# LAB REPORT: LAB 5

TNM079, MODELING AND ANIMATION

Viktor Sjögren
viksj950@student.liu.se

Sunday 30th May, 2021

## Abstract

We discuss level sets by formulating PDEs alongside implicit surface representations. Level sets allows for surfaces to be more dynamic by it being able to change in time and space. In this lab paper, we cover tasks that involves differential equations that allows for modification of an implicit surface in a grid space. Level set refinement through reinitilization is also looked at, and operations such as erosion and dilation is also implemented and applied to implicit surfaces to see the effects.

## 1 Background

In computer graphics, implicit surface representations can for example serve as a more reliable representation in physical simulations then what an explicit surface can offer. Furthermore if we incorporate partial differential equations to implicit surfaces, deformations, smoothing and other morphological operations can be applied to modify the surface in regards to time or space. Solving these PDEs makes the implicit surface a *level set*.

The equations that are derived from the level set surface are divided into time (temporal) and space (spatial) discretization parts. The temporal discretization can be computed to understand how the surface moves or evolves with a time step. Spatial discretization can be used to calculate properties such as curvature and gradients with space in mind.

The way a level set is defined are as follows:

$$S = \{x \in \mathbb{R}^d : \phi(x) = h\} \tag{1}$$

where S represents the interface of the level set, and $\phi$ the level set function. Through this we can also identify points on the inside or outside of the level $S$, by looking at values that fullfill $\phi(x) < h$ and $\phi(x) > h$ we define points inside or outside $S$ respectively.

To enable deformation operations on the implicit surface, time dependence has to be taken to account in equation (1). One method that doesn't bring with it to many constraints is to make the level set function vary, in regards to time. This means we define the level set as:

$$S(t) = \{x \in \mathbb{R}^d : \phi(x,t) = h\} \tag{2}$$

Often, with this equation the value of $h$ is kept to be 0, so that the description of inside or outside of $S$ is sign based. The motion of S can then be derived and represented as:

$$\frac{\partial \phi}{\partial t} = -\nabla\phi \cdot \frac{d\alpha}{dt} = -F|\nabla\phi| \tag{3}$$

where $\alpha$ correspond to a point located on $S$, and F represents the speed function of the level set. The speed in this case goes in the direction of the normal at the point $\alpha$.

As mentioned before, the level set equation is divided into two discretization parts handling time and space. The temporal part can be evolved through explicit or implicit schemes, however due to implicit schemes being computationally heavy, explicit schemes which are more simple to implement are often used instead. One such scheme is the forward Euler, which can be used at first order, or by using TVD Runge Kutta to higher orders to improve the accuracy.

The spatial discretization part can be represented by two PDEs, either by hyperbolic advection or parabolic diffusion. With the hyperbolic type equation (3) becomes:

$$\frac{\partial \phi}{\partial t} = -\boldsymbol{V} \cdot \nabla \phi = -F|\nabla \phi| \tag{4}$$

where $\boldsymbol{V}$ correspond to a vector field in which the level set can be moved in. The approximation for the PDE is then given by points which have not been reached yet which is also stated in CFL stability condition. Using an up-wind scheme serves this purpose and generates:

$$\frac{\partial \phi}{\partial t} \approx \{\phi_x^+ = (\phi_{i+1,j,k} - \phi_{i,j,k})/\Delta x\} \tag{5}$$

$$\frac{\partial \phi}{\partial t} \approx \{\phi_x^- = (\phi_{i,j,k} - \phi_{i-1,j,k})/\Delta x\} \tag{6}$$

where (5) is used when we are going from one direction, $V_x < 0$, and (6) in the opposite where $V_x > 0$. Because we are using an explicit method, the CFL stability condition requires the speed along the x-axis to be at least matching the speed of the physical wave $\boldsymbol{V}$ or $F$ in equation (4). This means that the time step has to fulfill:

$$\begin{gathered} \Delta t < \min\{\frac{\Delta x}{|V_x|}, \frac{\Delta y}{|V_y|}, \frac{\Delta z}{|V_z|}\}, \\ \Delta t < \min\{\frac{\Delta x, \Delta y, \Delta z}{|F|}\} \end{gathered} \tag{7}$$

If we instead rewrite equation (3) as parabolic diffusion, we obtain a version where the curvature $k$ is present. The equation becomes:

$$\frac{\partial \phi}{\partial t} = \alpha k |\nabla \phi| \tag{8}$$

where $\alpha$ is parameter that scales the curvature. Different to the hyperbolic advection method, there is no direction of the flow taken into account. Instead of using a upwind scheme, in this case a central difference scheme is used:

$$\frac{\partial \phi}{\partial t} \approx \phi_x^{+-} = \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{2\Delta x} \tag{9}$$

For second order the discretization becomes:

$$\frac{\partial^2 \phi}{\partial^2 x} \approx \frac{\phi_{i+1,j,k} - 2\phi_{i,j,k} + \phi_{i-1,j,k}}{\Delta x^2} \tag{10}$$

$$\frac{\partial^2 \phi}{\partial x \partial y} \approx \frac{\phi_{i+1,j+1,k} - \phi_{i+1,j-1,k} + \phi_{i-1,j-k,k} - \phi_{i-1,j+1,k}}{4\Delta x \Delta y} \tag{11}$$

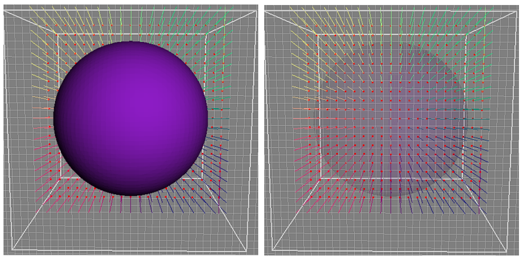
The first task involved implementing all the 15 differential equations in *Levelset.cpp*. This meant using the equations (5), (6), (9), (10) and (11) for each of the combinations of positive, negative, $x, y$ and $z$. When working with these equations all the access to points in the grid was done in object space $i$, $j$and $k$.

The second task was done through solely visual observation about the effects of reinitialization. The third task made it possible to perform erosion and dilation morphing. To do this, a stable timestep is required which meant equation (7) had to be and was therefore implemented in *OperatorDilateErode.h*. Along with this, an evaluation function was implemented which returned the speed/rate of change at any point. To compute this, Goudunov up-winding scheme was used specifically for the squared gradients.

A fourth task was done that performed advection in the class *OperatorAdvect.h*. Again, stability criterion from equation (7) had to be fullfilled for the time step. To make advection possible, the equation presented in (4) with the vector field $\boldsymbol{V}$ was used. As explained earlier, an upwind scheme as presented in (5) and (6) was used to discretize the equation.

## 2 Results

First we look at at conversion of an implicit surface to a level set, which can be seen in Figure 1. In 1(a) the gradient extends from the surface the expected way, in 1(b) the inside of the surface gradients are visible and it is seen for the middle point the gradient is close to zero.



(a) (b)

*Figure 1:* Implicit surface (sphere) that is converted to a level set using the PDEs which samples points of the surface in a grid which can be modified.

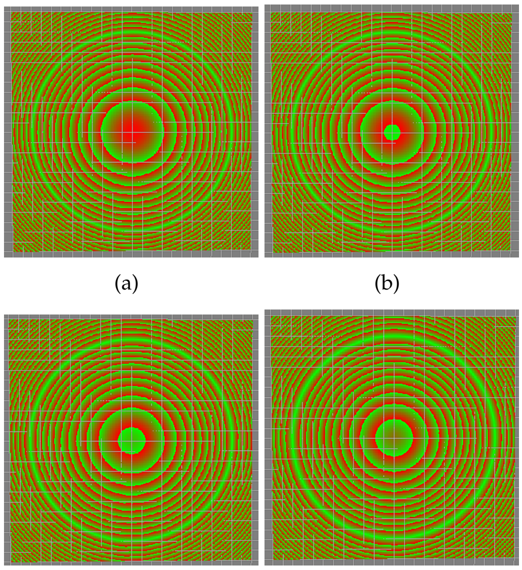
Visual observations of an implicit mesh that has been converted to a level set can be seen in Figure 2. As before, a implicit sphere has been converted, and set to a low opacity value so that the zero level set and all other possible level sets are visible. The images in 1 shows different reinitilization states, of which tries to maintain a signed distance (gradient function equal to one) field making each red and green line be uniform in width.

The results of the morphological operators *Dilation* and *Erode* is displayed in Figure 3.

Looking at the advection implementation, which the results of can be seen in Figure 4, the used vector field ***V*** is of an vortex. This makes the plane have the shape that is seen in Figure 4 where it has pinch point around the middle while the edges fold or swirl around this point.

## 3 Lab partner and grade

The lab was made in collaboration with Algot Sandahl. The report scope and content is aimed at grade **3**.

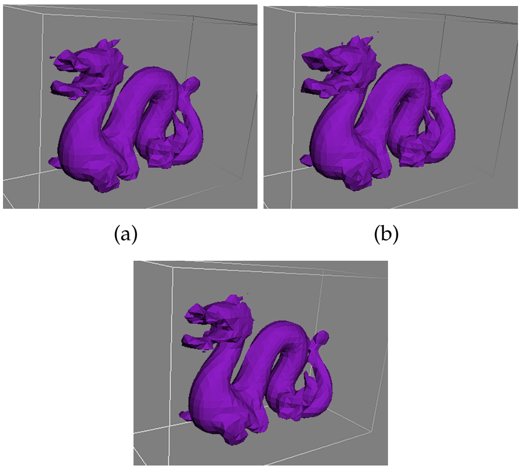


(a) (b) (c) (d)

*Figure 2:* Implicit surface (sphere) converted to a level set. A scalar cut plane is cut through the sphere and with visualisation mode *Iso countour* each level set is visualised. In the four images the level set has been reinitialized up to 8 times. In 2(a) no reinitilization is done, in 2(b) 2 reinitilization steps have been performed and 4 and 8 steps in 2(c) and 2(d) respectively.

## References

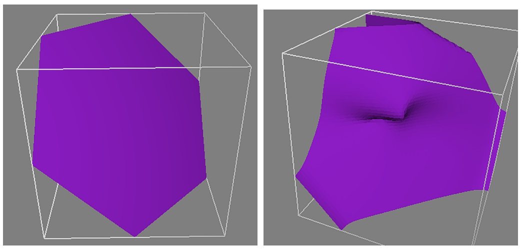
[1] M.E. Dieckmann. *Lectures 10-12* , tnm079, 2021.



(a) (b)

(c)

*Figure 3:* Implicit surface (Dragon) converted to a level set. 3(a) shows the surface unmodified, 3(b) shows the surface dilated twice and 3(c) shows the erode operator applied twice.



(a) (b)

*Figure 4:* Implicit surface (Plane) converted to a level set. 4(a) shows the surface unmodified, 4(b) shows the surface after 50 advection steps with the vortex field. (Disregard the cutting caused by the boundary box)