{min}} \)
- \( B2 = z_{xy} - z_{\text{max}} \)
- if \( B1 > 0 \) and \( B2 < 0 \)
- output \( z_{xy} \)
- else output \( z_{\text{med}} \)

Result of Adaptive Median filter

Reference

Noisy image (S&P, 0.35)

Median Filter (5x5)

Adaptive Median (\( S_{\text{max}} 9x9 \))
Restoration: Periodic noise reduction

\[ g(x, y) = f(x, y) + \eta(x, y) \]

- By frequency domain filtering
  - Bandreject filter
  - Notch filter \( \rightarrow \) optimum notch

Build \( H_{\text{NP}} \) (Notch Pass) by placing a notch pass filter at the location of each spike. Interference noise pattern is:

\[
N(u, v) = H_{\text{NP}}(u, v).G(u, v)
\]

then

\[
\eta(x, y) = \text{FT}^{-1}[N(u, v)]
\]

thus

\[
\hat{f}(x, y) = g(x, y) - w(x, y)\eta(x, y)
\]

Estimate of \( f(x, y) \)

Weighted function (minimizes the effect of components not present in the estimate of \( \eta \))

\( \Rightarrow \) How to select \( w(x, y) \) ?

- \( w(x, y) \)?

\( w(x, y) \) is selected so that the local variance of the estimate of \( f \) is minimized (optimum choice of \( w \))

\[
\frac{\partial \sigma^2(x, y)}{\partial w(x, y)} = 0
\]

\[
w(x, y) = \frac{g(x, y)\eta(x, y) - g(x, y)\bar{\eta}(x, y)}{\eta^2(x, y) - \bar{\eta}^2(x, y)}
\]

\( \Rightarrow \) Prove the validity of this equation

Hints:

\( \Rightarrow \) estimate the variance in a small neighborhood

\( \Rightarrow \) assume that \( w \) remains essentially constant over the neighborhood
Restoration: linear, position-invariant degradation

\[ g(x, y) = H[f(x, y)] + \eta(x, y) \]

- \( H \) is linear,
  \[ H[af_1(x, y) + bf_2(x, y)] = aH[f_1(x, y)] + bH[f_2(x, y)] \]
- If \( H \) is position invariant then (for any \( a \) and \( b \)):
  \[ H[\delta(x-a, y-b)] = h(x-a, y-b) \]

\( \Rightarrow g(x, y) \):

\[ g(x, y) = h(x, y) \otimes f(x, y) + \eta(x, y) \]
\[ G(u, v) = H(u, v)F(u, v) + N(u, v) \]

Estimating the degradation function

Blind deconvolution

- 3 principal ways
  - Observation
    - Small rectangular section containing samples structures (part of an object, background)
  - Experimentation
    - Obtain the impulse response of the degradation function by imaging an impulse (small dot of light)
  - Modeling
    - Mathematical model that take into account environmental conditions that cause degradation
    - Derive a mathematical model starting from basic principles
And after?

- Inverse filtering
  - Without noise
    \[ \hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \]
  - With noise
    - We have to know \( N \)!
    - What happens for small values of \( H(u,v) \)?

- Minimum Mean Square Error (Wiener) Filtering

- Constrained Least Squares Filtering

- …

Geometrical image modification

- Spatial transformations
  - Example
    - Shrink image to half its size
    \[ (x', y') = T\{(x, y)\} = (x/2, y/2) \]

- Affine transform:
  \[
  \begin{bmatrix}
  x' \\
  y'
  \end{bmatrix} = \begin{bmatrix}
  t_{11} & t_{12} & 0 \\
  t_{21} & t_{22} & 0 \\
  t_{31} & t_{32} & 1
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  y \\
  1
  \end{bmatrix}
  \]

- Higher order
  \[
  \begin{bmatrix}
  x' \\
  y'
  \end{bmatrix} = \begin{bmatrix}
  x, y, x^2, y^2, xy, ..., 1
  \end{bmatrix} \cdot T
  \]

- Estimate (or compute) the inverse matrix
- If needed, use interpolation
Higher order transforms

Applications:
Lens distortion correction, perspective

- Orthoscopic projection
- Barrel distortion
- Pincushion distortion

End …
- Before texture
Analyse de texture

Région ≠ zone de NG ou de couleur homogène

Seuillage !

Détection de contour !

Texture = information visuelle qualitative:
Grossière, fine, tachetée, marbrée, régulière, périodique...

Région homogène: Assemblage plus ou moins régulier
de primitives plus ou moins similaires.

Analyse de texture = formalisation de ces critères

Texture microscopique: Aspect chaotique mais régulier,
primitive de base réduite.

Texture macroscopique: primitive de base
évidente, assemblage régulier.
Méthodes d’analyse de texture:

**Structurelles:** recherche de primitives de base bien définies et de leur organisation (règles de placement)

*Méthodes peu utilisées*

**Stochastiques:** primitives mal définies et organisation +/- aléatoire.

Principe: évaluation d’un paramètre dans une petite région (fenêtre de taille dépendant de la texture (!)):

Analyse fréquentielle, statistiques, comptage d’événements, corrélation,....

Pas de modèle général de texture → Nombreuses méthodes ad-hoc.

---

**Exemple de méthode: Matrices de co-occurrence**

Statistique du second ordre:

\[
\Pr(f(i,j) = a \text{ et } f(i+k,j+l) = b) = p(k,l; a, b) = p(d, \theta; a, b)
\]

\[
\begin{array}{c|cccc}
 0 & 1 & 2 & 3 \\
 0 & 2 & 2 & 1 & 0 \\
 1 & 0 & 2 & 0 & 0 \\
 2 & 0 & 0 & 3 & 1 \\
 3 & 0 & 0 & 0 & 1 \\
\end{array}
\]

\[
\begin{array}{c|cccc}
 0 & 1 & 2 & 3 \\
 0 & 4 & 2 & 1 & 0 \\
 1 & 2 & 4 & 0 & 0 \\
 2 & 1 & 0 & 6 & 1 \\
 3 & 0 & 0 & 1 & 2 \\
\end{array}
\]

(d = 1 , \theta = 0^\circ \quad (k=1, l=0))

(én symétrique

\[
\theta = 0^\circ \quad d = 1 \text{ et } d = -1
\]

DÉPARTAMENT GE - DIP - THOMAS GRENIER
Quelques Paramètres extraits des matrices de co-occurrence

Moyenne locale:
\[
\sum_{i=1}^{NG} \sum_{j=1}^{i} (i + j)p(i, j) 
\]

Energie ou second moment:
\[
\sum_{i=1}^{NG} \sum_{j=1}^{i} p(i, j)^2 
\]

Inertie ou moment d’ordre deux des différences :
\[
\sum_{i=1}^{NG} \sum_{j=1}^{i} (i - j)^2 p(i, j) 
\]

Autocorrélation:
\[
\sum_{i=1}^{NG} \sum_{j=1}^{i} i.j p(i, j) 
\]

Contraste:
\[
\sum_{i=1}^{NG} \sum_{j=1}^{i} (i + j)^2 p(i, j) 
\]

- Il y en a d’autres ....
- L’interprétation visuelle est difficile.

Application de l’analyse de texture

Mesure de paramètres dans une fenêtre de taille K,L
Avec un pas de déplacement Dx, Dy
Application des matrices de co-occurrence

Fenêtre 16x16, pas 2x2, $k=1$, $l=0$

Influence des paramètres

- Exemple : Choix de la taille de la fenêtre

(Matrice de co-occurrence : Pseudo-variance)

Le choix et les réglages des paramètres sont difficiles. Il faut souvent faire de nombreux essais.

Les paramètres obtenus doivent être pertinents pour l’opération suivante de segmentation.