LAB REPORT: LAB 5

TNM079, MODELING AND ANIMATION

Alma Linder
almli825@student.liu.se

Friday 2nd May, 2025

Abstract

This laboratory work is about level set surfaces and how these are constructed. The work showcases how this kind of surface can be moved and altered in time and space using vector fields, solving partial derivatives. Operators such as dilation, erosion, advection and smoothing are implemented, and the use for reinitialization is highlighted. Applying dilation and erosion to level set surfaces expand respectively shrink the objects and thus serves as a way to mend holes or smoothen out spikes on surfaces. By advection it is possible to move surfaces either uniformly with a uniform vector field, or twist them with vortex-like vector fields. Mean curvature calculations allow for significant smoothing of a surface. Finally, the effects of reinitialization show how cycles of level sets in space are altered when ensuring stability.

This laboratory work aims for `grade 4`.

1 Background

Level set surfaces are the infinite amount of surfaces generated by implicit surfaces' equations ϕ. An iso value h equaling zero is commonly chosen so that an in- and outside can be defined ($\phi < 0$ and $\phi > 0$ respectively); this is the zero-level set surface. To allow this surface to move, the implicit function $\phi(x, y, z)$ is given a time dependency t, resulting in $\phi(x, y, z, t)$. The motion of the surface is thus governed by partial derivatives of ϕ with respect to both space and time.

To not get ahead of ourselves, the following must first be established: Since ϕ is defined everywhere in space - it is Lipschitz continuous - the surface will be infinitely thin or thick. It is therefore not possible to move the surface directly by altering ϕ as time evolves. By applying a (in this lab uniform) grid in space, and only using the grid line intersection points as concrete index (integer) points for the surface, a spatial discretization is achieved. The grid is illustrated in Figure 1, where it becomes evident that all surface points between grid line intersections must be found via interpolation. A simplified way of mapping surface points (x, y, z) to the closest grid point (i, j, k) was implemented by using standard rounding. Figure 1 also shows that a conversion of the surface points (x, y, z) from world space to grid space (i, j, k) is necessary first and foremost.

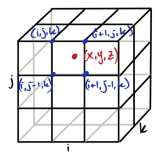


Figure 1: An excerpt of space of ϕ on which a uniform grid has been applied. Uniform means equal spacing between grid lines. A surface point (x, y, z) was mapped to the closest grid point $(i + 1, j, k)$ with standard rounding.

Actually, moving the surface can also introduce an external vector field $\mathbf{V}$. For a single dimension this concept is sketched out in Figure 2. It is possible to calculate the affects and

thus movement of the surface from the discrete grid points which $\mathbf{V}$ has visited. For example, if $\mathbf{V} > 0$, points $(i-1, j, k)$ and (i, j, k) bear historical information on how $\mathbf{V}$ has interacted with the surface.

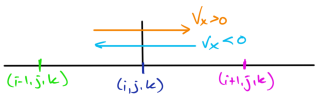


Figure 2: Vector field $\mathbf{V}$ allows a surface to move along $\mathbf{V}$. Both positive and negative direction of $\mathbf{V}$ are shown. Only historical discrete grid points can be considered when abiding to the CFL stability criterion.

Now that the spatial discretization has been obtained, is it appropriate to define the aforementioned partial derivatives that evaluate the rate of change of ϕ as the surface moves over time in space. Because there are 15 partial derivatives of ϕ in 3D for all possible directions and dimensions, these are presented separately in Appendix A. They were derived from the Courant-Freidrichs-Lewy (CFL) stability condition, which states that only visited points hold information from $\mathbf{V}$ or F are useful. $\mathbf{V}$ is an external vector field and F is the vector field of the surface. Using future points lead to unstable evolution of ϕ as there is no meaningful knowledge there. Since the grid is uniform the distances between grid points in all three directions are the same, and thus in Equations 6-8 Δy and Δz have been replaced by Δx.

As time is also continuous, it must be discretized into time steps Δt. Lingering on stability, Δt must fulfill the requirements in Equation 1. The first case is for when the direction of the vector flow is known, and the second case is for when the vector field direction is unknown. The grid spacings Δy and Δz could in practice be replaced by Δx since the grid spacing is uniform in all directions, leading to Equation 2 being implemented. If the curvature κ is considered for the change of ϕ rather than information flow along a vector field, the time constrain changes to Equation 3, assuming a uniform grid and α being the scaling parameter.

$$\begin{cases} \Delta t < \min\left(\frac{\Delta x}{|V_x|}, \frac{\Delta y}{|V_y|}, \frac{\Delta z}{|V_z|}\right) \\ \Delta t < \frac{\min(\Delta x, \Delta y, \Delta z)}{|F|} \end{cases} \tag{1}$$

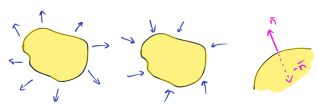
$$\begin{cases} \Delta t < \min\left(\frac{\Delta x}{|\mathbf{V}|}\right) \\ \Delta t < \frac{\min(\Delta x)}{|F|} \end{cases} \tag{2}$$

$$\Delta t < \frac{\Delta x^2}{6\alpha} \tag{3}$$

Should these time constraints fail, the evolution of ϕ will become unstable. Figuratively speaking, this means stepping over the grid in larger steps than there is new information added from the vector fields or curvature. This instability can manifest as numerical artifacts like oscillations or loss of interface definition for instance.

To ensure that ϕ remains stable as errors from both the spatial and temporal approximations via discretization are introduced, reinitialization with either marching- or sweeping algorithms can be implemented. The reinitialization ensures that the zero-level set surface maintains the properties of a signed distance function; namely that ϕ satisfies the Eikonal equation $|\nabla\phi| = 1$. If the surface change is unitary - i.e the greatest change of ϕ is controlled and uniform for fixed time steps Δt, then accuracy is preserved as errors are not allowed to accumulate over time or space.

Once the temporal and spatial discretization had been achieved in a stable way, surface modification by moving it could finally be implemented. The first surface operators to be implemented was dilation (expanding the surface) and erosion (shrinking the surface) - refer to Figure 3 - given by Equation 4a, where F is the surface speed vector field.



(a) Dilation (b) Erosion (c) Normal Directions

Figure 3: Dilation and erosion by moving the surface either with or directly against its normal.

Because these two operations move the surface in parallel with its own normal, the information about an external vector field $\mathbf{V}$ is cut off. Therefore, the second time constrain in Equation 2 was used, ensuring the `min` condition with a multiplication of 0.9. To accurately evaluate the partial derivatives for the gradient of ϕ the Godunov method was used for each surface point (i, j, k), as this method is a stable centered upwind scheme to solve partial differential equations.

Advection is the operation that moves a surface in space, where the movement is dictated by a constant, external vector field $\mathbf{V}$. The advection equation is given in Equation 4b. Since the external field $\mathbf{V}$ was known in this case, the first time limitation in Equation 2 was selected. Because $\mathbf{V}$ is external, the points (i, j, k) which $\mathbf{V}$ affect had to be converted to world space first. Then the upwind scheme for first order partial derivatives illustrated in Figure 2 and Equations 6a, 6b, 7a, 7b, 8a and 8b were evaluated for the gradient of ϕ.

Lastly, Equation 4c was used for the smoothing operation. α scales the curvature's influence on the smoothening effects. κ is the mean surface curvature, shown in Equation 5, where the partial derivatives correspond to the notation of Equations 6 to 8 in the Appendix.

$$\frac{\partial \phi}{\partial t} = -F|\nabla \phi| \tag{4a}$$

$$\frac{\partial \phi}{\partial t} = \mathbf{V} \cdot \nabla \phi \tag{4b}$$

$$\frac{\partial \phi}{\partial t} = \alpha \kappa |\nabla \phi| \tag{4c}$$

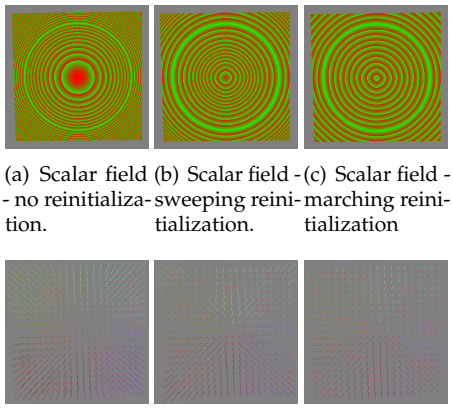
$$\begin{aligned} \kappa = & \frac{\phi_x^2(\phi_{yy} + \phi_{zz}) - 2\phi_y\phi_z\phi_{yz}}{2(\phi_x^2 + \phi_y^2 + \phi_z^2)^{3/2}} \\ & + \frac{\phi_y^2(\phi_{xx} + \phi_{zz}) - 2\phi_x\phi_z\phi_{xz}}{2(\phi_x^2 + \phi_y^2 + \phi_z^2)^{3/2}} \\ & + \frac{\phi_z^2(\phi_{xx} + \phi_{yy}) - 2\phi_x\phi_y\phi_{xy}}{2(\phi_x^2 + \phi_y^2 + \phi_z^2)^{3/2}} \end{aligned} \tag{5}$$

2 Results

Figure 4 shows the effects of applying reinitialization of a unit sphere. The time amounted to 0.05, and the number of iterations was eight for all tests in the figure. The first row depicts the sphere's scalar plane in iso colors so that each red-green ring is one iso-surface in the space the sphere is defined in. Visibly, the sphere is located at the thick outer green ring in Figure 4(a) as cycles shift direction at this zero-level due to the definition of an in- and outside of this level surface.

In Figure 4(b) fast sweeping reinitialization was applied. Evidently, this reinitialization method accurately finds a neighborhood to the sphere surface and boosts stability and accuracy in that region, as the green zero-level ring becomes sharper and thicker. The in- and exteriors far away from the surface become minimized.

In Figure 4(c) a fast marching approach was used instead. Clearly, this algorithm manages to apply level-set equation recalculation on all of the level sets uniformly. In most applications working with only a proximity to the zero-level surface is enough.



(a) Scalar field - no reinitialization. (b) Scalar field - sweeping reinitialization. (c) Scalar field - marching reinitialization

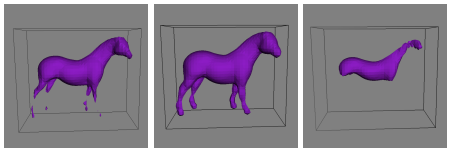
(d) Vector field - no reinitialization. (e) Vector field - sweeping reinitialization. (f) Vector field - marching reinitialization.

Figure 4: Results from sweeping- and marching-based reinitialization on a unit sphere and its scalar and vector fields.

The second row of Figure 4 shows how the sphere's positively one-sided vector field changes as the two reinitialization methods are deployed. Notably, the field vector lengths remain uniformly long for the fast marching

approach in 4(f), whereas the vectors not in close proximity to the sphere grow more for the sweeping technique. This again exemplifies the ability the sweeping method has to only control a neighborhood around the surface.

Here it is also noted that the gradient at the center of the sphere is zero if a centered differentiation scheme is utilized. When the one-sided scheme in Figure 4 is implemented, the center gradient is slightly larger than zero and skewed towards the on-sided direction. This is because only sampling information in one direction introduces a bias as not the full sphere symmetry is captured.

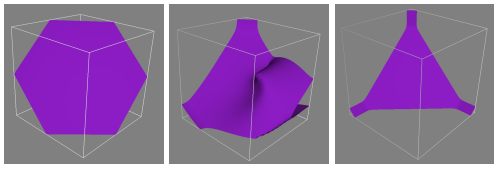


(a) Original horse. (b) Horse dilated. (c) Horse eroded.

Figure 5: Dilation and erosion results on a horse level set surface.

Figure 5 portrays a horse level set surface. Initially, the original horse surface shown in 5(a) underwent dilation for a duration of 0.02 time units over three iterations, resulting in the shape depicted in 5(b). This dilation process clearly expands the surface outwards, which effectively fuses gaps to disconnected components with the main body.

Following dilation, the surface was eroded over 0.03 time units across five iterations, producing the result shown in 5(c). Although the erosion is a bit aggressive, it successfully shrinks the object while still preserving key structural features of the original geometry, particularly around the back. This sequence of Figure 5 demonstrates how dilation can enhance connectivity and continuity in fragmented regions, while erosion can be used to restore finer detail or thin out over-expanded parts.

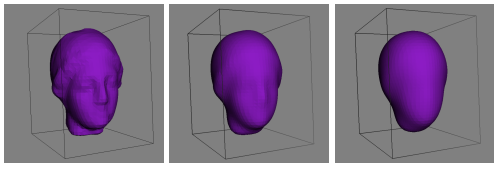


(a) Original plane. (b) Plane advection - vortex. (c) Plane advection - uniform.

Figure 6: Advection results on a plane level set surface using a vortex vector field.

Applying a vortex-like external vector field to a level set plane ($x + y + z = 0$) in Figure 6 illustrates how the surface shape can be modified in a controlled way - in this case spiraling the surface in the same directions as the vortex vector field's directions. In Figure 6(b) the duration was set to 0.03 and run 50 times.

Allowing a uniform vector field (simply $\{$`-1,-1,-1`$\}$) affect the plane instead pushes the plane uniformly in that direction in space, as Figure 6(c) shows, which is to be expected since all surface points are advected identically, resulting in a rigid shift of the entire level set. The parts of the plane leaving the bounding box in which the plane is defined are cut off, and hence the triangular shape.



(a) Original bust. (b) Bust smoothing 0.01, one iteration. (c) Bust smoothing, 0.01, another two iterations.

Figure 7: Results of smoothening on a head level set surface using mean curvature.

Finally, the results of smoothening are displayed in Figure 7, where it is evident that the level set surface gradually becomes smoother (a time of 0.01 was applied once in 7(b), and 0.01 was applied another two times in 7(c)). As for the dilation and erosion, the main features of the surface shape are preserved. This demonstrates how mean curvature flow can

effectively regularize a surface without ruining its overall structure. The extent of smoothing can thus be precisely controlled by adjusting the time step and number of iterations, offering a flexible trade-off between detail preservation and enforcing surface regularity.

A Discrete partial derivatives

$$\phi_x^+ = \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\Delta x} \tag{6a}$$

$$\phi_x^- = \frac{\phi_{i,j,k} - \phi_{i-1,j,k}}{\Delta x} \tag{6b}$$

$$\phi_x^\pm = \frac{\phi_{i+1,j,k} - \phi_{i-1,j,k}}{2\Delta x} \tag{6c}$$

$$\phi_{xx} = \frac{\phi_{i,j,k+1} - 2\phi_{i,j,k} + \phi_{i,j,k-1}}{\Delta x^2} \tag{6d}$$

$$\phi_{xy} = \frac{\phi_{i+1,j+1,k} - \phi_{i+1,j-1,k} + \phi_{i-1,j-1,k} - \phi_{i-1,j+1,k}}{4\Delta x^2} \tag{6e}$$

$$\phi_y^+ = \frac{\phi_{i,j+1,k} - \phi_{i,j,k}}{\Delta x} \tag{7a}$$

$$\phi_y^- = \frac{\phi_{i,j,k} - \phi_{i,j-1,k}}{\Delta x} \tag{7b}$$

$$\phi_y^\pm = \frac{\phi_{i,j+1,k} - \phi_{i,j-1,k}}{2\Delta x} \tag{7c}$$

$$\phi_{yy} = \frac{\phi_{i,j,k+1} - 2\phi_{i,j,k} + \phi_{i,j,k-1}}{\Delta x^2} \tag{7d}$$

$$\phi_{yz} = \frac{\phi_{i,j+1,k+1} - \phi_{i,j+1,k-1} + \phi_{i,j-1,k-1} - \phi_{i,j-1,k+1}}{4\Delta x^2} \tag{7e}$$

$$\phi_z^+ = \frac{\phi_{i,j,k+1} - \phi_{i,j,k}}{\Delta x} \tag{8a}$$

$$\phi_z^- = \frac{\phi_{i,j,k} - \phi_{i,j,k-1}}{\Delta x} \tag{8b}$$

$$\phi_z^\pm = \frac{\phi_{i,j,k+1} - \phi_{i,j,k-1}}{2\Delta x} \tag{8c}$$

$$\phi_{zz} = \frac{\phi_{i,j,k+1} - 2\phi_{i,j,k} + \phi_{i,j,k-1}}{\Delta x^2} \tag{8d}$$

$$\phi_{xz} = \frac{\phi_{i+1,j,k+1} - \phi_{i+1,j,k-1} + \phi_{i-1,j,k-1} - \phi_{i-1,j,k+1}}{4\Delta x^2} \tag{8e}$$