„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”

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

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

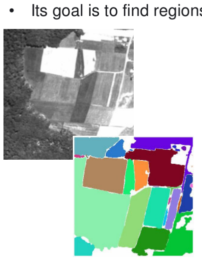
Fragments of the lecture were also provided by

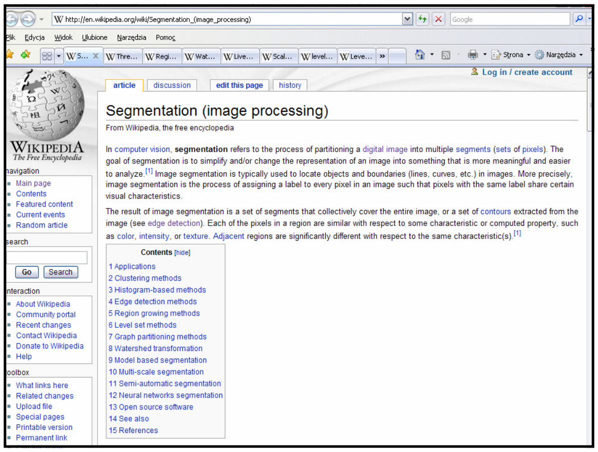
Andrzej Materka, Michał Strzelecki, Paweł Strumiłło, Piotr Romaniuk, Piotr Makowski

and were used with permission.

Segmentation goal

- Segmentation partitions image into areas.
- Segmentation groups pixels in separate sets.
- Its goal is to find regions (areas) of interest for further analysis.





W http://en.wikipedia.org/wiki/Segmentation_(image_processing)

Plik Edycja Widok Ulubione Narzędzia Pomoc

Log in / create account

article | discussion | edit this page | history

Segmentation (image processing)

From Wikipedia, the free encyclopedia

In computer vision, **segmentation** refers to the process of partitioning a digital image into multiple segments (sets of pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.[1] Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s).[1]

Contents [hide]

Segmentation

- Segmentation methods:
  - Histogram based
  - Edge detection
  - Region based
- Selected methods:
  - Grey-level thresholding
  - Top-hat transform
  - Watershed segmentation
  - Seed-growing (flood-filling) algorithm
  - Deformable models (level-set, active contour/surface)
- Segmentation aimed preprocessing
  - Template matching detection
  - Texture segmentation

Segmentation

- Segmentation methods:
  - Histogram based

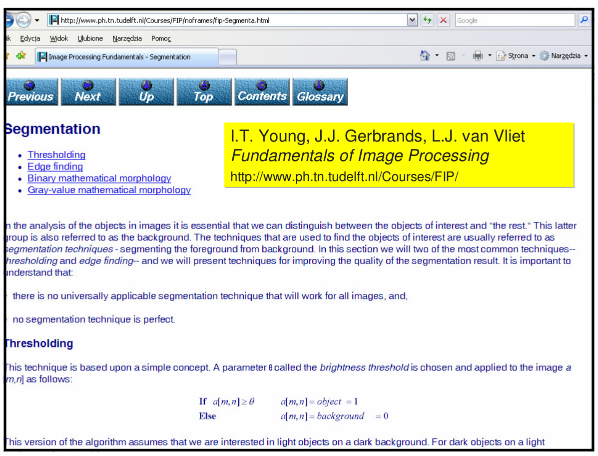
    Object pixels differ from background pixels with their gray-level
  - Region based

    Object area have specific characteristics we can compute…
    …and then differentiate such areas from everything else
  - Edge detection

    Object edges have specific characteristics we can compute…
    …and then differentiate edges from everything else

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


http://en.wikipedia.org/wiki/Adaptive_thresholding
article
discussion
edit this page
history
Log in / create account
Thresholding (image processing)
From Wikipedia, the free encyclopedia
(Redirected from Adaptive thresholding)
Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images (Shapiro, et al 2001:83).
Contents [hide]
1 Method
1.1 Adaptive Thresholding
2 Categorizing Thresholding Methods
3 Multiband Thresholding
4 See also
5 Citations
6 References and Further Reading
Example of a threshold effect used on an image.
Method [edit]
During the thresholding process, individual pixels in an image are marked as "object" pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as "background" pixels otherwise. This convention is known as threshold above. Variants include threshold below, which is opposite of threshold above; threshold inside, where a pixel is labeled "object" if its value is between two thresholds; and threshold outside, which is the opposite of threshold inside (Shapiro, et al 2001:83). Typically, an object pixel is given a value of "1" while a background pixel is given a value of "0." Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's label.
Original image.
Adaptive Thresholding [edit]


http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip-Segmenta.html
Image Processing Fundamentals - Segmentation
Previous
Next
Up
Top
Contents
Glossary
Segmentation
Thresholding
Edge finding
Binary mathematical morphology
Gray-value mathematical morphology
I.T. Young, J.J. Gerbrands, L.J. van Vliet
Fundamentals of Image Processing
http://www.ph.tn.tudelft.nl/Courses/FIP/
In the analysis of the objects in images it is essential that we can distinguish between the objects of interest and "the rest." This latter group is also referred to as the background. The techniques that are used to find the objects of interest are usually referred to as segmentation techniques - segmenting the foreground from background. In this section we will two of the most common techniques--thresholding and edge finding-- and we will present techniques for improving the quality of the segmentation result. It is important to understand that:
there is no universally applicable segmentation technique that will work for all images, and,
no segmentation technique is perfect.
Thresholding
This technique is based upon a simple concept. A parameter θ called the brightness threshold is chosen and applied to the image a[m,n] as follows:
If a[m,n] ≥ θ   a[m,n] = object = 1
Else   a[m,n] = background = 0
This version of the algorithm assumes that we are interested in light objects on a dark background. For dark objects on a light

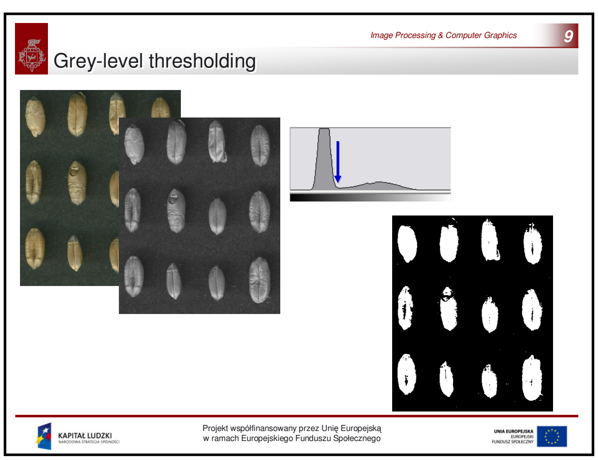
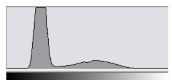


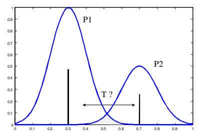
Image Processing & Computer Graphics
9
Grey-level thresholding
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

Optimum thresholding

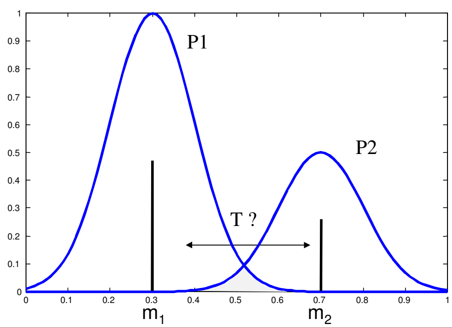
Suppose an image contains two intensity values m1, m2 combined with additive Gaussian noise N(0,s) that appear in an image with apriori probabilities P1 and P2 correspondingly P1 + P2=1.

The task is to define a threshold level T that would minimise the overall segmentation error





Optimum thresholding

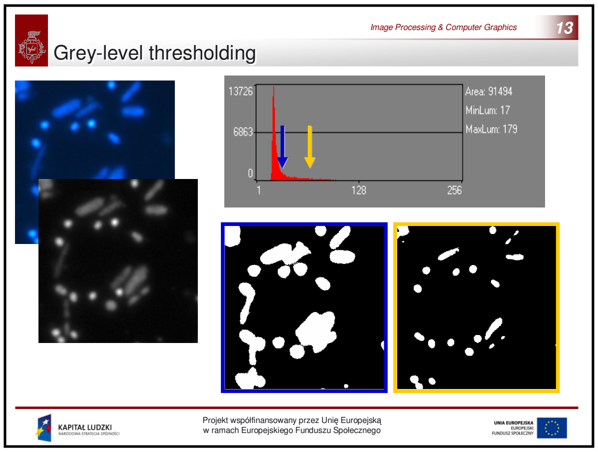


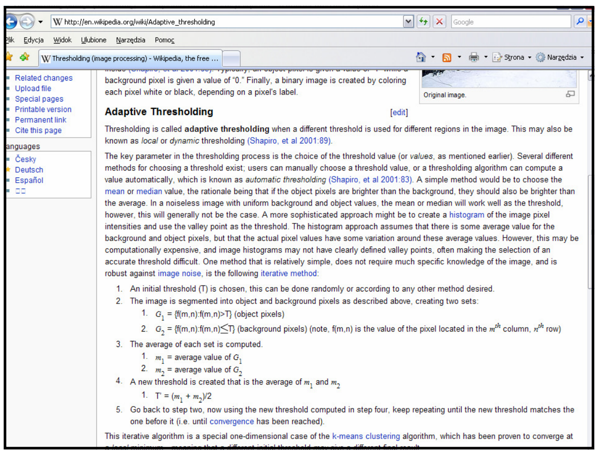
Optimum thresholding

For σ_1, σ_2 solution to this optimisation problem is the following:

$$\boxed{T = \frac{m_1 + m_2}{2} + \frac{\sigma^2}{m_1 - m_2} \ln\left(\frac{P_2}{P_1}\right)}$$

If $P_1 = P_2$, the optimal threshold is the average of the image intensities.





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

Thresholding (image processing) - Wikipedia, the free ...

background pixel is given a value of "0." Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's label.

Original image.

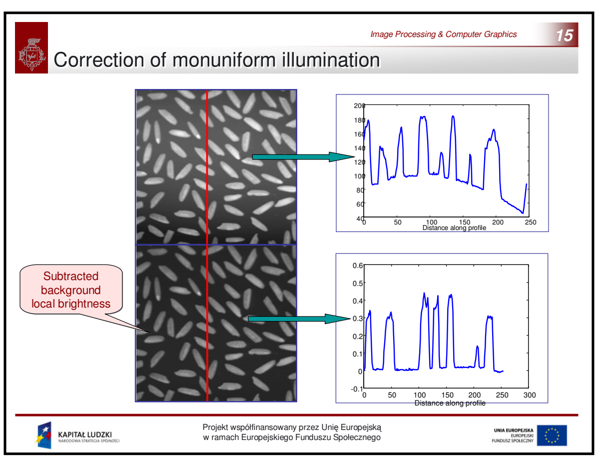
Adaptive Thresholding [edit]

Thresholding is called **adaptive thresholding** when a different threshold is used for different regions in the image. This may also be known as *local* or *dynamic* thresholding (Shapiro, et al 2001:89).

The key parameter in the thresholding process is the choice of the threshold value (or *values*, as mentioned earlier). Several different methods for choosing a threshold exist; users can manually choose a threshold value, or a thresholding algorithm can compute a value automatically, which is known as *automatic thresholding* (Shapiro, et al 2001:83). A simple method would be to choose the mean or median value, the rationale being that if the object pixels are brighter than the background, they should also be brighter than the average. In a noiseless image with uniform background and object values, the mean or median will work well as the threshold, however, this will generally not be the case. A more sophisticated approach might be to create a histogram of the image pixel intensities and use the valley point as the threshold. The histogram approach assumes that there is some average value for the background and object pixels, but that the actual pixel values have some variation around these average values. However, this may be computationally expensive, and image histograms may not have clearly defined valley points, often making the selection of an accurate threshold difficult. One method that is relatively simple, does not require much specific knowledge of the image, and is robust against image noise, is the following iterative method:

1. An initial threshold (T) is chosen, this can be done randomly or according to any other method desired.
2. The image is segmented into object and background pixels as described above, creating two sets:
   1. G_1 = {f(m,n):f(m,n)>T} (object pixels)
   2. G_2 = {f(m,n):f(m,n)≤T} (background pixels) (note, f(m,n) is the value of the pixel located in the m^{th} column, n^{th} row)
3. The average of each set is computed.
   1. m_1 = average value of G_1
   2. m_2 = average value of G_2
4. A new threshold is created that is the average of m_1 and m_2
   1. T' = $(m_1 + m_2)/2$
5. Go back to step two, now using the new threshold computed in step four, keep repeating until the new threshold matches the one before it (i.e. until convergence has been reached).

This iterative algorithm is a special one-dimensional case of the k-means clustering algorithm, which has been proven to converge at



http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip-Segmenta.html

Image Processing Fundamentals - Segmentation

Gray-value mathematical morphology

As we have seen in Section 10.1.2, gray-value morphological processing techniques can be used for practical problems such as shading correction. In this section several other techniques will be presented.

* *Top-hat transform* - The isolation of gray-value objects that are convex can be accomplished with the *top-hat transform* as developed by Meyer . Depending upon whether we are dealing with light objects on a dark background or dark objects on a light background, the transform is defined as:

Light objects - $TopHat(A,B) = A - (A \circ B) = A - \max_B\left(\min_B(A)\right)$

Dark objects - $TopHat(A,B) = (A \bullet B) - A = \min_B\left(\max_B(A)\right) - A$

where the structuring element ***B*** is chosen to be bigger than the objects in question and, if possible, to have a convex shape. Because of the properties given in eqs. and , *Topat*(***A***,***B***) >= 0. An example of this technique is shown in Figure 63.

The original image including shading is processed by a 15 x 1 structuring element as described in eqs. and to produce the desired result. Note that the transform for dark objects has been defined in such a way as to yield "positive" objects as opposed to "negative" objects. Other definitions are, of course, possible.

* *Thresholding* - A simple estimate of a locally-varying threshold surface can be derived from morphological processing as follows:

Threshold surface - $\theta[m,n] = \frac{1}{2}\left(\max(A) + \min(A)\right)$

Once again, we suppress the notation for the structuring element ***B*** under the *max* and *min* operations to keep the notation simple. Its use, however, is understood.

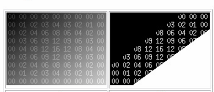
http://www.dca.fee.unicamp.br/~lotufo/Courses/ia-865-1996/mm_segm/html/c1/top-hat-gray/front-page.html

Top Hat: gray-scale case

Segmenting white numbers from a non-uniform intensity background

The goal in this experiment is to segment the foreground numbers which are on a non-uniform background. This image is a synthetic one and was made to illustrate the power of the gray-scale top hat.

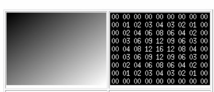
Below we show the original image and the result of a simple thresholding technique. As the overall illumination of the image is not uniform, we get the numbers properly segment in just one portion of the image.



a) *b)*

a)original; b)simple thresholding

The opening of the original image recovers the uneven background and the resultant top hat just shows the segmented numbers.



a) *b)*

a)opening; b)top hat

How many objects?

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

Region based segmentation

The main idea in region-based segmentation techniques is to identify different regions in an image that have similar features (gray level, color, texture, etc.). There are two main region-based image segmentation techniques:

- region growing (merging)
- region splitting

Region growing

In region growing the image is divided into ***atomic regions*** (e.g., pixels, templates). These "**seed**" points grow by appending to each point other points that have similar properties. The key problem lies in selecting proper criterion (predicate) for merging. A frequently used merging rule is:

Merge two regions if they are „similar"
(in terms of a predefined features)

Region growing – general formulation

Let R represent the entire image. Segmentation may be viewed as a process that partitions R into N disjoint regions, $R_1, R_1, \ldots, R_N$, such that:

a) $\sum_{i=1}^{N} R_i = R$

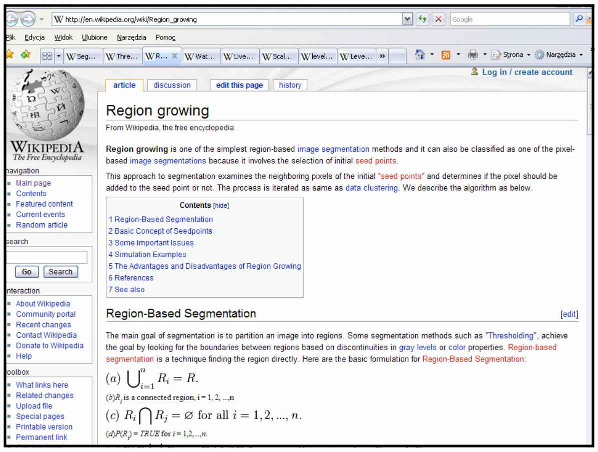
b) R_i is a connected region, $i = 1, 2, \ldots N,$

c) $R_i \cap R_j = \varnothing$ for all i and $j, i \neq j,$

d) $P(R_i) = TRUE$ for $i = 1, 2, \ldots N,$

e) $P(R_i \cup R_j) = FALSE$ for $i \neq j$

where $P(R_i)$ is a logical predicate over the points in set R_i and $\varnothing$ is the empty set.



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

Region growing

From Wikipedia, the free encyclopedia

Region growing is one of the simplest region-based image segmentation methods and it can also be classified as one of the pixel-based image segmentations because it involves the selection of initial seed points.

This approach to segmentation examines the neighboring pixels of the initial "seed points" and determines if the pixel should be added to the seed point or not. The process is iterated as same as data clustering. We describe the algorithm as below.

Contents [hide]

Region-Based Segmentation [edit]

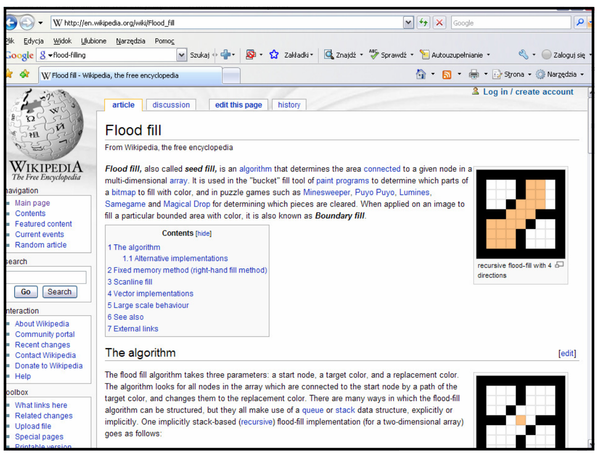
The main goal of segmentation is to partition an image into regions. Some segmentation methods such as "Thresholding", achieve the goal by looking for the boundaries between regions based on discontinuities in gray levels or color properties. Region-based segmentation is a technique finding the region directly. Here are the basic formulation for Region-Based Segmentation:

(a) $\bigcup_{i=1}^{n} R_i = R.$

(b) R_i is a connected region, $i = 1, 2, ..., n$

(c) $R_i \bigcap R_j = \varnothing$ for all $i = 1, 2, ..., n.$

(d) $P(R_i) = TRUE$ for $i = 1, 2, ..., n.$



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

Flood fill

From Wikipedia, the free encyclopedia

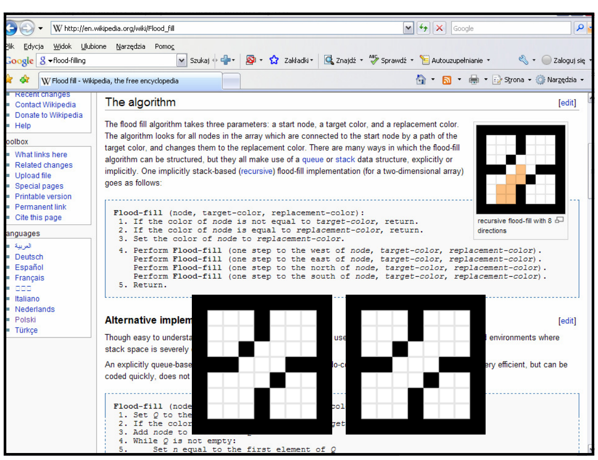
Flood fill, also called ***seed fill,*** is an algorithm that determines the area connected to a given node in a multi-dimensional array. It is used in the "bucket" fill tool of paint programs to determine which parts of a bitmap to fill with color, and in puzzle games such as Minesweeper, Puyo Puyo, Lumines, Samegame and Magical Drop for determining which pieces are cleared. When applied on an image to fill a particular bounded area with color, it is also known as ***Boundary fill***.

recursive flood-fill with 4 directions

Contents [hide]

The algorithm [edit]

The flood fill algorithm takes three parameters: a start node, a target color, and a replacement color. The algorithm looks for all nodes in the array which are connected to the start node by a path of the target color, and changes them to the replacement color. There are many ways in which the flood-fill algorithm can be structured, but they all make use of a queue or stack data structure, explicitly or implicitly. One implicitly stack-based (recursive) flood-fill implementation (for a two-dimensional array) goes as follows:



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

The algorithm [edit]

The flood fill algorithm takes three parameters: a start node, a target color, and a replacement color. The algorithm looks for all nodes in the array which are connected to the start node by a path of the target color, and changes them to the replacement color. There are many ways in which the flood-fill algorithm can be structured, but they all make use of a queue or stack data structure, explicitly or implicitly. One implicitly stack-based (recursive) flood-fill implementation (for a two-dimensional array) goes as follows:

recursive flood-fill with 8 directions

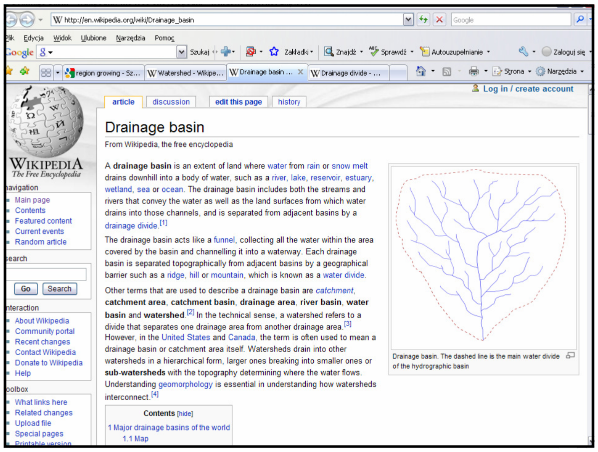
```
Flood-fill (node, target-color, replacement-color):
 1. If the color of node is not equal to target-color, return.
 2. If the color of node is equal to replacement-color, return.
 3. Set the color of node to replacement-color.
 4. Perform Flood-fill (one step to the west of node, target-color, replacement-color).
    Perform Flood-fill (one step to the east of node, target-color, replacement-color).
    Perform Flood-fill (one step to the north of node, target-color, replacement-color).
    Perform Flood-fill (one step to the south of node, target-color, replacement-color).
 5. Return.
```

Alternative implem [edit]

Though easy to understa … use … environments where stack space is severely …

An explicitly queue-base … o-c … ery efficient, but can be coded quickly, does not …

```
Flood-fill (node … col
 1. Set Q to the
 2. If the color … get
 3. Add node to
 4. While Q is not empty:
 5.     Set n equal to the first element of Q
```


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

Drainage basin

From Wikipedia, the free encyclopedia

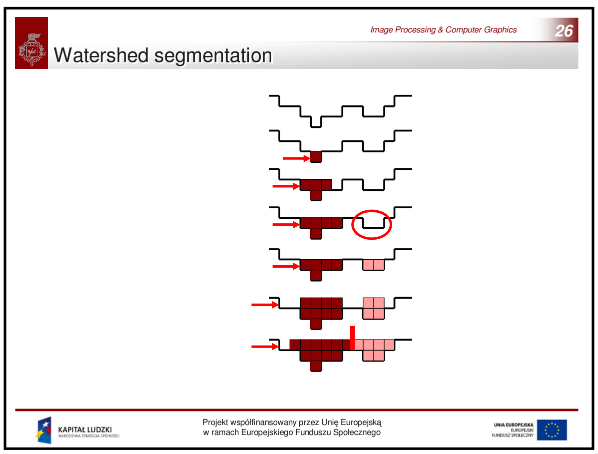
A **drainage basin** is an extent of land where water from rain or snow melt drains downhill into a body of water, such as a river, lake, reservoir, estuary, wetland, sea or ocean. The drainage basin includes both the streams and rivers that convey the water as well as the land surfaces from which water drains into those channels, and is separated from adjacent basins by a drainage divide.[1]

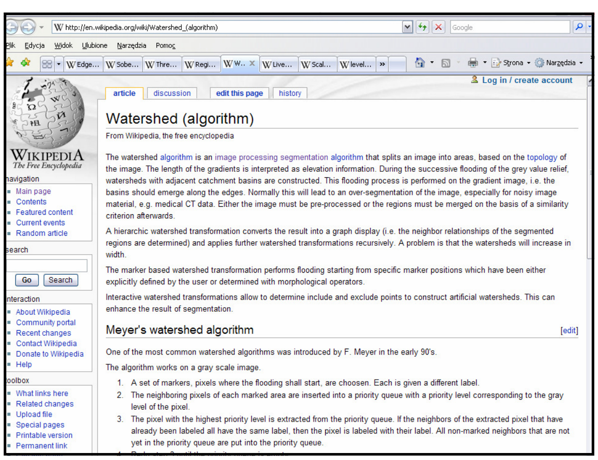
The drainage basin acts like a funnel, collecting all the water within the area covered by the basin and channelling it into a waterway. Each drainage basin is separated topographically from adjacent basins by a geographical barrier such as a ridge, hill or mountain, which is known as a water divide.

Other terms that are used to describe a drainage basin are *catchment*, **catchment area**, **catchment basin**, **drainage area**, **river basin**, **water basin** and **watershed**.[2] In the technical sense, a watershed refers to a divide that separates one drainage area from another drainage area.[3] However, in the United States and Canada, the term is often used to mean a drainage basin or catchment area itself. Watersheds drain into other watersheds in a hierarchical form, larger ones breaking into smaller ones or **sub-watersheds** with the topography determining where the water flows. Understanding geomorphology is essential in understanding how watersheds interconnect.[4]

Drainage basin. The dashed line is the main water divide of the hydrographic basin

Contents [hide]




http://en.wikipedia.org/wiki/Watershed_(algorithm)

Watershed (algorithm)

From Wikipedia, the free encyclopedia

The watershed algorithm is an image processing segmentation algorithm that splits an image into areas, based on the topology of the image. The length of the gradients is interpreted as elevation information. During the successive flooding of the grey value relief, watersheds with adjacent catchment basins are constructed. This flooding process is performed on the gradient image, i.e. the basins should emerge along the edges. Normally this will lead to an over-segmentation of the image, especially for noisy image material, e.g. medical CT data. Either the image must be pre-processed or the regions must be merged on the basis of a similarity criterion afterwards.

A hierarchic watershed transformation converts the result into a graph display (i.e. the neighbor relationships of the segmented regions are determined) and applies further watershed transformations recursively. A problem is that the watersheds will increase in width.

The marker based watershed transformation performs flooding starting from specific marker positions which have been either explicitly defined by the user or determined with morphological operators.

Interactive watershed transformations allow to determine include and exclude points to construct artificial watersheds. This can enhance the result of segmentation.

Meyer's watershed algorithm [edit]

One of the most common watershed algorithms was introduced by F. Meyer in the early 90's.

The algorithm works on a gray scale image.

1. A set of markers, pixels where the flooding shall start, are choosen. Each is given a different label.
2. The neighboring pixels of each marked area are inserted into a priority queue with a priority level corresponding to the gray level of the pixel.
3. The pixel with the highest priority level is extracted from the priority queue. If the neighbors of the extracted pixel that have already been labeled all have the same label, then the pixel is labeled with their label. All non-marked neighbors that are not yet in the priority queue are put into the priority queue.

Region splitting

Quad-tree decomposition

Initially, the image is subdivided into a set of arbitrary disjoint regions, then splits of the regions take place in attempt to satisfy region uniformity criteria (a-e).

An example split algorithm works as follows:

Subdivide the entire image successively into smaller and smaller quadrant regions until P(Ri) = TRUE for any region; that is if P is FALSE for any quadrant, subdivide the quadrant into sub-quadrants, and so on.

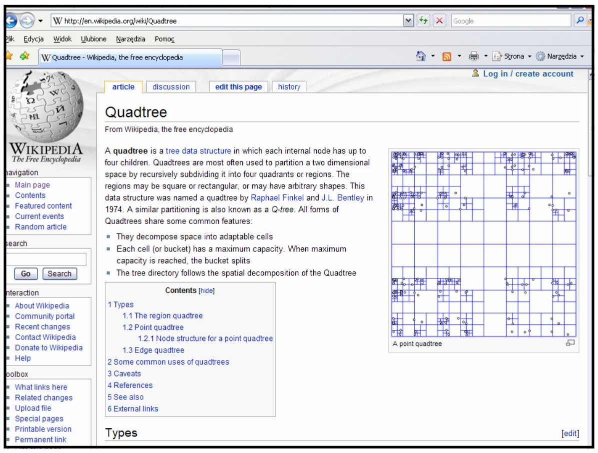
However, if only splitting is used the final result likely would contain adjacent regions with identical properties.

This can be improved by allowing merging as well as splitting. The following procedure can be implemented:

Split any region Ri if P(Ri) = FALSE,

Merge any adjacent regions for which P(Ri ∪ Rj) = TRUE,

Stop when no further merging or splitting is possible.



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

Quadtree - Wikipedia, the free encyclopedia

article | discussion | edit this page | history

Log in / create account

Quadtree

From Wikipedia, the free encyclopedia

A **quadtree** is a tree data structure in which each internal node has up to four children. Quadtrees are most often used to partition a two dimensional space by recursively subdividing it into four quadrants or regions. The regions may be square or rectangular, or may have arbitrary shapes. This data structure was named a quadtree by Raphael Finkel and J.L. Bentley in 1974. A similar partitioning is also known as a *Q-tree*. All forms of Quadtrees share some common features:

- They decompose space into adaptable cells
- Each cell (or bucket) has a maximum capacity. When maximum capacity is reached, the bucket splits
- The tree directory follows the spatial decomposition of the Quadtree

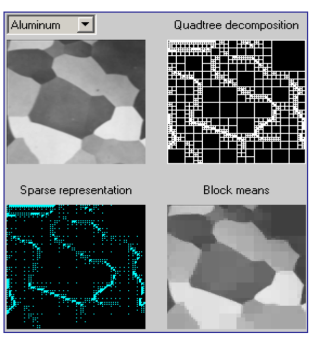
A point quadtree

Contents [hide]

1 Types
1.1 The region quadtree
1.2 Point quadtree
1.2.1 Node structure for a point quadtree
1.3 Edge quadtree
2 Some common uses of quadtrees
3 Caveats
4 References
5 See also
6 External links

Types [edit]

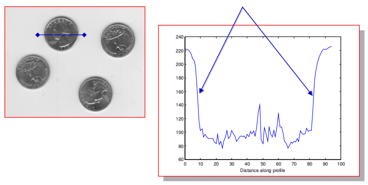
Quadtree decomposition



Edge detection methods

Boundary based segmentation (edge detection)

Changes (or discontinuous) in an image amplitude are important primitive characteristics of an image that carry information about object borders. Detection methods of image discontinuities are principal approaches to image segmentation and identification of objects in a scene. Local discontinuities in image intensity fall into three categories: points, lines, or edges



Point detection

Point detection mask:

-1	-1	-1
-1	8	-1
-1	-1	-1

The point is rendered if:

$$\boxed{|D| > T}$$

where D is a similarity measure between the image and the template, and T is a non-negative threshold.

KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA

Line detection

-1	-1	-1
2	2	2
-1	-1	-1

Horizontal

-1	2	-1
-1	2	-1
-1	2	-1

Vertical

-1	-1	2
-1	2	-1
2	-1	-1

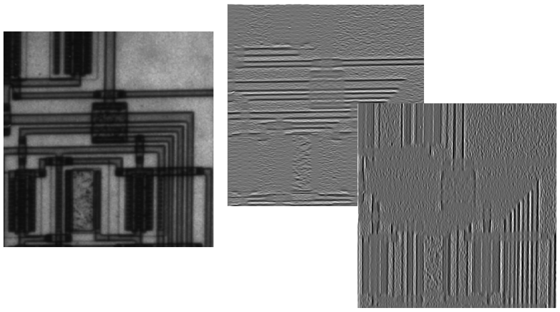
+45°

2	-1	-1
-1	2	-1
-1	-1	2

-45°

KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA

Line detection



KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA

Edge detection

An edge

is the boundary between two regions with distinct gray-level properties.

Edges characterize the physical extent of objects thus their accurate detection plays a key role in image analysis and pattern recognition problems.

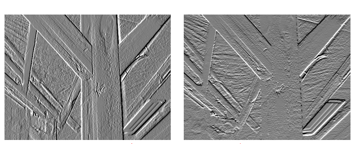
The main idea underlying most edge-detection techniques is the computation of a local derivative of an image.

KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA

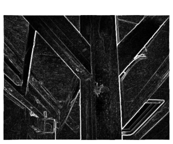
Edge detectors

Operators:
- Sobel
- Prewitt
- Roberts

Pairs of detectors
(gradient is a vector)

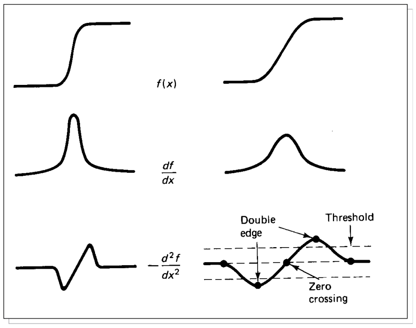


$$I_{edges}(i,j) = \sqrt{I_H^{\,2}(i,j) + I_V^{\,2}(i,j)}$$



KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA

Gradient operators (1D example)

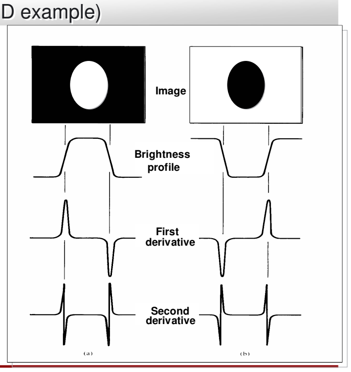


Gradient operators (1D example)

the magnitude of the first derivative can be used to detect the presence of an edge in an image,

the sign of the second derivative can be used to determine whether an edge pixel is on the dark or light side of an edge,

the second derivative has zero crossing at the midpoint of a grey-level transition.



Gradient (2D)

Gradient (vector field)

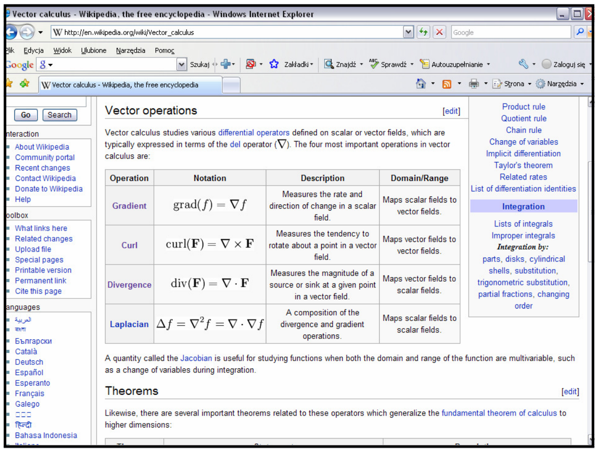
$$\boxed{\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}}$$

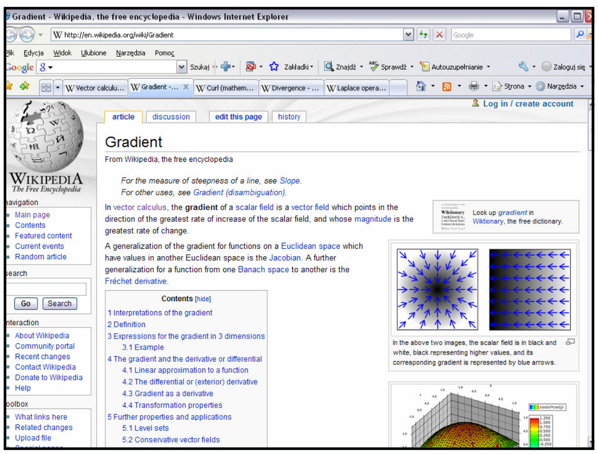
Second order derivative „equivalents”:

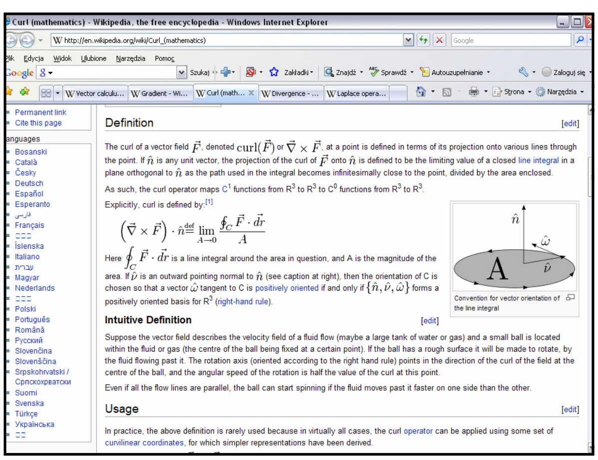
- **Gradient divergence**
- **Gradient curl (rotation)**
- **Laplacian vector field**

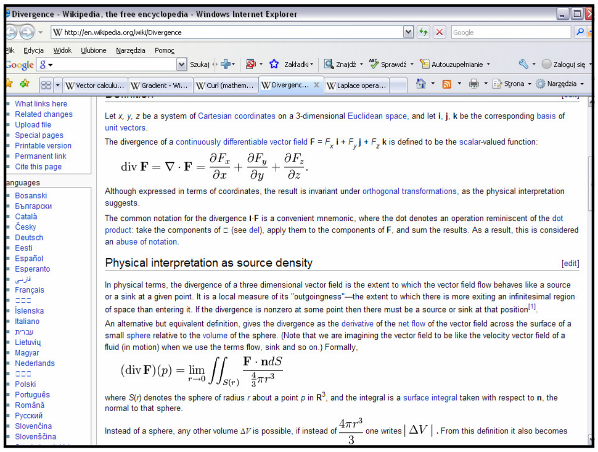
Gradient’s magnitude

$$\boxed{\nabla f = mag(\nabla f) = \left[G_x^2 + G_y^2\right]^{1/2} \approx |G_x| + |G_y|}$$


Vector calculus - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/Vector_calculus
Vector operations
[edit]
Vector calculus studies various differential operators defined on scalar or vector fields, which are typically expressed in terms of the del operator (∇). The four most important operations in vector calculus are:
Operation | Notation | Description | Domain/Range
Gradient | grad(f) = ∇f | Measures the rate and direction of change in a scalar field. | Maps scalar fields to vector fields.
Curl | curl(F) = ∇ × F | Measures the tendency to rotate about a point in a vector field. | Maps vector fields to vector fields.
Divergence | div(F) = ∇ · F | Measures the magnitude of a source or sink at a given point in a vector field. | Maps vector fields to scalar fields.
Laplacian | Δf = ∇²f = ∇ · ∇f | A composition of the divergence and gradient operations. | Maps scalar fields to scalar fields.
A quantity called the Jacobian is useful for studying functions when both the domain and range of the function are multivariable, such as a change of variables during integration.
Theorems
[edit]
Likewise, there are several important theorems related to these operators which generalize the fundamental theorem of calculus to higher dimensions:
Product rule
Quotient rule
Chain rule
Change of variables
Implicit differentiation
Taylor's theorem
Related rates
List of differentiation identities
Integration
Lists of integrals
Improper integrals
Integration by:
parts, disks, cylindrical shells, substitution, trigonometric substitution, partial fractions, changing order


Gradient - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/Gradient
Log in / create account
article
discussion
edit this page
history
Gradient
From Wikipedia, the free encyclopedia
For the measure of steepness of a line, see Slope.
For other uses, see Gradient (disambiguation).
In vector calculus, the gradient of a scalar field is a vector field which points in the direction of the greatest rate of increase of the scalar field, and whose magnitude is the greatest rate of change.
Look up gradient in Wiktionary, the free dictionary.
A generalization of the gradient for functions on a Euclidean space which have values in another Euclidean space is the Jacobian. A further generalization for a function from one Banach space to another is the Fréchet derivative.
Contents [hide]
1 Interpretations of the gradient
2 Definition
3 Expressions for the gradient in 3 dimensions
3.1 Example
4 The gradient and the derivative or differential
4.1 Linear approximation to a function
4.2 The differential or (exterior) derivative
4.3 Gradient as a derivative
4.4 Transformation properties
5 Further properties and applications
5.1 Level sets
5.2 Conservative vector fields
In the above two images, the scalar field is in black and white, black representing higher values, and its corresponding gradient is represented by blue arrows.


Curl (mathematics) - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/Curl_(mathematics)
Definition
[edit]
The curl of a vector field F, denoted curl(F) or ∇ × F, at a point is defined in terms of its projection onto various lines through the point. If n̂ is any unit vector, the projection of the curl of F onto n̂ is defined to be the limiting value of a closed line integral in a plane orthogonal to n̂ as the path used in the integral becomes infinitesimally close to the point, divided by the area enclosed.
As such, the curl operator maps C¹ functions from R³ to R³ to C⁰ functions from R³ to R³.
Explicitly, curl is defined by:[1]
(∇ × F) · n̂ ≝ lim_{A→0} ∮_C F · dr / A
Here ∮_C F · dr is a line integral around the area in question, and A is the magnitude of the area. If ν̂ is an outward pointing normal to n̂ (see caption at right), then the orientation of C is chosen so that a vector ω̂ tangent to C is positively oriented if and only if {n̂, ν̂, ω̂} forms a positively oriented basis for R³ (right-hand rule).
Convention for vector orientation of the line integral
Intuitive Definition
[edit]
Suppose the vector field describes the velocity field of a fluid flow (maybe a large tank of water or gas) and a small ball is located within the fluid or gas (the centre of the ball being fixed at a certain point). If the ball has a rough surface it will be made to rotate, by the fluid flowing past it. The rotation axis (oriented according to the right hand rule) points in the direction of the curl of the field at the centre of the ball, and the angular speed of the rotation is half the value of the curl at this point.
Even if all the flow lines are parallel, the ball can start spinning if the fluid moves past it faster on one side than the other.
Usage
[edit]
In practice, the above definition is rarely used because in virtually all cases, the curl operator can be applied using some set of curvilinear coordinates, for which simpler representations have been derived.



Divergence - Wikipedia, the free encyclopedia - Windows Internet Explorer

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

Let x, y, z be a system of Cartesian coordinates on a 3-dimensional Euclidean space, and let **i**, **j**, **k** be the corresponding basis of unit vectors.

The divergence of a continuously differentiable vector field $\mathbf{F} = F_x\,\mathbf{i} + F_y\,\mathbf{j} + F_z\,\mathbf{k}$ is defined to be the scalar-valued function:

$$\operatorname{div}\mathbf{F} = \nabla\cdot\mathbf{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}.$$

Although expressed in terms of coordinates, the result is invariant under orthogonal transformations, as the physical interpretation suggests.

The common notation for the divergence I·F is a convenient mnemonic, where the dot denotes an operation reminiscent of the dot product: take the components of ⊐ (see del), apply them to the components of **F**, and sum the results. As a result, this is considered an abuse of notation.

Physical interpretation as source density [edit]

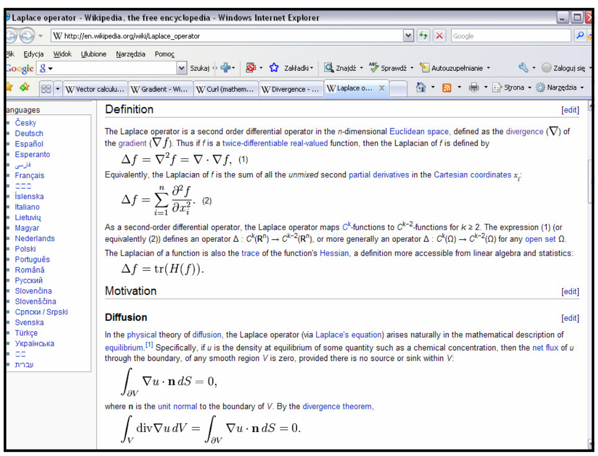
In physical terms, the divergence of a three dimensional vector field is the extent to which the vector field flow behaves like a source or a sink at a given point. It is a local measure of its "outgoingness"—the extent to which there is more exiting an infinitesimal region of space than entering it. If the divergence is nonzero at some point then there must be a source or sink at that position[1].

An alternative but equivalent definition, gives the divergence as the derivative of the net flow of the vector field across the surface of a small sphere relative to the volume of the sphere. (Note that we are imagining the vector field to be like the velocity vector field of a fluid (in motion) when we use the terms flow, sink and so on.) Formally,

$$(\operatorname{div}\mathbf{F})(p) = \lim_{r\to 0}\iint_{S(r)}\frac{\mathbf{F}\cdot\mathbf{n}\,dS}{\frac{4}{3}\pi r^3}$$

where $S(r)$ denotes the sphere of radius r about a point p in $\mathbf{R}^3$, and the integral is a surface integral taken with respect to **n**, the normal to that sphere.

Instead of a sphere, any other volume ΔV is possible, if instead of $\frac{4\pi r^3}{3}$ one writes $|\,\Delta V\,|$. From this definition it also becomes



Laplace operator - Wikipedia, the free encyclopedia - Windows Internet Explorer

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

Definition [edit]

The Laplace operator is a second order differential operator in the n-dimensional Euclidean space, defined as the divergence (∇) of the gradient (∇f). Thus if f is a twice-differentiable real-valued function, then the Laplacian of f is defined by

$$\Delta f = \nabla^2 f = \nabla\cdot\nabla f, \quad (1)$$

Equivalently, the Laplacian of f is the sum of all the *unmixed* second partial derivatives in the Cartesian coordinates x_i:

$$\Delta f = \sum_{i=1}^{n}\frac{\partial^2 f}{\partial x_i^2}. \quad (2)$$

As a second-order differential operator, the Laplace operator maps C^k-functions to C^{k-2}-functions for $k \ge 2$. The expression (1) (or equivalently (2)) defines an operator $\Delta : C^k(\mathbf{R}^n) \to C^{k-2}(\mathbf{R}^n)$, or more generally an operator $\Delta : C^k(\Omega) \to C^{k-2}(\Omega)$ for any open set Ω.

The Laplacian of a function is also the trace of the function's Hessian, a definition more accessible from linear algebra and statistics:

$$\Delta f = \operatorname{tr}(H(f)).$$

Motivation [edit]

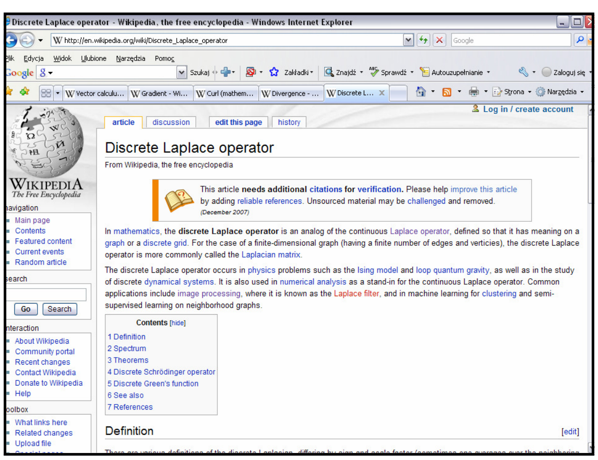
Diffusion [edit]

In the physical theory of diffusion, the Laplace operator (via Laplace's equation) arises naturally in the mathematical description of equilibrium.[1] Specifically, if u is the density at equilibrium of some quantity such as a chemical concentration, then the net flux of u through the boundary, of any smooth region V is zero, provided there is no source or sink within V:

$$\int_{\partial V}\nabla u\cdot\mathbf{n}\,dS = 0,$$

where **n** is the unit normal to the boundary of V. By the divergence theorem,

$$\int_V \operatorname{div}\nabla u\,dV = \int_{\partial V}\nabla u\cdot\mathbf{n}\,dS = 0.$$



Discrete Laplace operator - Wikipedia, the free encyclopedia - Windows Internet Explorer

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

article | discussion | edit this page | history

Log in / create account

Discrete Laplace operator

From Wikipedia, the free encyclopedia

This article **needs additional citations for verification.** Please help improve this article by adding reliable references. Unsourced material may be challenged and removed. *(December 2007)*

In mathematics, the **discrete Laplace operator** is an analog of the continuous Laplace operator, defined so that it has meaning on a graph or a discrete grid. For the case of a finite-dimensional graph (having a finite number of edges and verticies), the discrete Laplace operator is more commonly called the Laplacian matrix.

The discrete Laplace operator occurs in physics problems such as the Ising model and loop quantum gravity, as well as in the study of discrete dynamical systems. It is also used in numerical analysis as a stand-in for the continuous Laplace operator. Common applications include image processing, where it is known as the Laplace filter, and in machine learning for clustering and semi-supervised learning on neighborhood graphs.

Contents [hide]

Definition [edit]

Laplacian

Laplacian of a 2-D function f(x,y) is a second derivative defined as:

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

The Laplacian of a discrete image can be approximated by a difference equation:

$$\nabla^2 f \approx 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

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

Laplacian of a discrete image

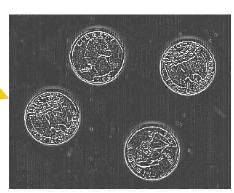
0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

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

Laplacian of a discrete image

-1	-1	-1
-1	8	-1
-1	-1	-1



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

LoG (Laplacian of Gaussian)

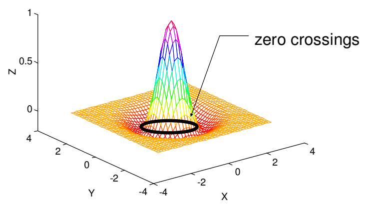
A more suitable use of the Laplacian is in finding the location of edges using its *zero-crossing* property. This concept is based on convolving an image with the Laplacian of a 2-D Gaussian function of the form:

$$h(x, y) = exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

where σ is the standard deviation. Assume $r^2=x^2+y^2$. Then, the Laplacian of h with respect to r is:

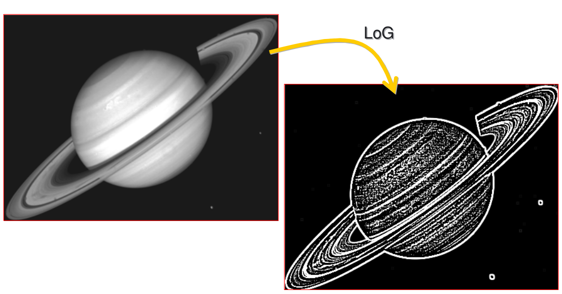
$$\nabla^2 h = \left(\frac{r^2 - \sigma^2}{\sigma^4}\right) exp\left(-\frac{r^2}{2\sigma^2}\right)$$

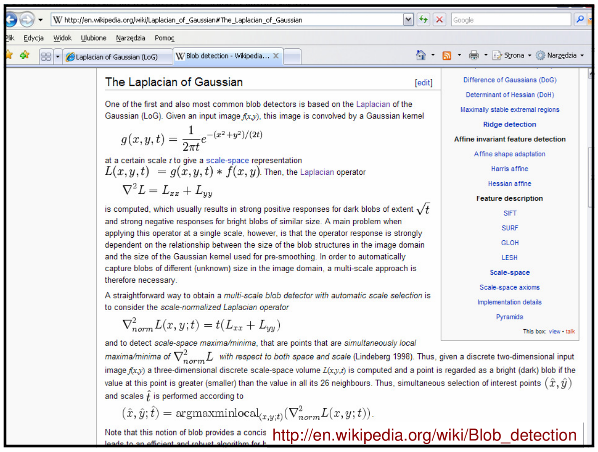
LoG (Laplacian of Gaussian)



Laplacian of a 2-D Gaussian function for $\sigma = 1$
(also called the Mexican hat function).

LoG (Laplacian of Gaussian)





The Laplacian of Gaussian

One of the first and also most common blob detectors is based on the Laplacian of the Gaussian (LoG). Given an input image $f(x,y)$, this image is convolved by a Gaussian kernel

$$g(x,y,t) = \frac{1}{2\pi t} e^{-(x^2+y^2)/(2t)}$$

at a certain scale t to give a scale-space representation $L(x,y,t) \ = g(x,y,t) * f(x,y)$. Then, the Laplacian operator

$$\nabla^2 L = L_{xx} + L_{yy}$$

is computed, which usually results in strong positive responses for dark blobs of extent $\sqrt{t}$ and strong negative responses for bright blobs of similar size. A main problem when applying this operator at a single scale, however, is that the operator response is strongly dependent on the relationship between the size of the blob structures in the image domain and the size of the Gaussian kernel used for pre-smoothing. In order to automatically capture blobs of different (unknown) size in the image domain, a multi-scale approach is therefore necessary.

A straightforward way to obtain a *multi-scale blob detector with automatic scale selection* is to consider the *scale-normalized Laplacian operator*

$$\nabla^2_{norm} L(x,y;t) = t(L_{xx} + L_{yy})$$

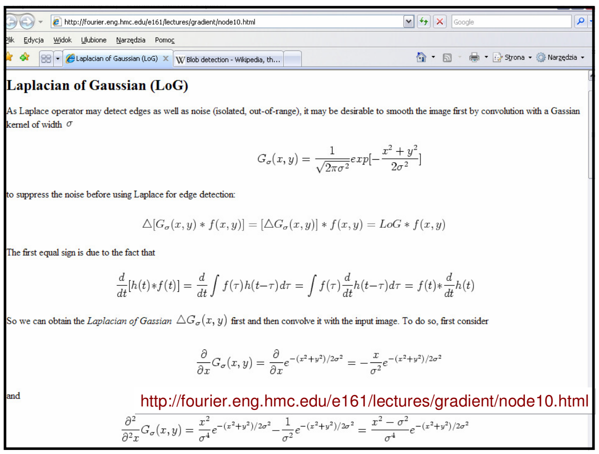
and to detect *scale-space maxima/minima*, that are points that are *simultaneously local maxima/minima of* $\nabla^2_{norm} L$ *with respect to both space and scale* (Lindeberg 1998). Thus, given a discrete two-dimensional input image $f(x,y)$ a three-dimensional discrete scale-space volume $L(x,y,t)$ is computed and a point is regarded as a bright (dark) blob if the value at this point is greater (smaller) than the value in all its 26 neighbours. Thus, simultaneous selection of interest points $(\hat{x},\hat{y})$ and scales $\hat{t}$ is performed according to

$$(\hat{x},\hat{y};\hat{t}) = \operatorname{argmaxminlocal}_{(x,y;t)}(\nabla^2_{norm} L(x,y;t)).$$

Note that this notion of blob provides a concis

Difference of Gaussians (DoG)
Determinant of Hessian (DoH)
Maximally stable extremal regions
Ridge detection
Affine invariant feature detection
Affine shape adaptation
Harris affine
Hessian affine
Feature description
SIFT
SURF
GLOH
LESH
Scale-space
Scale-space axioms
Implementation details
Pyramids

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



Laplacian of Gaussian (LoG)

As Laplace operator may detect edges as well as noise (isolated, out-of-range), it may be desirable to smooth the image first by convolution with a Gassian kernel of width σ

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} exp[-\frac{x^2+y^2}{2\sigma^2}]$$

to suppress the noise before using Laplace for edge detection:

$$\triangle[G_\sigma(x,y) * f(x,y)] = [\triangle G_\sigma(x,y)] * f(x,y) = LoG * f(x,y)$$

The first equal sign is due to the fact that

$$\frac{d}{dt}[h(t)*f(t)] = \frac{d}{dt}\int f(\tau)h(t-\tau)d\tau = \int f(\tau)\frac{d}{dt}h(t-\tau)d\tau = f(t)*\frac{d}{dt}h(t)$$

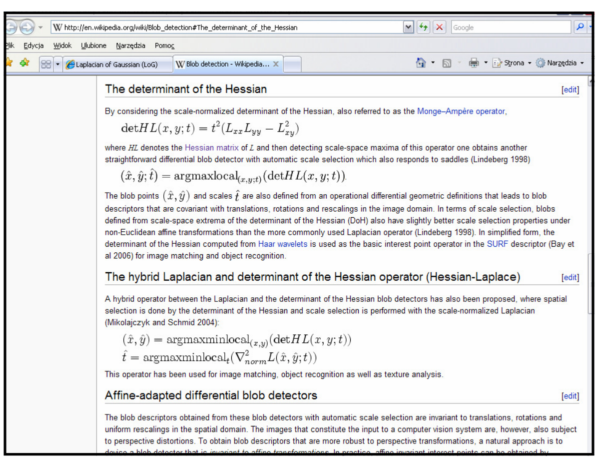
So we can obtain the *Laplacian of Gassian* $\triangle G_\sigma(x,y)$ first and then convolve it with the input image. To do so, first consider

$$\frac{\partial}{\partial x}G_\sigma(x,y) = \frac{\partial}{\partial x}e^{-(x^2+y^2)/2\sigma^2} = -\frac{x}{\sigma^2}e^{-(x^2+y^2)/2\sigma^2}$$

and

$$\frac{\partial^2}{\partial^2 x}G_\sigma(x,y) = \frac{x^2}{\sigma^4}e^{-(x^2+y^2)/2\sigma^2} - \frac{1}{\sigma^2}e^{-(x^2+y^2)/2\sigma^2} = \frac{x^2-\sigma^2}{\sigma^4}e^{-(x^2+y^2)/2\sigma^2}$$

http://fourier.eng.hmc.edu/e161/lectures/gradient/node10.html



The determinant of the Hessian

By considering the scale-normalized determinant of the Hessian, also referred to as the Monge–Ampère operator,

$$\det HL(x,y;t) = t^2(L_{xx}L_{yy} - L_{xy}^2)$$

where HL denotes the Hessian matrix of L and then detecting scale-space maxima of this operator one obtains another straightforward differential blob detector with automatic scale selection which also responds to saddles (Lindeberg 1998)

$$(\hat{x},\hat{y};\hat{t}) = \operatorname{argmaxlocal}_{(x,y;t)}(\det HL(x,y;t)).$$

The blob points $(\hat{x},\hat{y})$ and scales $\hat{t}$ are also defined from an operational differential geometric definitions that leads to blob descriptors that are covariant with translations, rotations and rescalings in the image domain. In terms of scale selection, blobs defined from scale-space extrema of the determinant of the Hessian (DoH) also have slightly better scale selection properties under non-Euclidean affine transformations than the more commonly used Laplacian operator (Lindeberg 1998). In simplified form, the determinant of the Hessian computed from Haar wavelets is used as the basic interest point operator in the SURF descriptor (Bay et al 2006) for image matching and object recognition.

The hybrid Laplacian and determinant of the Hessian operator (Hessian-Laplace)

A hybrid operator between the Laplacian and the determinant of the Hessian blob detectors has also been proposed, where spatial selection is done by the determinant of the Hessian and scale selection is performed with the scale-normalized Laplacian (Mikolajczyk and Schmid 2004):

$$(\hat{x},\hat{y}) = \operatorname{argmaxminlocal}_{(x,y)}(\det HL(x,y;t))$$
$$\hat{t} = \operatorname{argmaxminlocal}_t(\nabla^2_{norm} L(\hat{x},\hat{y};t))$$

This operator has been used for image matching, object recognition as well as texture analysis.

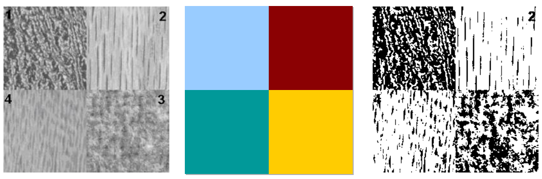
Affine-adapted differential blob detectors

The blob descriptors obtained from these blob detectors with automatic scale selection are invariant to translations, rotations and uniform rescalings in the spatial domain. The images that constitute the input to a computer vision system are, however, also subject to perspective distortions. To obtain blob descriptors that are more robust to perspective transformations, a natural approach is to

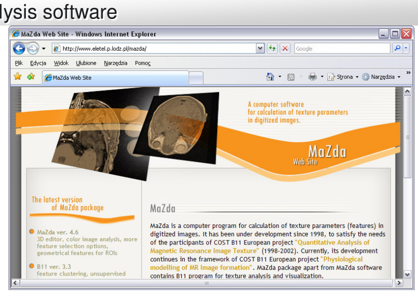
Texture problem

A texture

perceived by humans is a visualization of complex patterns composed of spatially organized, repeated subpatterns, which have a characteristic, somewhat uniform appearance. The local subpatterns within an image are perceived to demonstrate specific brightness, color size, roughness, directivity, randomness, smoothness, granulation, etc. A texture may carry substantial information about the structure of physical objects.

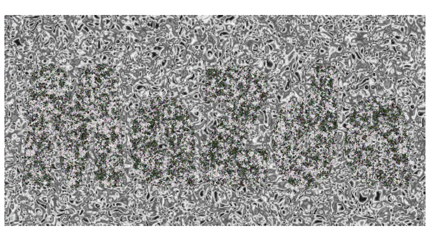


Texture analysis software

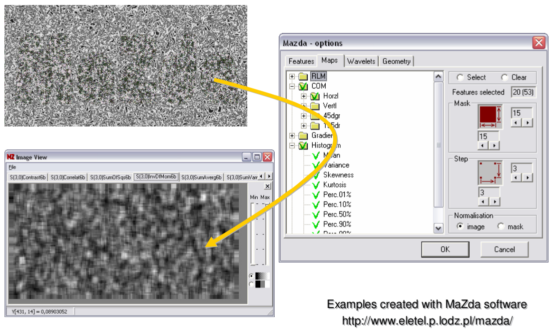


http://www.eletel.p.lodz.pl/mazda/
http://www.eletel.p.lodz.pl/programy/cost/cost_project.html
http://www.eletel.p.lodz.pl/programy/cost/pdf_1.pdf
http://dx.doi.org/10.1016/j.cmpb.2008.08.005

Texture problem



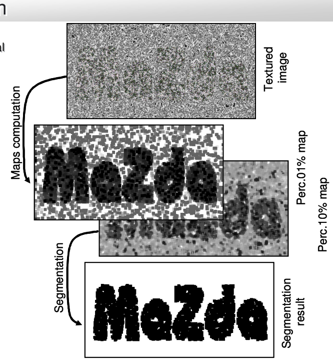
MaZda texture maps



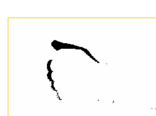
Examples created with MaZda software
http://www.eletel.p.lodz.pl/mazda/

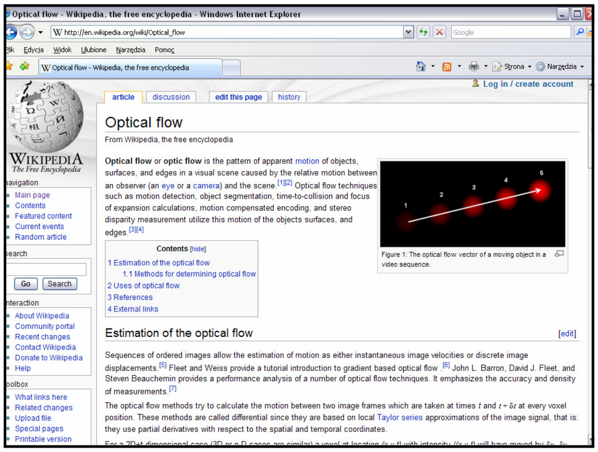
Texture segmentation

1. Convert a textured image into textural feature maps
2. Combine maps if needed
3. Use a gray-level thresholding



Segmentation from motion





Optical flow - Wikipedia, the free encyclopedia - Windows Internet Explorer

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

Optical flow

From Wikipedia, the free encyclopedia

Optical flow or **optic flow** is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene.[1][2] Optical flow techniques such as motion detection, object segmentation, time-to-collision and focus of expansion calculations, motion compensated encoding, and stereo disparity measurement utilize this motion of the objects surfaces, and edges.[3][4]

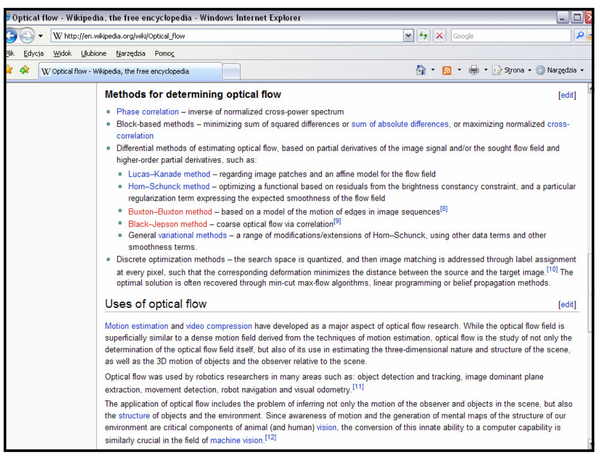
Figure 1: The optical flow vector of a moving object in a video sequence.

Contents [hide]

Estimation of the optical flow [edit]

Sequences of ordered images allow the estimation of motion as either instantaneous image velocities or discrete image displacements.[5] Fleet and Weiss provide a tutorial introduction to gradient based optical flow .[6] John L. Barron, David J. Fleet, and Steven Beauchemin provides a performance analysis of a number of optical flow techniques. It emphasizes the accuracy and density of measurements.[7]

The optical flow methods try to calculate the motion between two image frames which are taken at times t and $t+\delta t$ at every voxel position. These methods are called differential since they are based on local Taylor series approximations of the image signal, that is: they use partial derivatives with respect to the spatial and temporal coordinates.



Optical flow - Wikipedia, the free encyclopedia - Windows Internet Explorer

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

Methods for determining optical flow [edit]

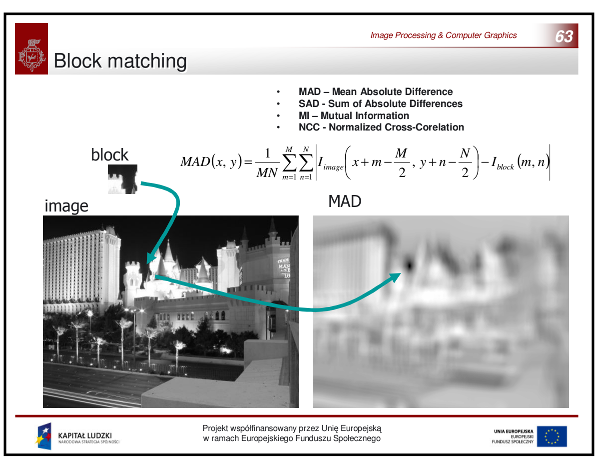
- Phase correlation – inverse of normalized cross-power spectrum
- Block-based methods – minimizing sum of squared differences or sum of absolute differences, or maximizing normalized cross-correlation
- Differential methods of estimating optical flow, based on partial derivatives of the image signal and/or the sought flow field and higher-order partial derivatives, such as:
  - Lucas–Kanade method – regarding image patches and an affine model for the flow field
  - Horn–Schunck method – optimizing a functional based on residuals from the brightness constancy constraint, and a particular regularization term expressing the expected smoothness of the flow field
  - Buxton–Buxton method – based on a model of the motion of edges in image sequences[8]
  - Black–Jepson method – coarse optical flow via correlation[9]
  - General variational methods – a range of modifications/extensions of Horn–Schunck, using other data terms and other smoothness terms.
- Discrete optimization methods – the search space is quantized, and then image matching is addressed through label assignment at every pixel, such that the corresponding deformation minimizes the distance between the source and the target image.[10] The optimal solution is often recovered through min-cut max-flow algorithms, linear programming or belief propagation methods.

Uses of optical flow [edit]

Motion estimation and video compression have developed as a major aspect of optical flow research. While the optical flow field is superficially similar to a dense motion field derived from the techniques of motion estimation, optical flow is the study of not only the determination of the optical flow field itself, but also of its use in estimating the three-dimensional nature and structure of the scene, as well as the 3D motion of objects and the observer relative to the scene.

Optical flow was used by robotics researchers in many areas such as: object detection and tracking, image dominant plane extraction, movement detection, robot navigation and visual odometry.[11]

The application of optical flow includes the problem of inferring not only the motion of the observer and objects in the scene, but also the structure of objects and the environment. Since awareness of motion and the generation of mental maps of the structure of our environment are critical components of animal (and human) vision, the conversion of this innate ability to a computer capability is similarly crucial in the field of machine vision.[12]

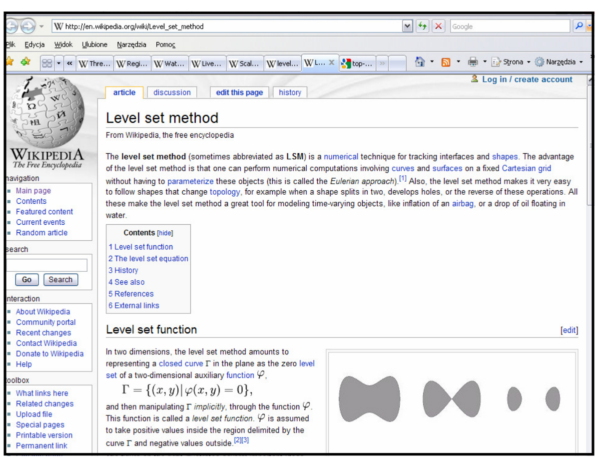


Segmentation form motion

- Divide video frame (n) into blocks,
- Calculate MAD functions of all the blocks by matching them with the next frame (n+1),
- For each block calculate vector from block coordinates to MAD maximum corresponding with the block,
- Find high magnitude vectors.

Deformable models

- Deformable models
  - Nonparametric
  - Parametric


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

Level set method

From Wikipedia, the free encyclopedia

The **level set method** (sometimes abbreviated as **LSM**) is a numerical technique for tracking interfaces and shapes. The advantage of the level set method is that one can perform numerical computations involving curves and surfaces on a fixed Cartesian grid without having to parameterize these objects (this is called the *Eulerian approach*).[1] Also, the level set method makes it very easy to follow shapes that change topology, for example when a shape splits in two, develops holes, or the reverse of these operations. All these make the level set method a great tool for modeling time-varying objects, like inflation of an airbag, or a drop of oil floating in water.

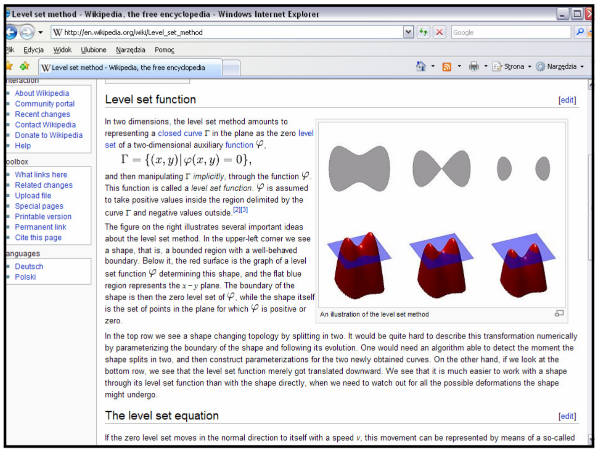
Contents [hide]

Level set function [edit]

In two dimensions, the level set method amounts to representing a closed curve Γ in the plane as the zero level set of a two-dimensional auxiliary function φ,

$$\Gamma = \{(x,y) \mid \varphi(x,y) = 0\},$$

and then manipulating Γ *implicitly*, through the function φ. This function is called a *level set function*. φ is assumed to take positive values inside the region delimited by the curve Γ and negative values outside.[2][3]



Level set method - Wikipedia, the free encyclopedia - Windows Internet Explorer

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

Plik Edycja Widok Ulubione Narzędzia Pomoc

Level set method - Wikipedia, the free encyclopedia

Strona Narzędzia

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
- Cite this page

Languages
- Deutsch
- Polski

Level set function [edit]

In two dimensions, the level set method amounts to representing a closed curve Γ in the plane as the zero level set of a two-dimensional auxiliary function φ,

$$\Gamma = \{(x, y) | \varphi(x, y) = 0\},$$

and then manipulating Γ *implicitly*, through the function φ. This function is called a *level set function*. φ is assumed to take positive values inside the region delimited by the curve Γ and negative values outside.[2][3]

The figure on the right illustrates several important ideas about the level set method. In the upper-left corner we see a shape, that is, a bounded region with a well-behaved boundary. Below it, the red surface is the graph of a level set function φ determining this shape, and the flat blue region represents the $x-y$ plane. The boundary of the shape is then the zero level set of φ, while the shape itself is the set of points in the plane for which φ is positive or zero.

An illustration of the level set method

In the top row we see a shape changing topology by splitting in two. It would be quite hard to describe this transformation numerically by parameterizing the boundary of the shape and following its evolution. One would need an algorithm able to detect the moment the shape splits in two, and then construct parameterizations for the two newly obtained curves. On the other hand, if we look at the bottom row, we see that the level set function merely got translated downward. We see that it is much easier to work with a shape through its level set function than with the shape directly, when we need to watch out for all the possible deformations the shape might undergo.

The level set equation [edit]

If the zero level set moves in the normal direction to itself with a speed v, this movement can be represented by means of a so-called

Leveset method

$\varphi(X, t)$ – function of image and time
X – coordinates in image space
t – time (function φ *evolves*)

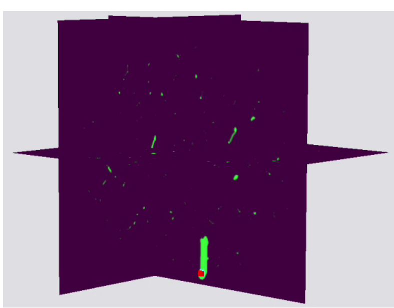
Function φ evolves according to some defined criteria
After the evolution completes equation $\varphi = 0$ is solved to find objects boundaries.

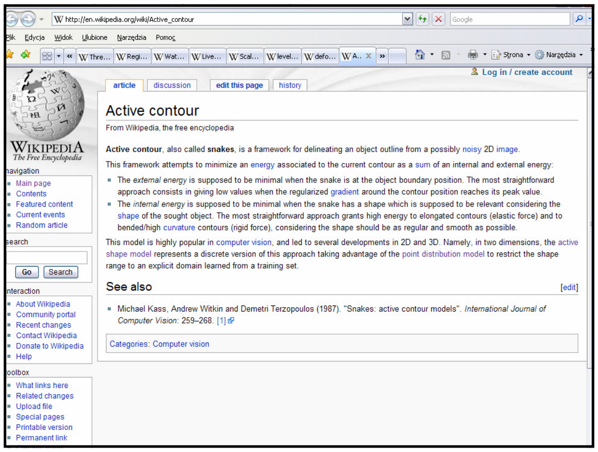
One of possible equations (implementations) to evolve function φ:

$$\frac{\partial \phi(\mathbf{x})}{\partial t} = \delta_\epsilon(\phi(\mathbf{x}))[\nu\kappa(\phi(\mathbf{x})) - \{(I(\mathbf{x}) - \mu_1)^2 - (I(\mathbf{x}) - \mu_0)^2\}]$$

δ_ϵ is a derivative of sign(φ),
I(**X**) is an image grey level in range <0, 1>,
μ_0, μ_1 are mean grey levels for object and background,
κ is divergence of normalized gradient of φ,
v is constant speed

Example





Active contour

From Wikipedia, the free encyclopedia

Active contour, also called **snakes**, is a framework for delineating an object outline from a possibly noisy 2D image.

This framework attempts to minimize an energy associated to the current contour as a sum of an internal and external energy:

- The *external energy* is supposed to be minimal when the snake is at the object boundary position. The most straightforward approach consists in giving low values when the regularized gradient around the contour position reaches its peak value.
- The *internal energy* is supposed to be minimal when the snake has a shape which is supposed to be relevant considering the shape of the sought object. The most straightforward approach grants high energy to elongated contours (elastic force) and to bended/high curvature contours (rigid force), considering the shape should be as regular and smooth as possible.

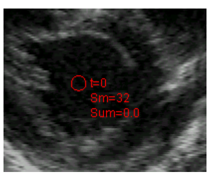
This model is highly popular in computer vision, and led to several developments in 2D and 3D. Namely, in two dimensions, the active shape model represents a discrete version of this approach taking advantage of the point distribution model to restrict the shape range to an explicit domain learned from a training set.

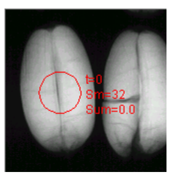
See also [edit]

- Michael Kass, Andrew Witkin and Demetri Terzopoulos (1987). "Snakes: active contour models". *International Journal of Computer Vision*. 259–268. [1]

Categories: Computer vision

Active contour (parametric model)





KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

Active contour equations

$$E_S = \int_0^{S_m} \left[E_i(\mathbf{v}(s)) + E_p(\mathbf{v}(s)) \right] ds$$ Models energy

$$E_p(s) = -\left|\nabla I(v(s))\right|$$ image component

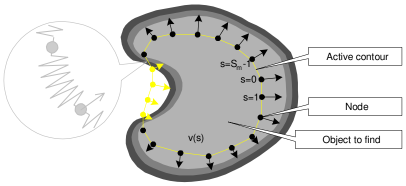
$$E_i(\mathbf{v}(s)) = \tau \left|\frac{d\mathbf{v}(s)}{ds}\right|^2 + \rho \left|\frac{d^2\mathbf{v}(s)}{ds^2}\right|^2$$ tension component

$$\mathbf{F}(s) = -\nabla E_S(s)$$ Force minimizing the energy

$$m\frac{\partial^2 \mathbf{v}(s,t)}{\partial t^2} + l\frac{\partial \mathbf{v}(s,t)}{\partial t} = \mathbf{F}(s,t)$$ Energy minimization

KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

Active contour (parametric model)



$$E_S = \sum_s \left[E_i(\mathbf{v}(s)) + E_p(\mathbf{v}(s)) \right]$$

Active contour in 3D (deformable surface)

$$E_S = \iint_{\Omega} \left[E_i(\mathbf{v}(\mathbf{s})) + E_p(\mathbf{v}(\mathbf{s})) \right] d\mathbf{s}$$

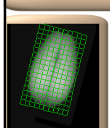
$$E_i = \tau \left[\left| \frac{\partial \mathbf{v}}{\partial i} \right|^2 + \left| \frac{\partial \mathbf{v}}{\partial j} \right|^2 \right] + \rho \left[\left| \frac{\partial^2 \mathbf{v}}{\partial i^2} \right|^2 + 2 \left| \frac{\partial^2 \mathbf{v}}{\partial i \partial j} \right|^2 + \left| \frac{\partial^2 \mathbf{v}}{\partial j^2} \right|^2 \right]$$

http://www.eletel.p.lodz.pl/pms/bada_en.html

Edycja Widok Ulubione Narzędzia Pomoc

Piotr M. Szczypiński

Piotr M. Szczypiński



English
Polski

About me
Research
Publications
Teaching
Downloads

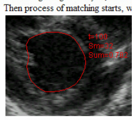
City and TU-Lodz
Lodz • City map
TU-Lodz • TU map
EE Dept. • Institute of Electronics

Search
Interia
Onet
Wirtualna Polska
Google • Netsprint
Altavista • Infoseek
Lycos • Szukacz
Go.com • Netscape

ACTIVE CONTOUR

Active contour is a mathematical model of deformable curve made up of abstract, elastic material. It is used for computer image analysis, for detection of object boundary that is presented in the computer image.

At the beginning of analysis, active contour (that has a simple, arbitrary chosen shape) is placed on the surface of the image. Then process of matching starts, which displace and deforms active contour to fit it to the shape of objects boundary.



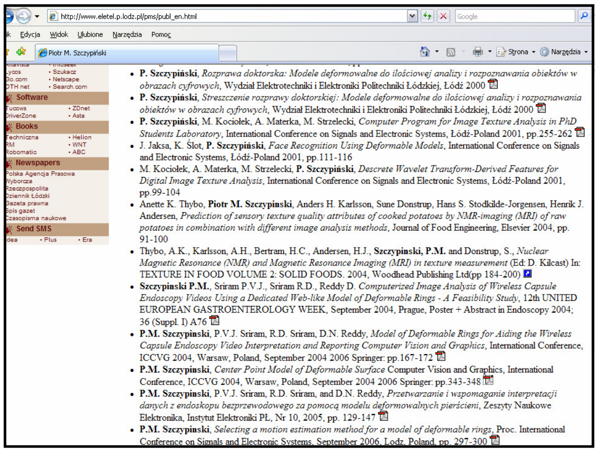
Process of active contour matching performed for ultrasound image of a human heart The purpose of the analysis is to detect a shape of left ventricle.

Some additional information on active contour may be found on the Publications page.

DEFORMABLE MODEL

Deformable (elastic) model is mathematical model for computer image analysis. It may be used for object recognition, for finding objects co-ordinates, size and orientation. In case of object in motion, deformable model may be used for tracking its movement.

Deformable model looks like a grid that is placed on the surface of computer image. At the first step, not deformed grid is placed on the surface of image presenting a reference (model) object. During this step, the information on local image features is stored at every node of a grid. We can say that grid learns how reference object looks like. In the next step the grid is placed on a surface of the image, which presents an object to be analysed. Then process of matching starts, which displace and deforms deformable model's grid to fit a stored image of reference object to the object being analysed.



- **P. Szczypiński**, *Rozprawa doktorska: Modele deformowalne do ilościowej analizy i rozpoznawania obiektów w obrazach cyfrowych*, Wydział Elektrotechniki i Elektroniki Politechniki Łódzkiej, Łódź 2000
- **P. Szczypiński**, *Streszczenie rozprawy doktorskiej: Modele deformowalne do ilościowej analizy i rozpoznawania obiektów w obrazach cyfrowych*, Wydział Elektrotechniki i Elektroniki Politechniki Łódzkiej, Łódź 2000
- **P. Szczypiński**, M. Kociołek, A. Materka, M. Strzelecki, *Computer Program for Image Texture Analysis in PhD Students Laboratory*, International Conference on Signals and Electronic Systems, Łódź-Poland 2001, pp.255-262
- J. Jaksa, K. Ślot, **P. Szczypiński**, *Face Recognition Using Deformable Models*, International Conference on Signals and Electronic Systems, Łódź-Poland 2001, pp.111-116
- M. Kociołek, A. Materka, M. Strzelecki, **P. Szczypiński**, *Descrete Wavelet Transform-Derived Features for Digital Image Texture Analysis*, International Conference on Signals and Electronic Systems, Łódź-Poland 2001, pp.99-104
- Anette K. Thybo, **Piotr M. Szczypinski**, Anders H. Karlsson, Sune Donstrup, Hans S. Stodkilde-Jorgensen, Henrik J. Andersen, *Prediction of sensory texture quality attributes of cooked potatoes by NMR-imaging (MRI) of raw potatoes in combination with different image analysis methods*, Journal of Food Engineering, Elsevier 2004, pp. 91-100
- Thybo, A.K., Karlsson, A.H., Bertram, H.C., Andersen, H.J., **Szczypinski, P.M.** and Donstrup, S., *Nuclear Magnetic Resonance (NMR) and Magnetic Resonance Imaging (MRI) in texture measurement* (Ed: D. Kilcast) In: TEXTURE IN FOOD VOLUME 2: SOLID FOODS. 2004, Woodhead Publishing Ltd(pp 184-200)
- **Szczypinski P.M.**, Sriram P.V.J., Sriram R.D., Reddy D. *Computerized Image Analysis of Wireless Capsule Endoscopy Videos Using a Dedicated Web-like Model of Deformable Rings - A Feasibility Study*, 12th UNITED EUROPEAN GASTROENTEROLOGY WEEK, September 2004, Prague, Poster + Abstract in Endoscopy 2004; 36 (Suppl. I) A76
- **P.M. Szczypinski**, P.V.J. Sriram, R.D. Sriram, D.N. Reddy, *Model of Deformable Rings for Aiding the Wireless Capsule Endoscopy Video Interpretation and Reporting Computer Vision and Graphics*, International Conference, ICCVG 2004, Warsaw, Poland, September 2004 2006 Springer: pp.167-172
- **P.M. Szczypinski**, *Center Point Model of Deformable Surface* Computer Vision and Graphics, International Conference, ICCVG 2004, Warsaw, Poland, September 2004 2006 Springer: pp.343-348
- **P.M. Szczypinski**, P.V.J. Sriram, R.D. Sriram, and D.N. Reddy, *Przetwarzanie i wspomaganie interpretacji danych z endoskopu bezprzewodowego za pomocą modelu deformowalnych pierścieni*, Zeszyty Naukowe Elektronika, Instytut Elektroniki PŁ, Nr 10, 2005, pp. 129-147
- **P.M. Szczypinski**, *Selecting a motion estimation method for a model of deformable rings*, Proc. International Conference on Signals and Electronic Systems, September 2006, Lodz, Poland, pp. 297-300

Summary, discussion and quiz

Summary, discussion and quiz

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

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