ugr Universidad de Granada

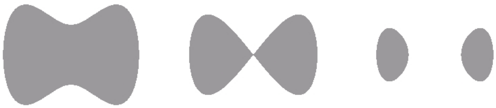
Dynamic Implicit Surfaces

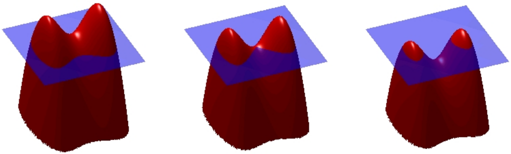
Jarno Ralli [1]

[1]Department of Computer Architecture and Technology
University of Granada, Spain

2012

Introduction





CONTENTS OF THE PRESENTATION

- Math primer
  - Domains, derivative, partial derivative, gradient etc.
- Level-set:s
  - What is a set?
  - What is a level-set?
  - Level-set:s for segmentation
- Dynamic implicit surfaces
  - How to move the interface?
  - Movement induced by external velocity
  - Movement induced by internal velocity
- An example of a concrete level-set based segmentation algorithm

MATH PRIMER

Mathematical Notation 1/2

$$\underbrace{NAME}_{\text{name of the mapping}} : \underbrace{DOMAIN}_{\text{input}} \underbrace{\rightarrow}_{\text{maps into}} \underbrace{CODOMAIN}_{\text{output}}$$

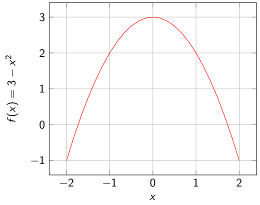
- $f : \mathbb{R} \rightarrow \mathbb{R}$, for example $f(x) = 3 - x^2$
- $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, for example $f(x, y) = 3 - x^2 - y^2$
- $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^{\mathbb{N}+}$, for example $I(x, y)$

Here $\times$ means Cartesian product (i.e. $X \times Y = \{(x, y)|x \in X,\ y \in Y\}$).

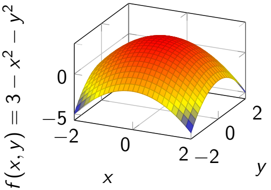
Mathematical Notation 2/2

DIFFERENT DOMAINS

Instead of writing $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ we can take a short cut and define the domain as $\Omega \in \mathbb{R} \times \mathbb{R}$ and write it as $f : \Omega \rightarrow \mathbb{R}$.



(a) $\Omega \in [-2 \ldots 2]$



(b) $\Omega \in [-2 \ldots 2] \times [-2 \ldots 2]$

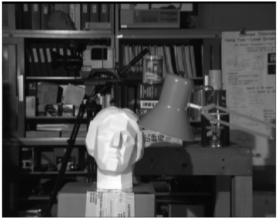
(c) $\Omega \in [1 \ldots m] \times [1 \ldots n]$

Notation

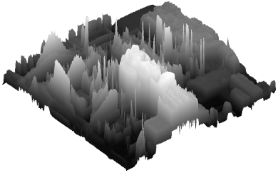
IMAGE AS A FUNCTION

Indeed, the image $I(x, y)$ can be see as a function $I : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ (i.e. each pair (x, y) gets mapped to a real value) and this is the basis of mathematical treatment of images.

What we obtain from a digital camera is a discretised version of the image: $I : +\mathbb{Z} \times +\mathbb{Z} \rightarrow +\mathbb{Z}$



(a) Tsukuba.

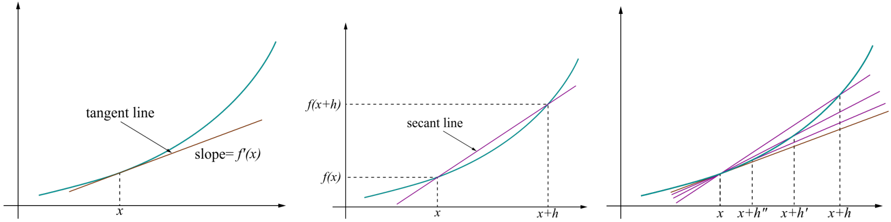


(b) Tsukuba.

Calculus

In calculus **derivative** is a measure how much a function changes as its input changes. Let f be a real valued function. Geometrically derivative of f at a point x is **tangent** to the graph of the function at $(x, f(x))$.
Formally, the *derivative* of the function f at x is the limit:

$$f(x)' = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$



(a) Tangent (b) Secant (c) Limit of secant

NOTATIONS FOR DERIVATIVES

- Leibniz's notation
  - first order: $\frac{dy}{dx}, \frac{df}{dx}(x), \frac{d}{dx}f(x)$
  - higher order: $\frac{d^n y}{dx^n}, \frac{d^n f}{dx^n}(x), \frac{d^n}{dx^n}f(x)$
- Lagrange's notation
  - first order: f'
  - higher order: $f'', f''', f^{(4)}$
- Newton's notation $\left(y = f(t)\right)$
  - first order: $\dot{y}$ (with respect to time)
  - higher order: $\ddot{y}$ (with respect to time)

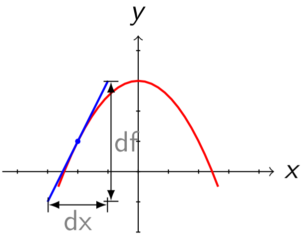


Figure : Leibniz's notation.

So far we have seen how to calculate how much a function of one input variable (e.g. $f(x)$) changes with respect as its input changes. A **partial derivative** tell us how a multi-variable function (e.g. $f(x, y)$) changes with respect to **one of the variables** while the **rest are kept constant**. The partial derivative with respect to x can be noted by: f'_x, f_x, $\partial_x f$ or $\dfrac{\partial}{\partial x}$.
For example:

$$\begin{cases} \dfrac{\partial}{\partial x} 3 - x^2 - y^2 = -2x \\ \dfrac{\partial}{\partial y} 3 - x^2 - y^2 = -2y \end{cases}$$

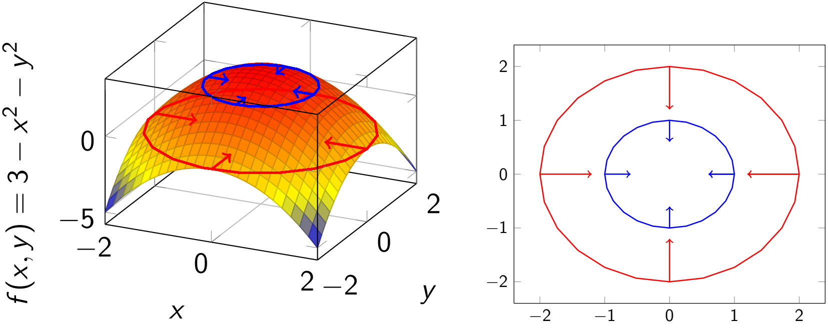
GRADIENT

- **Gradient** of a **scalar field** is a **vector field** that points in the direction of the greatest rate of increase of the scalar field.
- Gradient as operator: $\nabla := \left[\dfrac{\partial}{\partial x} \quad \dfrac{\partial}{\partial y} \right]$
- $|\vec{x}|$ means the Euclidean length of a vector $\vec{x}$
- Magnitude of rate of change: $|\nabla(f)| = \sqrt{f_x^2 + f_y^2}$

SCALAR AND VECTOR FIELDS

- Function $f(x, y)$ is a scalar field since it maps $\Omega := \mathbb{R} \times \mathbb{R}$ to a single value $z = f(x, y)$
- Gradient of this scalar maps two values (f_x and f_y) for every point $z = f(x, y)$ and, therefore, it is called a vector field

Gradient, $\nabla f = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y}\right]$, points in the direction of the greatest rate of increase.



(a) 3D plot of ∇f (b) Seen from above

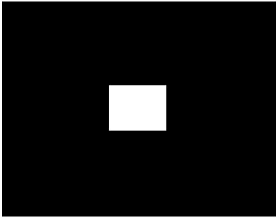
IMAGE DIFFUSION EQUATION

$$\boxed{I_t = DIV(\nabla I)}$$

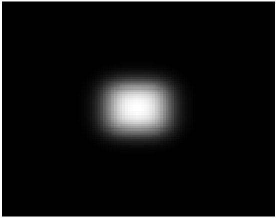
where $DIV(\nabla I) = DIV(\partial_x I\vec{i} + \partial_y I\vec{j})$ and $I := I(x, y)$ refers to the image. This equation can be read as: a temporal change in the image is due to 'movement' of particles due to diffusion. Therefore, the physical interpretation of the $DIV()$ operator is that of diffusion. If $\mathbf{F} = U\vec{i} + V\vec{j}$ is a continuously differentiable vector field, then:

$$\boxed{DIV(F) = \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y}}$$

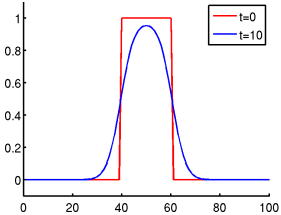
EXAMPLE OF DIFFUSION



(a) Time $t = 0$



(b) Time $t = 1$



(c) Graphs

Level-sets

SEGMENTATION USING LEVEL-SETS

CONTENTS OF THE PRESENTATION

- Math primer
  - Domains, derivative, partial derivative, gradient etc.
- **Level-set:s**
  - **What is a set?**
  - **What is a level-set?**
  - **Level-set:s for segmentation**
- Dynamic implicit surfaces
  - How to move the interface?
  - Movement induced by external velocity
  - Movement induced by internal velocity
- An example of a concrete level-set based segmentation algorithm

Level-sets

Segmentation is the process of joining individual pixels into 'meaningful' groups.

(a) Cones left image.   (b) Segmentation.

Here each group is assigned a different number and each pixel belonging to a particular group is displayed using the group's number.

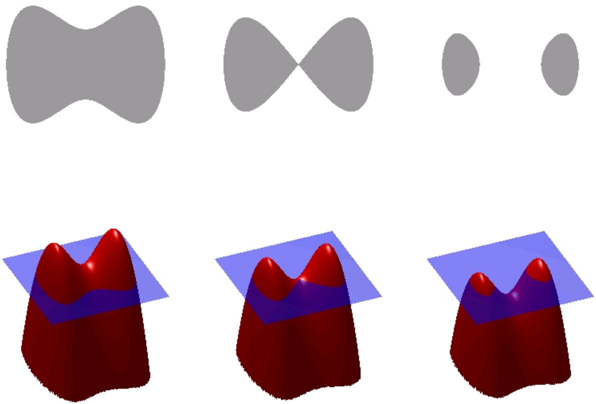
Roughly speaking, segmentation methods can de divided in the following 2 different categories:

- Contour based: different segments are identified by closed contours. Area enclosed inside the contour constitutes as a segment.
  - Explicit contour representation.
  - Implicit contour representation (e.g. level-set).
- Region based: segments are identified by area of the regions. Contour is just the outer part of the segment.

Level-sets

EXPLICIT representation	IMPLICIT representation
-Contour is directly available	-Contour has to be ‘searched’
-Inside of segment:	-Inside of segment:
...searching complicated	...searching trivial
-One segment per contour	-Several segments per contour
-Handling of topological changes	-Handling of topological changes
...via ad-hoc methods	...implicit
-Numerical stability:	-Numerical stability:
...depends on the curve	...depends on derivatives
-Implementation:	-Impelementation:
...depends on dimensionality	...extendible upto n-dims
-Numerically efficient	-Numerically more complex

Level-sets



A *set* can be defined by enclosing the set of members in curly brackets, e.g. $C = \{4, 2, 15\}$. Instead of explicitly writing down each and every member, we can identify the members based on a logical statement as follows:

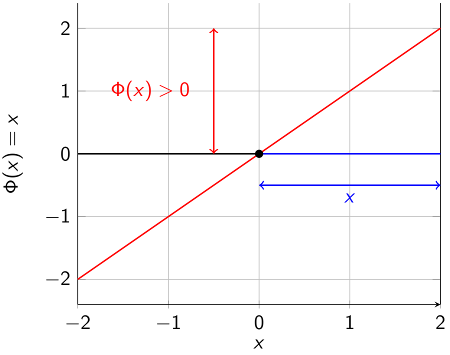
$$\boxed{\{x \,|\, P(x)\}}$$

, which means the set of all x for which $P(x)$ is true. We can use sets for describing the segments: $\{x \,|\, P(x)\}$ describes the set of points (in the domain) that belong to the segment in question.

Using the set notation and a function $\Phi(x)$, we can define the set as:

$$\boxed{\{x \,|\, \Phi(x) > 0\}}$$

, which is a set of values of x where $\Phi(x) > 0$.



Level-sets

We can identify the following sets on the domain based on the function $\Phi(x)$:

$$\boxed{\begin{aligned} interface(\Phi) &:= \{x \mid \Phi(x) = 0\} \\ outside(\Phi) &:= \{x \mid \Phi(x) < 0\} \\ inside(\Phi) &:= \{x \mid \Phi(x) > 0\} \end{aligned}}$$

, where *outside* is the area outside of the segment, *inside* is the area belonging to the segment, and *interface* contains those points separating the segments (called the interface).

EXPLICIT FUNCTION: explicit function is a function where the dependent variables are given explicitly in terms of the independent variables. For example $f(x) = x^2$.

IMPLICIT FUNCTION: implicit function is a function in which the dependent variables are not given explicitly in terms of the independent variable(s) OR it is a function in which the dependent variables are not expressed in terms of some independent variables. For example: $x^2 + y^2 - 3 = 0$.

IMPLICIT REPRESENTATION: $\Phi(x) = 0$ is the zero contour of the function $\Phi(x)$. Therefore, the level-set representation is said to be an 'implicit' representation.

Example of the implicit representation

Suppose that we have an explicit function of the form $\Phi(x) = 3 - x^2$. The function Φ clearly is an explicit function. However, the zero level-set is defined by $\Phi(x) = 0$, where $\Phi(x) = 3 - x^2$. Therefore, the zero level-set is given by:

$$\boxed{3 - x^2 = 0}$$

From this we can identify the zero level-set being as $x = \pm\sqrt{3}$

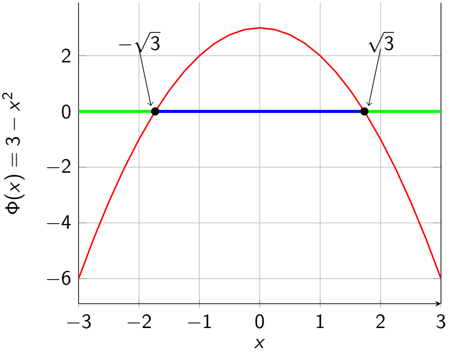
Level-sets

Unfortunately, there is a slight 'confusion' in the terminology. Even if $\Phi(x) = 3 - x^2$ clearly is an explicit function, due to the way it is being used implicitly, fathers of the level-set theorem have decided to call it implicit function. In this case, the implicit function is actually $3 - x^2$, since based on this we detect the *interface* and the *inside* and *outside* as follows:

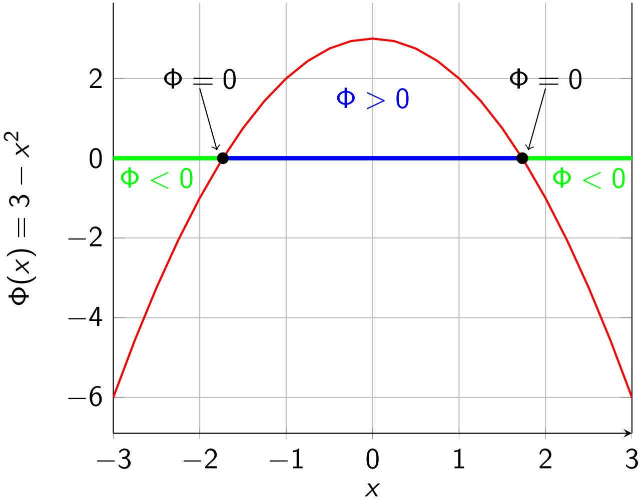
$$\boxed{\begin{aligned} \mathit{interface}(\Phi) &:= \{x \,|\, 3 - x^2 = 0\} \\ \mathit{outside}(\Phi) &:= \{x \,|\, 3 - x^2 < 0\} \\ \mathit{inside}(\Phi) &:= \{x \,|\, 3 - x^2 > 0\} \end{aligned}}$$

In this context better notation might be $\Phi(x) := 3 - x^2$ which means $\Phi(x)$ is another name for $3 - x^2$.

In the case of the explicit function $\Phi(x) = 3 - x^2$, we can divide the domain ($\mathbb{R}$) into three 'significant' sub-domains, namely $(-\inf, -\sqrt{3})$, $(-\sqrt{3}, \sqrt{3})$ and $(\sqrt{3}, \inf)$.



Level-sets



Level-sets

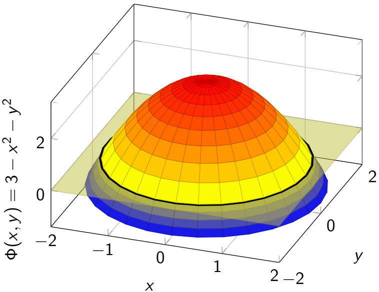
We can identify the following sets on the domain based on the function $\Phi(x, y)$:

$$\boxed{\begin{aligned} \mathit{interface}(\Phi) &:= \{(x, y) \,|\, \Phi(x, y) = 0\} \\ \mathit{outside}(\Phi) &:= \{(x, y) \,|\, \Phi(x, y) < 0\} \\ \mathit{inside}(\Phi) &:= \{(x, y) \,|\, \Phi(x, y) > 0\} \end{aligned}}$$

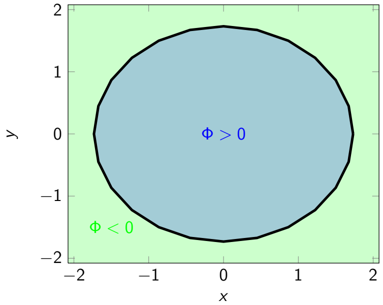
, where *outside* is the area outside of the segment, *inside* is the area belonging to the segment, and *interface* contains those points separating the segments (called the interface).

Level-sets

In the case of the explicit function $\Phi(x,y) = 3 - x^2 - y^2$, the domain $\mathbb{R} \times \mathbb{R}$ can be divided in the following segments based on the level-set function $\Phi(x,y) = 0$:

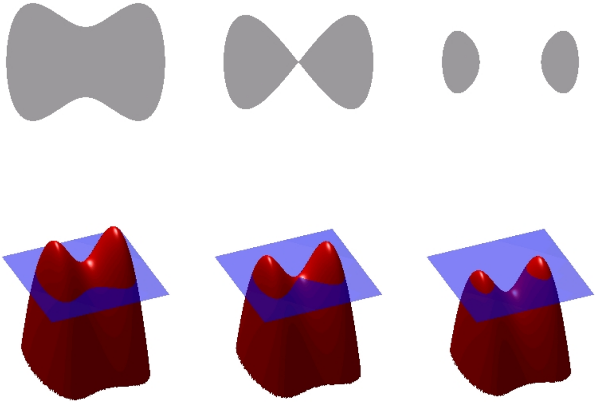


(a) 2D plot



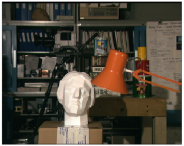
(b) Seen from above

Level-sets

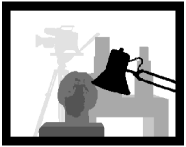


Level-sets

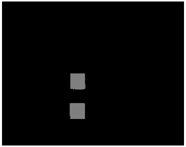
The segmentation process is started with two individual ‘seeds’ (t=0) with no connectivity. Approximately at t=13 these seeds ‘fuse’ together and, therefore, the topology has changed.



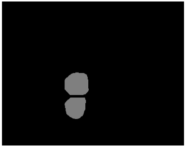
(a) Left.



(b) Disparity.



(c) t=0.



(d) t=12.

(e) t=14.

(f) t=199.

DYNAMIC IMPLICIT SURFACES

CONTENTS OF THE PRESENTATION

- Math primer
  - Domains, derivative, partial derivative, gradient etc.
- Level-set:s
  - What is a set?
  - What is a level-set?
  - Level-set:s for segmentation
- **Dynamic implicit surfaces**
  - **How to move the interface?**
  - **Movement induced by external velocity**
  - **Movement induced by internal velocity**
- An example of a concrete level-set based segmentation algorithm

Moving Interfaces

So far we have defined what are level-sets and how these can be used for our purpose of segmenting. Until now the level-sets have all been *static* (i.e. they don't move). Here, the idea is to define the necessary mathematical concepts in order to move the level-sets. Hence the name DYNAMIC IMPLICIT SURFACES.

Moving Interfaces

The interface is ‘moved’ automatically towards the borders of the object.

(a) Original interface.

(b) Interface after n iterations.

Now that we know how level-sets can be used for segmentation, we want to know how we can move the interface separating the segments. Here, $\vec{V} = u\vec{i} + v\vec{j}$ is an externally generated velocity field that we want to use for moving the interface. This can be achieved using a simple convection equation:

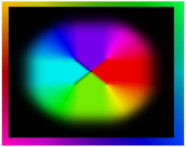
$$\boxed{\begin{array}{l} \dfrac{\partial \Phi}{\partial t} + \vec{V} \cdot \nabla \Phi = 0 \\ \Phi_t + u\Phi_x + v\Phi_y = 0 \end{array}}$$

, where the t subscript denotes temporal partial derivative, ∇ is the spatial gradient operator and $\cdot$ is the scalar product. This partial differential equation (PDE) defines the motion of the interface.

Moving Interfaces

$$\boxed{\vec{V} \cdot \nabla \Phi = 0}$$

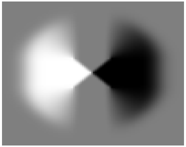
(a) $\vec{V}$



(b) $\nabla \Phi$

(c) u

$*$

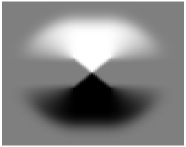


(d) Φ_x

$+$

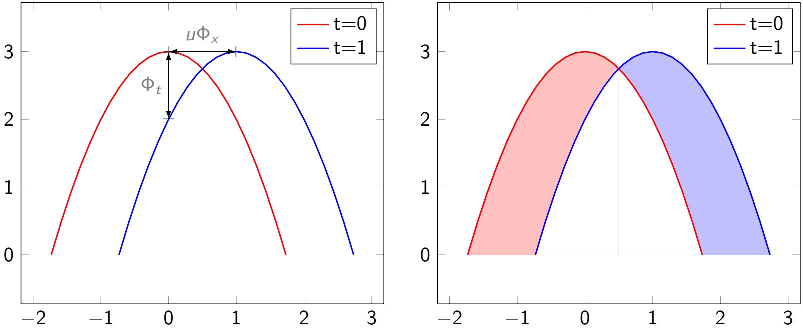
(e) v

$*$



(f) Φ_x

Convection 'intuitively' in 1D. $\Phi_t + u\Phi_x = 0$, therefore $\Phi_t = -u\Phi_x$.



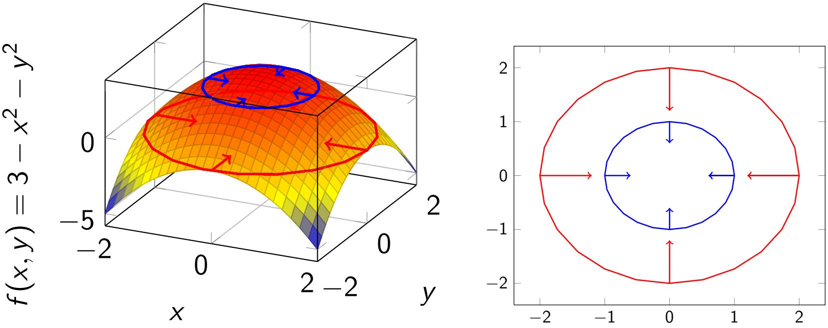
(a) (b)

Here, $\vec{V} = u\vec{i} + v\vec{j}$ is an externally generated velocity field.

$$\boxed{\begin{aligned} &\frac{\partial \Phi}{\partial t} + \vec{V} \cdot \nabla \Phi = 0 \\ &\Phi_t + u\Phi_x + v\Phi_y = 0 \end{aligned}}$$

Calculus

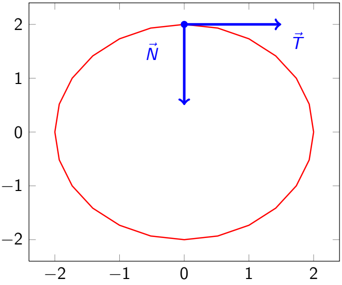
Normal unit vector can be expressed as: $\dfrac{\nabla\Phi}{|\nabla\Phi|}$ (note that $|\nabla\Phi|$ is the length of the vector).



(a) 3D plot of ∇f

(b) Seen from above

We can define movement in the direction of the normal of the level-set as follows:



$$\boxed{\begin{aligned} &\Phi_t + (v_t\vec{T} + v_n\vec{N})\cdot\nabla\Phi = 0 \\ &\Phi_t + (\underbrace{v_t\vec{T}}_{=0} + v_n\vec{N})\cdot\nabla\Phi = 0 \\ &\Phi_t + v_n\vec{N}\cdot\nabla\Phi = 0 \end{aligned}}$$

, where v_n and v_t are the velocities in the direction of the normal and the tangent.

We can define movement in the direction of the normal of the level-set as follows:

$$\boxed{\begin{aligned} \Phi_t + (v_t \vec{T} + v_n \vec{N}) \cdot \nabla\Phi = 0 \\ \Phi_t + (\underbrace{v_t \vec{T}}_{=0} + v_n \vec{N}) \cdot \nabla\Phi = 0 \\ \Phi_t + v_n \vec{N} \cdot \nabla\Phi = 0 \end{aligned}}$$

, where v_n and v_t are the velocities in the direction of the normal and the tangent. Since $\vec{N} = \nabla\Phi/|\nabla\Phi|$, we have:

$$\boxed{\begin{aligned} \Phi_t + v_n \frac{\nabla\Phi}{|\nabla\Phi|} \cdot \nabla\Phi = 0 \\ \Phi_t + v_n |\nabla\Phi| = 0 \end{aligned}}$$

Moving Interfaces

So far we have seen movement in external velocity field and movement in the normal direction. An interesting case of movement in the normal direction is so called Mean Curvature Motion (MCM), induced by the local curvature.

Motion by mean curvature is defined as follows:

$$\boxed{v_n = -\alpha DIV\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)}$$

, where α is simply a coefficient, typically varying between [0..1], defining how much of the local curvature is taken into account.

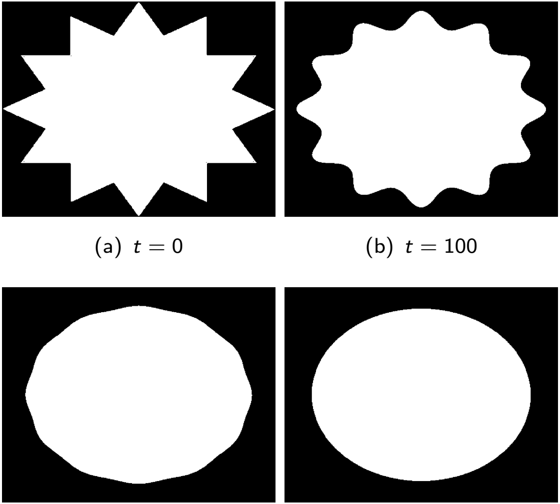
Movement in the normal direction:

$$\boxed{\Phi_t + v_n |\nabla \Phi| = 0}$$

By plugging in the the local curvature in the normal movement model, we obtain the following:

$$\boxed{\Phi_t - \alpha DIV \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) |\nabla \Phi| = 0}$$

Moving Interfaces



(a) $t = 0$ (b) $t = 100$

(c) $t = 0$ (d) $t = 100$

CONCRETE SEGMENTATION ALGORITHM

CONTENTS OF THE PRESENTATION

- Math primer
  - Domains, derivative, partial derivative, gradient etc.
- Level-set:s
  - What is a set?
  - What is a level-set?
  - Level-set:s for segmentation
- Dynamic implicit surfaces
  - How to move the interface?
  - Movement induced by external velocity
  - Movement induced by internal velocity
- **An example of a concrete level-set based segmentation algorithm**

Algorithm

'A GEOMETRIC MODEL FOR ACTIVE COUNTOURS IN IMAGE PROCESSING', Vicent Caselles et al., 1993

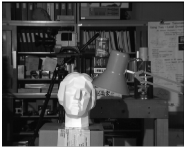
$$\Phi_t = \underbrace{g(|\nabla I|)DIV\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)|\nabla\Phi|}_{\text{minimises local curvature}} + \underbrace{g(|\nabla I|)c|\nabla\Phi|}_{\text{balloon force}}$$

$$\Phi_t = \underbrace{\left(g(\nabla I)DIV\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right) + g(|\nabla I|)c\right)}_{\text{normal velocity}}|\nabla\Phi|$$

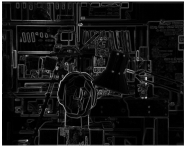
, where $g()$ is a monotonically descending function, I is the input image and c is a parameter defining the 'balloon' force.

Algorithm

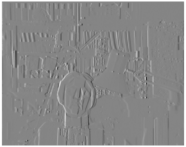
Function of the $g(|\nabla I|)$ is to stop movement of the contour once the contour reaches object edges (i.e. $|\nabla I|$ obtains 'big' value).



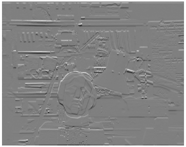
(a) Image.



(b) $|\nabla I|$



(c) I_x



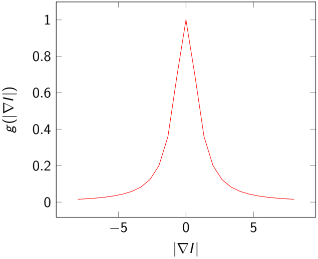
(d) I_y

Algorithm

Therefore, as $|\nabla I| \rightarrow \inf$, then $g(|\nabla I|) \rightarrow 0$. Once such function is:

$$\boxed{g(|\nabla I|) = \frac{1}{1 + \left(\frac{|\nabla I|}{\lambda}\right)^2}}$$

, where λ is a parameter that controls shape of the function and it is used for defining what strength (i.e. magnitude) of gradient is considered to be a border of an object.

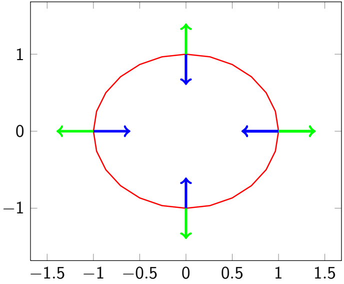


Algorithm

The 'balloon' force/movement is nothing more than constant movement in the direction defined by the gradient as seen previously.

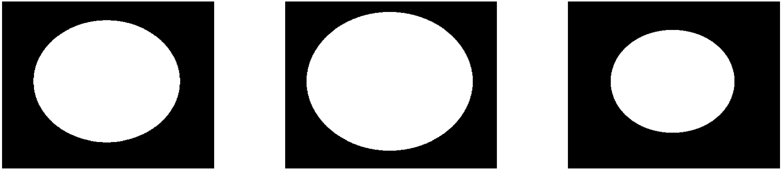
$$\boxed{\Phi_t = c|\nabla\Phi|}$$

If $c > 0$ the contour 'expands', if $c < 0$ the contour 'shrinks'



Algorithm

Example of the balloon force.



(a) $t = 0$ (b) $t = 50$, $c = 1.5$ (c) $t = 50$, $c = -1.5$

Algorithm

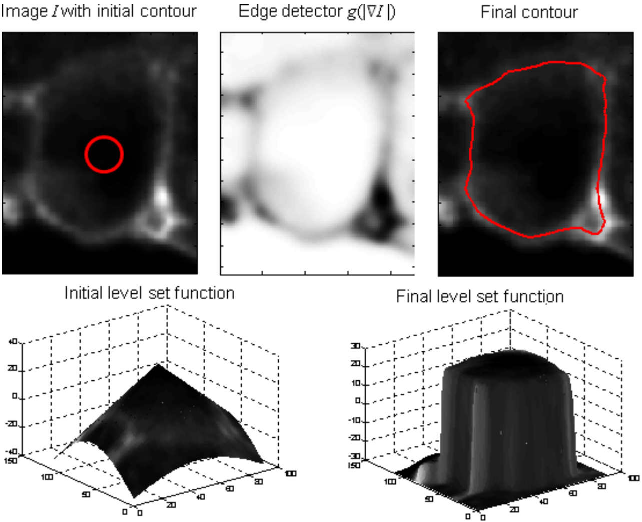


Image I with initial contour
Edge detector $g(|\nabla I|)$
Final contour
Initial level set function
Final level set function

BIBLIOGRAPHY

Bibliography

- 'Level set Methods and Dynamic Implicit Surfaces', S. Osher and R. Fedkiw
- www.math.ucla.edu/~sjo/
- www.jarnoralli.com

THANK YOU!