INFO0948
Image Processing

Renaud Detry

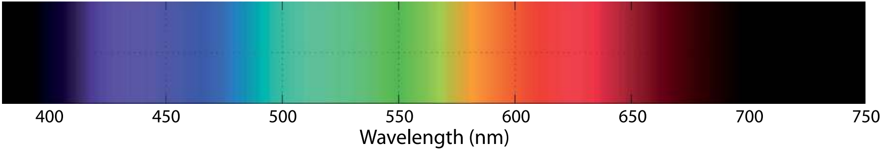
University of Liège, Belgium

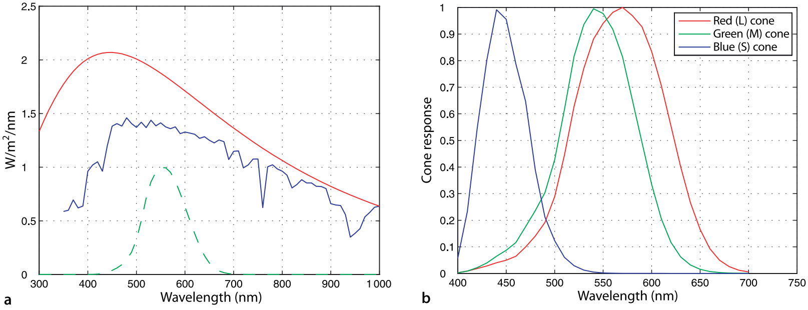
Updated November 20, 2013

These slides are based on Chapter 12 of the book *Robotics, Vision and Control: Fundamental Algorithms in MATLAB* by Peter Corke, published by Springer in 2011.

Plan

The Spectral Representation of Light, Color, and RGB





Human visual system, light and colors II

color	wavelength interval $\lambda\,[m]$	frequency interval $f\,[Hz]$
purple	$\sim$ 450–400 $[nm]$	$\sim$ 670–750 $[THz]$
blue	$\sim$ 490–450 $[nm]$	$\sim$ 610–670 $[THz]$
green	$\sim$ 560–490 $[nm]$	$\sim$ 540–610 $[THz]$
yellow	$\sim$ 590–560 $[nm]$	$\sim$ 510–540 $[THz]$
orange	$\sim$ 635–590 $[nm]$	$\sim$ 480–510 $[THz]$
red	$\sim$ 700–635 $[nm]$	$\sim$ 430–480 $[THz]$

Figure : Visible colors (remember that $\lambda = \frac{3\times10^8}{f}$).

Frequency representation of colors

$$\int_{\lambda} L(\lambda)\, d\lambda \tag{1}$$

Impossible from a practical perspective because this would require *one sensor for each wavelength.*

Solution: use colorspaces

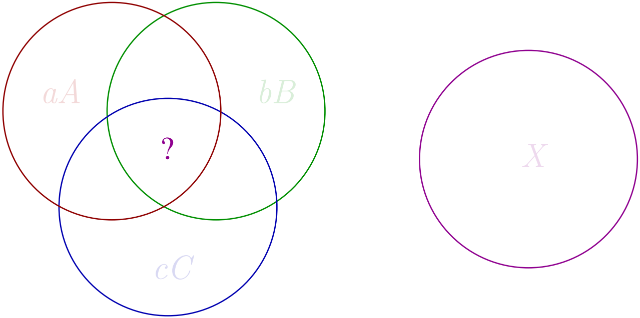


Figure : Equalization experiment for colors. The aim is to mix A, B, and C to get as close as possible to X.

Frequency representation of colors

$$\int_{\lambda} L(\lambda)\, d\lambda \tag{1}$$

Impossible from a practical perspective because this would require *one sensor for each wavelength.*

Solution: use colorspaces

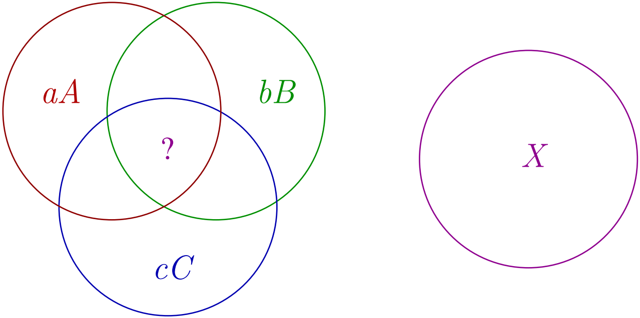


Figure : Equalization experiment for colors. The aim is to mix A, B, and C to get as close as possible to X.

The RGB additive colorspace

Three fundamental colors: red R $(700\,[nm])$, green G $(546,1\,[nm])$ and blue B $(435,8\,[nm])$,

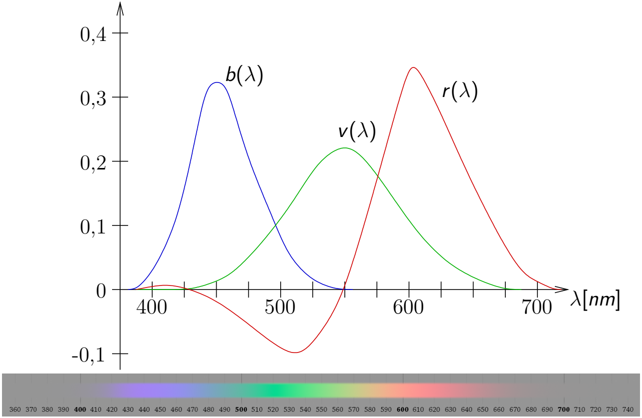


Figure : Equalization curves obtained by mixing the three fundamental colors to simulate a given color.

Images: Computer Representation

A color image is a 2D array of pixels.

Each pixel encodes a color, typically with a triplet (R, G, B).

A color pixel is usually encoded into 24 bits (8 bits per color).

As a result, the R, G, and B components take values between 0 and 255.

In image processing, it is not uncommon to represent colors with floating-point variables. In this case, R, G, and B usually scale between 0 and 1. Be careful that when converting back to 24-bit color, or writing to disk, the components must be scaled back to 0–255.

In Matlab, the commands `iread` and `idisp` read images from disk and display them.

Other colorspaces

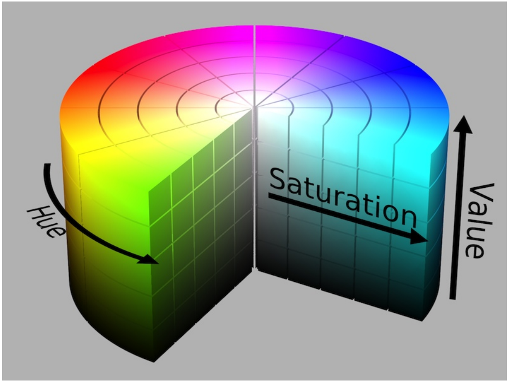
- a subtractive colorspace: Cyan, Magenta, and Yellow (CMY)
- Luminance + chrominances (YIQ, YUV or YC_bC_r)

In practice, most of the time, we use 8 bits to describe a color:

Hexadecimal				Decimal		
00	00	00		0	0	0
00	00	FF		0	0	255
00	FF	00		0	255	0
00	FF	FF		0	255	255
FF	00	00		255	0	0
FF	00	FF		255	0	255
FF	FF	00		255	255	0
FF	FF	FF		255	255	255

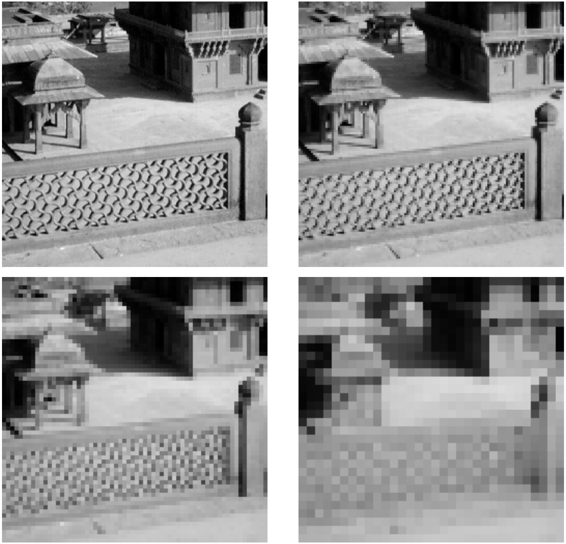
Table : Definition of RGB color values (8 bits) and conversion table between an hexadecimal notation and decimal notation.

Hue-Saturation-Value: an Intuitive Represenation of RGB



Hue is more robust to common changes of lighting conditions than saturation or value.

Resolution



The bitplanes of an image

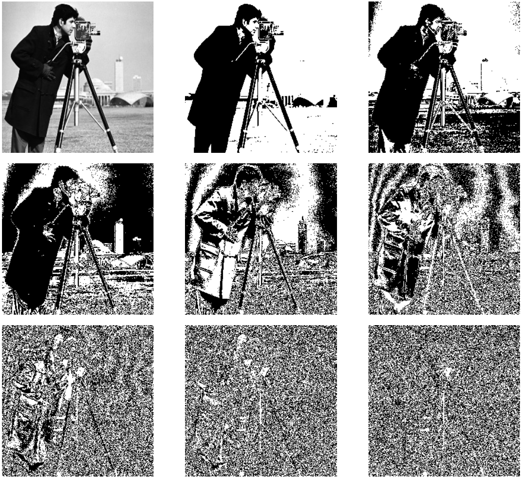


Table : An original image and its 8 bitplanes starting with the Most Significant Bitplane (MSB).

Technologies for dealing with 3D information

Acquisition

- Single monochromatic/color camera
- Multiple cameras (stereoscopy, network of cameras)
- Depth (range) cameras

Rendering

- Glasses
  - Color anaglyph systems

  - Polarization systems
- Display
  - Autostereoscopic display technologies

Depth cameras

There are two *acquisition* technologies for depth-cameras, also called range- or 3D-cameras:

- measurements of the deformations of a pattern sent on the scene (*structured light*).

  - first generation of the Kinects
- measurements by *time-of-flight* (ToF). Time to travel forth and back between the source led (camera) and the sensor (camera).

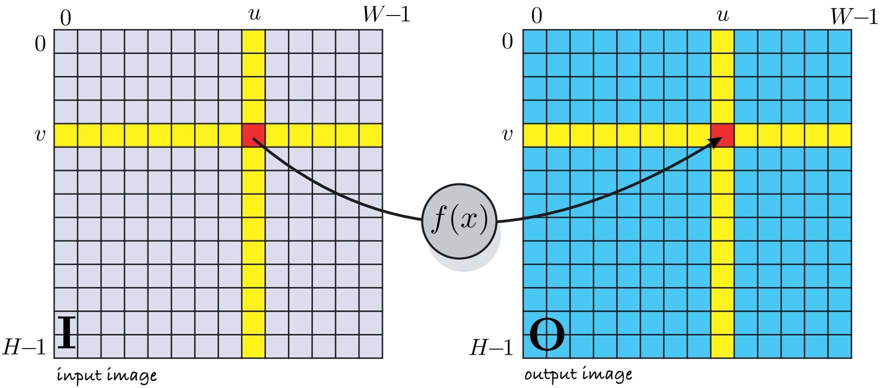
  - Mesa Imaging, PMD cameras
  - second generation of Kinects

Illustration of a depth map acquired with a range camera

Plan

Plan

Monadic Image Operations



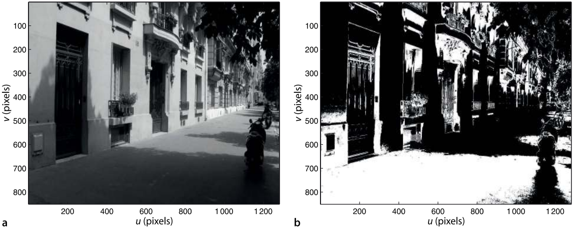
$$\boldsymbol{O}[u, v] = f\big(\boldsymbol{I}[u, v]\big), \quad \forall (u, v) \in \boldsymbol{I}$$

Simple Monadic Operations in Matlab

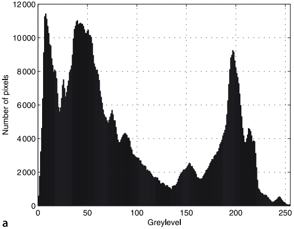
```
>> imd = idouble(im);
>> im = iint(imd);
>> grey = imono(im);
>> color = icolor(grey);
>> color = icolor(grey, [1 0 0]);
>> color = flipdim(im,3);
```

Intensity Histograms

```
>> street = iread('street.png');
>> ihist(street);
>> shadows = (street >= 30) & (street<= 80);
```



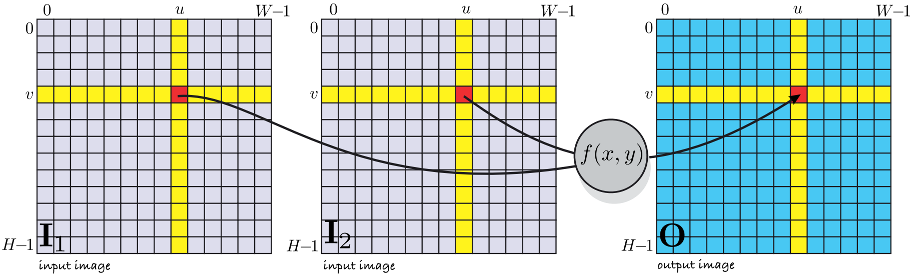
a b



a

Plan

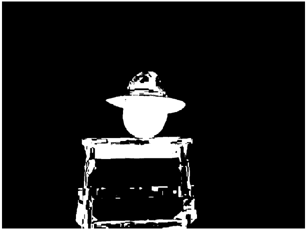
Diadic Operations



$$\boldsymbol{O}[u,v] = f\big(\boldsymbol{I}_1[u,v], \boldsymbol{I}_2[u,v]\big), \quad \forall (u,v) \in \boldsymbol{I}_1$$

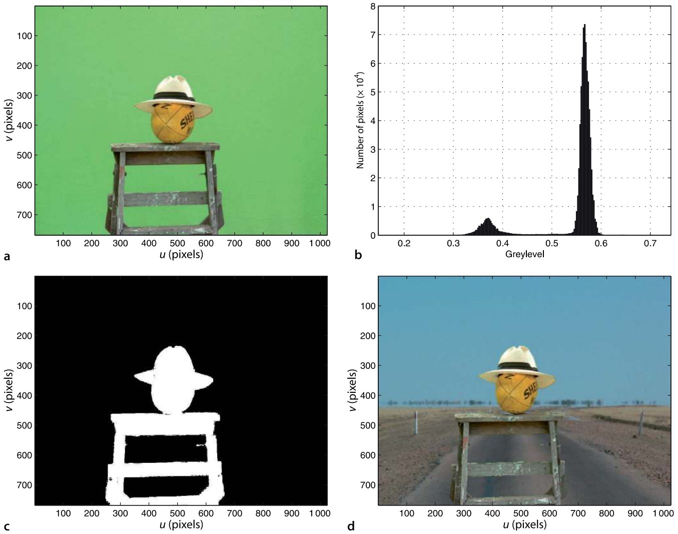
Chroma-keying

```
>> subject = iread('greenscreen.jpg', 'double');
>> r = subject(:,:,1);
>> g = subject(:,:,2);
>> b = subject(:,:,3);
>> mask = (g < r) | (g < b);
>> mask3 = icolor( idouble(mask) );
>> bg = isamesize(iread('road.png', 'double'), subject);
>> idisp( subject.*mask3 + bg.*(1-mask3) );
```



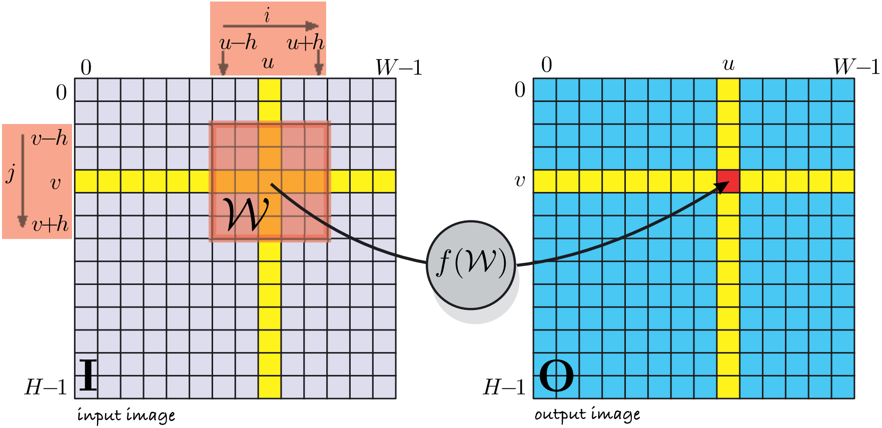
Chroma-keying

With a more sophisticated keying (see book)

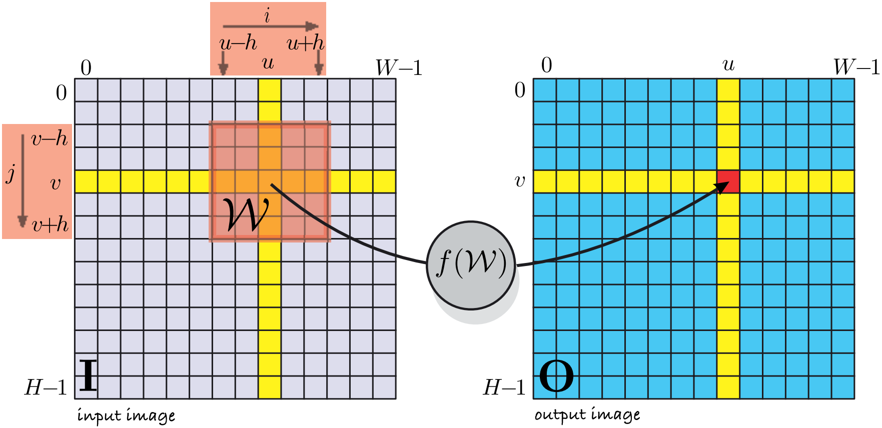


Plan

Spatial Operations



Convolution



$$\mathbf{O}[u,v] = \sum_{(i,j)\in\mathcal{W}} \boldsymbol{I}[u+i, v+j]\boldsymbol{K}[i,j], \quad \forall (u,v) \in \mathbf{I}$$

Computational cost: $O(h^2 W H)$

Convolution

Properties of convolution. Convolution obeys the familiar rules of algebra, it is commutative

$$A \otimes B = B \otimes A$$

associative

$$A \otimes B \otimes C = (A \otimes B) \otimes C = A \otimes (B \otimes C)$$

distributive (superposition applies)

$$A \otimes (B + C) = A \otimes B + A \otimes C$$

linear

$$A \otimes (\alpha B) = \alpha(A \otimes B)$$

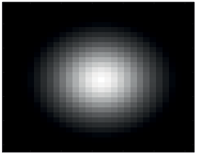
and shift invariant – if $S(\cdot)$ is a spatial shift then

$$A \otimes S(B) = S(A \otimes B)$$

that is, convolution with a shifted image is the same as shifting the result of the convolution with the unshifted image.

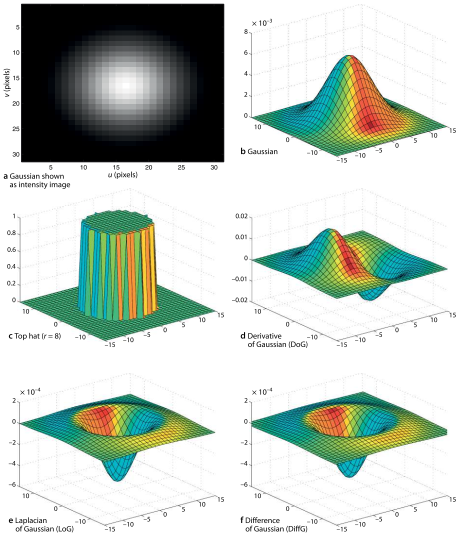
Smoothing

Gaussian kernel:



$$\mathbf{G}(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

Popular Kernels

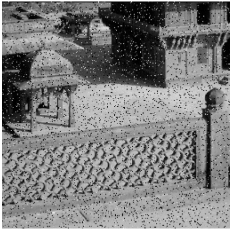


Median filter I

If n is odd, the $k = \frac{1}{2}(\sharp(B) + 1)$ choice leads to the definition of a self-dual operator, that is a filter that produces the same result as if applied on the dual function. This operator, denoted med_B, is the *median filter*.

$f(x)$	25	27	30	24	17	15	22	23	25	18	20
1		25	24	17	15	15	15	22	18	18	18
med_B	25	27	27	24	17	17	22	23	23	20	20
3	27	30	30	30	24	22	23	25	25	25	
$f \ominus B(x) = \min$		25	24	17	15	15	15	22	18	18	
$f \oplus B(x) = \max$		30	30	30	24	22	23	25	25	25	

Median filter II



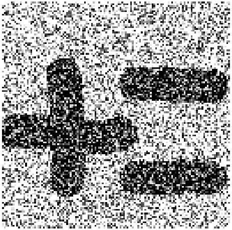
(a) Original image f + noise

(b) Opening with a 5×5 square

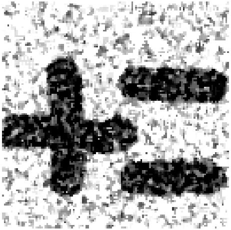
(c) Low-pass Butterworth ($f_c = 50$)

(d) Median with a 5×5 square

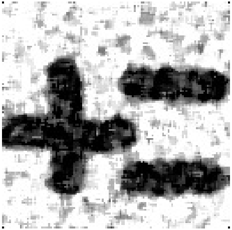
Effect of the size of the median filter



(a) Image f



(b) 3×3 median



(c) 5×5 median

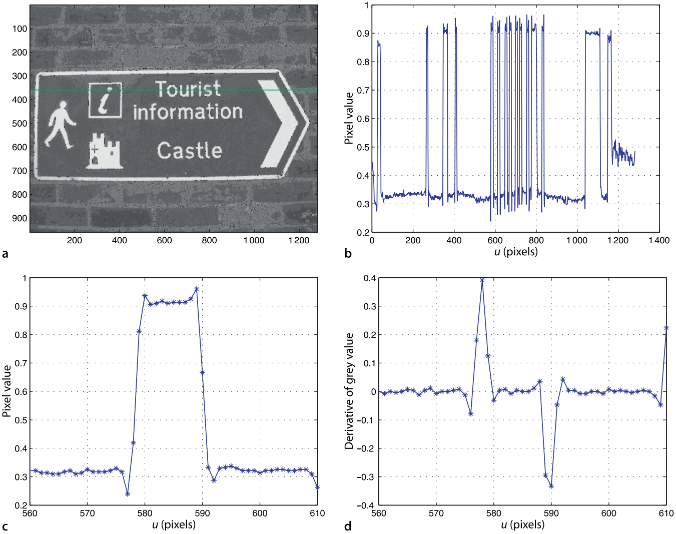
Edge Detection

$$p'[v] = p[v] - p[v-1]$$

$$p'[v] = \frac{1}{2}\big(p[v+1] - p[v-1]\big)$$

$$K = \begin{pmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$$

```
>> Du = ksobel
Du =
     -1         0         1
     -2         0         2
     -1         0         1
```



Practical expressions of gradient operators and convolution/multiplication masks I

Practical expression are based on the notion of convolution masks

$$\begin{bmatrix} +1 & -1 \end{bmatrix} \tag{190}$$

corresponds to the following non-centered approximation of the first derivate:

$$\frac{(-1) \times f(x,y) + (+1) \times f(x+h,y)}{h} \tag{191}$$

This "convolution mask" has an important drawback. Because it is not centered, the result is shifted by half a pixel. One usually prefers to use a centered (larger) convolution mask such as

$$\begin{bmatrix} +1 & 0 & -1 \end{bmatrix} \tag{192}$$

Practical expressions of gradient operators and convolution/multiplication masks II

In the y (vertical) direction, this becomes

$$\begin{bmatrix} +1 \\ 0 \\ -1 \end{bmatrix} \tag{193}$$

But then, it is also possible to use a diagonal derivate:

$$\begin{bmatrix} +1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{194}$$

Practical expressions of gradient operators and convolution/multiplication masks III

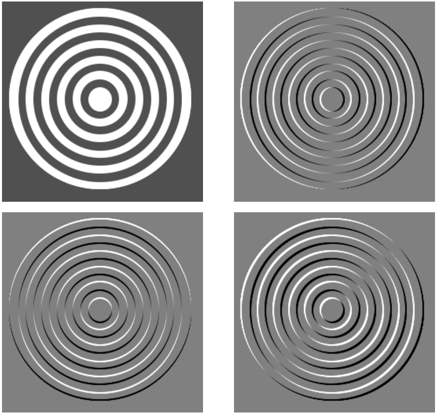


Figure : (a) original image, (b) after the application of a horizontal mask, (c) after the application of a vertical mask, and (d) mask oriented at 135^0.

Prewitt gradient filters

$$[h_x] = \frac{1}{3}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \tag{195}$$

$$[h_y] = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \tag{196}$$

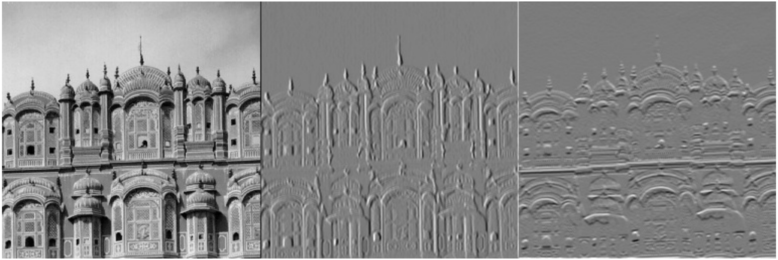


Figure : Original image, and images filtered with a horizontal and vertical Prewitt filter respectively.

Sobel gradient filters

$$[h_x] = \frac{1}{4}\left[\begin{array}{ccc} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{array}\right] = \frac{1}{4}\left[\begin{array}{c} 1 \\ 2 \\ 1 \end{array}\right]\left[\begin{array}{ccc} 1 & 0 & -1 \end{array}\right] \tag{197}$$

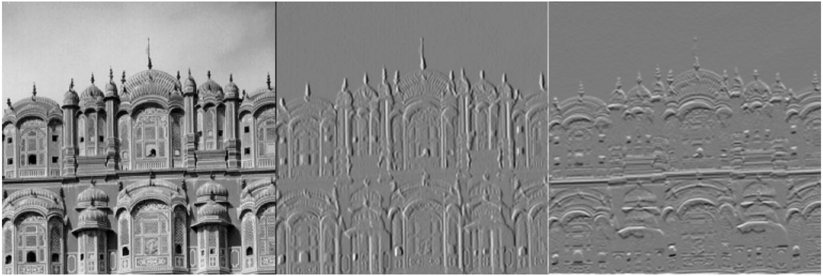
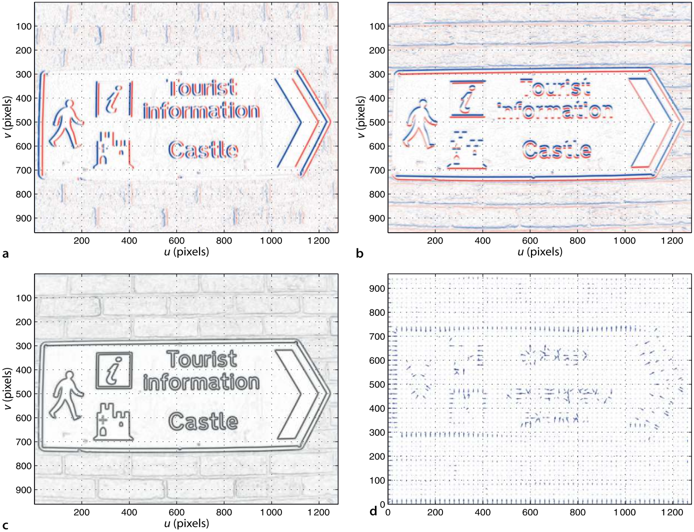


Figure : Original image, and images filtered with a horizontal and vertical Sobel filter respectively.

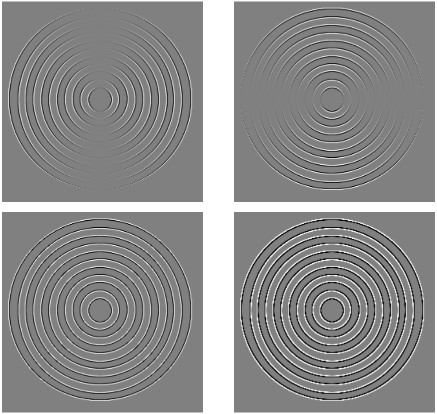
Edge Detection

```
>> Du = ksobel
Du =
    -1     0     1
    -2     0     2
    -1     0     1
```



Second derivate: basic filter expressions

$$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Noise-robust Edge Detection with Smoothing

- Noise is a stationary random process
- Edge pixels are correlated over large regions

$\rightarrow$ we can reduce the effect of noise with spatial smoothing.

$$\mathbf{I}_u = \mathbf{D} \otimes \big(\mathbf{G}(\sigma) \otimes \mathbf{I}\big)$$

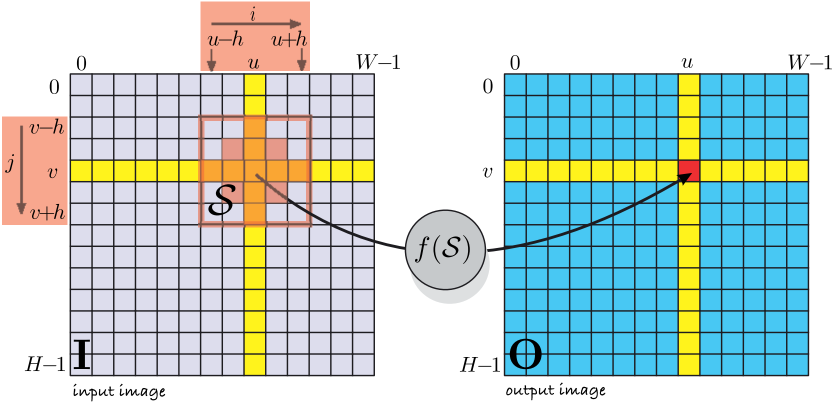
$$\nabla \mathbf{I} = \mathbf{D} \otimes \big(\mathbf{G}(\sigma) \otimes I\big) = \underbrace{\big(\mathbf{D} \otimes \mathbf{G}(\sigma)\big)}_{\text{DoG}} \otimes \mathbf{I}$$

$$\mathbf{G}_u(u, v) = -\frac{u}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

σ can be chosen to

- remove noise
- extract edges of different scales

Mathematical Morphology



$$\mathbf{O}[u,v] = f\big(\mathbf{I}[u+i, v+j]\big),\ \ \forall (i,j) \in \mathcal{S},\ \ \forall (u,v) \in \mathbf{I}$$

Opening: Erosion then Dilatation

Erosion:
$\boldsymbol{O} = \boldsymbol{I} \ominus \mathcal{S}$

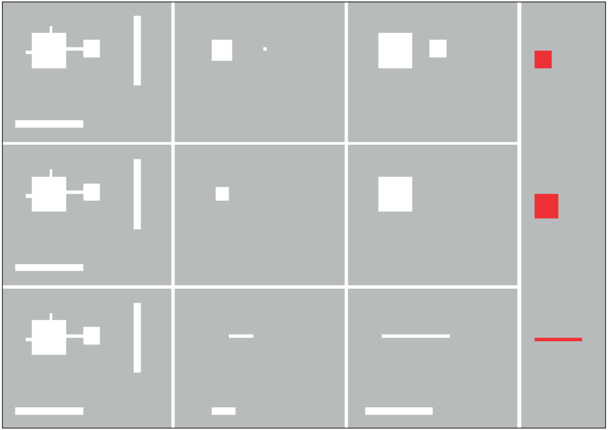
Dilatation:
$\boldsymbol{O} = \boldsymbol{I} \oplus \mathcal{S}$

Relation between erosion and dilatation:

$A \oplus B = \overline{\overline{A} \ominus B}$

$(A \oplus B) \oplus C = A \oplus (B \oplus C)$

$(A \ominus B) \ominus C = A \ominus (B \oplus C)$



$\boldsymbol{I} \circ \boldsymbol{S} = (\boldsymbol{I} \ominus \boldsymbol{S}) \oplus \boldsymbol{S}$

Basic morphological operators I

Erosion

Definition (*Morphological erosion*)

$$X \ominus B = \{z \in \mathcal{E} | B_z \subseteq X\}. \tag{42}$$

The following algebraic expression is equivalent to the previous definition:

Definition (*Alternative definition for the morphological erosion*)

$$X \ominus B = \bigcap_{b \in B} X_{-b}. \tag{43}$$

B is named "**structuring element**".

Erosion with a disk

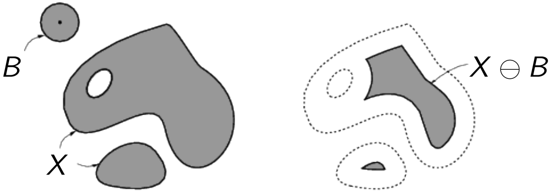


Figure : Erosion of X with a disk B. The origin of the structuring element is drawn at the center of the disk (with a black dot).

Dilation I

Definition (Dilation)

From an algebraic perspective, the *dilation* (*dila***ta***tion* in French!), is the union of translated version of X:

$$X \oplus B = \bigcup_{b \in B} X_b = \bigcup_{x \in X} B_x = \{x + b | x \in X,\ b \in B\}. \qquad (44)$$

Morphological opening I

Definition (Opening)

The *opening* results from cascading an erosion and a dilation with the same structuring element:

$$X \circ B = (X \ominus B) \oplus B. \tag{47}$$

Interpretation of openings (alternative definition)

The interpretation of the opening operator (which can be seen as an alternative definition) is based on

$$X \circ B = \bigcup \{B_z | z \in \mathcal{E} \text{ and } B_z \subseteq X\}. \tag{48}$$

In other words, the opening of a set by structuring element B is the set of all the elements of X that are covered by a translated copy of B when it moves inside of X.

Morphological opening I

Definition (Opening)

The *opening* results from cascading an erosion and a dilation with the same structuring element:

$$X \circ B = (X \ominus B) \oplus B. \tag{47}$$

Opening: Erosion then Dilatation

Erosion:
$\boldsymbol{O} = \boldsymbol{I} \ominus \mathcal{S}$

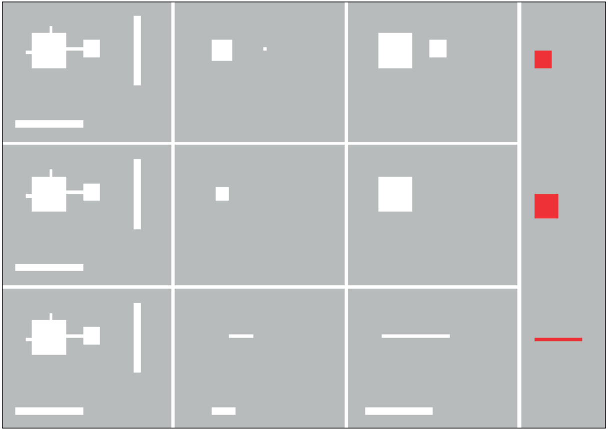
Dilatation:
$\boldsymbol{O} = \boldsymbol{I} \oplus \mathcal{S}$

Relation between erosion and dilatation:

$A \oplus B = \overline{\overline{A} \ominus B}$

$(A \oplus B) \oplus C = A \oplus (B \oplus C)$

$(A \ominus B) \ominus C = A \ominus (B \oplus C)$



$\boldsymbol{I} \circ \boldsymbol{S} = (\boldsymbol{I} \ominus \boldsymbol{S}) \oplus \boldsymbol{S}$

Closing: Dilatation then Erosion

Erosion:
$\boldsymbol{O} = \boldsymbol{I} \ominus \mathcal{S}$

Dilatation:
$\boldsymbol{O} = \boldsymbol{I} \oplus \mathcal{S}$

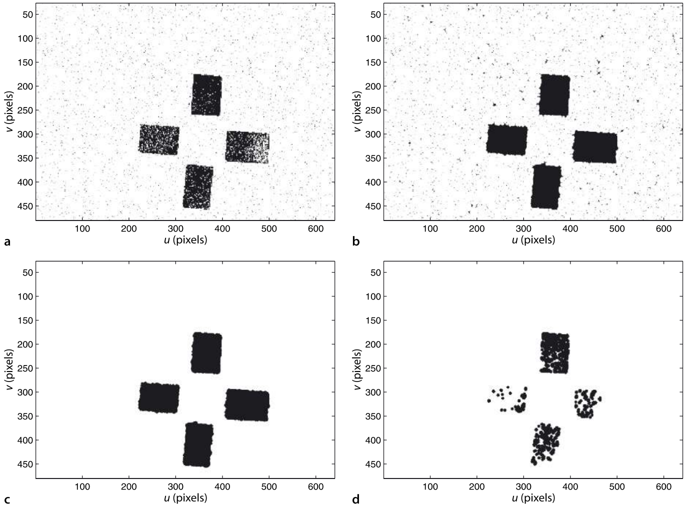
Relation between erosion and dilatation:
$A \oplus B = \overline{\overline{A} \ominus B}$

$(A \oplus B) \oplus C = A \oplus (B \oplus C)$

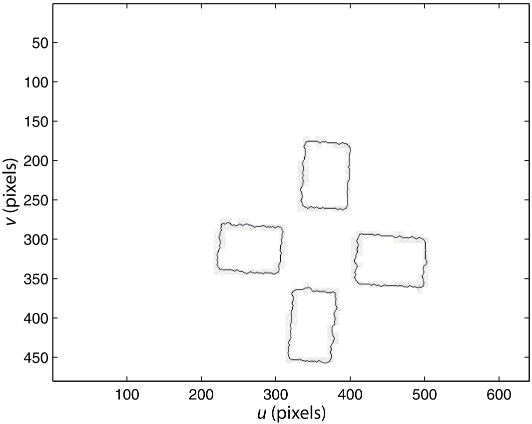
$(A \ominus B) \ominus C = A \ominus (B \oplus C)$

$\boldsymbol{I} \bullet \boldsymbol{S} = (\boldsymbol{I} \oplus \boldsymbol{S}) \ominus \boldsymbol{S}$

Noise Removal: Closing then Opening



Boundary Detection: Erosion then Substraction



Neighboring transforms

The *Hit or Miss transform* *is defined such as*

$$X \Uparrow (B, C) = \{x | B_x \subseteq X,\ C_x \subseteq X^c\} \tag{62}$$

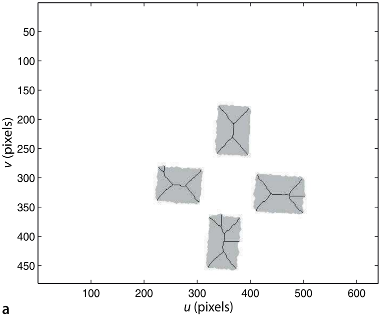
If $C = \emptyset$ the transform reduces to an erosion of X by B.

Hit and Miss Transform, Skeletonization

	1	
0	1	1
0	0	

1	1	0
0	1	1
0	0	1

1	1	0
0	1	1
1	0	1



a

The morphological skeleton to describe a shape II

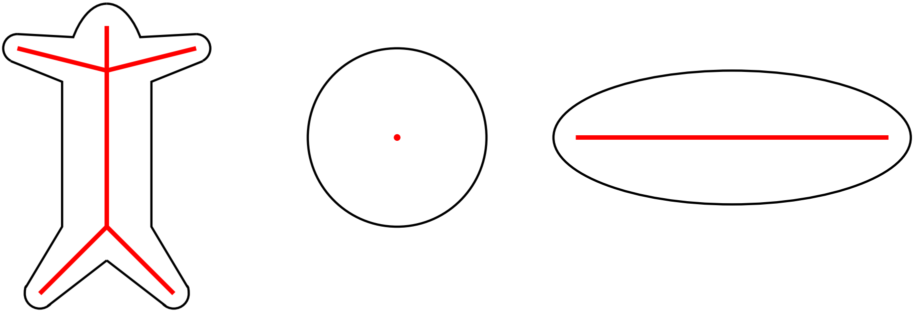


Figure : Shapes and their skeleton $S(X)$.

Skeletons are sensitive to noise on the object.

Formal definition of the skeleton II

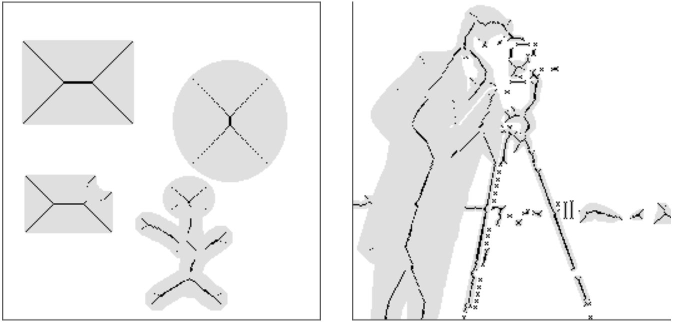


Figure : Some skeletons obtained with Lantuéjoul's formula.

Alternative skeleton formulas III

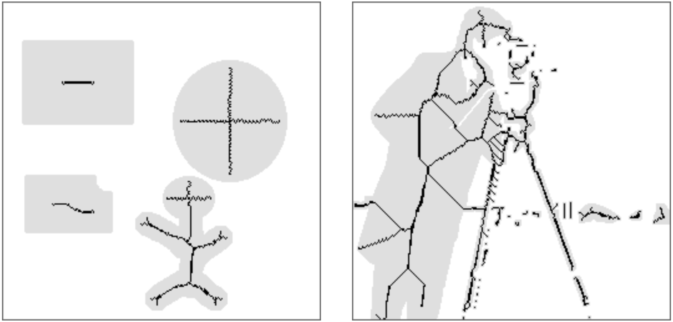


Figure : Some skeletons obtained with Vincent's algorithm.

Geodesy and reconstruction I

Geodesic dilation

A geodesic dilation is always based on two sets (images).

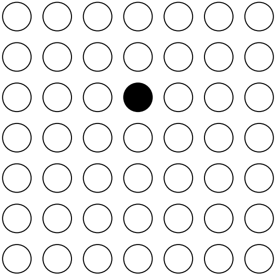
Definition

The *geodesic dilation of size* 1 of X conditionally to Y, denoted $D_Y^{(1)}(X)$, is defined as the intersection of the dilation of X and Y:

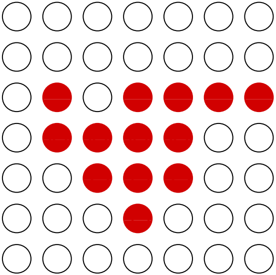
$$\forall X \subseteq Y, \;\; D_Y^{(1)}(X) = (X \oplus B) \cap Y \tag{63}$$

where B is usually chosen according to the frame connectivity (a 3×3 square for a 8-connected grid).

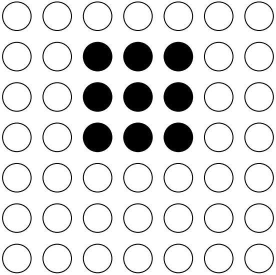
Geodesy and reconstruction II

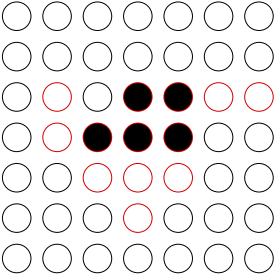


(a) Set to be dilated



(b) Geodesic mask



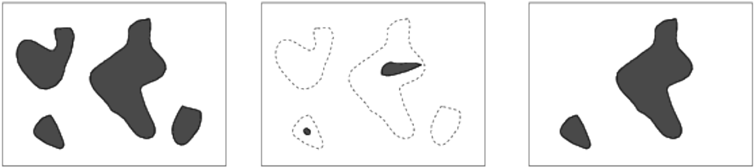


Morphological reconstruction

Definition

The *reconstruction* of X conditionally to Y is the geodesic dilation of X until idempotence. Let i be the iteration during which idempotence is reached, then the reconstruction of X is given by

$$R_Y(X) = D_Y^{(i)}(X) \text{ with } D_Y^{(i+1)}(X) = D_Y^{(i)}(X). \tag{65}$$



(a) Blobs (b) Marking blobs (c) Reconstructed blobs

Figure : Blob extraction by marking and reconstruction.

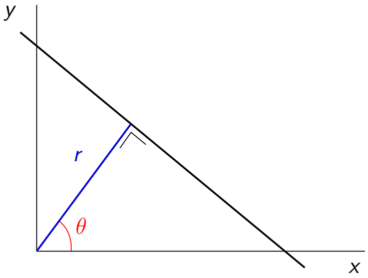
Detecting lines

Challenge: detect lines in an image

Towards the Hough transform

- Difficulty: matching a set of points arranged as a line
- Idea: instead of considering the family of points (x, y) that belong to a line $y = ax + b$, consider the two parameters
  1. the slope parameter a (but a is unbounded for vertical lines)
  2. the intercept parameter b (that is for $x = 0$)

Definition of the Hough transform I



With the Hough transform, we consider the (r, θ) pair where

- the parameter r represents the distance between the line and the origin,
- while θ is the angle of the vector from the origin to this closest point

Definition of the Hough transform II

We have several ways to characterize a line:

1. Slope a and b, such that $y = ax + b$.
2. The two parameters (r, θ), with $\theta \in [0, 2\pi[$ and $r \geq 0$.

Link between these characterizations:

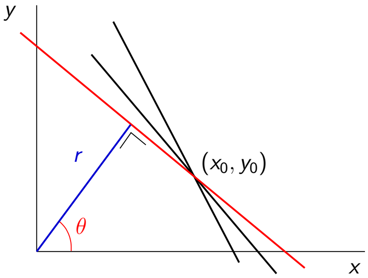
The equation of the line becomes

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right) x + \left(\frac{r}{\sin\theta}\right) \tag{200}$$

Check:

- For $x = 0$, $r = y \sin\theta \rightarrow$ ok.
- For $x = r\cos\theta$, $y = r\sin\theta \rightarrow$ ok.

Families of lines passing through a given point (x_0, y_0)

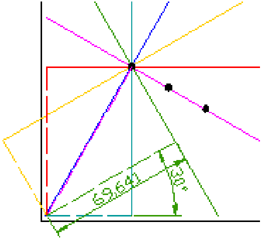


By re-arranging terms of $y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$, we get that, for an arbitrary point on the image plane with coordinates, e.g., (x_0, y_0), the family of lines passing through it are given by

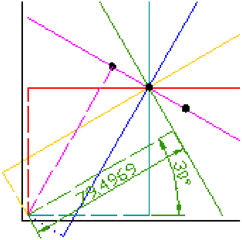
$$r = x_0 \cos\theta + y_0 \sin\theta \tag{201}$$

Example

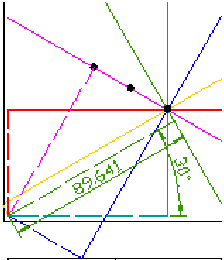
For three points (x_0, y_0), we explore the Hough's space.
That is, we compute r for a given set of orientations θ:



Angle	Dist.
0	40
30	69.6
60	81.2
90	70
120	40.6
150	0.4

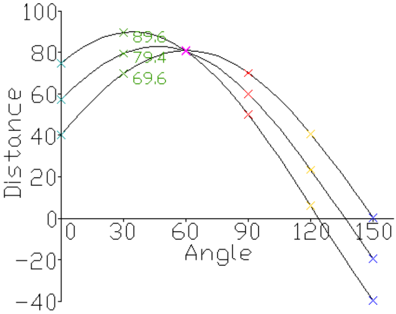


Angle	Dist.
0	57.1
30	79.5
60	80.5
90	60
120	23.4
150	−19.5



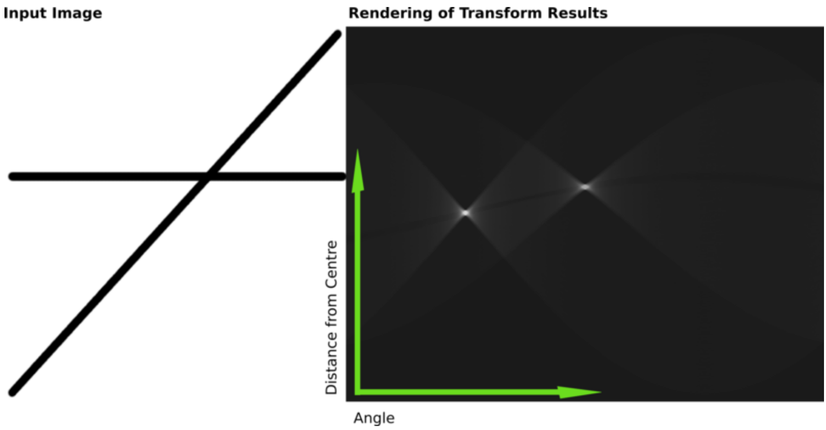
Angle	Dist.
0	74.6
30	89.6
60	80.6
90	50
120	6.0
150	−39.6

Hough space



Thus, the problem of detecting colinear points can be converted to the problem of finding concurrent curves.

Hough space

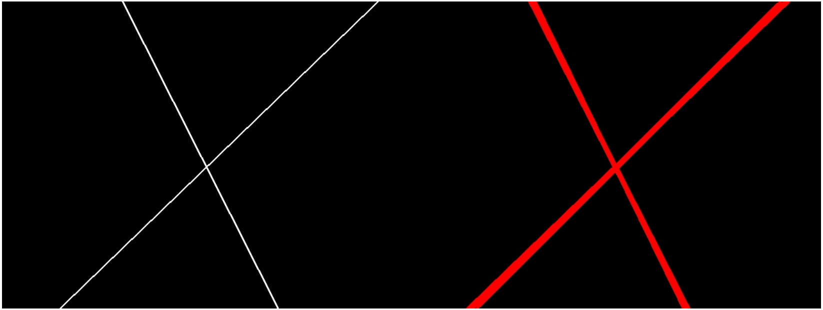

Input Image
Rendering of Transform Results
Distance from Centre
Angle

Algorithm for detecting lines

Algorithm

1. Detect edges in the original image.
2. Select "strong edges" for which there is enough evidence that they belong to lines.
3. For each point, accumulate values in the corresponding bins of the Hough space.
4. Threshold the accumulator function to select bins that correspond to lines in the original image.
5. Draw the corresponding lines in the original image.

"Toy" example



Real example

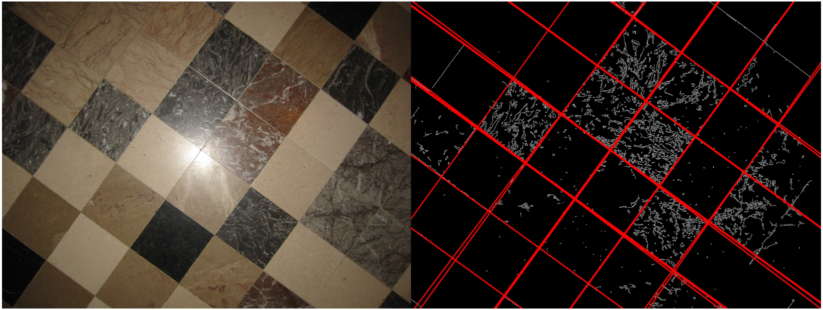
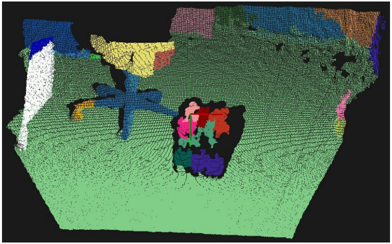


Image segmentation I

Segmentation of a color image



Segmentation of a depth image

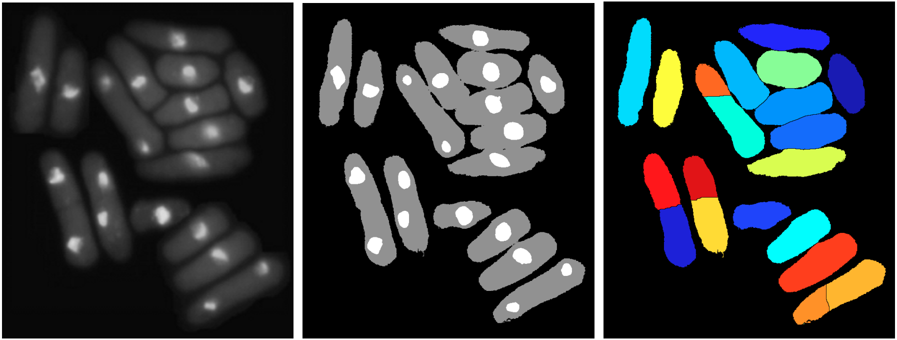
- Problem statement
- Segmentation by thresholding
- Segmentation by region detection (region growing)
  - Watershed
- Segmentation by classification (semantic classification)

Problem statement II

Segmented image

Labelled image

Illustration: segmentation of cells



[Source]

Semantic segmentation (based on deep learning)

- Based on classification techniques and machine learning
- Pixel-based
- A series of semantic notions (persons, cars, bicycles, etc)

Original image (hover to highlight segmented parts)

Semantic segmentation

Objects appearing in the image:

Bicycle | Person

Objects not appearing in the image:

Aeroplane | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow

Dining table | Dog | Horse | Motorbike | Potted plant | Sheep | Sofa | Train | TV/Monitor