Computer Vision

Images in the Spatial Domain

Prof. Flávio Cardeal – DECOM / CEFET-MG

cardeal@decom.cefetmg.br

Abstract

- This lecture introduces basic *notation* and *mathematical concepts* for describing an image in a regular grid in the *spatial domain*.

Digital Images

- A digital image is defined by *integrating* and *sampling* continuous data in a spatial domain.

Sampling: Measuring at uniformly spaced locations, e.g. as given by CCD or CMOS image sensor matrices.

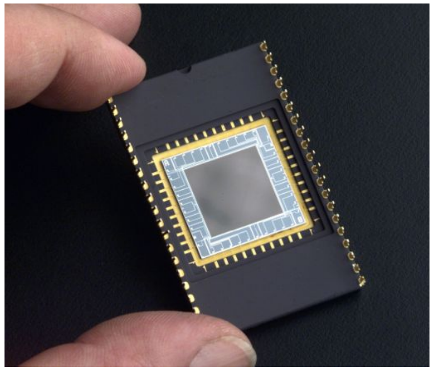
Integrating: A sensor cell converts measured light into an electric charge, represented by a single number (sample) in the created image.

CCD X CMOS Sensors

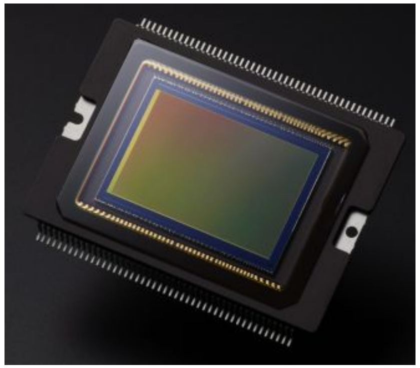
- Digital images are captured on a silicon semiconductor referred to as a *digital sensor*.
- This sensor is composed of an *array of photosensitive diodes* that capture photons and convert them to electrons.

CCD X CMOS Sensors

- There are two main types of digital sensors: CCD (*Charged Coupling Device*) and CMOS (*Complementary Metal Oxide Semiconductor*).



CCD Sensor



CMOS Sensor

CCD X CMOS Sensors

- In the past, CCDs have been considered superior to CMOS because of their quality.
- CCDs have traditionally offered higher *dynamic range* and *higher resolution*.
- However: frame rates are slower; require more power dissipation; cost more to manufacture.

CCD X CMOS Sensors

- Recently, CMOS sensors have shown significant improvements in quality.
- In fact, CMOS sensor resolutions and data quality are approaching those of CCDs.
- In addition, they have higher speed, lower power requirements and higher integration potential.

Digital Images

- It consists of a rectangular array of elements (x, y, u), each combining a location $(x, y) \in Z^2$ and a value u, the sample at location (x, y).

- Z is the set of all integers. Locations $(x, y) \in Z^2$ form a *regular grid*.

Array of Numbers

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

Source: S. Narasimhan

Image Carrier

- An image I is defined on a rectangular set, the carrier:

$$\Omega = \left\{ \left(x, y\right): \quad 1 \le x \le N_{cols} \quad \wedge \quad 1 \le y \le N_{rows} \right\} \subset Z^2$$

of I containing the grid points or *pixel locations* for $N_{cols} \ge 1$ and $N_{rows} \ge 1$.

Image Coordinate System

- We assume a left-hand coordinate system as shown in the figure below.

Source: R. Klette

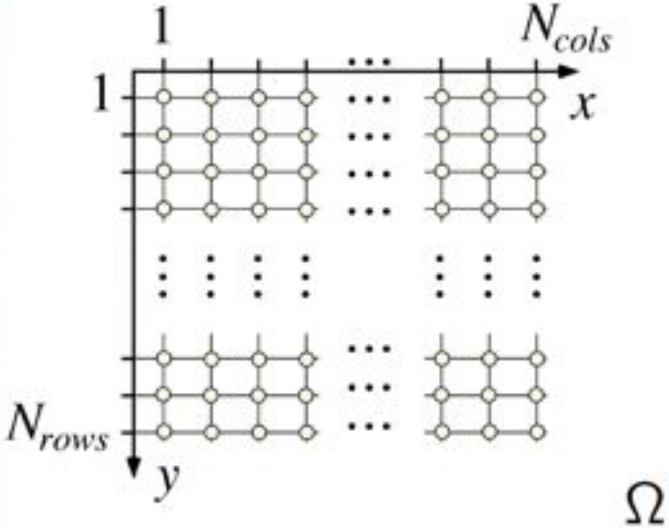


Image Coordinate System

- Row y contains grid points:

$$\{(1,y),(2,y),\ldots,(N_{cols},y)\} \text{ for } 1 \le y \le N_{rows}$$

- Column x contains grid points:

$$\{(x,1),(x,2),\ldots,(x,N_{rows})\} \text{ for } 1 \le x \le N_{cols}$$

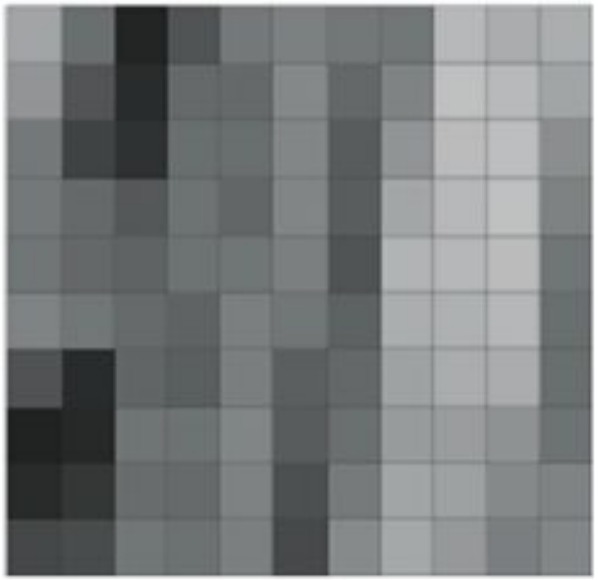
Pixels

- The term *pixel* is short for **picture element**. Pixels are the "atomic elements" of an image.

- Two ways of thinking about the geometric representations of pixels: (1) as *shaded grid squares* or (2) as *labeled grid points*.

Pixels

Shaded Grid Squares



Different shades represent values in a chosen set of image values.

Source: R. Klette

Labeled Grid Points

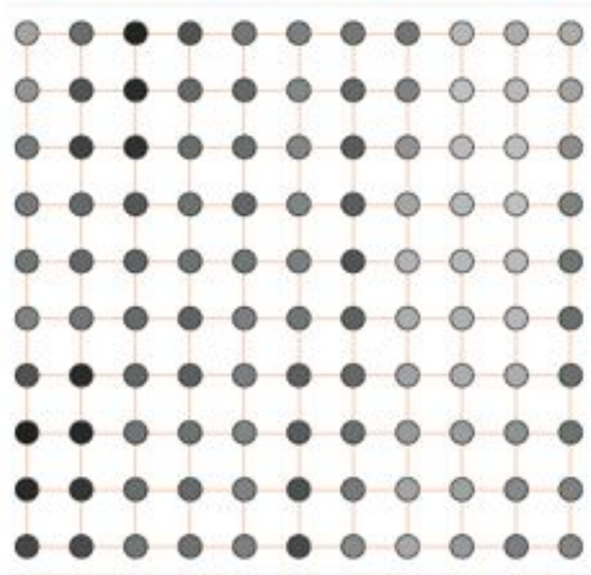


Image values as labels at grid points (centers of grid squares).

Grid Cells and Grid Points

- In the *grid cell model*, a pixel is considered as a homogeneously shaded square cell.
- On the other hand, in the *grid point model*, a pixel is considered a *labelled grid point*.

Image Windows

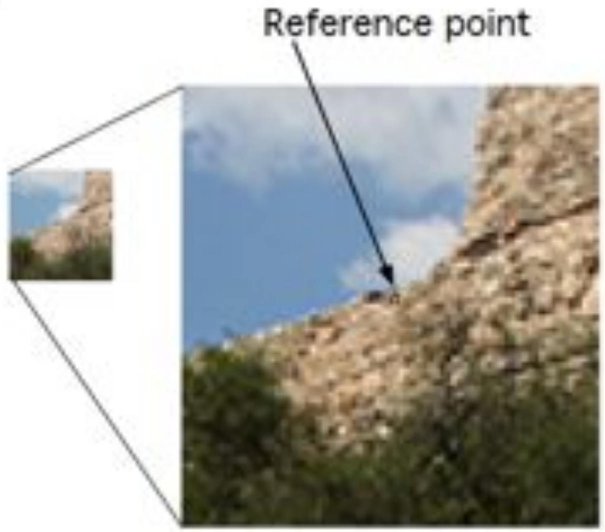
- A window $W_p^{m,n}(I)$ is a subimage of image I of size $m \times n$ positioned with respect to a reference point p (i. e., a pixel location).

- The default is that $m = n$ is an odd number and p is the center location in the window.

- We can simplify notation to W_p.

Image Windows

I $\qquad\qquad$ $W^{73,77}_{(453,134)}(I)$



Source: R. Klette

Image Values

- An image value u is taken in a discrete set of possible values.

- **Scalar Image**: values are integers

$$u \in \{0,1,\ldots,2^a-1\}$$

<u>Usually</u>: a = 8 (i.e. one byte) or a = 16 (new standard).
Let $G_{max} = 2^a - 1$ be the general maximum image value.

Grey Level Images

- A *grey level image* is a scalar image where scalar values represent *grey levels*, with 0 = black and $2^a - 1$ = white.

- All other grey-levels are linearly interpolated between black and white.

Examples of Scalar Images

Grey-Level Image

Range image

Source: R. Klette

Binary Images

- A *binary image* has only two values at its pixels, commonly 0 = black and 1 = white, meaning white objects on a black background.

Source: R. Klette

Vector-Valued Images

- A *vector-valued image* has more than one channel or band.

- Image values are vectors of length $N_{channels}$:

$$\left[u_1,\ldots,u_{N_{channels}}\right]^{\mathrm{T}}$$

- Ex: color images in RGB color model. Values u_i in each channel are in the set $\left\{0,1,\ldots,G_{max}\right\}$.

Example of Color Image

Each channel is just like a grey level image

Red Channel

Green Channel

Blue Channel

Source: R. Klette

Basic Statistics - Mean

- Given an $N_{cols} \times N_{rows}$ image I, we define its mean (i.e., the average grey level) as:

$$\mu_I = \frac{1}{N_{cols} \cdot N_{rows}} \sum_{x=1}^{N_{cols}} \sum_{y=1}^{N_{rows}} I(x,y) = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} I(x,y)$$

in which $|\Omega| = N_{cols} \times N_{rows}$ is the cardinality of carrier Ω for all pixel locations.

Variance and St. Deviation

- The variance of image I is defined as

$$\sigma_I^2 = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} \left[I(x,y) - \mu_I \right]^2$$

Its root σ_I is the standard deviation of image I.

Variance and St. Deviation

- Consider now the alternative formula for variance:

$$\sigma_I^2 = \left[\frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} I(x,y)^2 \right] - \mu_I^2$$

- Note that, in this case, mean and variance may be calculated by running through image I only once.
- Why?

Histograms

- A histogram represents tabulated *frequencies*, typically by using bars in a graphical diagram.

- Histograms are used for representing *value frequencies* of a scalar image, or of one channel or band of a vector-valued image.

Absolute Frequencies

- Given a scalar image I with pixels (x, y, u), where $0 \le u \le G_{max}$, the absolute frequency of u (count of appearances of u in Ω) is given by:

$$H_I(u) = \left| \left\{ \ (x, y) \in \Omega \quad : \quad I(x, y) = u \ \right\} \right|$$

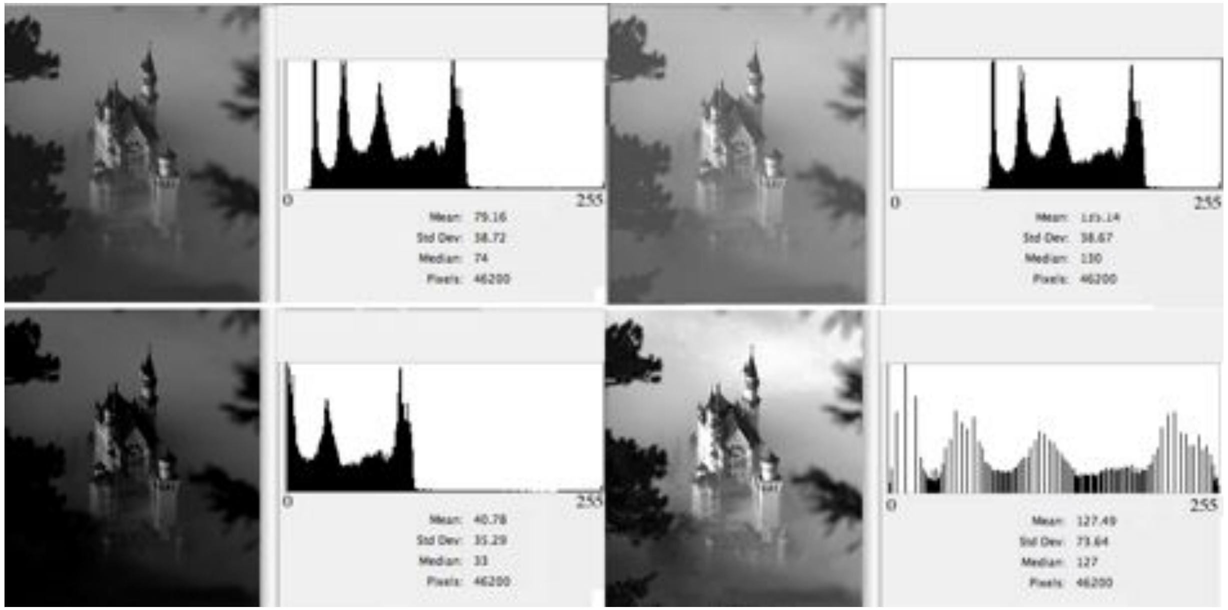
in which $|\cdot|$ denotes the cardinality of a set.

Grey Level Histogram

- The values $H_I(0), H_I(1), \ldots, H_I(G_{max})$ define the (absolute) *grey level histogram* of image I.

Original Image

Brighter Version



Darker Version

After Equalization

Source: R. Klette

Relative Frequencies

- Relative frequencies (values between 0 and 1) define a relative histogram:

$$h_I(u) = \frac{H_I(u)}{|\Omega|}$$

- **Observation**: relative frequencies are comparable to the probability density function $P[I(x,y)=u]$ of discrete random numbers $I(x,y)$.

Mean and Variance

- We can compute the mean and variance also based on relative frequencies as follows:

$$\mu_I = \sum_{u=0}^{G_{max}} u \cdot h_I(u) \qquad \sigma_I^2 = \sum_{u=0}^{G_{max}} \left[u - \mu_I\right]^2 \cdot h_I(u)$$

- **Observation**: this provides a speed-up if the histogram was already computed.

Cumulative Frequencies

- Absolute and relative cumulative frequencies are defined as follows, respectively:

$$C_I(u) = \sum_{v=0}^{u} H_I(v)$$

$$c_I(u) = \sum_{v=0}^{u} h_I(v)$$

- **Observation**: relative cumulative frequencies are comparable to the probability function $P\left[I(x,y) \leq u\right]$ of discrete random numbers $I(x,y)$.

Value Statistics in Windows

- Consider a window $W = W_p^{n,n}(I)$ with $n = 2k + 1$ and $p = (x, y)$. The mean is given by

$$\mu_W = \frac{1}{n^2} \sum_{i=-k}^{+k} \sum_{j=-k}^{+k} I(x+i, y+j)$$

- **Observation**: formulas for the variance, and so forth, can be adapted analogously.

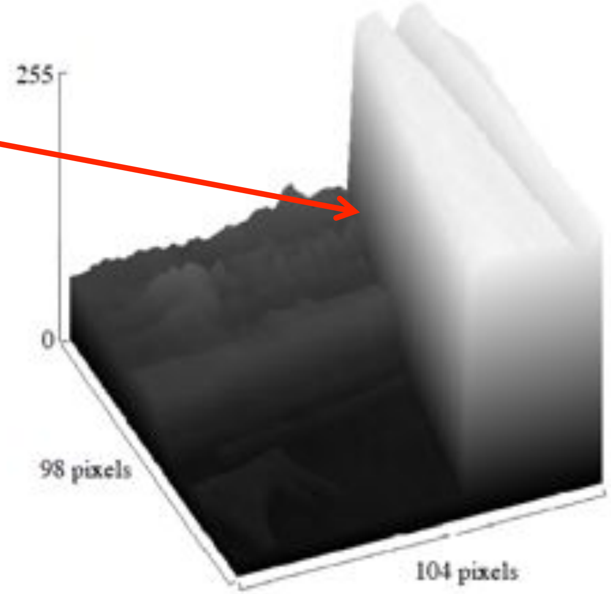
Examples of Windows

Source: R. Klette

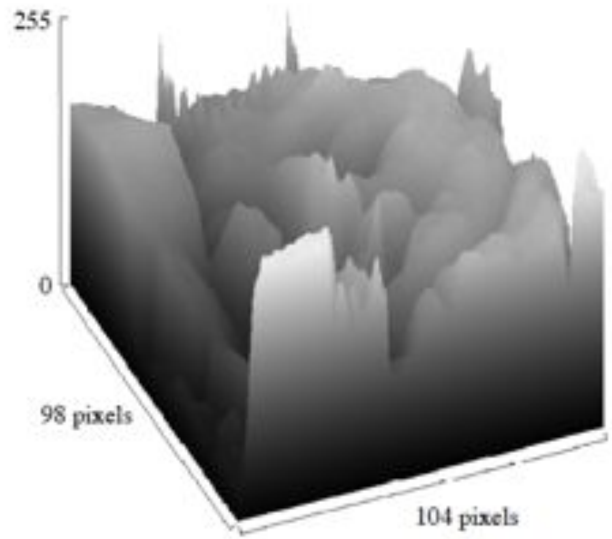
3D Views of Grey Level Images

- A 3D view of grey levels shows the different "degrees of homogeneity" in an image.

The steep slope between plateaus is a typical edge in an image



W_2



W_1

Source: R. Klette

Contrast of an Image

- In image analysis, we commonly need to classify windows into categories, such as:
  - Within a homogeneous region;
  - Of low contrast;
  - Showing an edge between two different regions;
  - Of high contrast.

Contrast of an Image

- The *contrast* of an image I is the mean absolute difference between pixel values and the mean value at adjacent pixels, that is:

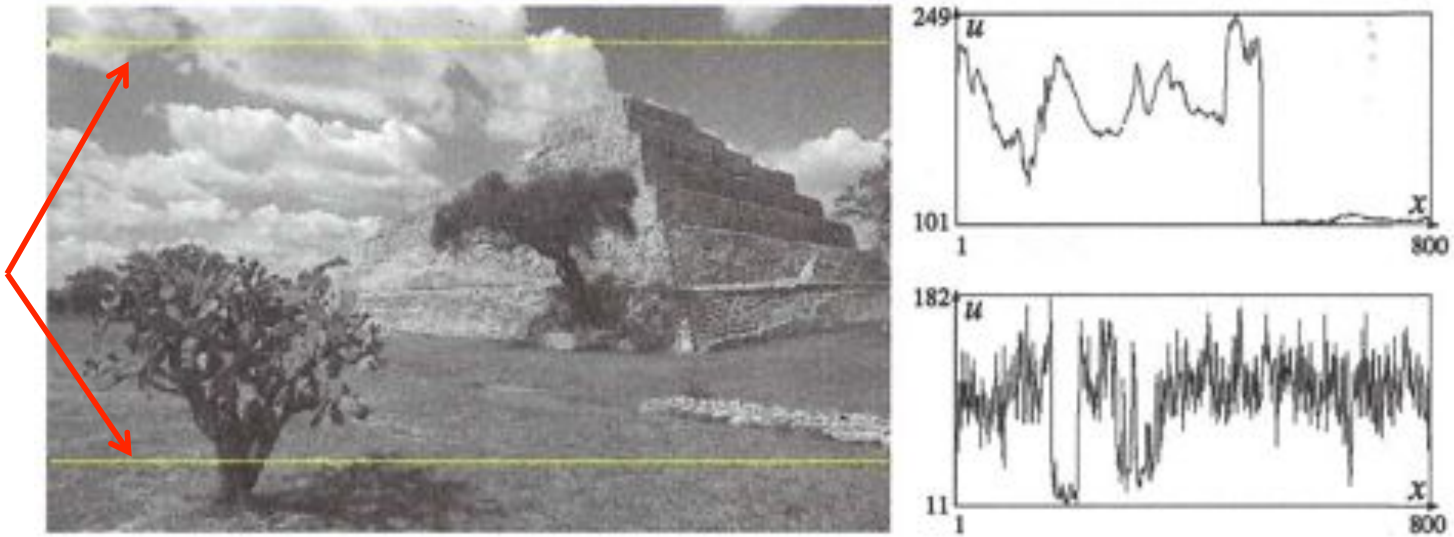
$$C(I) = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} \left| I(x,y) - \mu_{A(x,y)} \right|$$

in which $\mu_{A(x,y)}$ is the mean value of pixel locations adjacent to location (x, y).

Intensity Profiles

- *Intensity profiles* in images are defined by 1D cuts through the given scalar data arrays.

Two selected image rows in the intensity channel



Source: R. Klette

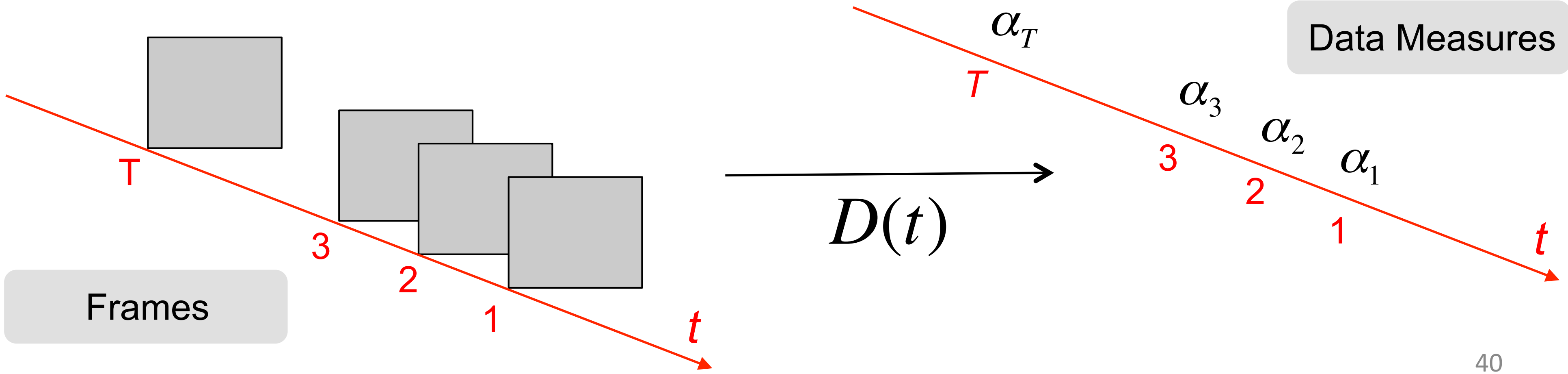
Spatial Value Statistics

- Intensity profiles and histograms are examples of *spatial value statistics*.

- For example:
  - Intensity profiles for rows 1 to N_{rows} in image I define a sequence of discrete functions, which can be compared with the corresponding one of image J.

Temporal Value Statistics

- Assume now an image sequence consisting of frames I_t for $t = 1, 2, \ldots, T$ and a function $D(t)$ that maps frames in *data measures*.



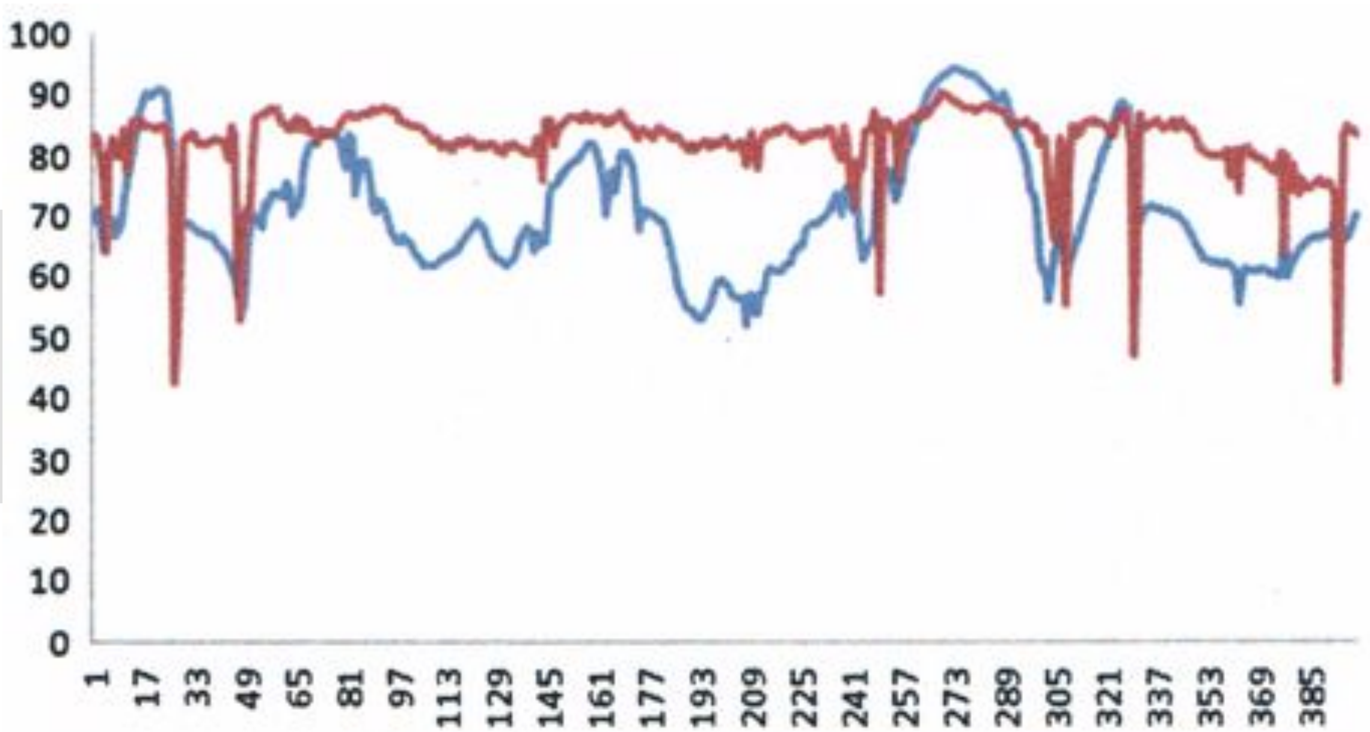
Temporal Value Statistics

- The data measures $\{\alpha_1, \quad ... \quad \alpha_T\}$ refer to different discrete time instants and, thus, support *temporal value statistics*.

- For example, $D(t)$ could be the contrast, mean or the variance of an image captured at time t.

Temporal Value Statistics

The plot of a data measure (e.g. mean) for an image sequence before and after its processing.

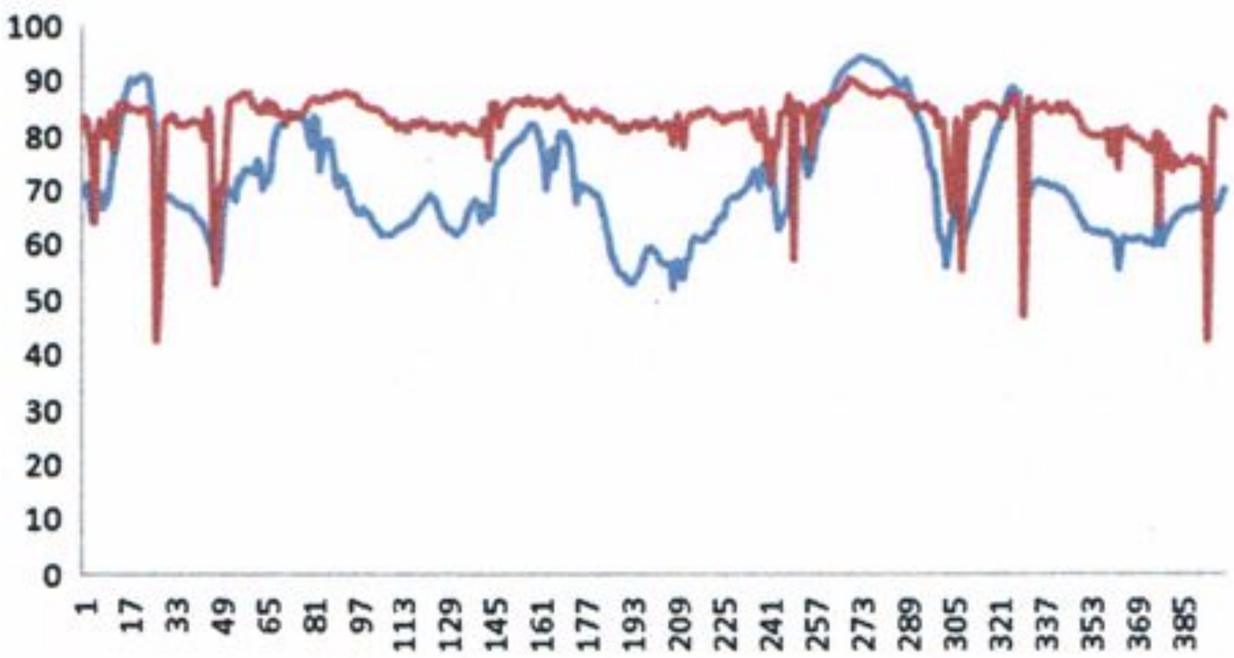


Source: R. Klette

Temporal Value Statistics

A plot of two data measures for an image sequence.

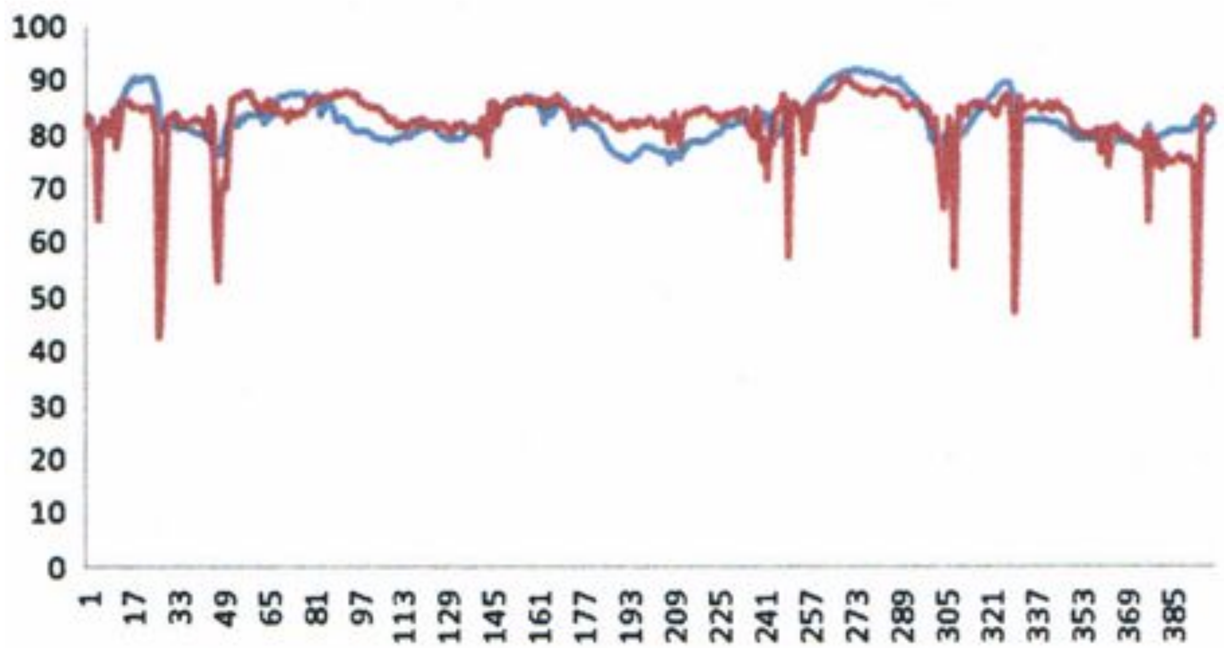
Both measures have their individual range across the image sequence.



Source: R. Klette

Temporal Value Statistics

For a better comparison of distinct data measures, it is interesting to map both onto functions having identical mean and variance.



Source: R. Klette

Normalization of Functions

- Let μ_f, μ_g and σ_f ,σ_g be the mean and standard deviation values, respectively, for two functions and f and g.

- How can we obtain a new function g_{new} with the same mean and standard deviation values of f ?

Normalization of Functions

- To achieve this goal, we compute two parameters α and β, as follows:

$$\alpha = \frac{\sigma_g}{\sigma_f}\mu_f - \mu_g \quad \text{and} \quad \beta = \frac{\sigma_f}{\sigma_g}$$

- So that:

$$g_{new}(x) = \beta\big(g(x) + \alpha\big)$$

Distance between Functions

- The distance between two real-valued functions may be computed as follows:

$$d_1(f, g) = \frac{1}{T} \sum_{x=1}^{T} \left| f(x) - g(x) \right|$$

$$d_2(f, g) = \frac{1}{T} \sqrt{\sum_{x=1}^{T} \left(f(x) - g(x) \right)^2}$$

Distance between Functions

- Both distances $d_1(\cdot)$ and $d_2(\cdot)$ are metrics satisfying the following axioms:

1. $f = g$ iff $d(f, g) = 0$,
2. $d(f, g) = d(g, f)$ (symmetry), and
3. $d(f, g) \leq d(f, h) + d(h, g)$ for a third function h

Structural Similarity

- Assume two different spatial or temporal data measures f and g on the same domain.
- Assume that g has been mapped into g_{new} such that both measures have now identical mean and variance.
- Compute now the distance between f and g_{new} by using either $d_1(\cdot)$ or $d_2(\cdot)$.

Structural Similarity

- Two measures f and g_{new} are *structurally similar* iff the resulting distance between f and g_{new} is close to zero.

- Structurally similar measures take their local maxima or minima at about the same arguments.

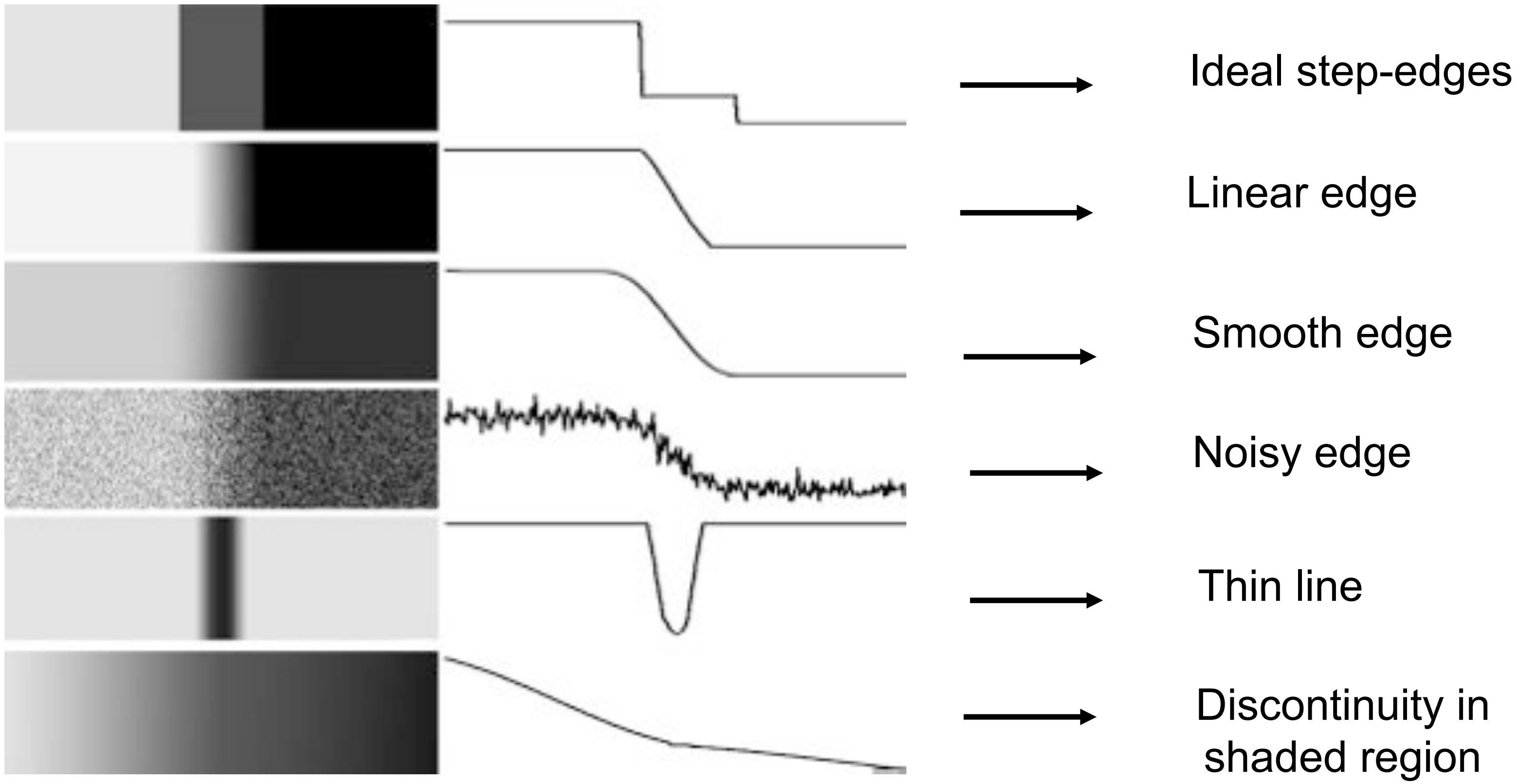
Step-Edges

- *Discontinuities* are features that are often useful for analyzing images.
- They can occur in small windows (e.g. noisy pixels) or define *edges* between image regions, which are important visual cues.

What is an edge?

- Edges may be defined as changes in local derivatives (*step-edge model*).
- Next slide illustrates a possible diversity of edges in images by sketches of 1D cuts through the intensity profile of an image.

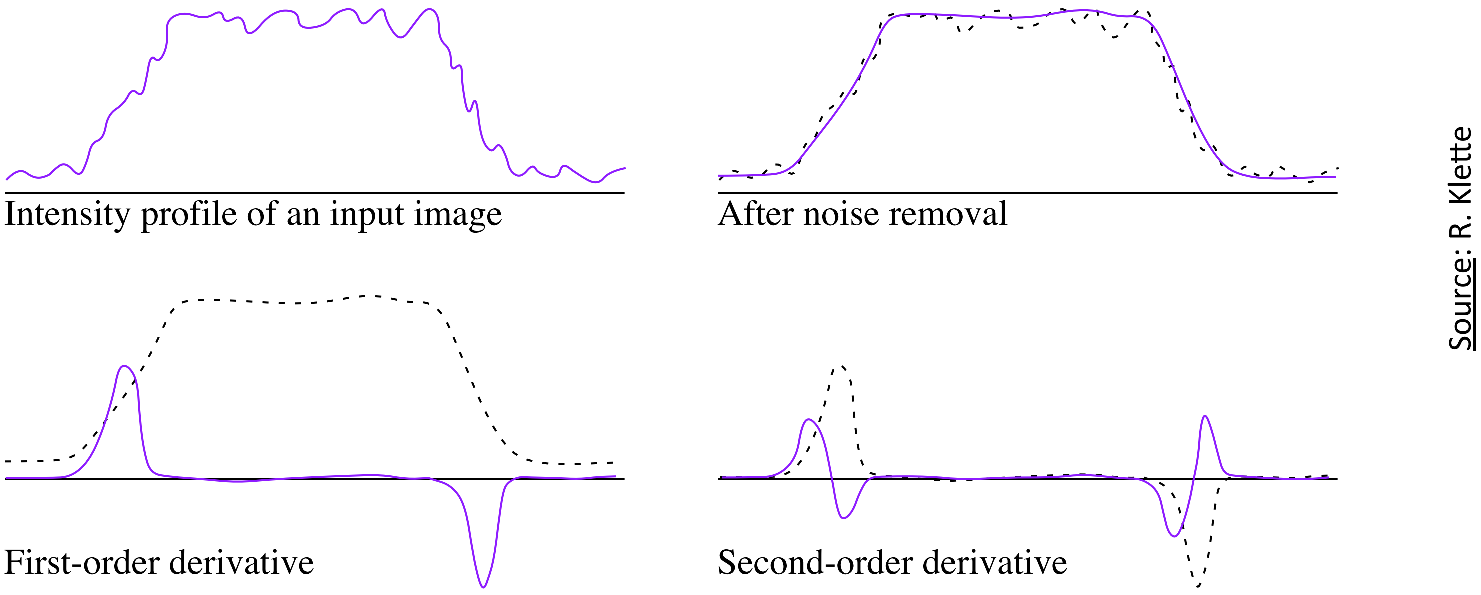
Step-Edge Model



Source: R. Klette

Step-Edge Model

- So, in this case, edges may be detected by first-order and second-order derivatives.

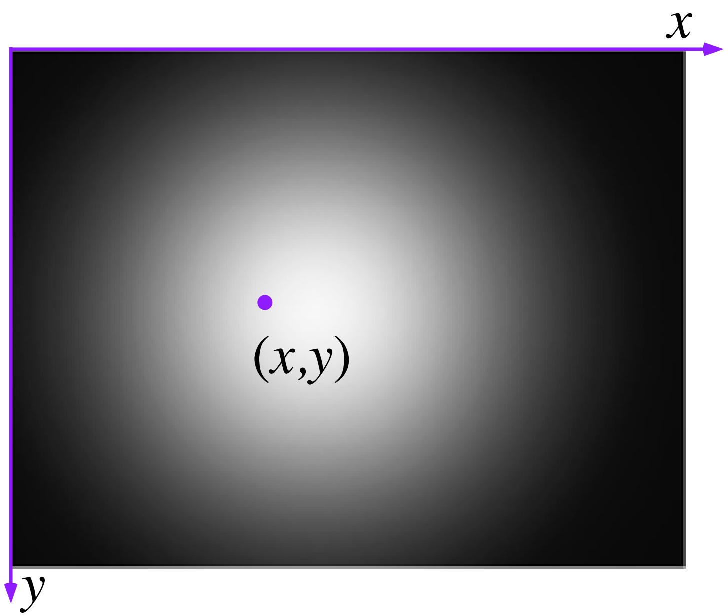


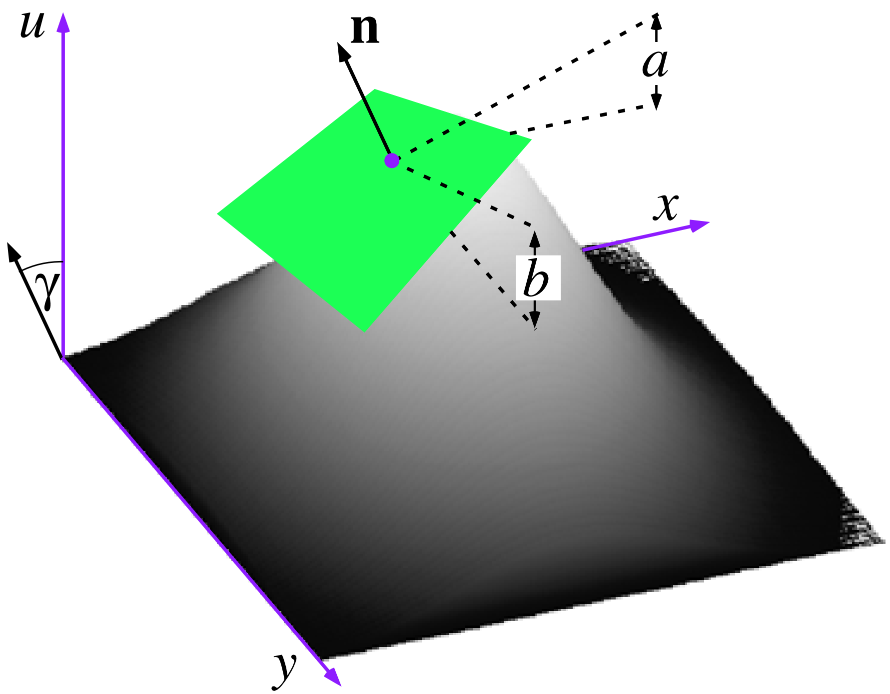
Continuous Surface

- Intensity values in an image I can be seen as defining a *continuous surface* having different elevations at pixel locations.

- In this interpretation, an image I represents valleys, plateaus, gentle or steep slopes, etc.

Continuous Surface





Source: R. Klette

First-Order Derivatives

- The *gradient* of an image I at a point $p = (x, y)$ is given by:

$$\nabla I = \mathbf{grad}\, I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]^{\top}$$

First-Order Derivatives

- The normal vector $\mathbf{n}$, which is orthogonal to the tangential plane at a pixel $(x, y, I(x, y))$ is:

$$\mathbf{n} = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, +1 \right]^\top$$

- It can point either into the positive or negative direction of the *u*-axis. We use the positive one.

First-Order Derivatives

- The first-order derivatives allow us to calculate the length (or *magnitude*) of gradient and normal:

$$\|\mathbf{grad}\, I\|_2 = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

$$\|\mathbf{n}\|_2 = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 + 1}$$

First-Order Derivatives

- Based on the figure in Slide 54 and on the related discussion, we conclude that:

- It appears to be meaningful to detect edges at locations where the magnitudes $\|\mathbf{grad}\, I\,\|_2$ or $\|\mathbf{n}\|_2$ define a local maximum.

Second-Order Derivatives

- Second-order derivatives at a point $p = (x, y)$ are combined into the *Laplacian* of I, as:

$$\triangle I = \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

- Note that the Laplacian is a scalar and not a vector like the gradient or the normal.

Second-Order Derivatives

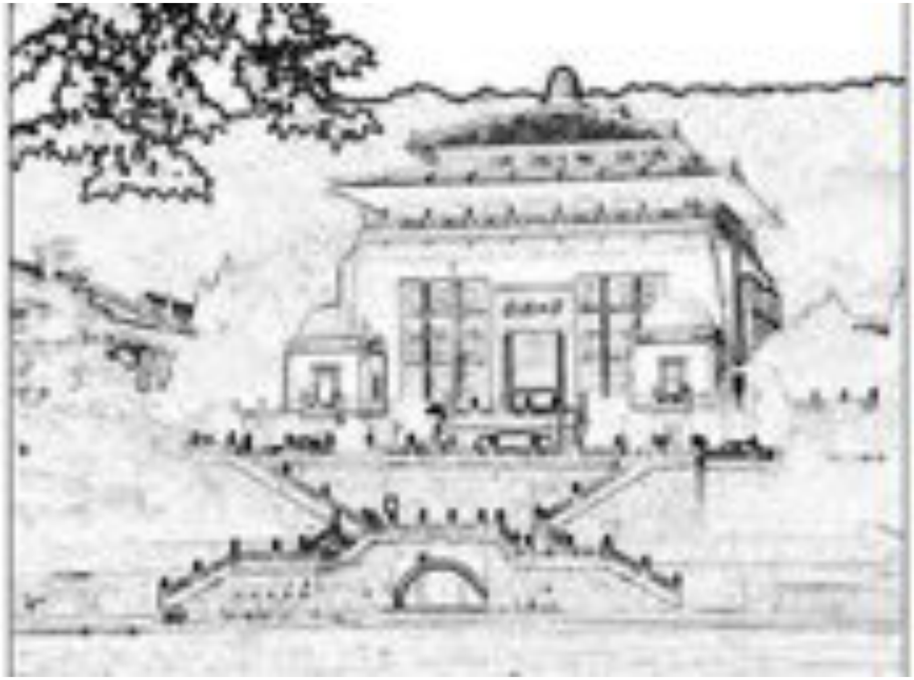
- Based on the figure in Slide 54 and on the related discussion, we conclude that:

- It appears to be meaningful to detect edges at locations where the *Laplacian* $\triangle I$ defines a zero-crossing.

Edge Maps

- Operators for detecting *edges* map images into *edge images* or *edge maps*.
- There is no general edge definition, and there is no general edge detector.

Edge Maps



Source: R. Klette

Next Lecture

- Lab Class

Matlab tutorial.

- Online resources

- Getting started with Matlab: tutorial by Stefan Roth.
- Matlab Image Processing Toolbox.
- Matlab Functions for Computer Vision by Peter Kovesi.