Smoothed Particle Hydrodynamics

A short Introduction to Principles and Ideas

Kenny Erleben
kenny@diku.dk

Department of Computer Science
University of Copenhagen

2010

Starting Point is Interpolation

The Integral Interpolant of any function $A(\mathbf{x})$ (This is just a convolution/smoothing/regularization)

$$A_I(\mathbf{x}) = \int A(\mathbf{x}')W(\| \mathbf{x} - \mathbf{x}' \|, h)d\mathbf{x}'$$

Interpolating Kernel

$$\int W(\| \mathbf{x} - \mathbf{x}' \|, h)d\mathbf{x}' = 1$$

and

$$\lim_{h\to 0} W(\| \mathbf{x} - \mathbf{x}' \|, h) = \delta(\| \mathbf{x} - \mathbf{x}' \|)$$

h is the kernel support.

More Kernel Properties

Usually we want

- Rotational (Symmetry) Invariant Kernels

$$W(\mathbf{x}_j - \mathbf{x}_i, h) = W(\mathbf{x}_i - \mathbf{x}_j, h)$$

- Non-negative kernels

$$W(\| \mathbf{x} - \mathbf{x}' \|, h) \geq 0$$

Discretization

By definition

$$A_I(\mathbf{x}) = \int A(\mathbf{x}')W(\| \mathbf{x} - \mathbf{x}' \|, h)d\mathbf{x}'$$

Discretization (Simpson/mid-point rule)

$$A_I(\mathbf{x}) \approx A_S(\mathbf{x}) = \sum_j A(\mathbf{x}_j)W(\| \mathbf{x} - \mathbf{x}_j' \|, h)\Delta V_j$$

Since $\Delta V_j = m_j/\rho_j$ we have

$$A_S(\mathbf{x}) = \sum m_j \frac{A(\mathbf{x}_j)}{\rho_j} W(\| \mathbf{x} - \mathbf{x}_j' \|, h)$$

SPH in a Nutshell

Summation Interpolant

$$A_S(\mathbf{x}) = \sum_j \frac{m_j A_j}{\rho_j} W(\| \mathbf{x} - \mathbf{x}_j \|, h)$$

Gradient

$$\nabla A_S(\mathbf{x}) = \sum_j \frac{m_j A_j}{\rho_j} \nabla W(\| \mathbf{x} - \mathbf{x}_j \|, h)$$

Laplacian

$$\nabla^2 A_S(\mathbf{x}) = \sum_j \frac{m_j A_j}{\rho_j} \nabla^2 W(\| \mathbf{x} - \mathbf{x}_j \|, h)$$

The Golden Rules of SPH

According to Monaghan

First: If you want to find a physical interpretation then it is always best to assume the kernel is Gaussian

Second: Rewrite formulas with density inside operators.

Last rule may be tricky so lets see how it can be done.

Putting Density inside Operators

Let us study the pressure gradient by definition

$$\nabla P_i = \nabla P(x_i) = \sum_j m_j \frac{P_j}{\rho_j} \nabla W(\| \mathbf{x}_i - \mathbf{x}_j \|, h)$$

Non-symmetric forces. Rewrite (using differentiation rules)

$$\nabla \left(\frac{P}{\rho} \right) = \left(\frac{\nabla P}{\rho} \right) - \left(\frac{P}{\rho^2} \right) \nabla \rho$$

Then

$$\left(\frac{\nabla P}{\rho} \right) = \nabla \left(\frac{P}{\rho} \right) + \left(\frac{P}{\rho^2} \right) \nabla \rho$$

Now density is inside operators.

Putting Density inside Operators (Cont'd)

So

$$\left(\frac{\nabla P}{\rho}\right) = \nabla\left(\frac{P}{\rho}\right) + \left(\frac{P}{\rho^2}\right)\nabla\rho$$

Using definition of summation interpolant

$$\nabla\left(\frac{P_i}{\rho_i}\right) = \sum_j m_j \frac{P_j}{\rho_j^2}\nabla_i W(\| \mathbf{x}_i - \mathbf{x}_j \|, h)$$

$$\left(\frac{P_i}{\rho_i^2}\right)\nabla\rho_i = \frac{P_i}{\rho_i^2}\sum_j m_j \nabla_i W(\| \mathbf{x}_i - \mathbf{x}_j \|, h)$$

Putting it together we have

$$\left(\frac{\nabla P_i}{\rho_i}\right) = \sum_j m_j \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2}\right)\nabla_i W(\| \mathbf{x}_i - \mathbf{x}_j \|, h)$$

A symmetrical central force between pairs of particles.

The Real World Problems

- The choice of kernel $W(\cdot)$ has great influence on accuracy and stability
- Using the Gradient and Laplacian equations may result in break-down of symmetry laws known from physics

A Simple Navier–Stokes Solver

Step 1 Compute density

$$\rho_i = \rho(\mathbf{x}_i) = \sum_j m_j W(\| x_i - x_j \|, h)$$

Step 2 Compute pressure

$$p_i = p(\mathbf{x}_i) = k(\rho(\mathbf{x}_i) - \rho_0)$$

Any problems here?

Step 3 Compute Interaction (Symmetric) Forces

$$\mathbf{f}^p(\mathbf{x}_i) = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\| \mathbf{x}_i - \mathbf{x}_j \|, h)$$

$$\mathbf{f}^v(\mathbf{x}_i) = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(\| \mathbf{x}_i - \mathbf{x}_j \|, h)$$

sum up $\mathbf{f}_i = \mathbf{f}_i^p + \mathbf{f}_i^v + \mathbf{g}$

Navier–Stokes Solver (Cont'd)

Step 4 Solve ODE (typical leap-frog scheme is used)

$$\dot{\mathbf{x}}_i = \mathbf{v}_i$$
$$\dot{\mathbf{v}}_i = \mathbf{f}_i / \rho_i$$

Step 5 Goto step 1

Discuss

- How to compute surface tension forces?
- How to deal with solid boundaries?

The Poly6 Kernel

$$W_{\text{poly6}}(r,h) = \frac{315}{64\pi h^4}\begin{cases}(h^2 - r^2)^3 & ; 0 \le r \le h \\ 0 & ;\end{cases}$$

Try

- Plot $W_{\text{poly6}}(r,h)$ as a function of r.
- Prove $\int W_{\text{poly6}}(r,h)dr = 1$ (Hint: consider if you are in 2D or 3D)
- Prove $W_{\text{poly6}}(r,h) \ge 0$
- Compute $\nabla_r W_{\text{poly6}}(r,h)$ and plot it
- Compute $\nabla_r^2 W_{\text{poly6}}(r,h)$ and plot it

The Spiky Kernel

$$W_{\text{spiky}}(r,h) = \frac{15}{\pi h^6} \begin{cases} (h-r)^3 & ; 0 \leq r \leq h \\ 0 & ; \end{cases}$$

Try

- Plot $W_{\text{spiky}}(r,h)$ as a function of r.
- Compute $\nabla_r W_{\text{spiky}}(r,h)$ and plot it
- Compute $\nabla_r^2 W_{\text{spiky}}(r,h)$ and plot it

Acceleration Techniques

- Finite Kernel Support $\Rightarrow$ only sum over neighbors that contribute
- Given support radius h find number of neighbors K for kNN

```
find–K(x, h, K)
  while
    [I, D] = knnsearch(x, x, K)
    if ‖ h ‖ > max D
      K ← K + ΔK
    else
      return K
    end
  end
end
```

- Spatial Hashing (Later)

Choosing Kernel Support Radius

- Proportional to the "volume" of the particle

$$h_i = \sigma \left(\frac{m_i}{\rho_i} \right)^{1/d}$$

where d is dimension and $\sigma \sim 1.3$ is a constant.

- Adjust h so each particle has a constant number of neighbors
- Choice h consistent with Density summation equation

$$
\begin{aligned}
&1: \textbf{while not converged} \\
&2: \quad \rho_i \leftarrow \textstyle\sum_j m_j W(\| \mathbf{x}_i - \mathbf{x}_j \|, h_i); \quad \forall i \\
&3: \quad h_i \leftarrow \sigma \left(\frac{m_i}{\rho_i} \right)^{1/d}; \quad \forall i \\
&4: \textbf{end}
\end{aligned}
$$

A fixed-point problem

Particle Setup

Given ρ_0 and some global h-value

- Make d-dimensional grid with cell sides $2h$
- Make mass m_i of the i^{th} particle correspond to the cell size

$$m_i = (2h)^d \rho_0$$

Body-Centered-Cubic (BCC) Lattice may result in better "compactness"

Visualization or Fluid

- Think of particles as balls of material having volume

$$V_i = \frac{m_i}{\rho_i}$$

  and radius

$$r_i = \sqrt[3]{\left(\frac{3V_i}{4\pi}\right)}$$

  Draw particle as ball with center $\mathbf{x}_i$ and radius r_i
- Think of particles as a blob (distribution) of material so draw them as balls with radius equal to h.
- Use color field and marching cubes or tetrahedra
- Use point rendering/splating

That is It!

Questions?

Further Reading

- J. J. Monaghan: Smoothed Particle Hydrodynamics, In: Annual review of astronomy and astrophysics. Vol. 30, p. 543-574. 1992.
- M. Desbrun and M. Gascuel: Smoothed particles: a new paradigm for animating highly deformable bodies, Proceedings of the Eurographics workshop on Computer animation and simulation 1996.
- M. Müller, D. Charypar and M. Gross: Particle-based fluid simulation for interactive applications, Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation.
- M. Müller, B. Solenthaler, R. Keiser and M. Gross: Particle-based fluid-fluid interaction, Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation.

More Reading

- J. J. Monaghan: Smoothed Particle Hydrodynamics, Reports on Progress in Physics, Volume 68, Number 8, 2005
- M. Kelager:Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics, DIKU graduate project, 2006.
- M. Becker and M. Teschner: Weakly compressible SPH for free surface flows, Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation.
- B. Adams, M. Pauly, R. Keiser and J.L. Guibas: Adaptively sampled particle fluids, ACM SIGGRAPH 2007
- B. Solenthaler and R. Pajarola, Predictive-corrective incompressible SPH,ACM Trans. Graph., vol. 28, no. 3, 2009.
- H. Lee and S. Han: Solving the Shallow Water equations using 2D SPH particles for interactive applications, The Visual Computer Volume 26, Numbers 6-8, 2010.

Study Groups

From the literature given

- List values used for the gas constant
- List the different kernels used
- List all the different methods for selecting/controlling the kernel radius
- List all the different methods for controlling the time-step size
- List the methods for computing pressure forces
- List the equation of state
- Identify problems with using the classical formula for density estimation
- Identify issues in using time-spatial varying support radii

Basic Programming Assignment

- Implement the M4 spline, the Poly 6 and the Spiky kernels in 2D
- Using numerical integration verify that your kernels are normalized
- Plot the kernels and verify that they are symmetrical (even) functions
- Implement the derivatives of the kernels, plot them and visually verify that they corresponds to the “slopes” of the kernel functions.

Intermediate Programming Assignment

Given a grid-based initialization of a water-like particles investigate the numerical properties of the density estimation only[1]. Plot the density profile along a horizontal line going through particles.

- Based on everyday experience reflect over how a real-world density profile should look
- Based on knowledge from convolution reflect over how a regularized density profile should look
- Examine how the density profile changes as a function of the support radius
- Examine how the density profile changes as a function of the number of particles
- Examine how the computing time changes as a function of the support radius and number of particles

[1] Turn of all forces and time integration

Advanced Programming Assignment

Use SPH to solve the problem

$$\frac{d\mathbf{v}}{dt} = -\nu\mathbf{v} - \frac{\nabla P}{\rho} + \mathbf{r}$$

where $\nu > 0$ and $\dot{\mathbf{r}} = \mathbf{v}$ and $\mathbf{r}, \mathbf{v} \in \mathbb{R}^2$ and $P, \rho \in \mathbb{R}$.

- Work out the SPH formulas for the problem
- Determine what kernels you want to use and how to compute kernel size
- If time permits try to play around with using different kernels, different strategies for choosing kernel size and different ways of initializing the problem.
- If time permits try to investigate what happens if the number of particles is increased.

Nice Matlab Functions to Know

Read about these things in your Matlab help

- **quad2d**
- **knnsearch**
- **repmat**
- **struct**
- Logical subscripting of matrices
- (Anonymous) function handles

They are used everywhere in the sample code.