IMPLICIT SURFACES AND RAYMARCHING

CIS 566 - Jan 25, 2018

Explicit vs. Implicit Surfaces (Briefly)

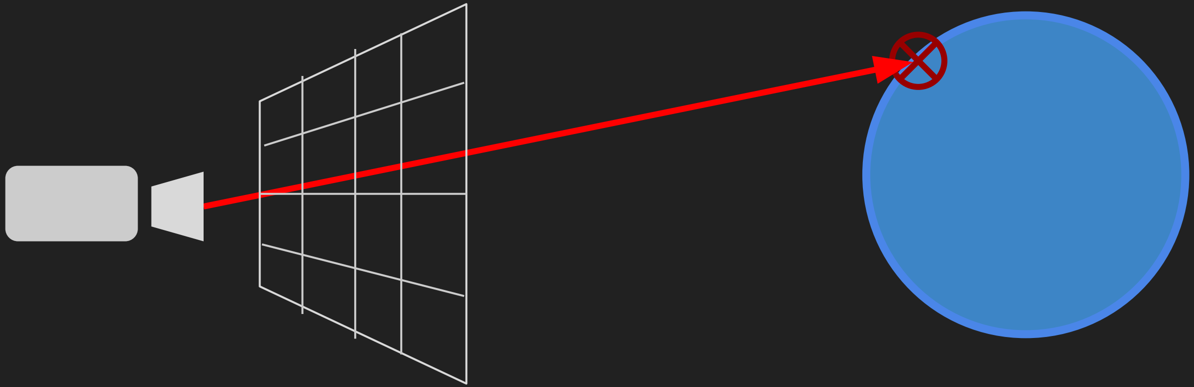
- Explicit surfaces are defined by parameterization functions, e.g. meshes or specific shapes
- Implicit surfaces are the set of all solutions to some function F(x, y, z) = 0, where x or y or z are unsolved.
- Many shapes can be expressed both implicitly and explicitly, but implicit surfaces are far more flexible (any F where solutions exist)

Explicit Circle: `f(t) = {r * cos(t), r * sin(t)}`

Implicit Circle: `F(x, y) = sqrt(x * x + y * y) - r`

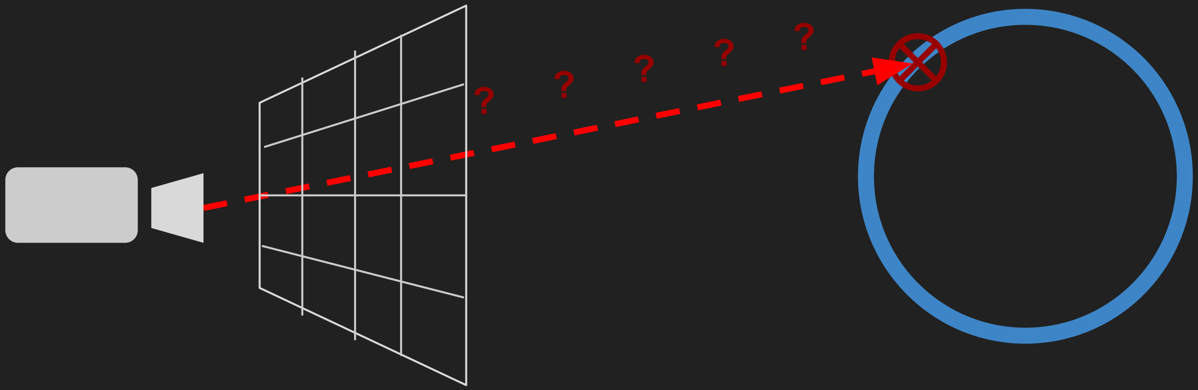
Finding an explicit surface: Ray-Tracing

Given a ray and a specific shape, use that shape's specific intersection test to determine whether the ray intersects it, and if so, where.



Finding an implicit surface: Ray-Marching

Given a ray and an implicit surface function F(x, y, z), test points along the ray to see if they solve F = 0.

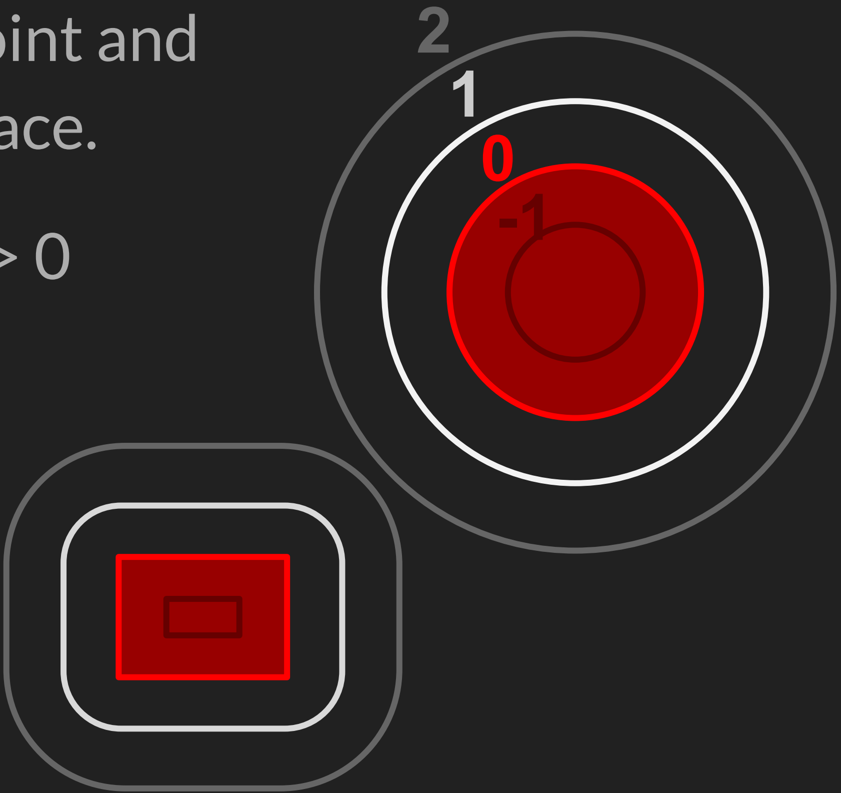


Implicit Surface case: Signed Distance Fields (SDF)

An SDF is a function that takes a point and returns a distance to a shape's surface.

SDF evaluates to 0 on the surface, > 0 outside, and < 0 inside.

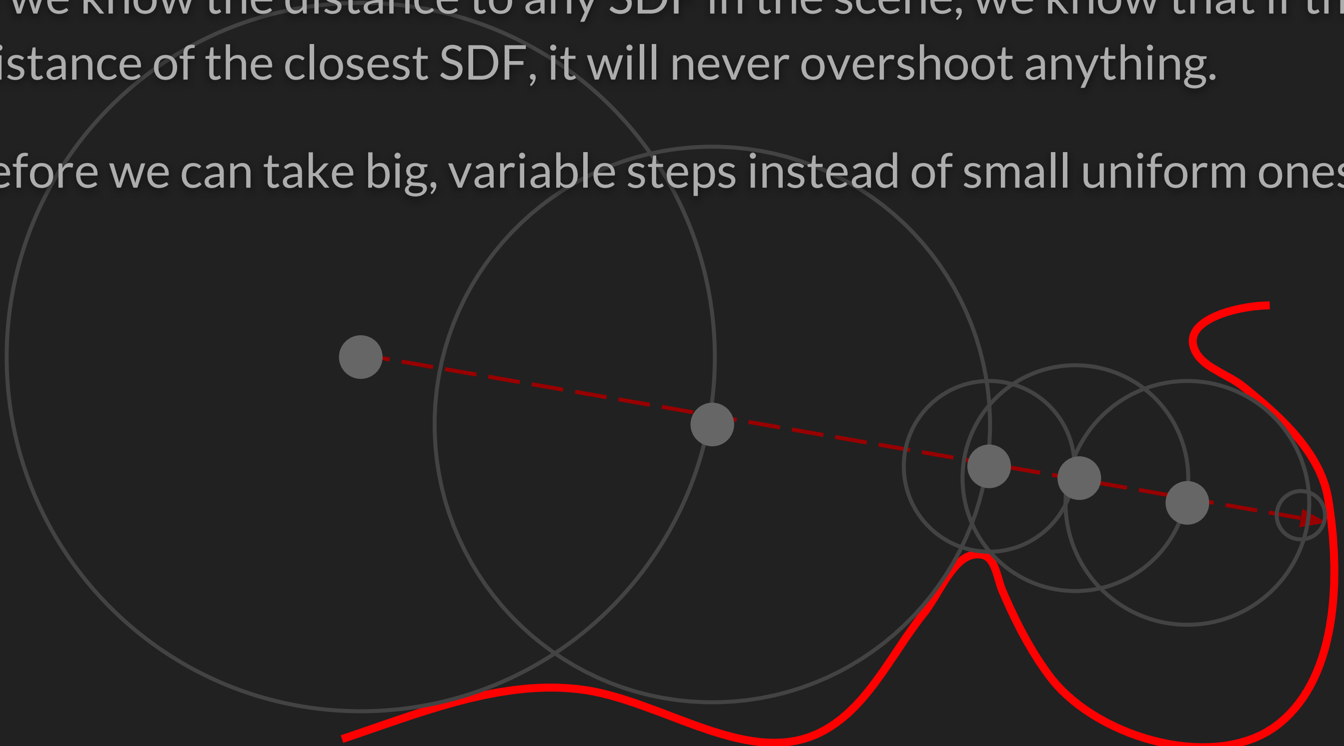
An SDF must always be linear.



Better marching for SDF: Sphere-Marching

Since we know the distance to any SDF in the scene, we know that if the ray marches the distance of the closest SDF, it will never overshoot anything.

Therefore we can take big, variable steps instead of small uniform ones.



Meet the Reference Guy: I Q

Your new dictionary:

http://www.iquilezles.org/www/index.htm

Made every article on this subject

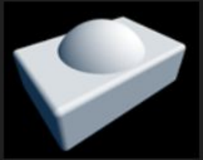
Code can be nigh-incomprehensible

Eyes too terrible to behold

Modeling with SDFs: from Inigo Quilez

Union

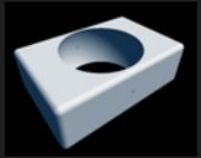
```
float opU( float d1, float d2 )
{
    return min(d1,d2);
}
```



Subtraction

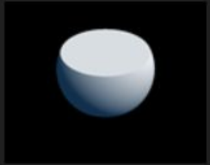
~~**Substraction**~~

```
float opS( float d1, float d2 )
{
    return max(-d1,d2);
}
```



Intersection

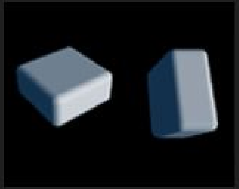
```
float opI( float d1, float d2 )
{
    return max(d1,d2);
}
```



Properties of SDFs: Domain Transformation (Still IQ)

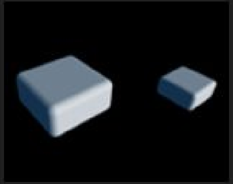
Rotation/Translation

```
vec3 opTx( vec3 p, mat4 m )
{
    vec3 q = invert(m)*p;
    return primitive(q);
}
```



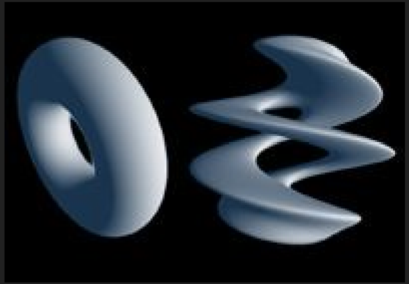
Scale

```
float opScale( vec3 p, float s )
{
    return primitive(p/s)*s;
}
```

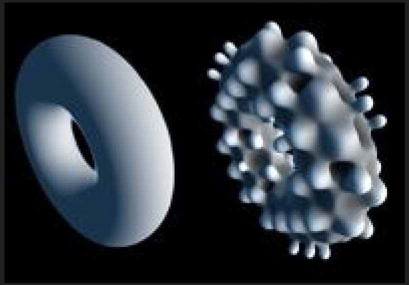


Domain Deformations: Danger Zone

You can add non-uniform transformations to the SDF such as twists, noises or bends that depend on the location.



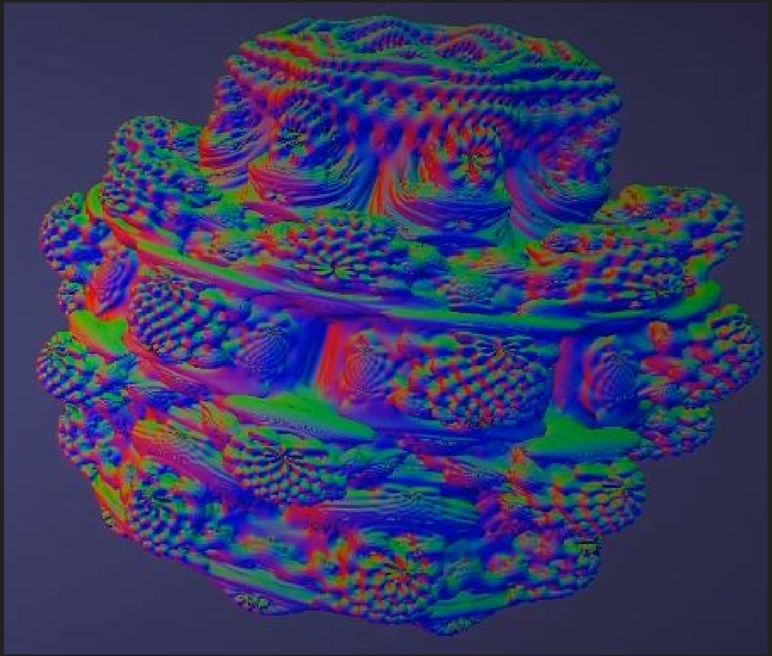
But this makes the distance field non-constant and can create bad artifacts if you sphere-march carelessly!



Properties of SDFs: Surface Normals

- Gradient Method - central differences
- Jitter x, y, and z for each component of normal, then normalize

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)$$



Properties of SDFs: Ambient Occlusion (IQ again)

- “5-tap” AO
- Sample the SDF along the normal with exponentially decreasing weights

$$ao = 1 - k \cdot \sum_{i=1}^{5} \frac{1}{2^i} \left(i \cdot \Delta - distfield\left(p + n \cdot i \cdot \Delta \right) \right)$$

Making a Ray-Marcher

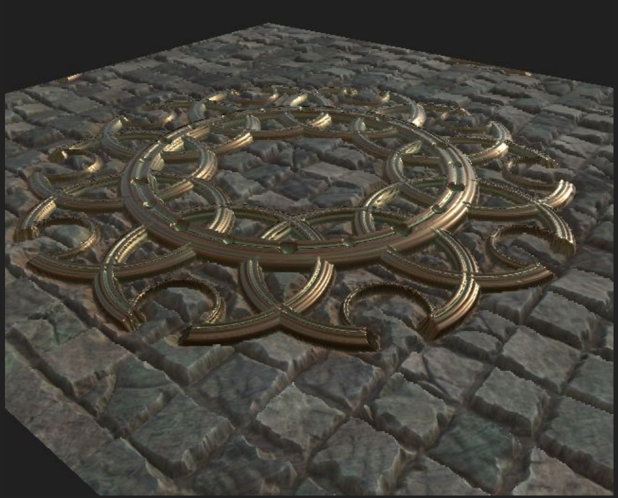
- Demo

Optimizing a Ray-Marcher

- Bounding volume
- 4-sample normal computation
- Raytracing when appropriate
- Step size proportional to t (aka distance) (details decay according to 1/t)
- Cone marching (low-resolution marching)

Ray-Marching beyond SDFs

- Volumes: https://vimeo.com/252453243
- Non-linear Implicit surfaces: https://www.clicktorelease.com/code/bumpy-metaballs/
- Parallax Occlusion Mapping:



IQ Starter Pack

Distance Functions:
http://www.iquilezles.org/www/articles/distfunctions/distfunctions.htm

Smooth Minimum:
www.iquilezles.org/www/articles/smin/smin.htm

Penumbra Shadows:
http://www.iquilezles.org/www/articles/rmshadows/rmshadows.htm

Useful Little Functions (good for motion and color):
http://www.iquilezles.org/www/articles/functions/functions.htm

Extra Stuff / References

- https://www.facebook.com/mariano.merchante/videos/10159669480995618/ (cool raymarching shader from CGGT grad Mariano Merchante)
- https://cis700-procedural-graphics.github.io/files/implicit_surfaces_2_21_17.pdf (Rachel Hwang’s presentation from Procedurals in Spring 2017)
- http://jamie-wong.com/2016/07/15/ray-marching-signed-distance-functions/
- http://www.iquilezles.org/www/material/nvscene2008/rwwtt.pdf