

# A Summary of Implicit Snake Formulations

Yongjik Kim

## Abstract

This is a simple summary of implicit snake formulations to show (hopefully) every different formulations of implicit snake I've read so far, in a uniform framework. It's chief purpose is to remind myself, so that I won't ever worry again 'is this term correct?', 'does this  $I$  in paper A correspond to this  $\varphi$  in paper B?', or all sort of such questions. Therefore, important discussions would be frequently omitted, and mathematical derivations would be left lengthy, showing as many steps as possible. *Warning : this document is under construction.*

## 1 Formulation of 2D Implicit Snakes

The most clear formulation of implicit snakes I've found is in [Gd], and following is largely an excerpt from it. First, we will deal with 2-D case.

### 1.1 Formulation of geodesic curve

The geodesic snake  $\mathbf{C}(u) = (C_x, C_y)$  has an associated energy function to be minimized:

$$E[\mathbf{C}] = E_g[\mathbf{C}] + E_{\text{area}}[\mathbf{C}] = \oint g(\mathbf{C}) ds + \alpha \int dA,$$

where  $g$  is the metric,  $ds$  is arc length, and  $\int dA$  is total area inside  $\mathbf{C}$ . Obviously,  $E_g[\mathbf{C}]$  tries to minimize the integral of  $g$  over the curve, while  $E_{\text{area}}[\mathbf{C}]$  tries to minimize or maximize (depending on the sign of  $\alpha$ ) the total area. Here we are assuming that  $\mathbf{C}$  is a closed simple curve over a plane. Now we calculate  $\delta E[\mathbf{C}]$ .

$$E_g[\mathbf{C}] = \oint g(\mathbf{C}) ds = \oint g(\mathbf{C}) |\mathbf{C}'| du,$$

where  $u$  is an arbitrary parametrization of  $\mathbf{C} = \mathbf{C}(u)$ , and  $\mathbf{C}' = d\mathbf{C}/du$ . Therefore,

$$\begin{aligned} \delta E_g[\mathbf{C}] &= \oint \delta g(\mathbf{C}) |\mathbf{C}'| du + \oint g(\mathbf{C}) \delta |\mathbf{C}'| du \\ &= \oint \nabla g \cdot \delta \mathbf{C} |\mathbf{C}'| du + \oint g(\mathbf{C}) \frac{\mathbf{C}' \cdot \delta \mathbf{C}'}{|\mathbf{C}'|} du \\ &= \oint \nabla g \cdot \delta \mathbf{C} ds + \left[ \left( g(\mathbf{C}) \frac{\mathbf{C}'}{|\mathbf{C}'|} \right) \cdot \delta \mathbf{C} \right]_{\text{a cycle}} - \oint \frac{d}{du} \left( g(\mathbf{C}) \frac{\mathbf{C}'}{|\mathbf{C}'|} \right) \cdot \delta \mathbf{C} du, \end{aligned}$$

where we used integration by part, and the fact that  $\mathbf{C}$  is a closed curve. We define the local curvature as  $\kappa$ , and normal vector as  $\mathbf{N}$ . Furthermore, we assume that the curve cycles counterclockwise as parameter  $u$  cycles, and let  $\mathbf{N}$  always point to the outward direction. Finally, we define the sign of  $\kappa$  as:

$$\kappa \mathbf{N} = \frac{d^2 \mathbf{C}}{ds^2} = \frac{d}{ds} \left( \frac{du}{ds} \mathbf{C}' \right) = \frac{d}{ds} \left( \frac{\mathbf{C}'}{|\mathbf{C}'|} \right) = \frac{1}{|\mathbf{C}'|} \frac{d}{du} \left( \frac{\mathbf{C}'}{|\mathbf{C}'|} \right).$$

Therefore,  $\kappa \mathbf{N}$  always points to the concave side of the curve, i.e., towards the center of the local curvature. It has the unhappy consequence that  $\kappa$  is negative in 'ordinary' case, e.g., in a circle. But such kind of confusion is unavoidable however we define all kinds of signs.

In that case,

$$\frac{d}{du} \left( \frac{\mathbf{C}'}{|\mathbf{C}'|} \right) = (\nabla g \cdot \mathbf{C}') \frac{\mathbf{C}'}{|\mathbf{C}'|} + g(\mathbf{C}) \frac{d}{du} \left( \frac{\mathbf{C}'}{|\mathbf{C}'|} \right) = (\nabla g \cdot \mathbf{C}') \frac{\mathbf{C}'}{|\mathbf{C}'|} + g(\mathbf{C}) |\mathbf{C}'| \kappa N.$$

Therefore,

$$\delta E_g[\mathbf{C}] = \oint \left[ \nabla g - \frac{(\nabla g \cdot \mathbf{C}') \mathbf{C}'}{|\mathbf{C}'|^2} - \kappa g(\mathbf{C}) \mathbf{N} \right] \cdot \delta \mathbf{C} ds = \oint [\nabla g \cdot \mathbf{N} - \kappa g(\mathbf{C})] \mathbf{N} \cdot \delta \mathbf{C} ds,$$

where we used the fact that  $\mathbf{C}'/|\mathbf{C}'|$  and  $\mathbf{N}$  form a basis in the 2D plane.

Turning to  $E_{\text{area}}[\mathbf{C}]$ , we have

$$E_{\text{area}}[\mathbf{C}] = \alpha \int dA = \alpha \oint C_x C'_y du.$$

Therefore,

$$\begin{aligned} \delta E_{\text{area}}[\mathbf{C}] &= \alpha \left[ \oint \delta C_x C'_y du + \oint C_x \delta C'_y du \right] \\ &= \alpha \oint \delta C_x C'_y - C'_x \delta C_y du \\ &= \alpha \oint (C'_y, -C'_x) \cdot (\delta C_x, \delta C_y) du \\ &= \alpha \oint \mathbf{N} \cdot \delta \mathbf{C} |\mathbf{C}'| du = \alpha \oint \mathbf{N} \cdot \delta \mathbf{C} ds, \end{aligned}$$

where we used the definition that  $\mathbf{N}$  is parallel to the vector obtained by rotating  $\mathbf{C}'$  by 90 degrees clockwise, which is  $(C'_y, -C'_x)$ . Combining these two result, we get

$$\delta E[\mathbf{C}] = \oint (\nabla g \cdot \mathbf{N} - \kappa g(\mathbf{C}) + \alpha) \mathbf{N} \cdot \delta \mathbf{C} ds.$$

Therefore, the value of  $\delta \mathbf{C}$  that decreases  $E[\mathbf{C}]$  most rapidly is given as:

$$\frac{d\mathbf{C}}{dt} = (\kappa g(\mathbf{C}) - \nabla g \cdot \mathbf{N} - \alpha) \mathbf{N}.$$

## 1.2 Level set method : imbedding in an implicit function

If we define  $\varphi$  as an implicit function such that  $\{\mathbf{x} \mid \varphi(\mathbf{x}) = 0\} = \{\mathbf{C}(u)\}$ , and  $\varphi$  is negative inside the curve, and positive outside the curve, it is easily shown that

$$\frac{d\varphi}{dt} = -V |\nabla \varphi|, \quad \text{where} \quad \frac{d\mathbf{C}}{dt} = V \mathbf{N}.$$

Also,

$$\mathbf{N} = \frac{\nabla \varphi}{|\nabla \varphi|} \quad \text{and} \quad \kappa = -\nabla \cdot \left( \frac{\nabla \varphi}{|\nabla \varphi|} \right).$$

Therefore,

$$\frac{d\varphi}{dt} = \left[ g \nabla \cdot \left( \frac{\nabla \varphi}{|\nabla \varphi|} \right) + (\nabla g) \cdot \frac{\nabla \varphi}{|\nabla \varphi|} + \alpha \right] |\nabla \varphi|.$$

Now, there are three terms. The first is the MCF (Mean Curvature Flow) term, which smooths out the curve in the absence of boundary. The second is the  $\nabla g$  term, which attracts the curve into the boundary. The third is the  $\alpha$  term, which is a constant velocity.

Alternatively,

$$\frac{d\varphi}{dt} = \left[ \nabla \cdot \left( g \frac{\nabla \varphi}{|\nabla \varphi|} \right) + \alpha \right] |\nabla \varphi|. \quad [\text{Gd}]$$

### 1.3 Modulation of each terms

Each of these three terms can be modulated by  $g(\mathbf{C})$  to slow down curve evolution at the boundary. Furthermore, we can have  $\nabla^2\varphi$  term to smooth  $\varphi$  over time in order to avoid numerical instability. Hence we get the following formula:

$$\frac{d\varphi}{dt} = (g)^{r_{\text{MCF}}+1} \nabla \cdot \left( \frac{\nabla\varphi}{|\nabla\varphi|} \right) |\nabla\varphi| + (g)^{r_{\nabla g}} (\nabla g) \cdot \nabla\varphi + (g)^{r_c} \alpha |\nabla\varphi| + c_s \cdot (g)^{r_s} \nabla^2\varphi,$$

where we introduced four nonnegative integer constants  $r_{\text{MCF}}$ ,  $r_{\nabla g}$ ,  $r_c$ , and  $r_s$ , and one real coefficient  $c_s$ . This formula can serve as a unified framework to describe various different formulations shown in various papers.

### 1.4 Table of formulations

1. Goldenberg's version [Gd]

$$* \quad r_{\text{MCF}} : 1 \quad r_{\nabla g} : 1 \quad r_c : 1 \quad c_s : 0 \quad r_s : -$$

$$\begin{aligned} \frac{d\varphi}{dt} &= g(x, y) \left( \alpha + \operatorname{div} \left( g(x, y) \frac{\nabla\varphi}{|\nabla\varphi|} \right) \right) |\nabla\varphi| \\ &= g(x, y) (\alpha + \operatorname{div}(g\nabla\varphi)) \quad (\text{because } |\nabla\varphi| = 1) \end{aligned}$$

- uses additive operator splitting (implicit integration scheme in time)
- has discussions about multi-color images & 2D movies
- reinitialize  $\varphi$  to be distance map every step

2. Caselles' version [Ca] (named 'Sapiro version' in [Sch&Tu])

$$* \quad r_{\text{MCF}} : 0 \quad r_{\nabla g} : 0 \quad r_c : 1 \quad c_s : 0 \quad r_s : -$$

$$\frac{\partial u}{\partial t} = |\nabla u| \operatorname{div} \left( g(I) \frac{\nabla u}{|\nabla u|} \right) + cg(I) |\nabla u| \quad [\text{Ca}] \quad (9)$$

- $u = \varphi$ ,  $c = \alpha$ .
- has tons of mathematics — should read carefully later

3. 'Turello version' in [Sch&Tu]

$$* \quad r_{\text{MCF}} : 0 \quad r_{\nabla g} : 1 \quad r_c : 2 \quad c_s : 1/2^N = 1/4 \quad r_s : 2$$

$$\frac{\partial u}{\partial t} = g(I) |\nabla u| \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right) + g(I) (\nabla g(I) \cdot \nabla u) + cg(I)^2 |\nabla u| + \frac{1}{2^N} g(I)^2 \nabla^2 u \quad [\text{Sch\&Tu}] \quad (2.36)$$

4. 'Schlegel version' in [Sch&Tu]

$$* \quad r_{\text{MCF}} : 1 \quad r_{\nabla g} : 1 \quad r_c : 2 \quad c_s : 1 \quad r_s : 2$$

$$\frac{\partial u}{\partial t} = g(I) \left[ |\nabla u| \operatorname{div} \left( g(I) \frac{\nabla u}{|\nabla u|} \right) + cg(I) |\nabla u| + g(I) \nabla^2 u \right] \quad [\text{Sch\&Tu}] \quad (2.37)$$

## 1.5 Similarity between Mean Curvature Flow and Gaussian diffusion

Rather unexpectedly, MCF and Gaussian diffusion equations behave similarly. If we have  $|\nabla\varphi| = \text{const}$ , which is the ideal case, the two terms are equivalent as shown below.

$$\text{MCF} = \nabla \cdot \left( \frac{\nabla\varphi}{|\nabla\varphi|} \right) |\nabla\varphi| = \nabla \cdot \nabla\varphi = \text{Gaussian diffusion}$$

Therefore, Gaussian diffusion can be regarded as an approximation of Mean Curvature Flow. The following comparison in 2D shows that, despite their first appearance, they are indeed quite similar mathematically.

$$\begin{aligned} \text{MCF} &= \frac{\varphi_x^2 \varphi_{yy} - 2\varphi_x \varphi_y \varphi_{xy} + \varphi_y^2 \varphi_{xx}}{|\nabla\varphi|^2} \\ \text{Gaussian diffusion} &= \varphi_{xx} + \varphi_{yy} \\ &= \frac{\varphi_x^2 \varphi_{xx} + \varphi_x^2 \varphi_{yy} + \varphi_y^2 \varphi_{xx} + \varphi_y^2 \varphi_{yy}}{|\nabla\varphi|^2} \\ \text{MCF} - (\text{Gaussian diffusion}) &= -\frac{\varphi_x^2 \varphi_{xx} + 2\varphi_x \varphi_y \varphi_{xy} + \varphi_y^2 \varphi_{yy}}{|\nabla\varphi|^2} \end{aligned}$$

From this, we might conjecture that Gaussian diffusion and MCF may show numerically similar behavior. This is an important observation, because Gaussian diffusion can be made unconditionally stable by adopting “implicit scheme” [NR, Ch. 19].

## 1.6 Stability analysis of Mean Curvature Flow equation

In this section we will develop fast and stable method to evaluate MCF equation. First, we need some definitions.

We denote the value of  $\varphi$  at grid point  $(r, s)$  at time step  $n$  by  $\varphi_{r,s}^{(n)}$ . Its derivatives are calculated as shown below.

$$\begin{aligned} \varphi &= \varphi_{r,s}^{(n)} \\ \varphi_x &= \varphi_{x; r,s}^{(n)} = \frac{\varphi_{r+1,s}^{(n)} - \varphi_{r-1,s}^{(n)}}{2\Delta x} \\ \varphi_{xx} &= \varphi_{xx; r,s}^{(n)} = \frac{\varphi_{r-1,s}^{(n)} - 2\varphi_{r,s}^{(n)} + \varphi_{r+1,s}^{(n)}}{(\Delta x)^2} \\ \varphi_{xy} &= \varphi_{xy; r,s}^{(n)} = \frac{\varphi_{r-1,s-1}^{(n)} - \varphi_{r-1,s+1}^{(n)} - \varphi_{r+1,s-1}^{(n)} + \varphi_{r+1,s+1}^{(n)}}{4(\Delta x)^2} \quad (\Delta x = \Delta y) \end{aligned} \tag{1}$$

(From now on, we will assume that  $\Delta x = \Delta y = 1$ , unless otherwise noted.)

In FTCS (Forward Time, Centered Space) scheme, the MCF and Gaussian diffusion equations are respectively written as the following:

$$\begin{aligned} \text{MCF} &: \frac{\varphi_{r,s}^{(n+1)} - \varphi_{r,s}^{(n)}}{\Delta t} = \left[ \frac{\varphi_x^2 \varphi_{yy} - 2\varphi_x \varphi_y \varphi_{xy} + \varphi_y^2 \varphi_{xx}}{|\nabla\varphi|} \right]_{r,s}^{(n)} \\ \text{Gaussian diffusion} &: \frac{\varphi_{r,s}^{(n+1)} - \varphi_{r,s}^{(n)}}{\Delta t} = [\varphi_{xx} + \varphi_{yy}]_{r,s}^{(n)} \end{aligned}$$

Unfortunately, the Gaussian diffusion becomes unstable when  $\Delta t > 1/2$  [NR, Ch. 19], and the MCF equations behaves no better empirically. Therefore, we need other formulations to solve MCF equation.

It is not easy to analyze stability of MCF equation, especially because it is nonlinear. But we can do a quick hack and make a simplified equation that is “*at least as good as*” MCF equation. That is, if the simplified version is unstable, MCF is almost certain to be unstable. Moreover, experiment shows that when the simplified version is stable, MCF is usually also stable. The simplified version is:

$$\frac{\partial \varphi}{\partial t} = A\varphi_{xx} + B\varphi_{xy} + C\varphi_{yy}, \quad \text{where } A, B, \text{ and } C \text{ are constants} \quad (2)$$

This is indeed a radical simplification, and as far as I know, it can be justified only empirically, not theoretically. Note that this simplified equation is used only for stability analysis. Once we do stability analysis, we create corresponding numerical method for MCF equation, and see if it behaves as predicted by analysis of (2).

In general, equation (2) is not even analytically stable. But if it is to simulate MCF equation, we have constraints on its coefficients:

$$A = \frac{\varphi_y^2}{|\nabla \varphi|^2}, \quad B = -\frac{2\varphi_x \varphi_y}{|\nabla \varphi|^2}, \quad C = \frac{\varphi_x^2}{|\nabla \varphi|^2}.$$

Hence,

$$B^2 - 4AC = 0 \quad \text{and} \quad A^2 + C^2 = 1.$$

Therefore, the coefficients can be parametrized by a single parameter  $\theta$ :

$$A = \sin^2 \theta, \quad B = -2 \cos \theta \sin \theta, \quad C = \cos^2 \theta. \quad (3)$$

In real MCF equation,  $\theta$  is defined as  $\tan^{-1} \varphi_y / \varphi_x$ , i.e., the direction of  $\nabla \varphi$ . In (2), this constraint represents what kind of values of  $A$ ,  $B$ , and  $C$  can be regarded as a simplified possible case of MCF equation. (Remember, there is no mathematical justification.) If we impose this constraint, equation (2) does become analytically stable. Since the equation is linear, we can analyze numerical stability by analyzing behavior of single sine wave [NR, Ch. 19].

We define  $\varphi$  as:

$$\varphi_{r,s}^{(n)} = \xi^n e^{i(k_1 r + k_2 s)}.$$

A numerical method is stable only if  $|\xi| \leq 1$ . From this and (1), we have

$$\begin{aligned} \varphi_{xx; r,s}^{(n)} &= \frac{e^{ik_1} + e^{-ik_1} - 2}{(\Delta x)^2} \varphi_{r,s}^{(n)} = \frac{2}{(\Delta x)^2} (\cos k_1 - 1) \varphi_{r,s}^{(n)}, \\ \varphi_{xy; r,s}^{(n)} &= \frac{(e^{ik_1} - e^{-ik_1})(e^{ik_2} - e^{-ik_2})}{4(\Delta x)^2} \varphi_{r,s}^{(n)} = -\frac{1}{(\Delta x)^2} (\sin k_1)(\sin k_2) \varphi_{r,s}^{(n)}, \\ \varphi_{yy; r,s}^{(n)} &= \frac{e^{ik_2} + e^{-ik_2} - 2}{(\Delta x)^2} \varphi_{r,s}^{(n)} = \frac{2}{(\Delta x)^2} (\cos k_2 - 1) \varphi_{r,s}^{(n)}. \end{aligned}$$

Now we list various numerical schemes and evaluate their stability.

### 1. fully explicit scheme

$$\begin{aligned} \frac{\varphi_{r,s}^{(n+1)} - \varphi_{r,s}^{(n)}}{\Delta t} &= \left[ A\varphi_{xx}^{(n)} + B\varphi_{xy}^{(n)} + C\varphi_{yy}^{(n)} \right]_{r,s} \\ \xi &= 1 + \frac{\Delta t}{(\Delta x)^2} [2A(\cos k_1 - 1) - B \sin k_1 \sin k_2 + 2C(\cos k_2 - 1)] \\ &= 1 + \frac{\Delta t}{(\Delta x)^2} X \end{aligned}$$

Although it is possible to analytically find maxima and minima of  $\xi(k_1, k_2, \theta)$  for a given value of  $\Delta t$ , it is rather tedious. A better solution is to use matlab and draw how  $\xi$  depends on these variables. (We assume  $\Delta x = 1$ .) In this way, it is easily shown that  $-4 \leq X \leq 0$ . Therefore this method is stable (i.e.,  $|\xi| \leq 1$ ) only if  $\Delta t \leq 1/2$ .

2. 2nd order implicit scheme : impilcit in  $\varphi_{xx}$ ,  $\varphi_{xy}$ , and  $\varphi_{yy}$

$$\begin{aligned} \frac{\varphi_{r,s}^{(n+1)} - \varphi_{r,s}^{(n)}}{\Delta t} &= \left[ A\varphi_{xx}^{(n+1)} + B\varphi_{xy}^{(n+1)} + C\varphi_{yy}^{(n+1)} \right]_{r,s} \\ \xi &= \frac{1}{1 - \frac{\Delta t}{(\Delta x)^2} [2A(\cos k_1 - 1) - B \sin k_1 \sin k_2 + 2C(\cos k_2 - 1)]} \\ &= \frac{1}{1 - \frac{\Delta t}{(\Delta x)^2} X} \end{aligned}$$

Since  $-4 \leq X \leq 0$ , this scheme is unconditionally stable. Unfortunately, it requires solving a system of linear equations over the whole mesh. Although the matrix is very sparse, it is still computationally expensive. Hence we seek for other unconditionally stable schemes.

3. semi-implicit scheme : impilcit in  $\varphi_{xx}$  and  $\varphi_{yy}$

$$\begin{aligned} \frac{\varphi_{r,s}^{(n+1)} - \varphi_{r,s}^{(n)}}{\Delta t} &= \left[ A\varphi_{xx}^{(n+1)} + B\varphi_{xy}^{(n)} + C\varphi_{yy}^{(n+1)} \right]_{r,s} \\ \xi &= \frac{1 - \frac{\Delta t}{(\Delta x)^2} B \sin k_1 \sin k_2}{1 - \frac{\Delta t}{(\Delta x)^2} [2A(\cos k_1 - 1) + 2C(\cos k_2 - 1)]} \end{aligned}$$

This scheme is also unconditionally stable, as shown in the following plot of the range of  $\xi$  for  $\Delta t = 2$  case. Unfortunately, this scheme still requires full size sparse matrix. But soon we will show that we can split operators in  $x$  and  $y$  directions and get stable scheme.

4. 2nd order implicit scheme with additive operator splitting

Since it is computationally expensive to solve a system of linear equations with size of the whole mesh, we must devise another approximation. The 2nd order implicit scheme uses 8 adjacent mesh points to calculate next time step. If we decompose it into 4 different operators, each using only 2 adjacent mesh points (respectively,  $x$  direction,  $y$  direction, and two diagonal directions), then we have 4 operators, each being represented by tridiagonal matrix. Then we average the result of 4 operators to get final result.

$$\begin{aligned} \frac{\varphi_{r,s}^{(n+1,1)} - \varphi_{r,s}^{(n)}}{\Delta t} &= 4A\varphi_{xx; r,s}^{(n+1,1)} \\ &= \frac{8A}{(\Delta x)^2} (\cos k_1 - 1) \varphi_{r,s}^{(n+1,1)} \\ \frac{\varphi_{r,s}^{(n+1,2)} - \varphi_{r,s}^{(n)}}{\Delta t} &= 4C\varphi_{yy; r,s}^{(n+1,2)} \\ &= \frac{8C}{(\Delta x)^2} (\cos k_2 - 1) \varphi_{r,s}^{(n+1,2)} \\ \frac{\varphi_{r,s}^{(n+1,3)} - \varphi_{r,s}^{(n)}}{\Delta t} &= 4B \frac{[\varphi_{r-1,s-1} - 2\varphi_{r,s} + \varphi_{r+1,s+1}]^{(n+1,3)}}{4(\Delta x)^2} \\ &= \frac{2B}{(\Delta x)^2} (\cos(k_1 + k_2) - 1) \varphi_{r,s}^{(n+1,3)} \end{aligned}$$

$$\begin{aligned}
\frac{\varphi_{r,s}^{(n+1,4)} - \varphi_{r,s}^{(n)}}{\Delta t} &= -4B \frac{[\varphi_{r-1,s+1} - 2\varphi_{r,s} + \varphi_{r+1,s-1}]^{(n+1,3)}}{4(\Delta x)^2} \\
&= -\frac{2B}{(\Delta x)^2} (\cos(k_1 - k_2) - 1) \varphi_{r,s}^{(n+1,4)} \\
\varphi_{r,s}^{(n+1)} &= \frac{1}{4} \left[ \varphi_{r,s}^{(n+1,1)} + \varphi_{r,s}^{(n+1,2)} + \varphi_{r,s}^{(n+1,3)} + \varphi_{r,s}^{(n+1,4)} \right]
\end{aligned}$$

---

Unfortunately, this scheme is numerically *unconditionally unstable*, as shown in the following figure. (Note that  $\xi$  can be larger than 1.) Therefore, we need another scheme that can use tridiagonal matrix.

## References

- [Gd] Roman Goldenberg, Ron Kimmel, Ehud Rivlin, Michael Rudzsky / Fast Geodesic Active Contours
- [Ca] Vincent Caselles, Ron Kimmel, Guillermo Sapiro / Geodesic Snakes
- [Sch&Tu] Joachim Schlegel, Silvio Turello / Bildsegmentierung mit deformierbaren Modellen veränderlicher Topologie
- [NR] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery / Numerical Recipes in C (2nd Ed.) / Cambridge University Press



