KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

„Image Processing and Computer Graphics"

Prezentacja multimedialna współfinansowana przez Unię Europejską w ramach Europejskiego Funduszu Społecznego w projekcie pt.

„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej - zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do zatrudniania osób niepełnosprawnych"

Politechnika Łódzka

Politechnika Łódzka, ul. Żeromskiego 116, 90-924 Łódź, tel. (042) 631 28 83
www.kapitalludzki.p.lodz.pl

Contribution

Lecturer:

Piotr M. Szczypiński, Dr inż.
Institute of Electronics, Technical University of Lodz
Wolczanska 211/215, 90-924 Lodz, Poland
tel. +48426312638
http://www.eletel.p.lodz.pl/pms
pms@p.lodz.pl

Fragments of the lecture were also provided by

Andrzej Materka, Michal Strzelecki, Paweł Strumiłło, Piotr Romaniuk, Piotr Makowski

and were used with permission.

Image Enhancement

- Image histogram
- Point processing
- Spectral filtering
- Spatial filtering
- Image resampling

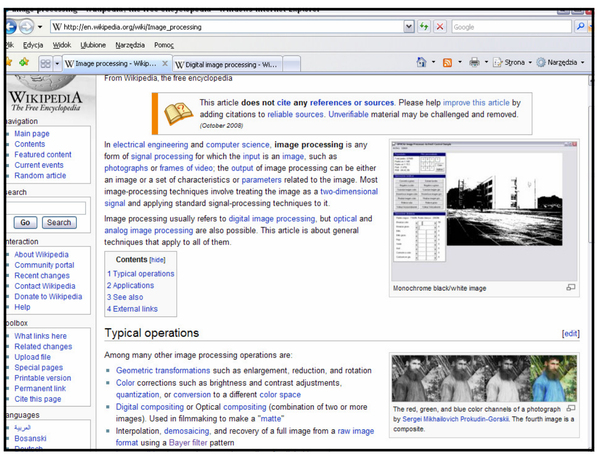
Image enhancement

Image enhancement belongs to image pre-processing methods.

Objective of **image enhancement** is to process the image so that it is better suited for viewing, further processing or analysis.

Image enhancement methods are based on subjective image quality criteria.

No objective mathematical criteria are used for optimizing processing results.



http://en.wikipedia.org/wiki/Image_processing

From Wikipedia, the free encyclopedia

This article **does not cite any references or sources**. Please help improve this article by adding citations to reliable sources. Unverifiable material may be challenged and removed. *(October 2008)*

In electrical engineering and computer science, **image processing** is any form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it.

Image processing usually refers to digital image processing, but optical and analog image processing are also possible. This article is about general techniques that apply to all of them.

Contents [hide]
1 Typical operations
2 Applications
3 See also
4 External links

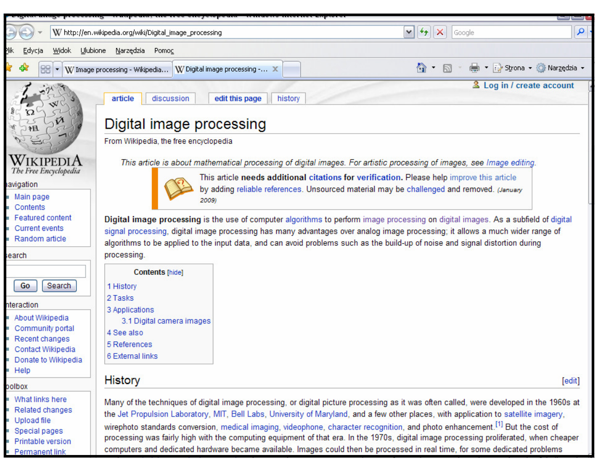
Monochrome black/white image

Typical operations [edit]

Among many other image processing operations are:

- Geometric transformations such as enlargement, reduction, and rotation
- Color corrections such as brightness and contrast adjustments, quantization, or conversion to a different color space
- Digital compositing or Optical compositing (combination of two or more images). Used in filmmaking to make a "matte"
- Interpolation, demosaicing, and recovery of a full image from a raw image format using a Bayer filter pattern

The red, green, and blue color channels of a photograph by Sergei Mikhailovich Prokudin-Gorskii. The fourth image is a composite.



http://en.wikipedia.org/wiki/Digital_image_processing

Digital image processing

From Wikipedia, the free encyclopedia

This article is about mathematical processing of digital images. For artistic processing of images, see Image editing.

This article **needs additional citations for verification.** Please help improve this article by adding reliable references. Unsourced material may be challenged and removed. *(January 2009)*

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subfield of digital signal processing, digital image processing has many advantages over analog image processing; it allows a much wider range of algorithms to be applied to the input data, and can avoid problems such as the build-up of noise and signal distortion during processing.

Contents [hide]
1 History
2 Tasks
3 Applications
3.1 Digital camera images
4 See also
5 References
6 External links

History [edit]

Many of the techniques of digital image processing, or digital picture processing as it was often called, were developed in the 1960s at the Jet Propulsion Laboratory, MIT, Bell Labs, University of Maryland, and a few other places, with application to satellite imagery, wirephoto standards conversion, medical imaging, videophone, character recognition, and photo enhancement.[1] But the cost of processing was fairly high with the computing equipment of that era. In the 1970s, digital image processing proliferated, when cheaper computers and dedicated hardware became available. Images could then be processed in real time, for some dedicated problems

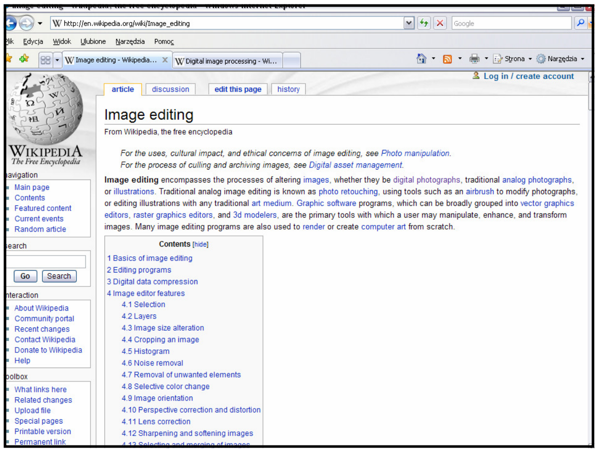


Image histogram and point processing

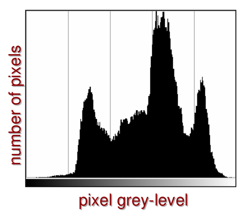
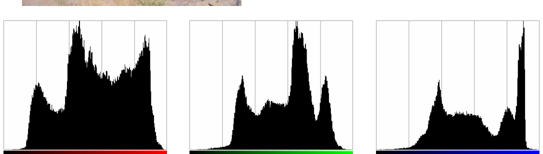
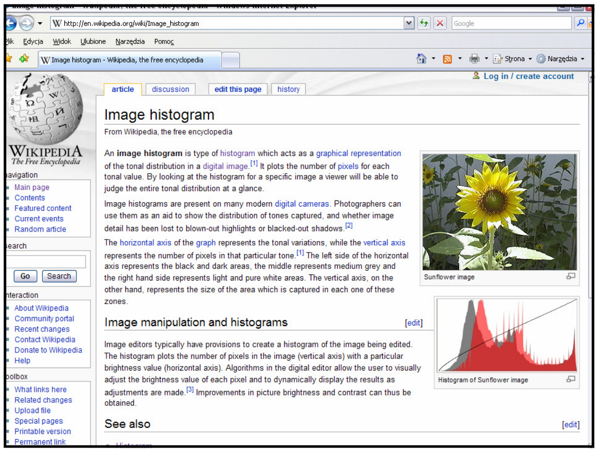


Image histogram




http://en.wikipedia.org/wiki/Image_histogram
Image histogram - Wikipedia, the free encyclopedia
article
discussion
edit this page
history
Log in / create account
Image histogram
From Wikipedia, the free encyclopedia
An image histogram is type of histogram which acts as a graphical representation of the tonal distribution in a digital image.[1] It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance.
Image histograms are present on many modern digital cameras. Photographers can use them as an aid to show the distribution of tones captured, and whether image detail has been lost to blown-out highlights or blacked-out shadows.[2]
The horizontal axis of the graph represents the tonal variations, while the vertical axis represents the number of pixels in that particular tone.[1] The left side of the horizontal axis represents the black and dark areas, the middle represents medium grey and the right hand side represents light and pure white areas. The vertical axis, on the other hand, represents the size of the area which is captured in each one of these zones.
Sunflower image
Histogram of Sunflower image
Image manipulation and histograms
[edit]
Image editors typically have provisions to create a histogram of the image being edited. The histogram plots the number of pixels in the image (vertical axis) with a particular brightness value (horizontal axis). Algorithms in the digital editor allow the user to visually adjust the brightness value of each pixel and to dynamically display the results as adjustments are made.[3] Improvements in picture brightness and contrast can thus be obtained.
See also
[edit]
Main page
Contents
Featured content
Current events
Random article
Go
Search
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Donate to Wikipedia
Help
What links here
Related changes
Upload file
Special pages
Printable version
Permanent link

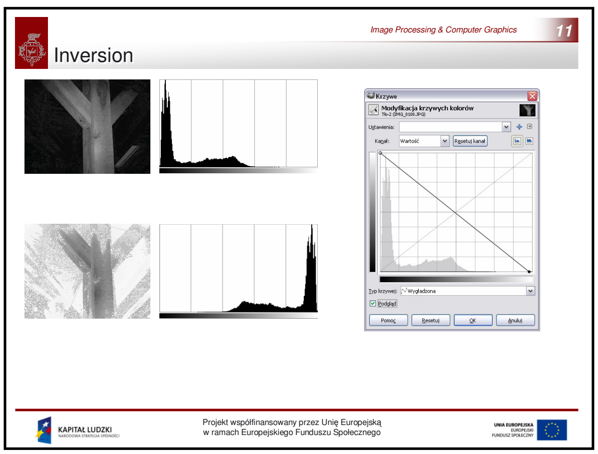


Image Processing & Computer Graphics
11
Inversion
Krzywe
Modyfikacja krzywych kolorów
Ustawienia:
Kanał:
Wartość
Resetuj kanał
Typ krzywej:
Wygładzona
Podgląd
Pomoc
Resetuj
OK
Anuluj
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA

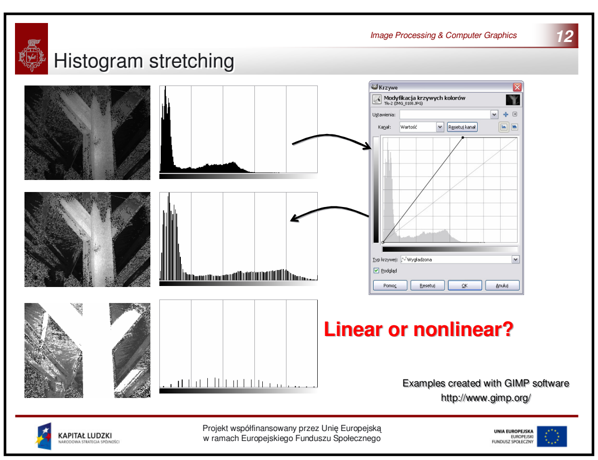
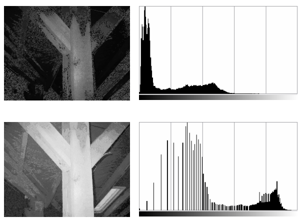
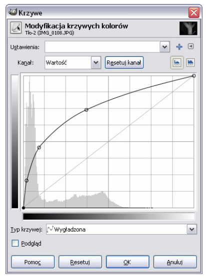


Image Processing & Computer Graphics
12
Histogram stretching
Krzywe
Modyfikacja krzywych kolorów
Ustawienia:
Kanał:
Wartość
Resetuj kanał
Typ krzywej:
Wygładzona
Podgląd
Pomoc
Resetuj
OK
Anuluj
Linear or nonlinear?
Examples created with GIMP software
http://www.gimp.org/
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA

Gamma correction





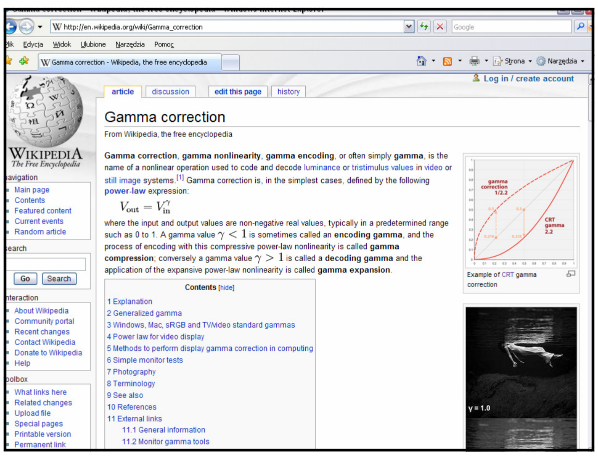
$$I_{out}(i,j) = I_{in}^{\gamma}(i,j)$$

$$I_{in} \in <0,1>$$

Linear or nonlinear?

Examples created with GIMP software http://www.gimp.org/

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY



http://en.wikipedia.org/wiki/Gamma_correction

Gamma correction - Wikipedia, the free encyclopedia

Log in / create account

article | discussion | edit this page | history

WIKIPEDIA The Free Encyclopedia

navigation: Main page, Contents, Featured content, Current events, Random article

search: Go, Search

interaction: About Wikipedia, Community portal, Recent changes, Contact Wikipedia, Donate to Wikipedia, Help

toolbox: What links here, Related changes, Upload file, Special pages, Printable version, Permanent link

Gamma correction

From Wikipedia, the free encyclopedia

Gamma correction, **gamma nonlinearity**, **gamma encoding**, or often simply **gamma**, is the name of a nonlinear operation used to code and decode luminance or tristimulus values in video or still image systems.[1] Gamma correction is, in the simplest cases, defined by the following **power-law** expression:

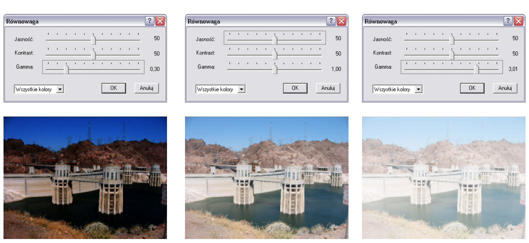
$$V_{out} = V_{in}^{\gamma}$$

where the input and output values are non-negative real values, typically in a predetermined range such as 0 to 1. A gamma value $\gamma < 1$ is sometimes called an **encoding gamma**, and the process of encoding with this compressive power-law nonlinearity is called **gamma compression**; conversely a gamma value $\gamma > 1$ is called a **decoding gamma** and the application of the expansive power-law nonlinearity is called **gamma expansion**.

Example of CRT gamma correction

Contents [hide]

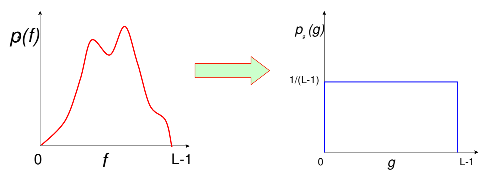
Gamma correction



Examples created with Microsoft Photo Editor

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

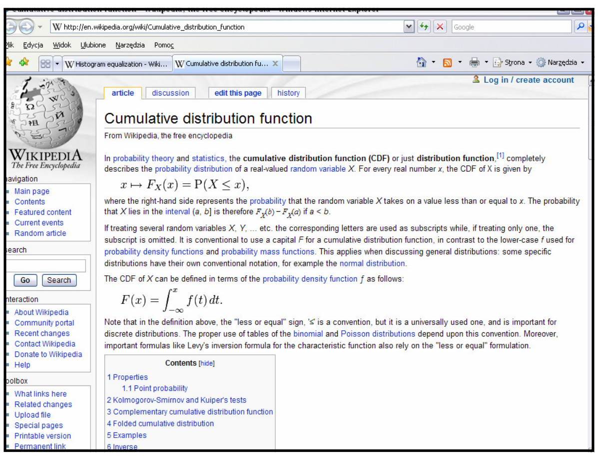
Histogram equalization (idea)



$p_f(f)$=hist[f]/MN

$p_g(g)$=1/(L-1)

Histogram equalization aims at obtaining uniform statistical distribution of image gray levels (uniform probability density function)



http://en.wikipedia.org/wiki/Cumulative_distribution_function

Cumulative distribution function

From Wikipedia, the free encyclopedia

In probability theory and statistics, the **cumulative distribution function (CDF)** or just **distribution function**,[1] completely describes the probability distribution of a real-valued random variable X. For every real number x, the CDF of X is given by

$$x \mapsto F_X(x) = \mathrm{P}(X \leq x),$$

where the right-hand side represents the probability that the random variable X takes on a value less than or equal to x. The probability that X lies in the interval $(a, b]$ is therefore $F_X(b) - F_X(a)$ if $a < b$.

If treating several random variables X, Y, ... etc. the corresponding letters are used as subscripts while, if treating only one, the subscript is omitted. It is conventional to use a capital F for a cumulative distribution function, in contrast to the lower-case f used for probability density functions and probability mass functions. This applies when discussing general distributions: some specific distributions have their own conventional notation, for example the normal distribution.

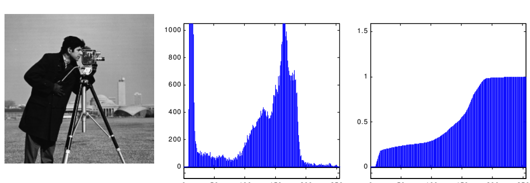
The CDF of X can be defined in terms of the probability density function f as follows:

$$F(x) = \int_{-\infty}^{x} f(t)\,dt.$$

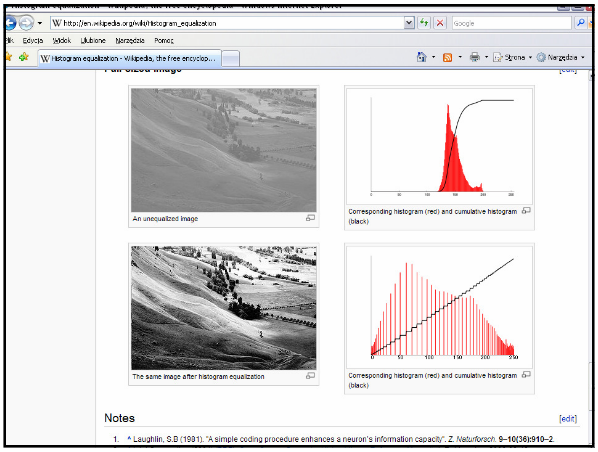
Note that in the definition above, the "less or equal" sign, '≤' is a convention, but it is a universally used one, and is important for discrete distributions. The proper use of tables of the binomial and Poisson distributions depend upon this convention. Moreover, important formulas like Levy's inversion formula for the characteristic function also rely on the "less or equal" formulation.

Contents [hide]

Cumulative histogram



Histogram **Cumulative histogram**


http://en.wikipedia.org/wiki/Histogram_equalization
Histogram equalization - Wikipedia, the free encyclop...
An unequalized image
Corresponding histogram (red) and cumulative histogram (black)
The same image after histogram equalization
Corresponding histogram (red) and cumulative histogram (black)
Notes
[edit]
1. ^ Laughlin, S.B (1981). "A simple coding procedure enhances a neuron's information capacity". Z. Naturforsch. 9–10(36):910–2.

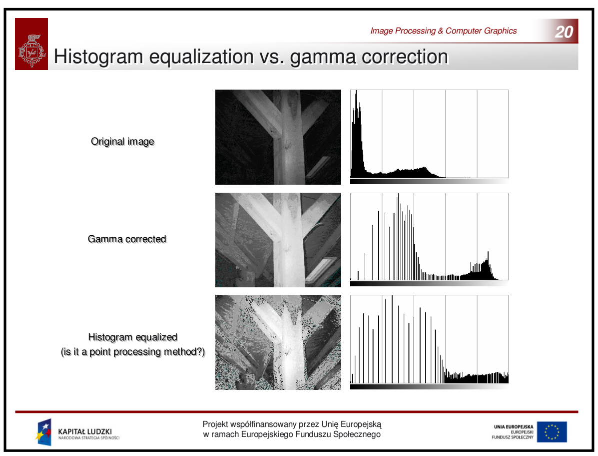


Image Processing & Computer Graphics
20
Histogram equalization vs. gamma correction
Original image
Gamma corrected
Histogram equalized
(is it a point processing method?)
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

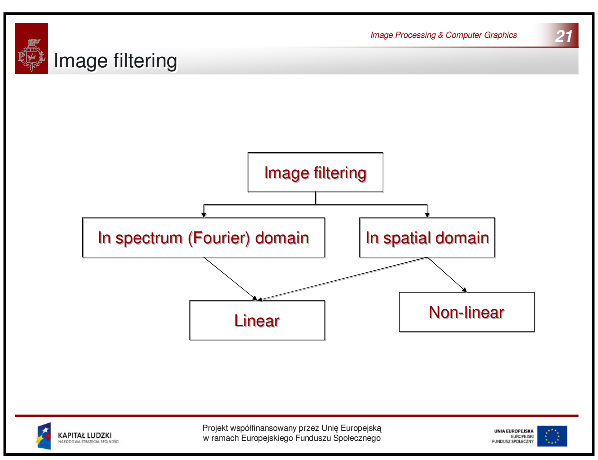
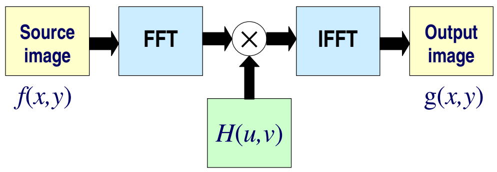


Image Processing & Computer Graphics
21
Image filtering
Image filtering
In spectrum (Fourier) domain
In spatial domain
Linear
Non-linear
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

Image filtering in spectrum domain

$$g(x,y) = IFFT\{\ H(u,v)\ FFT\{f(x,y)\}\ \}$$



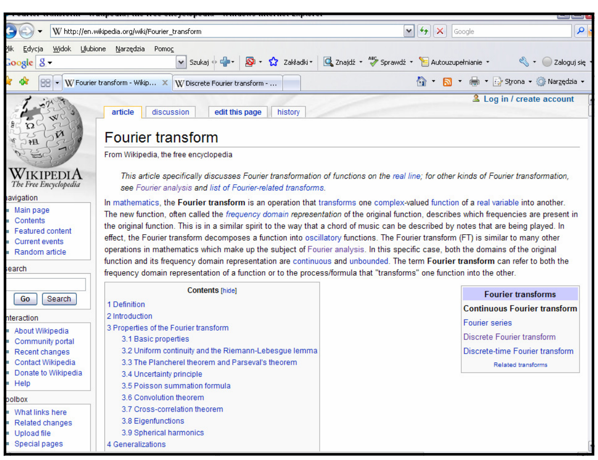
Fourier transform

Joseph Fourier
(1768-1830)

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$ **forward**

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$ **inverse**

$F(u)$ - Fourier coefficients



http://en.wikipedia.org/wiki/Fourier_transform

Plik Edycja Widok Ulubione Narzędzia Pomoc

article | discussion | edit this page | history

Log in / create account

Fourier transform

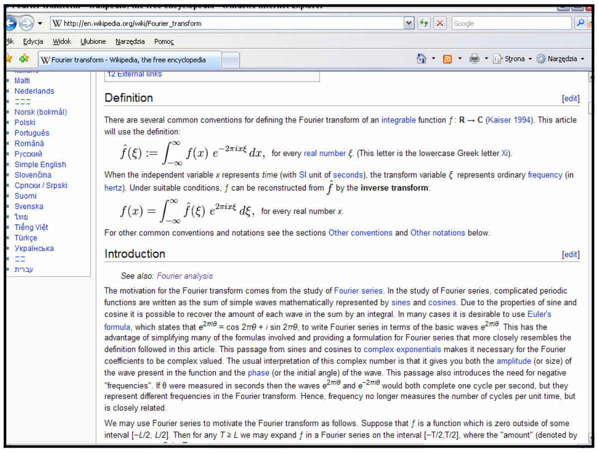
From Wikipedia, the free encyclopedia

This article specifically discusses Fourier transformation of functions on the real line; for other kinds of Fourier transformation, see Fourier analysis and list of Fourier-related transforms.

In mathematics, the **Fourier transform** is an operation that transforms one complex-valued function of a real variable into another. The new function, often called the *frequency domain representation* of the original function, describes which frequencies are present in the original function. This is in a similar spirit to the way that a chord of music can be described by notes that are being played. In effect, the Fourier transform decomposes a function into oscillatory functions. The Fourier transform (FT) is similar to many other operations in mathematics which make up the subject of Fourier analysis. In this specific case, both the domains of the original function and its frequency domain representation are continuous and unbounded. The term **Fourier transform** can refer to both the frequency domain representation of a function or to the process/formula that "transforms" one function into the other.

Contents [hide]

Fourier transforms
Continuous Fourier transform
Fourier series
Discrete Fourier transform
Discrete-time Fourier transform
Related transforms



http://en.wikipedia.org/wiki/Fourier_transform

Fourier transform - Wikipedia, the free encyclopedia

12 External links

Definition [edit]

There are several common conventions for defining the Fourier transform of an integrable function $f : \mathbf{R} \to \mathbf{C}$ (Kaiser 1994). This article will use the definition:

$$\hat{f}(\xi) := \int_{-\infty}^{\infty} f(x)\ e^{-2\pi i x \xi}\, dx,$$ for every real number ξ. (This letter is the lowercase Greek letter Xi).

When the independent variable x represents *time* (with SI unit of seconds), the transform variable ξ represents ordinary frequency (in hertz). Under suitable conditions, f can be reconstructed from $\hat{f}$ by the **inverse transform**:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\ e^{2\pi i x \xi}\, d\xi,$$ for every real number x.

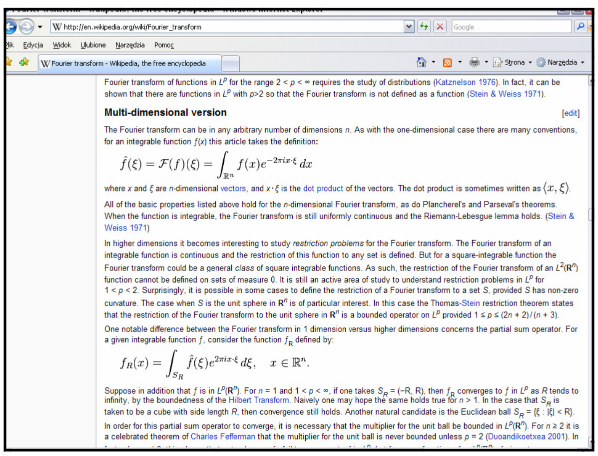
For other common conventions and notations see the sections Other conventions and Other notations below.

Introduction [edit]

See also: Fourier analysis

The motivation for the Fourier transform comes from the study of Fourier series. In the study of Fourier series, complicated periodic functions are written as the sum of simple waves mathematically represented by sines and cosines. Due to the properties of sine and cosine it is possible to recover the amount of each wave in the sum by an integral. In many cases it is desirable to use Euler's formula, which states that $e^{2\pi i\theta} = \cos 2\pi\theta + i \sin 2\pi\theta$, to write Fourier series in terms of the basic waves $e^{2\pi i\theta}$. This has the advantage of simplifying many of the formulas involved and providing a formulation for Fourier series that more closely resembles the definition followed in this article. This passage from sines and cosines to complex exponentials makes it necessary for the Fourier coefficients to be complex valued. The usual interpretation of this complex number is that it gives you both the amplitude (or size) of the wave present in the function and the phase (or the initial angle) of the wave. This passage also introduces the need for negative "frequencies". If θ were measured in seconds then the waves $e^{2\pi i\theta}$ and $e^{-2\pi i\theta}$ would both complete one cycle per second, but they represent different frequencies in the Fourier transform. Hence, frequency no longer measures the number of cycles per unit time, but is closely related.

We may use Fourier series to motivate the Fourier transform as follows. Suppose that f is a function which is zero outside of some interval $[-L/2, L/2]$. Then for any $T \geq L$ we may expand f in a Fourier series on the interval $[-T/2, T/2]$, where the "amount" (denoted by



http://en.wikipedia.org/wiki/Fourier_transform

Fourier transform - Wikipedia, the free encyclopedia

Fourier transform of functions in L^p for the range $2 < p < \infty$ requires the study of distributions (Katznelson 1976). In fact, it can be shown that there are functions in L^p with $p>2$ so that the Fourier transform is not defined as a function (Stein & Weiss 1971).

Multi-dimensional version [edit]

The Fourier transform can be in any arbitrary number of dimensions n. As with the one-dimensional case there are many conventions, for an integrable function $f(x)$ this article takes the definition:

$$\hat{f}(\xi) = \mathcal{F}(f)(\xi) = \int_{\mathbb{R}^n} f(x) e^{-2\pi i x\cdot\xi}\, dx$$

where x and ξ are n-dimensional vectors, and $x \cdot \xi$ is the dot product of the vectors. The dot product is sometimes written as $\langle x, \xi \rangle$.

All of the basic properties listed above hold for the n-dimensional Fourier transform, as do Plancherel's and Parseval's theorems. When the function is integrable, the Fourier transform is still uniformly continuous and the Riemann-Lebesgue lemma holds. (Stein & Weiss 1971)

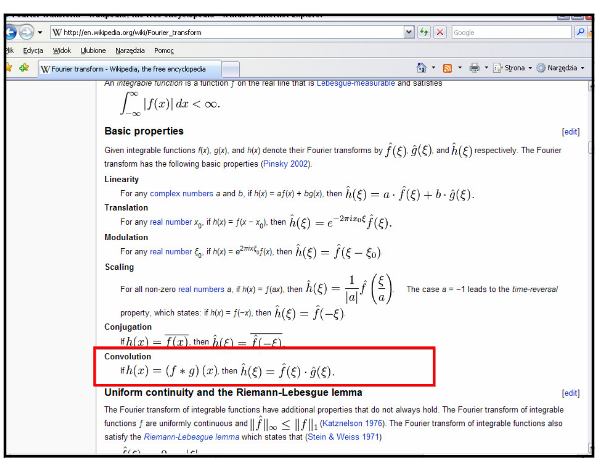
In higher dimensions it becomes interesting to study *restriction problems* for the Fourier transform. The Fourier transform of an integrable function is continuous and the restriction of this function to any set is defined. But for a square-integrable function the Fourier transform could be a general *class* of square integrable functions. As such, the restriction of the Fourier transform of an $L^2(\mathbf{R}^n)$ function cannot be defined on sets of measure 0. It is still an active area of study to understand restriction problems in L^p for $1 < p < 2$. Surprisingly, it is possible in some cases to define the restriction of a Fourier transform to a set S, provided S has non-zero curvature. The case when S is the unit sphere in $\mathbf{R}^n$ is of particular interest. In this case the Thomas-Stein restriction theorem states that the restriction of the Fourier transform to the unit sphere in $\mathbf{R}^n$ is a bounded operator on L^p provided $1 \leq p \leq (2n + 2)/(n + 3)$.

One notable difference between the Fourier transform in 1 dimension versus higher dimensions concerns the partial sum operator. For a given integrable function f, consider the function f_R defined by:

$$f_R(x) = \int_{S_R} \hat{f}(\xi) e^{2\pi i x\cdot\xi}\, d\xi, \quad x \in \mathbb{R}^n.$$

Suppose in addition that f is in $L^p(\mathbf{R}^n)$. For $n = 1$ and $1 < p < \infty$, if one takes $S_R = (-R, R)$, then f_R converges to f in L^p as R tends to infinity, by the boundedness of the Hilbert Transform. Naively one may hope the same holds true for $n > 1$. In the case that S_R is taken to be a cube with side length R, then convergence still holds. Another natural candidate is the Euclidean ball $S_R = \{\xi : |\xi| < R\}$.

In order for this partial sum operator to converge, it is necessary that the multiplier for the unit ball be bounded in $L^p(\mathbf{R}^n)$. For $n \geq 2$ it is a celebrated theorem of Charles Fefferman that the multiplier for the unit ball is never bounded unless $p = 2$ (Duoandikoetxea 2001). In



http://en.wikipedia.org/wiki/Fourier_transform

Fourier transform - Wikipedia, the free encyclopedia

An integrable function is a function f on the real line that is Lebesgue-measurable and satisfies

$$\int_{-\infty}^{\infty} |f(x)|\, dx < \infty.$$

Basic properties [edit]

Given integrable functions $f(x)$, $g(x)$, and $h(x)$ denote their Fourier transforms by $\hat{f}(\xi)$, $\hat{g}(\xi)$, and $\hat{h}(\xi)$ respectively. The Fourier transform has the following basic properties (Pinsky 2002).

Linearity
For any complex numbers a and b, if $h(x) = af(x) + bg(x)$, then $\hat{h}(\xi) = a \cdot \hat{f}(\xi) + b \cdot \hat{g}(\xi)$.

Translation
For any real number x_0, if $h(x) = f(x - x_0)$, then $\hat{h}(\xi) = e^{-2\pi i x_0 \xi} \hat{f}(\xi)$.

Modulation
For any real number ξ_0, if $h(x) = e^{2\pi i x \xi_0} f(x)$, then $\hat{h}(\xi) = \hat{f}(\xi - \xi_0)$.

Scaling
For all non-zero real numbers a, if $h(x) = f(ax)$, then $\hat{h}(\xi) = \frac{1}{|a|}\hat{f}\left(\frac{\xi}{a}\right)$. The case $a = -1$ leads to the *time-reversal* property, which states: if $h(x) = f(-x)$, then $\hat{h}(\xi) = \hat{f}(-\xi)$.

Conjugation
If $h(x) = \overline{f(x)}$, then $\hat{h}(\xi) = \overline{\hat{f}(-\xi)}$.

Convolution
If $h(x) = (f * g)(x)$, then $\hat{h}(\xi) = \hat{f}(\xi) \cdot \hat{g}(\xi)$.

Uniform continuity and the Riemann-Lebesgue lemma [edit]

The Fourier transform of integrable functions have additional properties that do not always hold. The Fourier transform of integrable functions f are uniformly continuous and $\|\hat{f}\|_\infty \leq \|f\|_1$ (Katznelson 1976). The Fourier transform of integrable functions also satisfy the *Riemann-Lebesgue lemma* which states that (Stein & Weiss 1971)

Important properties of the Fourier transform

Suppose we are given two functions f and g, with Fourier transforms F and G, and suppose that a and b are constants. Then

- The Fourier transform is linear, that is,

$$\mathcal{F}(af(x)+bg(y)) = aF(\omega)+bG(\omega).$$

- Changing spatial scale inversely affects frequency and amplitude, that is,

$$\mathcal{F}(f(ax)) = \frac{1}{a}F(\frac{\omega}{a}).$$

- Shifting the function only changes the phase of the spectrum, that is,

$$\mathcal{F}(f(x-a)) = F(\omega)e^{2\pi i\omega a}.$$

We can also take advantage of symmetries in the spatial and frequency domains as follows:

- if $f(x)$ is real, then $F(-\omega) = F(\omega)^*$
- if $f(x)$ is imaginary, then $F(-\omega) = -F(\omega)^*$
- if $f(x)$ is even, then $F(-\omega) = F(\omega)$
- if $f(x)$ is odd, then $F(-\omega) = -F(\omega)$.

Real part is an even function
Imaginary part is an odd function

Here, the notation * indicates the complex conjugate operation. Thus, if $f(x)$ is real and even, then $F(\omega)$ is real and even, and if $f(x)$ is real and odd, then

http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT4/node2.html
Fourier transform theory

In many applications only the amplitude information is needed and the phase information is discarded. However, despite this common practice, phase information should not be ignored. In images, as in sound signals, phase carries considerable information [3]. Openheim and Lim have shown that if we construct synthetic images made from the amplitude information of one image and the phase information of another, it is the image corresponding to the phase data that we perceive, if somewhat degraded.

Figure 5: The amplitude data are taken from the vdu image and the phase data are taken from the face image.

http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT4/node2.html
Fourier transform theory

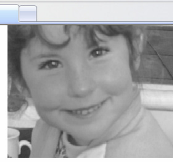
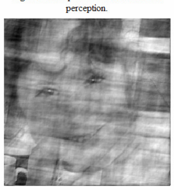
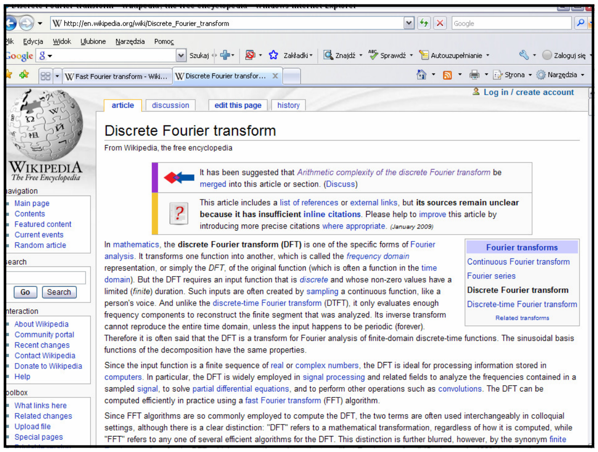


Figure 6: The phase data dominates our perception.





Discrete Fourier transform

From Wikipedia, the free encyclopedia

It has been suggested that *Arithmetic complexity of the discrete Fourier transform* be merged into this article or section. (Discuss)

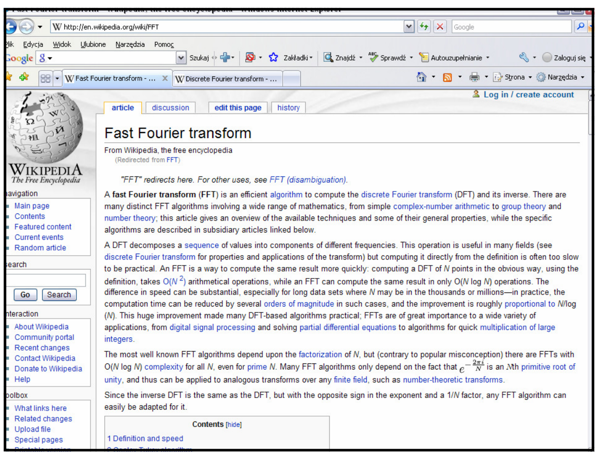
This article includes a list of references or external links, but **its sources remain unclear because it has insufficient inline citations**. Please help to improve this article by introducing more precise citations where appropriate. *(January 2009)*

In mathematics, the **discrete Fourier transform (DFT)** is one of the specific forms of Fourier analysis. It transforms one function into another, which is called the *frequency domain* representation, or simply the *DFT*, of the original function (which is often a function in the time domain). But the DFT requires an input function that is *discrete* and whose non-zero values have a limited (*finite*) duration. Such inputs are often created by sampling a continuous function, like a person's voice. And unlike the discrete-time Fourier transform (DTFT), it only evaluates enough frequency components to reconstruct the finite segment that was analyzed. Its inverse transform cannot reproduce the entire time domain, unless the input happens to be periodic (forever). Therefore it is often said that the DFT is a transform for Fourier analysis of finite-domain discrete-time functions. The sinusoidal basis functions of the decomposition have the same properties.

Fourier transforms
Continuous Fourier transform
Fourier series
Discrete Fourier transform
Discrete-time Fourier transform
Related transforms

Since the input function is a finite sequence of real or complex numbers, the DFT is ideal for processing information stored in computers. In particular, the DFT is widely employed in signal processing and related fields to analyze the frequencies contained in a sampled signal, to solve partial differential equations, and to perform other operations such as convolutions. The DFT can be computed efficiently in practice using a fast Fourier transform (FFT) algorithm.

Since FFT algorithms are so commonly employed to compute the DFT, the two terms are often used interchangeably in colloquial settings, although there is a clear distinction: "DFT" refers to a mathematical transformation, regardless of how it is computed, while "FFT" refers to any one of several efficient algorithms for the DFT. This distinction is further blurred, however, by the synonym finite



Fast Fourier transform

From Wikipedia, the free encyclopedia
(Redirected from FFT)

"FFT" redirects here. For other uses, see FFT (disambiguation).

A **fast Fourier transform** (**FFT**) is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. There are many distinct FFT algorithms involving a wide range of mathematics, from simple complex-number arithmetic to group theory and number theory; this article gives an overview of the available techniques and some of their general properties, while the specific algorithms are described in subsidiary articles linked below.

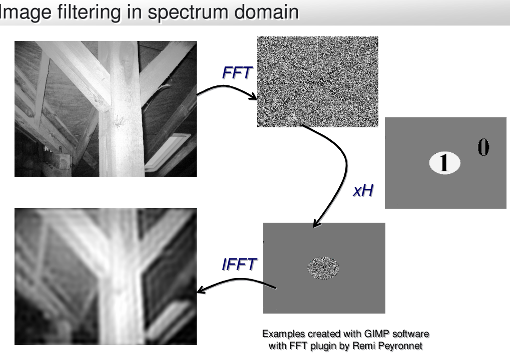
A DFT decomposes a sequence of values into components of different frequencies. This operation is useful in many fields (see discrete Fourier transform for properties and applications of the transform) but computing it directly from the definition is often too slow to be practical. An FFT is a way to compute the same result more quickly: computing a DFT of N points in the obvious way, using the definition, takes $O(N^2)$ arithmetical operations, while an FFT can compute the same result in only $O(N \log N)$ operations. The difference in speed can be substantial, especially for long data sets where N may be in the thousands or millions—in practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to $N/\log(N)$. This huge improvement made many DFT-based algorithms practical; FFTs are of great importance to a wide variety of applications, from digital signal processing and solving partial differential equations to algorithms for quick multiplication of large integers.

The most well known FFT algorithms depend upon the factorization of N, but (contrary to popular misconception) there are FFTs with $O(N \log N)$ complexity for all N, even for prime N. Many FFT algorithms only depend on the fact that $e^{-\frac{2\pi i}{N}}$ is an Nth primitive root of unity, and thus can be applied to analogous transforms over any finite field, such as number-theoretic transforms.

Since the inverse DFT is the same as the DFT, but with the opposite sign in the exponent and a $1/N$ factor, any FFT algorithm can easily be adapted for it.

Contents [hide]
1 Definition and speed

Image filtering in spectrum domain



Examples created with GIMP software
with FFT plugin by Remi Peyronnet

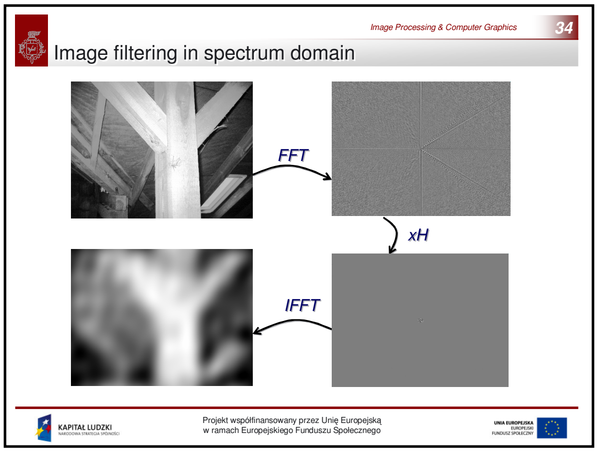


Image Processing & Computer Graphics
34
Image filtering in spectrum domain
FFT
xH
IFFT
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

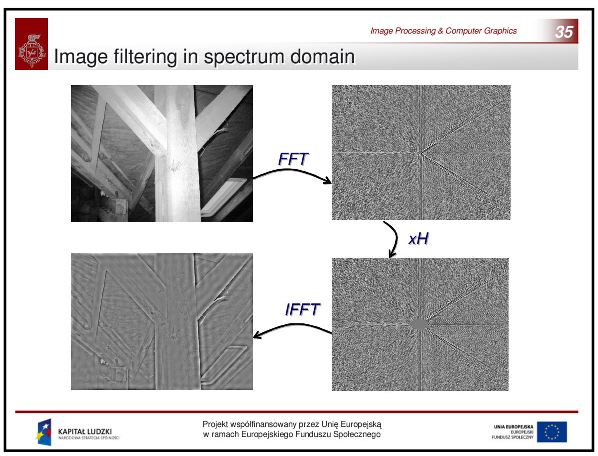


Image Processing & Computer Graphics
35
Image filtering in spectrum domain
FFT
xH
IFFT
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

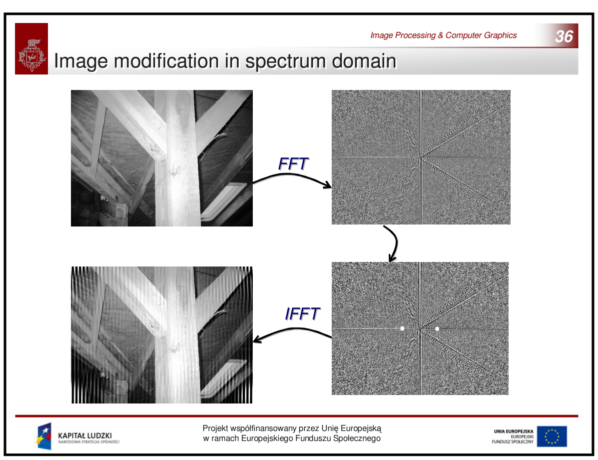
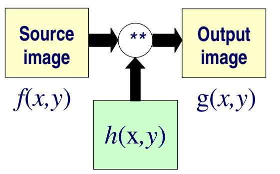


Image Processing & Computer Graphics
36
Image modification in spectrum domain
FFT
IFFT
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

Image filtering in spatial domain

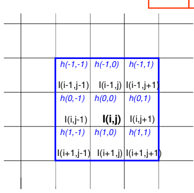


$$g(x,y) = h(x,y) ** f(x,y)$$

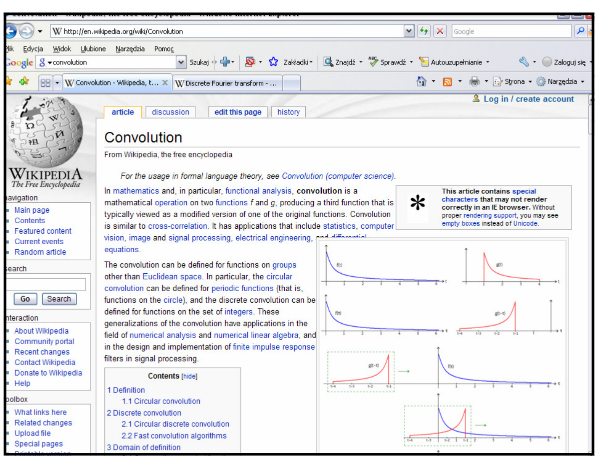
Image and filter mask convolution

h(-1,-1)	h(-1,0)	h(-1,1)
h(0,-1)	h(0,0)	h(0,1)
h(1,-1)	h(1,0)	h(1,1)

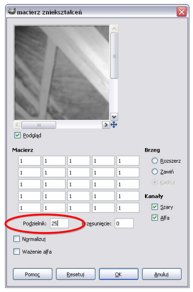
3x3 mask example



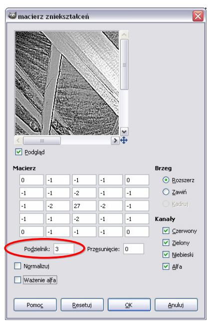
G(i,j) =
I(i-1,j-1)h(-1,-1) + I(i-1,j)h(-1,0) + I(i-1,j+1)h(-1,1)+
I(i,j-1)h(0,-1) + **I(i,j)**h(0,0) + I(i,j+1)h(0,1)+
I(i+1,j-1)h(1,-1) + I(i+1,j)h(1,0) + I(i+1,j+1)h(1,1)



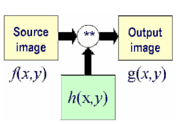
Convolution example (averaging/bluring)

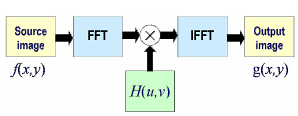


Convolution example (sharpening)



Convolution vs. Fourier transform





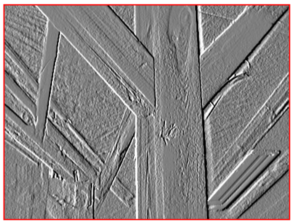
$$g(x,y) = IF\{\ H(u,v)\ F\{f(x,y)\}\ \} =$$

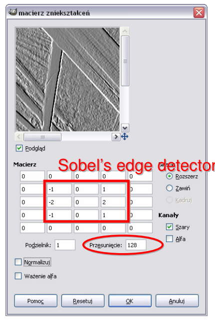
$$IF\ \{H(u,v)\} ** IF\ \{F\ \{f(x,y)\}\ \} = h(x,y) ** f(x,y)$$

$$h(x,y) = IF\ \{H(u,v)\}$$

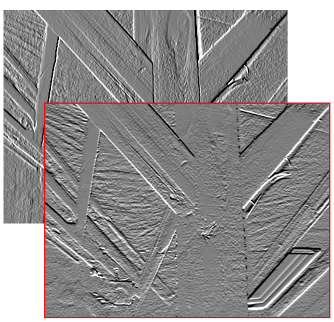
$$H(u,v) = F\{h(x,y)\}$$

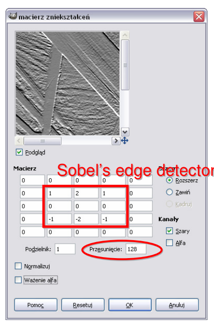
Sobel's edge (gradient) detection





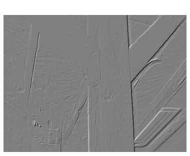
Sobel's edge (gradient) detection



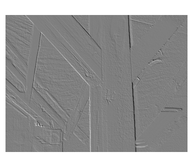


Roberts edge detection

-1	0
0	1



0	1
-1	0



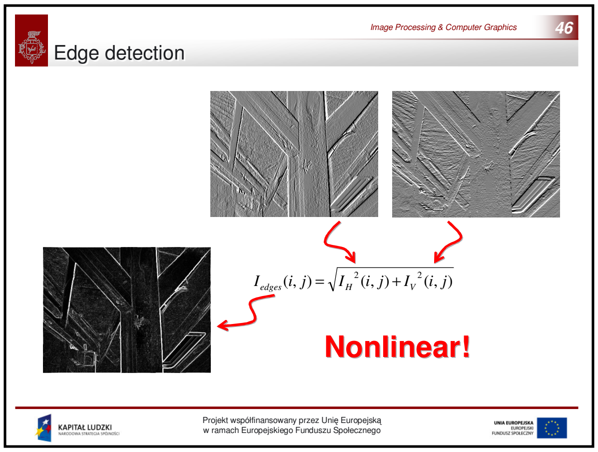
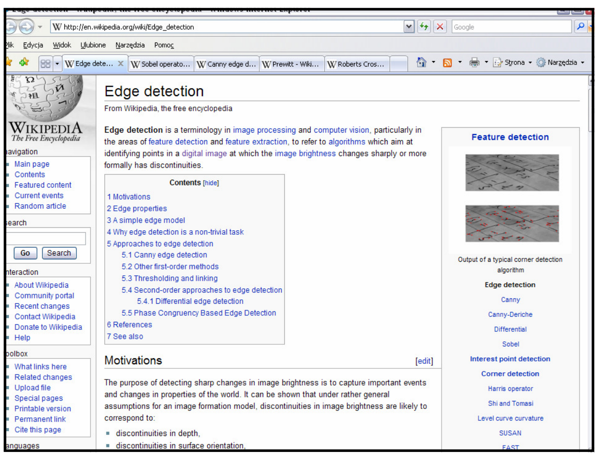
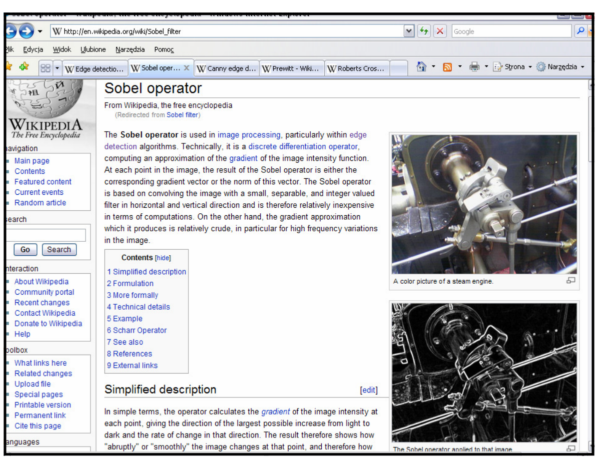


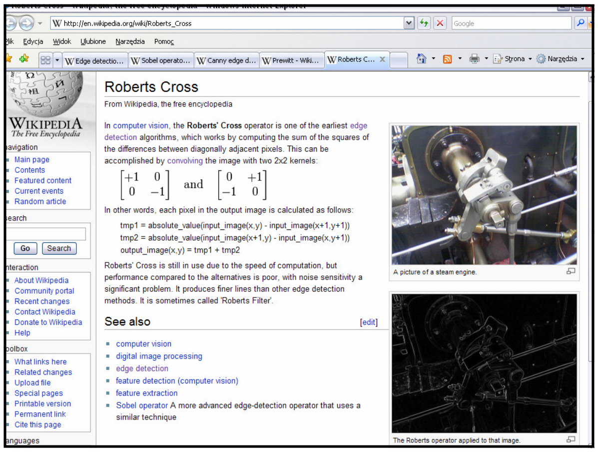
Image Processing & Computer Graphics
46
Edge detection
$I_{edges}(i,j)=\sqrt{I_H^2(i,j)+I_V^2(i,j)}$
Nonlinear!
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA



http://en.wikipedia.org/wiki/Edge_detection
Edge detection
From Wikipedia, the free encyclopedia
Edge detection is a terminology in image processing and computer vision, particularly in the areas of feature detection and feature extraction, to refer to algorithms which aim at identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities.
Contents [hide]
1 Motivations
2 Edge properties
3 A simple edge model
4 Why edge detection is a non-trivial task
5 Approaches to edge detection
5.1 Canny edge detection
5.2 Other first-order methods
5.3 Thresholding and linking
5.4 Second-order approaches to edge detection
5.4.1 Differential edge detection
5.5 Phase Congruency Based Edge Detection
6 References
7 See also
Motivations [edit]
The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. It can be shown that under rather general assumptions for an image formation model, discontinuities in image brightness are likely to correspond to:
discontinuities in depth,
discontinuities in surface orientation,
Feature detection
Output of a typical corner detection algorithm
Edge detection
Canny
Canny-Deriche
Differential
Sobel
Interest point detection
Corner detection
Harris operator
Shi and Tomasi
Level curve curvature
SUSAN



http://en.wikipedia.org/wiki/Sobel_filter
Sobel operator
From Wikipedia, the free encyclopedia
(Redirected from Sobel filter)
The Sobel operator is used in image processing, particularly within edge detection algorithms. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. The Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation which it produces is relatively crude, in particular for high frequency variations in the image.
Contents [hide]
1 Simplified description
2 Formulation
3 More formally
4 Technical details
5 Example
6 Scharr Operator
7 See also
8 References
9 External links
Simplified description [edit]
In simple terms, the operator calculates the gradient of the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. The result therefore shows how "abruptly" or "smoothly" the image changes at that point, and therefore how
A color picture of a steam engine.



http://en.wikipedia.org/wiki/Roberts_Cross

Roberts Cross

From Wikipedia, the free encyclopedia

In computer vision, the **Roberts' Cross** operator is one of the earliest edge detection algorithms, which works by computing the sum of the squares of the differences between diagonally adjacent pixels. This can be accomplished by convolving the image with two 2x2 kernels:

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$

In other words, each pixel in the output image is calculated as follows:

```
tmp1 = absolute_value(input_image(x,y) - input_image(x+1,y+1))
tmp2 = absolute_value(input_image(x+1,y) - input_image(x,y+1))
output_image(x,y) = tmp1 + tmp2
```

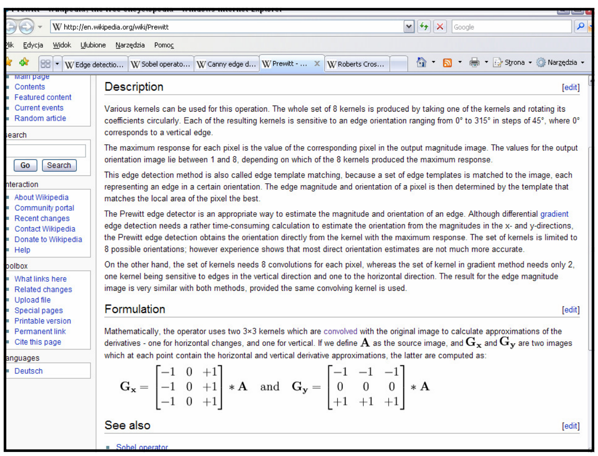
Roberts' Cross is still in use due to the speed of computation, but performance compared to the alternatives is poor, with noise sensitivity a significant problem. It produces finer lines than other edge detection methods. It is sometimes called 'Roberts Filter'.

A picture of a steam engine.

The Roberts operator applied to that image.

See also [edit]

- computer vision
- digital image processing
- edge detection
- feature detection (computer vision)
- feature extraction
- Sobel operator A more advanced edge-detection operator that uses a similar technique



http://en.wikipedia.org/wiki/Prewitt

Description [edit]

Various kernels can be used for this operation. The whole set of 8 kernels is produced by taking one of the kernels and rotating its coefficients circularly. Each of the resulting kernels is sensitive to an edge orientation ranging from 0° to 315° in steps of 45°, where 0° corresponds to a vertical edge.

The maximum response for each pixel is the value of the corresponding pixel in the output magnitude image. The values for the output orientation image lie between 1 and 8, depending on which of the 8 kernels produced the maximum response.

This edge detection method is also called edge template matching, because a set of edge templates is matched to the image, each representing an edge in a certain orientation. The edge magnitude and orientation of a pixel is then determined by the template that matches the local area of the pixel the best.

The Prewitt edge detector is an appropriate way to estimate the magnitude and orientation of an edge. Although differential gradient edge detection needs a rather time-consuming calculation to estimate the orientation from the magnitudes in the x- and y-directions, the Prewitt edge detection obtains the orientation directly from the kernel with the maximum response. The set of kernels is limited to 8 possible orientations; however experience shows that most direct orientation estimates are not much more accurate.

On the other hand, the set of kernels needs 8 convolutions for each pixel, whereas the set of kernel in gradient method needs only 2, one kernel being sensitive to edges in the vertical direction and one to the horizontal direction. The result for the edge magnitude image is very similar with both methods, provided the same convolving kernel is used.

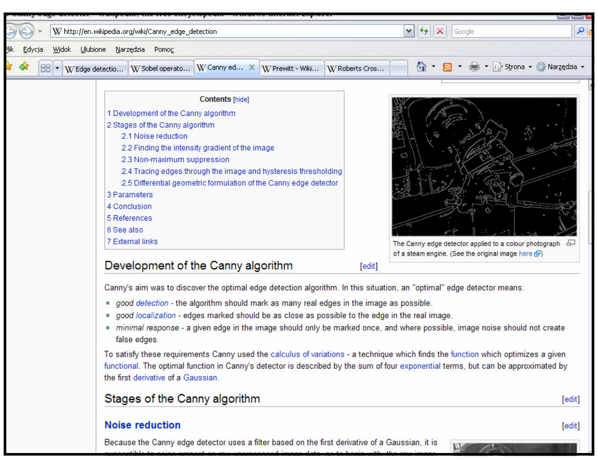
Formulation [edit]

Mathematically, the operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical. If we define $\mathbf{A}$ as the source image, and $\mathbf{G_x}$ and $\mathbf{G_y}$ are two images which at each point contain the horizontal and vertical derivative approximations, the latter are computed as:

$$\mathbf{G_x} = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G_y} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * \mathbf{A}$$

See also [edit]

- Sobel operator



http://en.wikipedia.org/wiki/Canny_edge_detection

Contents [hide]

The Canny edge detector applied to a colour photograph of a steam engine. (See the original image here)

Development of the Canny algorithm [edit]

Canny's aim was to discover the optimal edge detection algorithm. In this situation, an "optimal" edge detector means:

- *good detection* - the algorithm should mark as many real edges in the image as possible.
- *good localization* - edges marked should be as close as possible to the edge in the real image.
- *minimal response* - a given edge in the image should only be marked once, and where possible, image noise should not create false edges.

To satisfy these requirements Canny used the calculus of variations - a technique which finds the function which optimizes a given functional. The optimal function in Canny's detector is described by the sum of four exponential terms, but can be approximated by the first derivative of a Gaussian.

Stages of the Canny algorithm [edit]

Noise reduction [edit]

Because the Canny edge detector uses a filter based on the first derivative of a Gaussian, it is

Nonlinear filters

~~G(i,j) =
I(i-1,j-1)h(-1,-1) + I(i-1,j)h(-1,0) + I(i-1,j+1)h(-1,1)+
I(i,j-1)h(0,-1) + **I(i,j)**h(0,0) + I(i,j+1)h(0,1)+
I(i+1,j-1)h(1,-1) + I(i+1,j)h(1,0) + I(i+1,j+1)h(1,1)~~

~~$h(x,y) = IF\{H(u,v)\}$~~

~~$H(u,v) = F\{h(x,y)\}$~~

Median filter

The median $\boldsymbol{m}$ of a set of values (e.g. image pixels in the filtering mask) is such that half the elements in the set are less than $\boldsymbol{m}$ and other half are grater than $\boldsymbol{m}$.

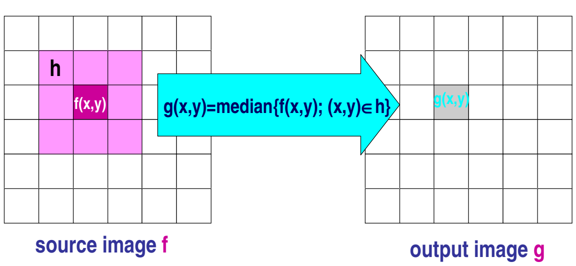
$x(n)=\{1, 5, -7, 101, -25, 3, 0, 11, 7\}$

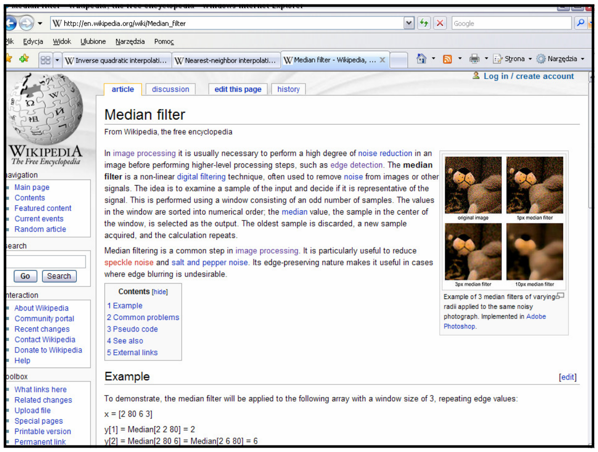
Sorted sequence of elements:

$x_s(n)=\{-25, -7, 0, 1, \mathbf{3}, 5, 7, 11, 101\}$

median

Median filter





Median filter

From Wikipedia, the free encyclopedia

In image processing it is usually necessary to perform a high degree of noise reduction in an image before performing higher-level processing steps, such as edge detection. The **median filter** is a non-linear digital filtering technique, often used to remove noise from images or other signals. The idea is to examine a sample of the input and decide if it is representative of the signal. This is performed using a window consisting of an odd number of samples. The values in the window are sorted into numerical order; the median value, the sample in the center of the window, is selected as the output. The oldest sample is discarded, a new sample acquired, and the calculation repeats.

Median filtering is a common step in image processing. It is particularly useful to reduce speckle noise and salt and pepper noise. Its edge-preserving nature makes it useful in cases where edge blurring is undesirable.

Example of 3 median filters of varying radii applied to the same noisy photograph. Implemented in Adobe Photoshop.

Contents [hide]

Example

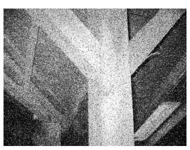
To demonstrate, the median filter will be applied to the following array with a window size of 3, repeating edge values:

x = [2 80 6 3]

y[1] = Median[2 2 80] = 2

y[2] = Median[2 80 6] = Median[2 6 80] = 6

Median filter



Examples created with Corel Photo-Paint 8

Image morphology

Median filter…

What about minimum or maximum filters?

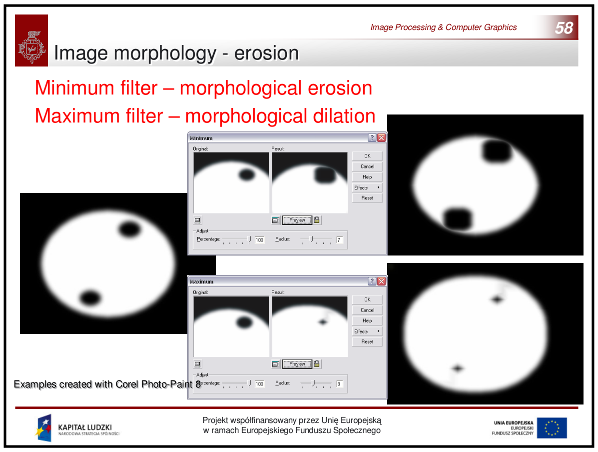


Image Processing & Computer Graphics
58
Image morphology - erosion
Minimum filter – morphological erosion
Maximum filter – morphological dilation
Examples created with Corel Photo-Paint 8
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

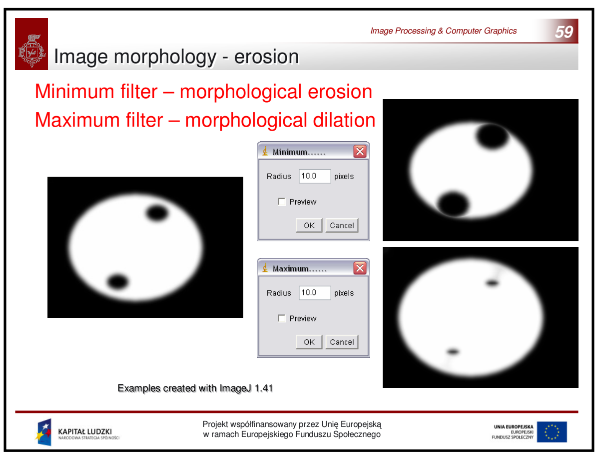
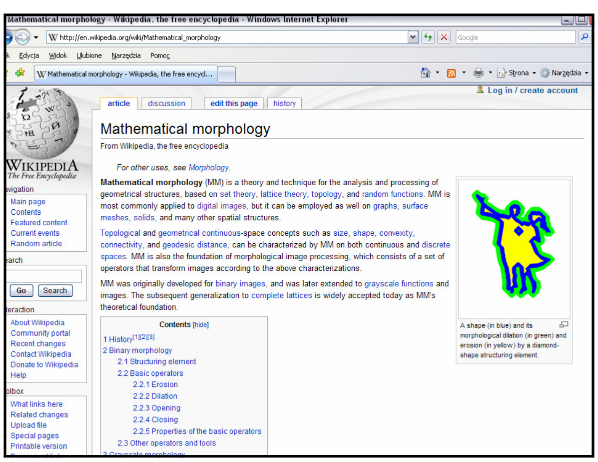
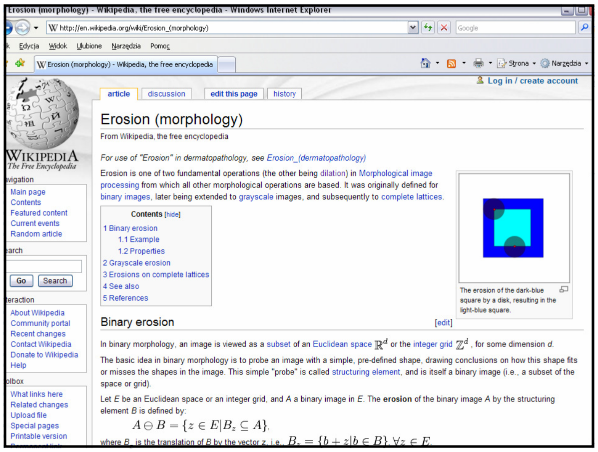
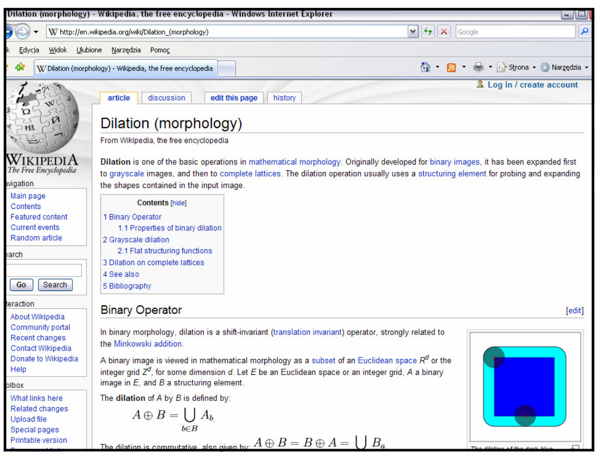


Image Processing & Computer Graphics
59
Image morphology - erosion
Minimum filter – morphological erosion
Maximum filter – morphological dilation
Minimum......
Radius 10.0 pixels
Preview
OK Cancel
Maximum......
Radius 10.0 pixels
Preview
OK Cancel
Examples created with ImageJ 1.41
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego


Mathematical morphology - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/Mathematical_morphology
article
discussion
edit this page
history
Log in / create account
Mathematical morphology
From Wikipedia, the free encyclopedia
For other uses, see Morphology.
Mathematical morphology (MM) is a theory and technique for the analysis and processing of geometrical structures, based on set theory, lattice theory, topology, and random functions. MM is most commonly applied to digital images, but it can be employed as well on graphs, surface meshes, solids, and many other spatial structures.
Topological and geometrical continuous-space concepts such as size, shape, convexity, connectivity, and geodesic distance, can be characterized by MM on both continuous and discrete spaces. MM is also the foundation of morphological image processing, which consists of a set of operators that transform images according to the above characterizations.
MM was originally developed for binary images, and was later extended to grayscale functions and images. The subsequent generalization to complete lattices is widely accepted today as MM's theoretical foundation.
A shape (in blue) and its morphological dilation (in green) and erosion (in yellow) by a diamond-shape structuring element.
Contents [hide]
1 History
2 Binary morphology
2.1 Structuring element
2.2 Basic operators
2.2.1 Erosion
2.2.2 Dilation
2.2.3 Opening
2.2.4 Closing
2.2.5 Properties of the basic operators
2.3 Other operators and tools
WIKIPEDIA
The Free Encyclopedia
Main page
Contents
Featured content
Current events
Random article
Go
Search
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Donate to Wikipedia
Help
What links here
Related changes
Upload file
Special pages
Printable version



Erosion (morphology) - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/Erosion_(morphology)
Log in / create account
article
discussion
edit this page
history
Erosion (morphology)
From Wikipedia, the free encyclopedia
For use of "Erosion" in dermatopathology, see Erosion_(dermatopathology)
Erosion is one of two fundamental operations (the other being dilation) in Morphological image processing from which all other morphological operations are based. It was originally defined for binary images, later being extended to grayscale images, and subsequently to complete lattices.
Contents [hide]
1 Binary erosion
1.1 Example
1.2 Properties
2 Grayscale erosion
3 Erosions on complete lattices
4 See also
5 References
The erosion of the dark-blue square by a disk, resulting in the light-blue square.
Binary erosion
[edit]
In binary morphology, an image is viewed as a subset of an Euclidean space $\mathbb{R}^d$ or the integer grid $\mathbb{Z}^d$, for some dimension d.
The basic idea in binary morphology is to probe an image with a simple, pre-defined shape, drawing conclusions on how this shape fits or misses the shapes in the image. This simple "probe" is called structuring element, and is itself a binary image (i.e., a subset of the space or grid).
Let E be an Euclidean space or an integer grid, and A a binary image in E. The erosion of the binary image A by the structuring element B is defined by:
$A \ominus B = \{z \in E | B_z \subseteq A\}$,



Dilation (morphology) - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/Dilation_(morphology)
Log in / create account
article
discussion
edit this page
history
Dilation (morphology)
From Wikipedia, the free encyclopedia
Dilation is one of the basic operations in mathematical morphology. Originally developed for binary images, it has been expanded first to grayscale images, and then to complete lattices. The dilation operation usually uses a structuring element for probing and expanding the shapes contained in the input image.
Contents [hide]
1 Binary Operator
1.1 Properties of binary dilation
2 Grayscale dilation
2.1 Flat structuring functions
3 Dilation on complete lattices
4 See also
5 Bibliography
Binary Operator
[edit]
In binary morphology, dilation is a shift-invariant (translation invariant) operator, strongly related to the Minkowski addition.
A binary image is viewed in mathematical morphology as a subset of an Euclidean space R^d or the integer grid Z^d, for some dimension d. Let E be an Euclidean space or an integer grid, A a binary image in E, and B a structuring element.
The dilation of A by B is defined by:
$A \oplus B = \bigcup_{b \in B} A_b$

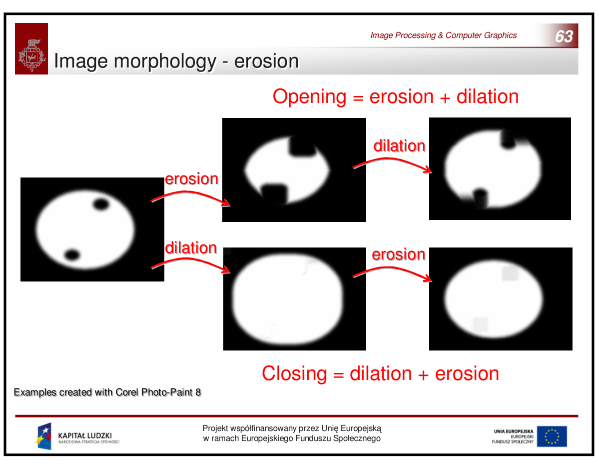
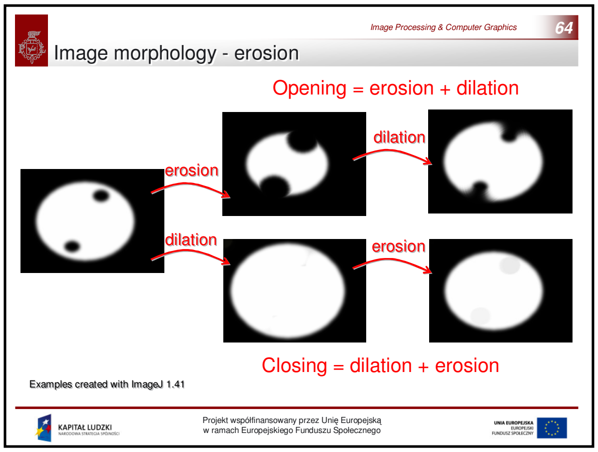
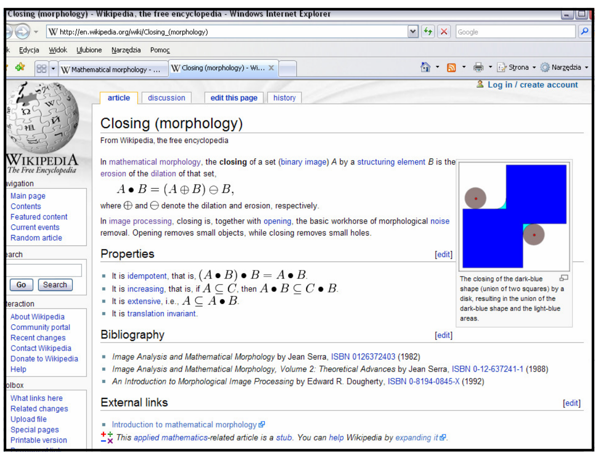


Image Processing & Computer Graphics
63
Image morphology - erosion
Opening = erosion + dilation
erosion
dilation
dilation
erosion
Closing = dilation + erosion
Examples created with Corel Photo-Paint 8
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego





Closing (morphology)

From Wikipedia, the free encyclopedia

In mathematical morphology, the **closing** of a set (binary image) *A* by a structuring element *B* is the erosion of the dilation of that set,

$$A \bullet B = (A \oplus B) \ominus B,$$

where $\oplus$ and $\ominus$ denote the dilation and erosion, respectively.

In image processing, closing is, together with opening, the basic workhorse of morphological noise removal. Opening removes small objects, while closing removes small holes.

The closing of the dark-blue shape (union of two squares) by a disk, resulting in the union of the dark-blue shape and the light-blue areas.

Properties [edit]

- It is idempotent, that is, $(A \bullet B) \bullet B = A \bullet B$.
- It is increasing, that is, if $A \subseteq C$, then $A \bullet B \subseteq C \bullet B$.
- It is extensive, i.e., $A \subseteq A \bullet B$.
- It is translation invariant.

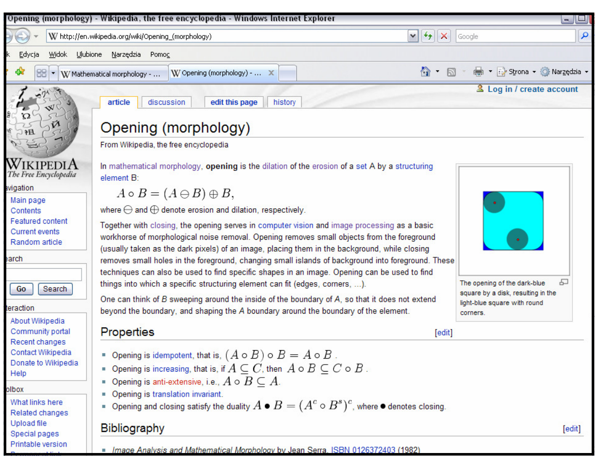
Bibliography [edit]

- *Image Analysis and Mathematical Morphology* by Jean Serra, ISBN 0126372403 (1982)
- *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances* by Jean Serra, ISBN 0-12-637241-1 (1988)
- *An Introduction to Morphological Image Processing* by Edward R. Dougherty, ISBN 0-8194-0845-X (1992)

External links [edit]

- Introduction to mathematical morphology

This applied mathematics-related article is a stub. You can help Wikipedia by expanding it.



Opening (morphology)

From Wikipedia, the free encyclopedia

In mathematical morphology, **opening** is the dilation of the erosion of a set A by a structuring element B:

$$A \circ B = (A \ominus B) \oplus B,$$

where $\ominus$ and $\oplus$ denote erosion and dilation, respectively.

Together with closing, the opening serves in computer vision and image processing as a basic workhorse of morphological noise removal. Opening removes small objects from the foreground (usually taken as the dark pixels) of an image, placing them in the background, while closing removes small holes in the foreground, changing small islands of background into foreground. These techniques can also be used to find specific shapes in an image. Opening can be used to find things into which a specific structuring element can fit (edges, corners, ...).

One can think of *B* sweeping around the inside of the boundary of *A*, so that it does not extend beyond the boundary, and shaping the *A* boundary around the boundary of the element.

The opening of the dark-blue square by a disk, resulting in the light-blue square with round corners.

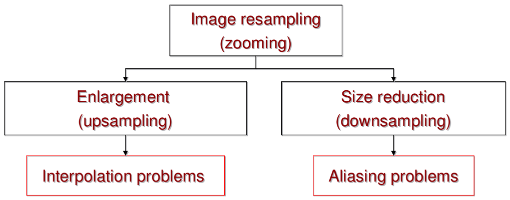
Properties [edit]

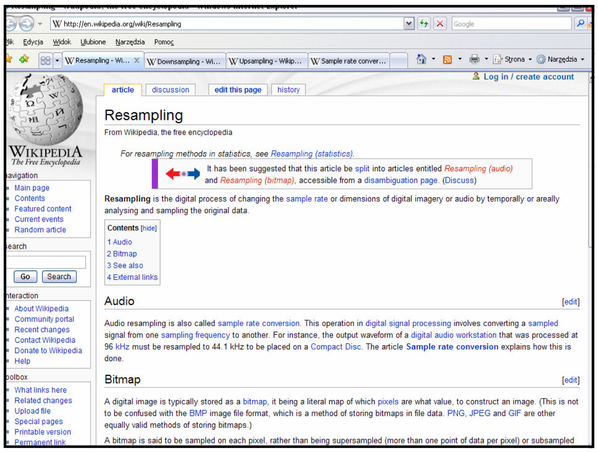
- Opening is idempotent, that is, $(A \circ B) \circ B = A \circ B$.
- Opening is increasing, that is, if $A \subseteq C$, then $A \circ B \subseteq C \circ B$.
- Opening is anti-extensive, i.e., $A \circ B \subseteq A$.
- Opening is translation invariant.
- Opening and closing satisfy the duality $A \bullet B = (A^c \circ B^s)^c$, where $\bullet$ denotes closing.

Bibliography [edit]

- *Image Analysis and Mathematical Morphology* by Jean Serra, ISBN 0126372403 (1982)

Image resampling (zooming)





http://en.wikipedia.org/wiki/Resampling

Resampling

From Wikipedia, the free encyclopedia

For resampling methods in statistics, see Resampling (statistics).

It has been suggested that this article be split into articles entitled *Resampling (audio)* and *Resampling (bitmap)*, accessible from a disambiguation page. (Discuss)

Resampling is the digital process of changing the sample rate or dimensions of digital imagery or audio by temporally or areally analysing and sampling the original data.

Contents [hide]

Audio [edit]

Audio resampling is also called sample rate conversion. This operation in digital signal processing involves converting a sampled signal from one sampling frequency to another. For instance, the output waveform of a digital audio workstation that was processed at 96 kHz must be resampled to 44.1 kHz to be placed on a Compact Disc. The article **Sample rate conversion** explains how this is done.

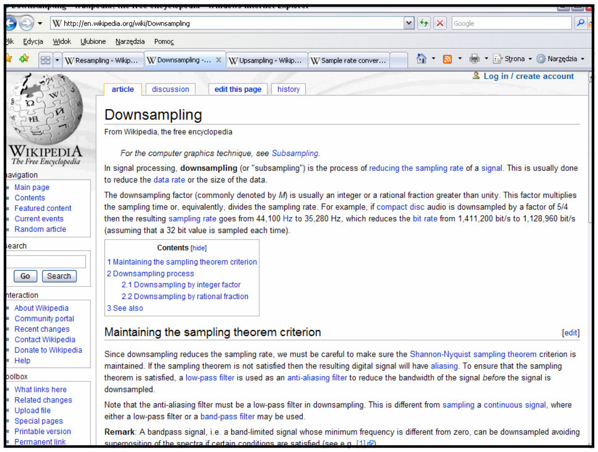
Bitmap [edit]

A digital image is typically stored as a bitmap, it being a literal map of which pixels are what value, to construct an image. (This is not to be confused with the BMP image file format, which is a method of storing bitmaps in file data. PNG, JPEG and GIF are other equally valid methods of storing bitmaps.)

A bitmap is said to be sampled on each pixel, rather than being supersampled (more than one point of data per pixel) or subsampled

Downsampling

Just take every n-th pixel!
SIMPLE!?



http://en.wikipedia.org/wiki/Downsampling

Downsampling

From Wikipedia, the free encyclopedia

For the computer graphics technique, see Subsampling.

In signal processing, **downsampling** (or "subsampling") is the process of reducing the sampling rate of a signal. This is usually done to reduce the data rate or the size of the data.

The downsampling factor (commonly denoted by *M*) is usually an integer or a rational fraction greater than unity. This factor multiplies the sampling time or, equivalently, divides the sampling rate. For example, if compact disc audio is downsampled by a factor of 5/4 then the resulting sampling rate goes from 44,100 Hz to 35,280 Hz, which reduces the bit rate from 1,411,200 bit/s to 1,128,960 bit/s (assuming that a 32 bit value is sampled each time).

Contents [hide]

Maintaining the sampling theorem criterion [edit]

Since downsampling reduces the sampling rate, we must be careful to make sure the Shannon-Nyquist sampling theorem criterion is maintained. If the sampling theorem is not satisfied then the resulting digital signal will have aliasing. To ensure that the sampling theorem is satisfied, a low-pass filter is used as an anti-aliasing filter to reduce the bandwidth of the signal *before* the signal is downsampled.

Note that the anti-aliasing filter must be a low-pass filter in downsampling. This is different from sampling a continuous signal, where either a low-pass filter or a band-pass filter may be used.

Remark: A bandpass signal, i.e. a band-limited signal whose minimum frequency is different from zero, can be downsampled avoiding superposition of the spectra if certain conditions are satisfied (see e.g. [1])

http://en.wikipedia.org/wiki/Nyquist_frequency

Nyquist frequency

From Wikipedia, the free encyclopedia

Not to be confused with Nyquist rate.

The **Nyquist frequency**, named after the Swedish-American engineer Harry Nyquist or the Nyquist–Shannon sampling theorem, is half the sampling frequency of a discrete signal processing system.[1][2] It is sometimes called the **folding frequency**, or the **cut-off frequency** of a sampling system.[3]

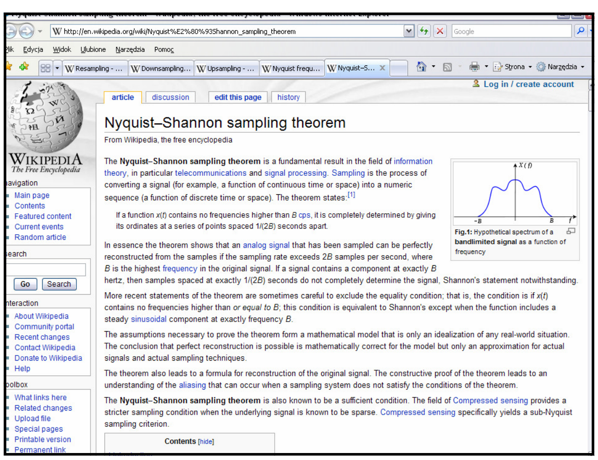
The sampling theorem shows that aliasing can be avoided if the Nyquist frequency is greater than the bandwidth, or maximum component frequency, of the signal being sampled.

The Nyquist frequency should not be confused with the *Nyquist rate*, which is the lower bound of the sampling frequency that satisfies the Nyquist sampling criterion for a given signal or family of signals. This lower bound is twice the bandwidth or maximum component frequency of the signal. *Nyquist rate*, as commonly used with respect to sampling, is a property of a continuous-time signal, not of a system, whereas *Nyquist frequency* is a property of a discrete-time system, not of a signal. The domain of the signals does not have to be time, though that is common, leading to Nyquist frequency in hertz; for example, an image sampling system has a Nyquist frequency expressed in units such as cycles per meter.

Contents [hide]

The aliasing problem [edit]

In principle, a Nyquist frequency just larger than the signal bandwidth is sufficient to allow perfect reconstruction of the signal from the samples. However, this reconstruction requires an unattainable filter that passes some frequencies unchanged while suppressing all others completely (commonly called a brickwall filter). When attainable filters are used, some degree of oversampling is necessary to accommodate the practical constraints on anti-aliasing filters. That is, frequencies close to the Nyquist frequency may be distorted in



Nyquist–Shannon sampling theorem

From Wikipedia, the free encyclopedia

The **Nyquist–Shannon sampling theorem** is a fundamental result in the field of information theory, in particular telecommunications and signal processing. Sampling is the process of converting a signal (for example, a function of continuous time or space) into a numeric sequence (a function of discrete time or space). The theorem states:[1]

If a function *x*(*t*) contains no frequencies higher than *B* cps, it is completely determined by giving its ordinates at a series of points spaced 1/(2*B*) seconds apart.

In essence the theorem shows that an analog signal that has been sampled can be perfectly reconstructed from the samples if the sampling rate exceeds 2*B* samples per second, where *B* is the highest frequency in the original signal. If a signal contains a component at exactly *B* hertz, then samples spaced at exactly 1/(2*B*) seconds do not completely determine the signal, Shannon's statement notwithstanding.

More recent statements of the theorem are sometimes careful to exclude the equality condition; that is, the condition is if *x*(*t*) contains no frequencies higher than *or equal to B*; this condition is equivalent to Shannon's except when the function includes a steady sinusoidal component at exactly frequency *B*.

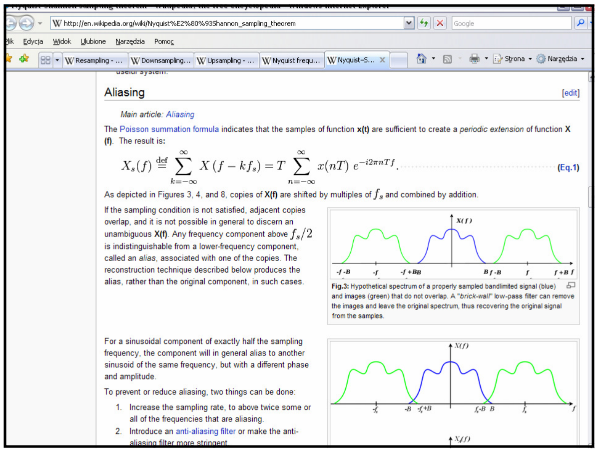
The assumptions necessary to prove the theorem form a mathematical model that is only an idealization of any real-world situation. The conclusion that perfect reconstruction is possible is mathematically correct for the model but only an approximation for actual signals and actual sampling techniques.

The theorem also leads to a formula for reconstruction of the original signal. The constructive proof of the theorem leads to an understanding of the aliasing that can occur when a sampling system does not satisfy the conditions of the theorem.

The **Nyquist–Shannon sampling theorem** is also known to be a sufficient condition. The field of Compressed sensing provides a stricter sampling condition when the underlying signal is known to be sparse. Compressed sensing specifically yields a sub-Nyquist sampling criterion.

Fig.1: Hypothetical spectrum of a **bandlimited signal** as a function of frequency

Contents [hide]



http://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem

Aliasing

Main article: Aliasing

The Poisson summation formula indicates that the samples of function **x(t)** are sufficient to create a *periodic extension* of function **X(f)**. The result is:

$$X_s(f) \stackrel{\mathrm{def}}{=} \sum_{k=-\infty}^{\infty} X(f - kf_s) = T \sum_{n=-\infty}^{\infty} x(nT)\, e^{-i2\pi nTf}. \quad \text{(Eq.1)}$$

As depicted in Figures 3, 4, and 8, copies of **X(f)** are shifted by multiples of f_s and combined by addition.

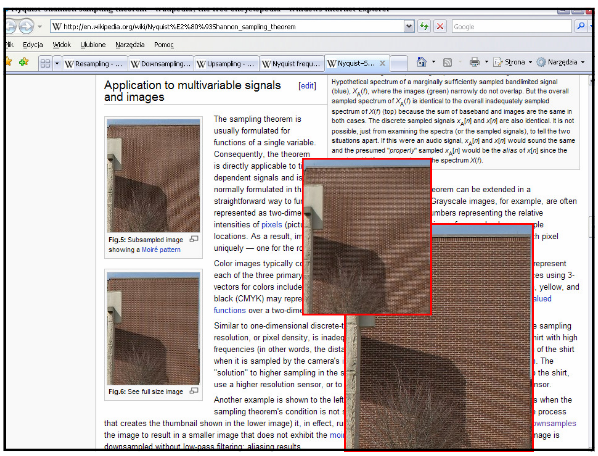
If the sampling condition is not satisfied, adjacent copies overlap, and it is not possible in general to discern an unambiguous **X(f)**. Any frequency component above $f_s/2$ is indistinguishable from a lower-frequency component, called an *alias*, associated with one of the copies. The reconstruction technique described below produces the alias, rather than the original component, in such cases.

Fig.3: Hypothetical spectrum of a properly sampled bandlimited signal (blue) and images (green) that do not overlap. A "*brick-wall*" low-pass filter can remove the images and leave the original spectrum, thus recovering the original signal from the samples.

For a sinusoidal component of exactly half the sampling frequency, the component will in general alias to another sinusoid of the same frequency, but with a different phase and amplitude.

To prevent or reduce aliasing, two things can be done:

1. Increase the sampling rate, to above twice some or all of the frequencies that are aliasing.
2. Introduce an anti-aliasing filter or make the anti-aliasing filter more stringent.



http://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem

Hypothetical spectrum of a marginally sufficiently sampled bandlimited signal (blue), $X_A(f)$, where the images (green) narrowly do not overlap. But the overall sampled spectrum of $X_A(f)$ is identical to the overall inadequately sampled spectrum of $X(f)$ (top) because the sum of baseband and images are the same in both cases. The discrete sampled signals $x_A[n]$ and $x[n]$ are also identical. It is not possible, just from examining the spectra (or the sampled signals), to tell the two situations apart. If this were an audio signal, $x_A[n]$ and $x[n]$ would sound the same and the presumed "*properly*" sampled $x_A[n]$ would be the *alias* of $x[n]$ since the ... the spectrum $X(f)$.

Application to multivariable signals and images

Fig.5: Subsampled image showing a Moiré pattern

Fig.6: See full size image

The sampling theorem is usually formulated for functions of a single variable. Consequently, the theorem is directly applicable to ti... dependent signals and is normally formulated in th... straightforward way to fu... represented as two-dime... intensities of pixels (pict... locations. As a result, im... uniquely — one for the ro...

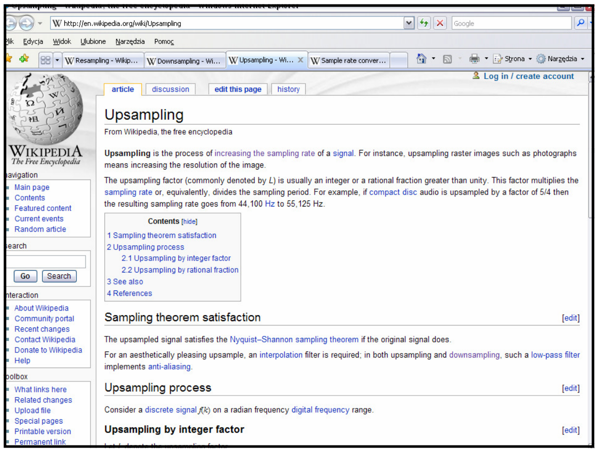
Color images typically c... each of the three primary... vectors for colors include... black (CMYK) may repre... functions over a two-dime...

Similar to one-dimensional discrete-t... resolution, or pixel density, is inadeq... frequencies (in other words, the dista... when it is sampled by the camera's ... "solution" to higher sampling in the s... use a higher resolution sensor, or to ...

Another example is shown to the left... sampling theorem's condition is not ... that creates the thumbnail shown in the lower image) it, in effect, ru... the image to result in a smaller image that does not exhibit the moi... downsampled without low-pass filtering, aliasing results.

Upsampling

This must be simple!
Let's multiply pixels!


http://en.wikipedia.org/wiki/Upsampling
Upsampling
From Wikipedia, the free encyclopedia
Upsampling is the process of increasing the sampling rate of a signal. For instance, upsampling raster images such as photographs means increasing the resolution of the image.
The upsampling factor (commonly denoted by L) is usually an integer or a rational fraction greater than unity. This factor multiplies the sampling rate or, equivalently, divides the sampling period. For example, if compact disc audio is upsampled by a factor of 5/4 then the resulting sampling rate goes from 44,100 Hz to 55,125 Hz.
Contents
1 Sampling theorem satisfaction
2 Upsampling process
2.1 Upsampling by integer factor
2.2 Upsampling by rational fraction
3 See also
4 References
Sampling theorem satisfaction
The upsampled signal satisfies the Nyquist–Shannon sampling theorem if the original signal does.
For an aesthetically pleasing upsample, an interpolation filter is required; in both upsampling and downsampling, such a low-pass filter implements anti-aliasing.
Upsampling process
Consider a discrete signal f(k) on a radian frequency digital frequency range.
Upsampling by integer factor

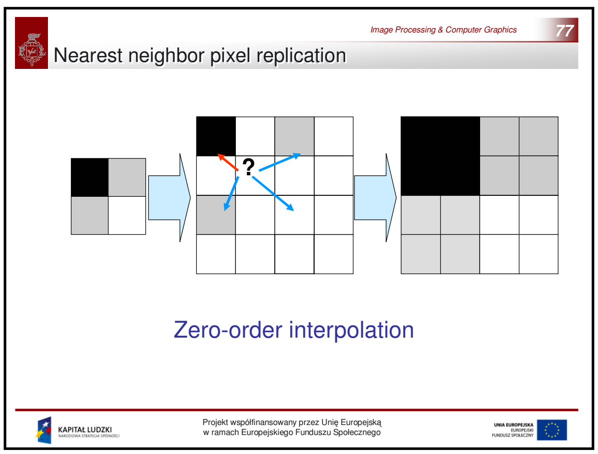
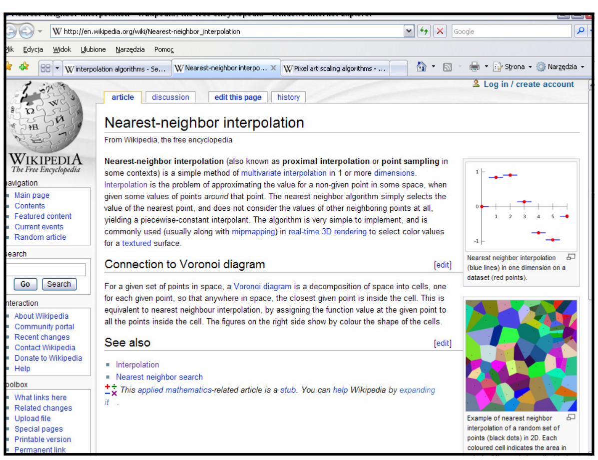
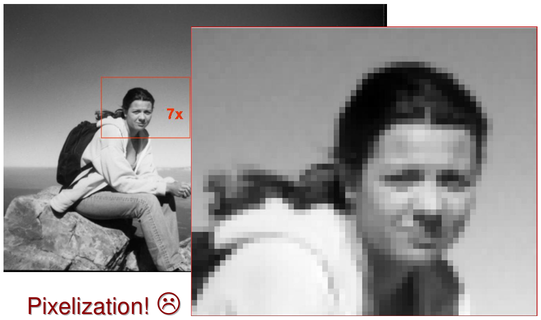


Image Processing & Computer Graphics
77
Nearest neighbor pixel replication
?
Zero-order interpolation
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

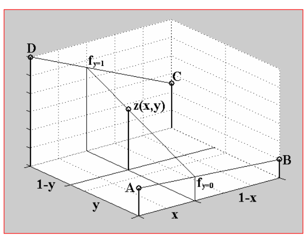

http://en.wikipedia.org/wiki/Nearest-neighbor_interpolation
Nearest-neighbor interpolation
From Wikipedia, the free encyclopedia
Nearest-neighbor interpolation (also known as proximal interpolation or point sampling in some contexts) is a simple method of multivariate interpolation in 1 or more dimensions.
Interpolation is the problem of approximating the value for a non-given point in some space, when given some values of points around that point. The nearest neighbor algorithm simply selects the value of the nearest point, and does not consider the values of other neighboring points at all, yielding a piecewise-constant interpolant. The algorithm is very simple to implement, and is commonly used (usually along with mipmapping) in real-time 3D rendering to select color values for a textured surface.
Connection to Voronoi diagram
For a given set of points in space, a Voronoi diagram is a decomposition of space into cells, one for each given point, so that anywhere in space, the closest given point is inside the cell. This is equivalent to nearest neighbour interpolation, by assigning the function value at the given point to all the points inside the cell. The figures on the right side show by colour the shape of the cells.
See also
Interpolation
Nearest neighbor search
Nearest neighbor interpolation (blue lines) in one dimension on a dataset (red points).
Example of nearest neighbor interpolation of a random set of points (black dots) in 2D. Each coloured cell indicates the area in

Nearest neighbor pixel replication

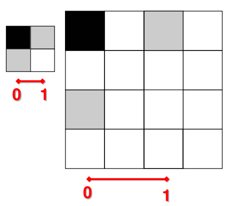


Pixelization! ☹

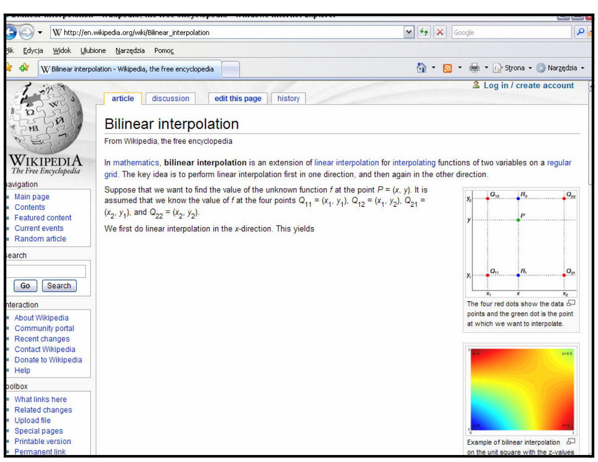
Bilinear interpolation



x ∈ <0,1>
y ∈ <0,1>



$$z(x,y) = A(1-x)(1-y) + B(x)(1-y) + C(x)(y) + D(1-x)(y)$$



http://en.wikipedia.org/wiki/Bilinear_interpolation

Bilinear interpolation

From Wikipedia, the free encyclopedia

In mathematics, **bilinear interpolation** is an extension of linear interpolation for interpolating functions of two variables on a regular grid. The key idea is to perform linear interpolation first in one direction, and then again in the other direction.

Suppose that we want to find the value of the unknown function *f* at the point $P = (x, y)$. It is assumed that we know the value of *f* at the four points $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$, and $Q_{22} = (x_2, y_2)$.

We first do linear interpolation in the *x*-direction. This yields

The four red dots show the data points and the green dot is the point at which we want to interpolate.

Example of bilinear interpolation on the unit square with the z-values


Image Processing & Computer Graphics
82
Bilinear interpolation
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

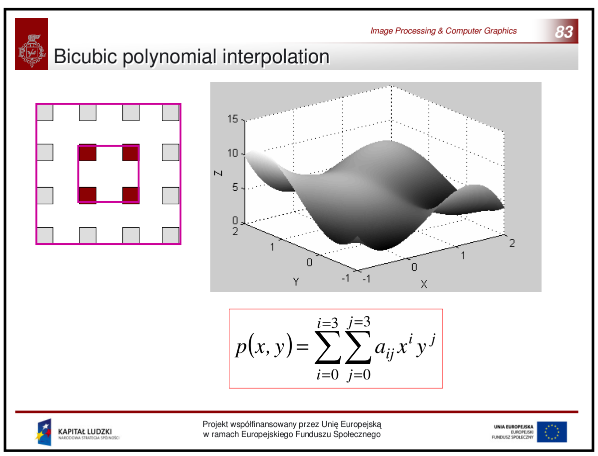
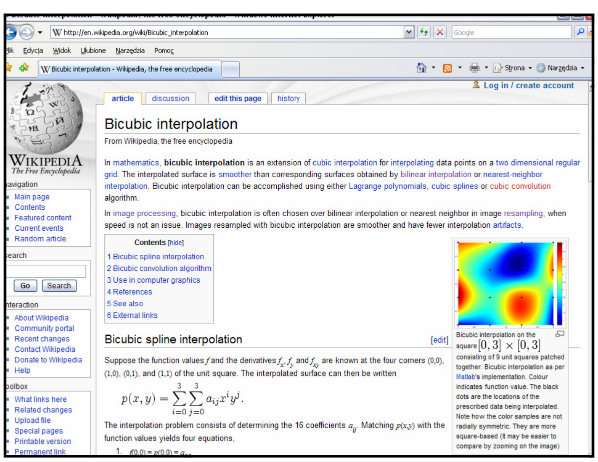
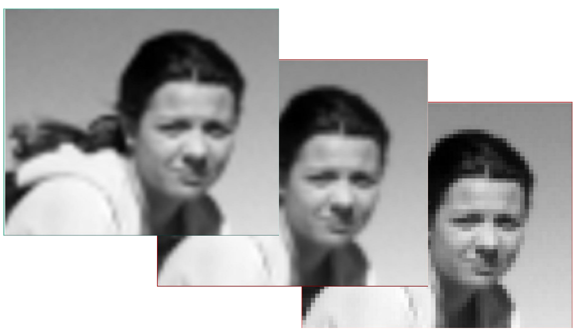


Image Processing & Computer Graphics
83
Bicubic polynomial interpolation
15
10
5
0
Z
Y
X
$p(x,y)=\sum_{i=0}^{i=3}\sum_{j=0}^{j=3}a_{ij}x^iy^j$
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

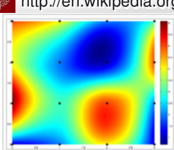

http://en.wikipedia.org/wiki/Bicubic_interpolation
Edycja Widok Ulubione Narzędzia Pomoc
Bicubic interpolation - Wikipedia, the free encyclopedia
Strona Narzędzia
Log in / create account
article discussion edit this page history
Bicubic interpolation
From Wikipedia, the free encyclopedia
In mathematics, bicubic interpolation is an extension of cubic interpolation for interpolating data points on a two dimensional regular grid. The interpolated surface is smoother than corresponding surfaces obtained by bilinear interpolation or nearest-neighbor interpolation. Bicubic interpolation can be accomplished using either Lagrange polynomials, cubic splines or cubic convolution algorithm.
In image processing, bicubic interpolation is often chosen over bilinear interpolation or nearest neighbor in image resampling, when speed is not an issue. Images resampled with bicubic interpolation are smoother and have fewer interpolation artifacts.
Contents [hide]
1 Bicubic spline interpolation
2 Bicubic convolution algorithm
3 Use in computer graphics
4 References
5 See also
6 External links
Bicubic spline interpolation [edit]
Suppose the function values f and the derivatives f_x, f_y and f_{xy} are known at the four corners (0,0), (1,0), (0,1), and (1,1) of the unit square. The interpolated surface can then be written
$p(x,y)=\sum_{i=0}^{3}\sum_{j=0}^{3}a_{ij}x^iy^j.$
The interpolation problem consists of determining the 16 coefficients a_{ij}. Matching $p(x,y)$ with the function values yields four equations,
Bicubic interpolation on the square $[0,3]\times[0,3]$ consisting of 9 unit squares patched together. Bicubic interpolation as per Matlab's implementation. Colour indicates function value. The black dots are the locations of the prescribed data being interpolated. Note how the color samples are not radially symmetric. They are more square-based (It may be easier to compare by zooming on the image)
WIKIPEDIA
The Free Encyclopedia
Main page
Contents
Featured content
Current events
Random article
Go Search
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Donate to Wikipedia
Help
What links here
Related changes
Upload file
Special pages
Printable version
Permanent link

Bicubic polynomial interpolation

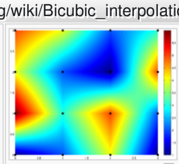
bicubic, bilinear, nearest neighbor (comparison)



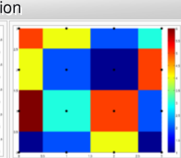
http://en.wikipedia.org/wiki/Bicubic_interpolation



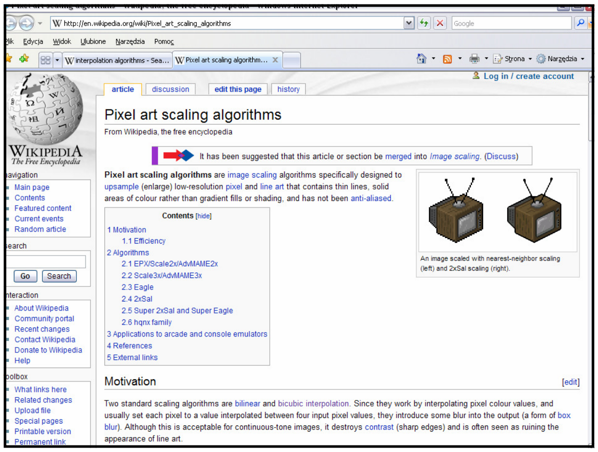
Bicubic interpolation on the square $[0,3]\times[0,3]$ consisting of 9 unit squares patched together. Bicubic interpolation as per Matlab's implementation. Colour indicates function value. The black dots are the locations of the prescribed data being interpolated. Note how the color samples are not radially symmetric. They are more square-based (It may be easier to compare by zooming on the image)



Bilinear interpolation on the same dataset as above. Derivatives of the surface are not continuous over the square boundaries.



Nearest-neighbor interpolation on the same dataset as above. Note that the information content in all these three examples is equivalent.



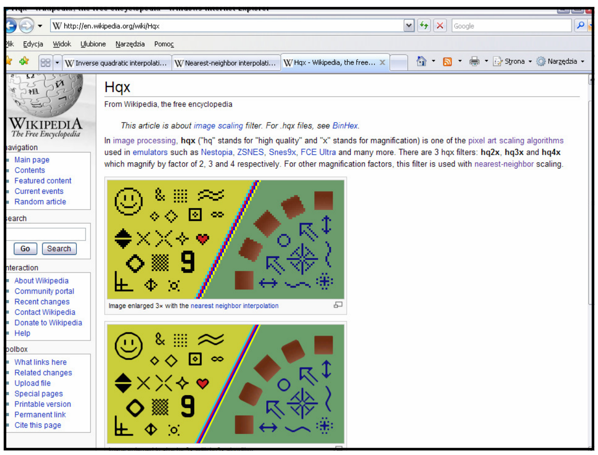


Image Processing & Computer Graphics 90

Summary, discussion and quiz

Summary, discussion and quiz

KAPITAŁ LUDZKI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
„ Image Processing and Computer Graphics”
Prezentacja multimedialna współfinansowana przez
Unię Europejską w ramach
Europejskiego Funduszu Społecznego w projekcie pt.
„Innowacyjna dydaktyka bez ograniczeń - zintegrowany
rozwój Politechniki Łódzkiej - zarządzanie Uczelnią,
nowoczesna oferta edukacyjna i wzmacniania zdolności
do zatrudniania osób niepełnosprawnych”
Politechnika Łódzka
Politechnika Łódzka, ul. Żeromskiego 116, 90-924 Łódź, tel. (042) 631 28 83
www.kapitalludzki.p.lodz.pl