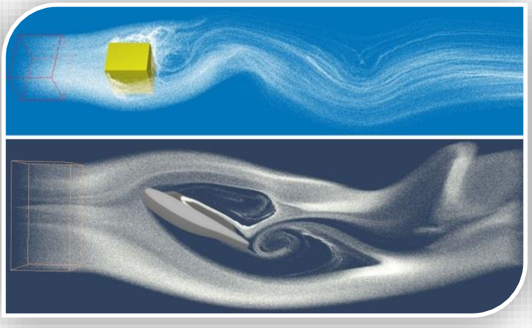
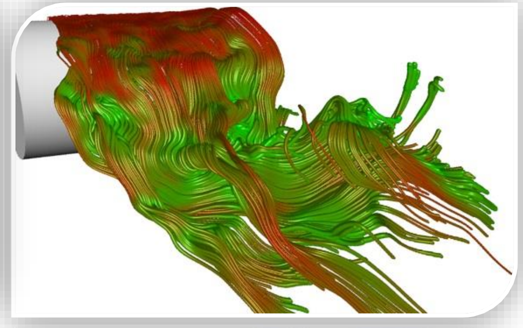
Master Practical Course

Interactive Visual Data Analysis

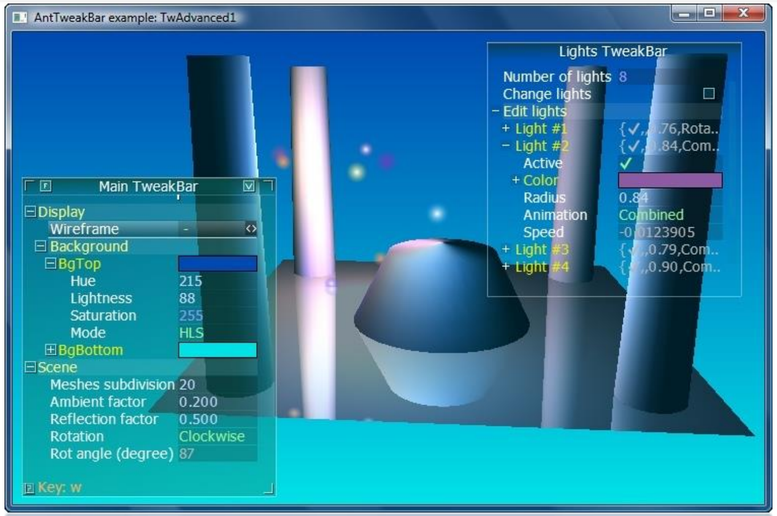



tum.3D

computer graphics & visualization

Today

- Assignment 3
  - Simple GUI
  - Iso-surface ray casting



Lights TweakBar

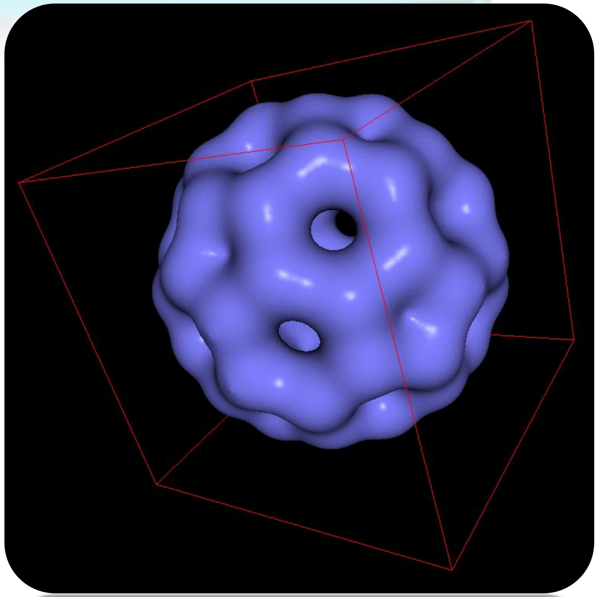
Number of lights	8
Change lights	☐
− Edit lights	
+ Light #1	{✓,0.76,Rota..
− Light #2	{✓,0.84,Com..
Active	✓
+ Color	🟪
Radius	0.84
Animation	Combined
Speed	-0.0123905
+ Light #3	{✓,0.79,Com..
+ Light #4	{✓,0.90,Com..

Main TweakBar

⊟ Display	
Wireframe	-
⊟ Background	
⊟ BgTop	🟦
Hue	215
Lightness	88
Saturation	255
Mode	HLS
⊞ BgBottom	🟦
⊟ Scene	
Meshes subdivision	20
Ambient factor	0.200
Reflection factor	0.500
Rotation	Clockwise
Rot angle (degree)	87

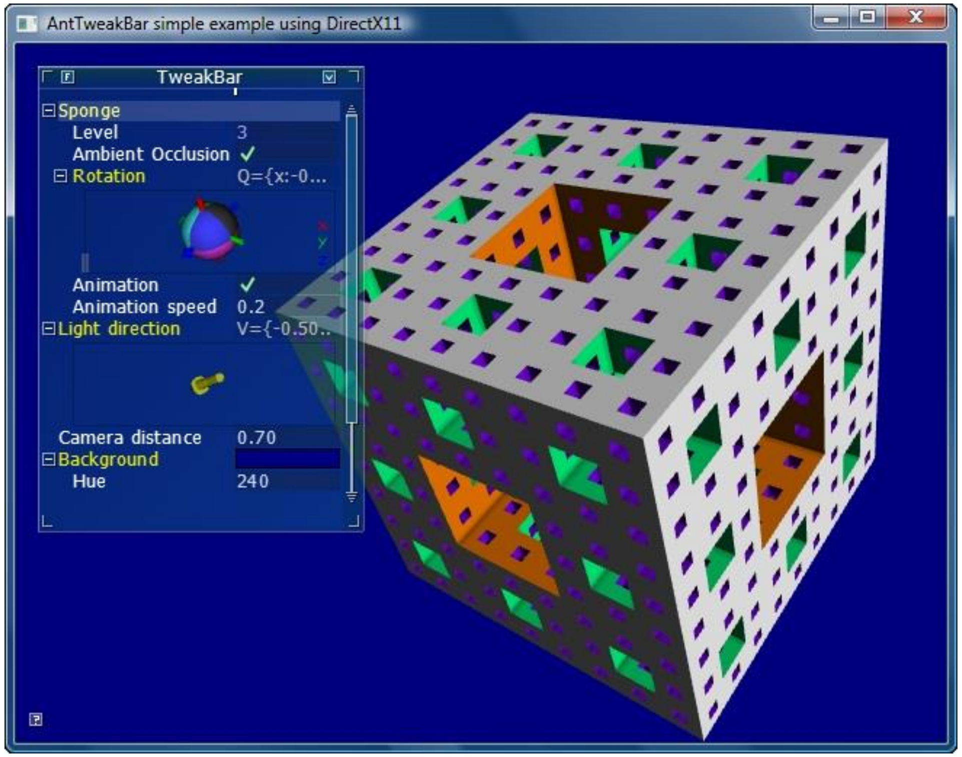
Key: w

Magic!



AntTweakBar

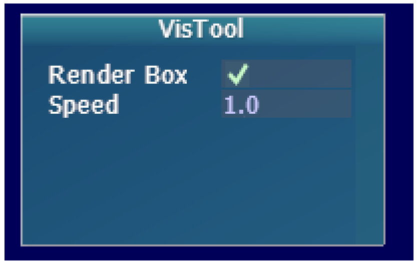
http://www.antisphere.com/Wiki/tools:anttweakbar :
"AntTweakBar is a small and easy-to-use C/C++ library that allows programmers to quickly add a light and intuitive graphical user interface into graphic applications based on OpenGL (compatibility and core profiles), DirectX 9, DirectX 10 or DirectX 11 to interactively tweak parameters on-screen."



⊟Sponge	
Level	3
Ambient Occlusion	✓
⊟ Rotation	Q={x:-0...
Animation	✓
Animation speed	0.2
⊟Light direction	V={-0.50..
Camera distance	0.70
⊟Background	
Hue	240

AntTweakBar

- AntTweakBar is minimalistic
  - Extremely easy and fast to use
  - Supports variables and buttons
- Integrate by inserting calls into the appropriate callback functions
  - E.g. TwInit ←→ OnCreateDevice
    TwWindowSize ←→ OnD3D11ResizedSwapChain
  - Follow the online tutorial
- Create a bar and add variables

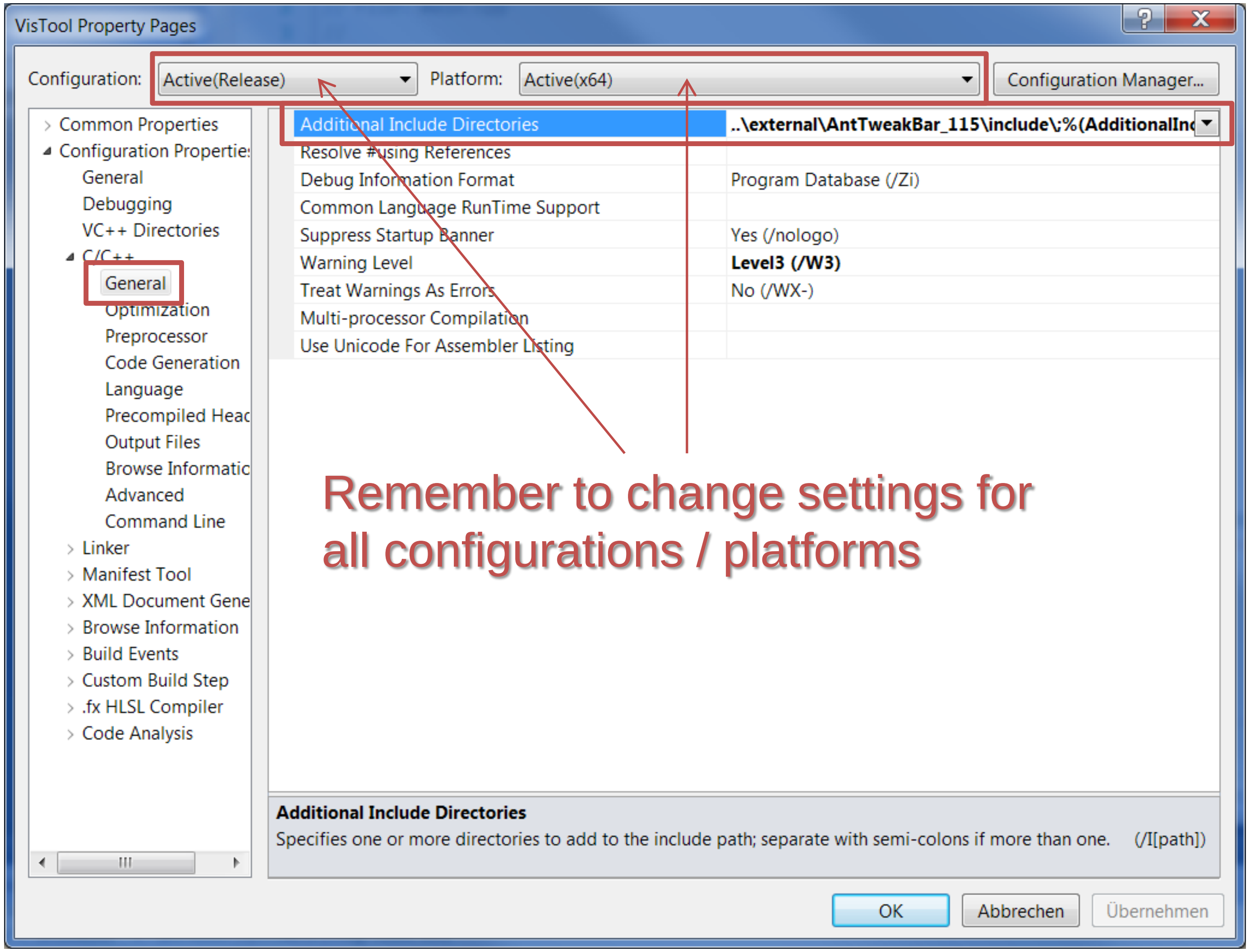

VisTool

Render Box ✓
Speed 1.0

```
g_Bar = TwNewBar("VisTool");
TwAddVarRW(g_Bar, "Render Box", TW_TYPE_BOOLCPP, &g_renderBox, "");
TwAddVarRW(g_Bar, "Speed", TW_TYPE_FLOAT, &g_speed, "min=1 max=5.5 step=0.1");
```

Adding External Libraries in VS

- Add include directory for compiler



VisTool Property Pages

Configuration: Active(Release) Platform: Active(x64) Configuration Manager...

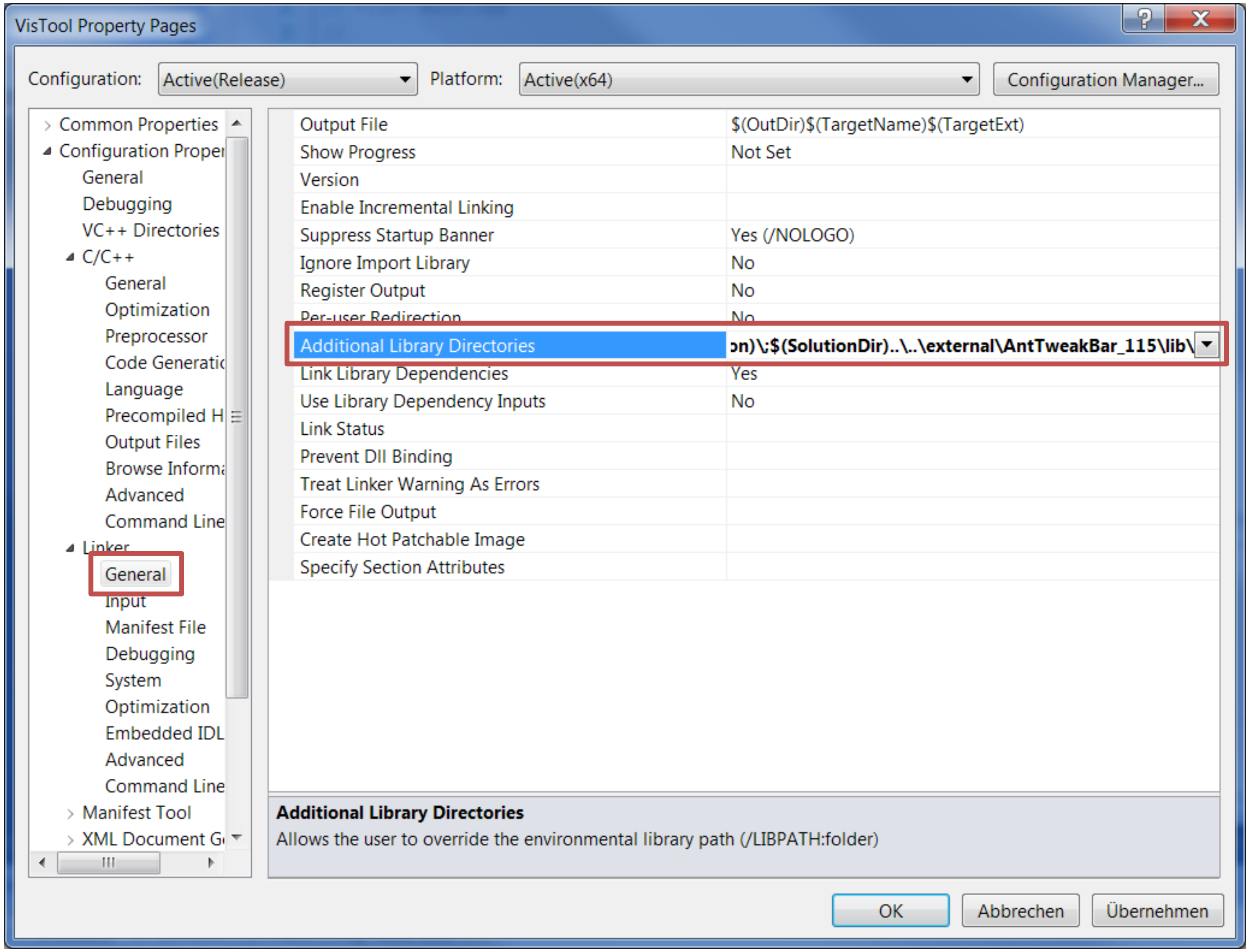
Additional Include Directories	..\external\AntTweakBar_115\include\;%(AdditionalInc
Resolve #using References	
Debug Information Format	Program Database (/Zi)
Common Language RunTime Support	
Suppress Startup Banner	Yes (/nologo)
Warning Level	**Level3 (/W3)**
Treat Warnings As Errors	No (/WX-)
Multi-processor Compilation	
Use Unicode For Assembler Listing	

Remember to change settings for all configurations / platforms

Additional Include Directories
Specifies one or more directories to add to the include path; separate with semi-colons if more than one. (/I[path])

Adding External Libraries in VS

- Add library directory for linker



VisTool Property Pages

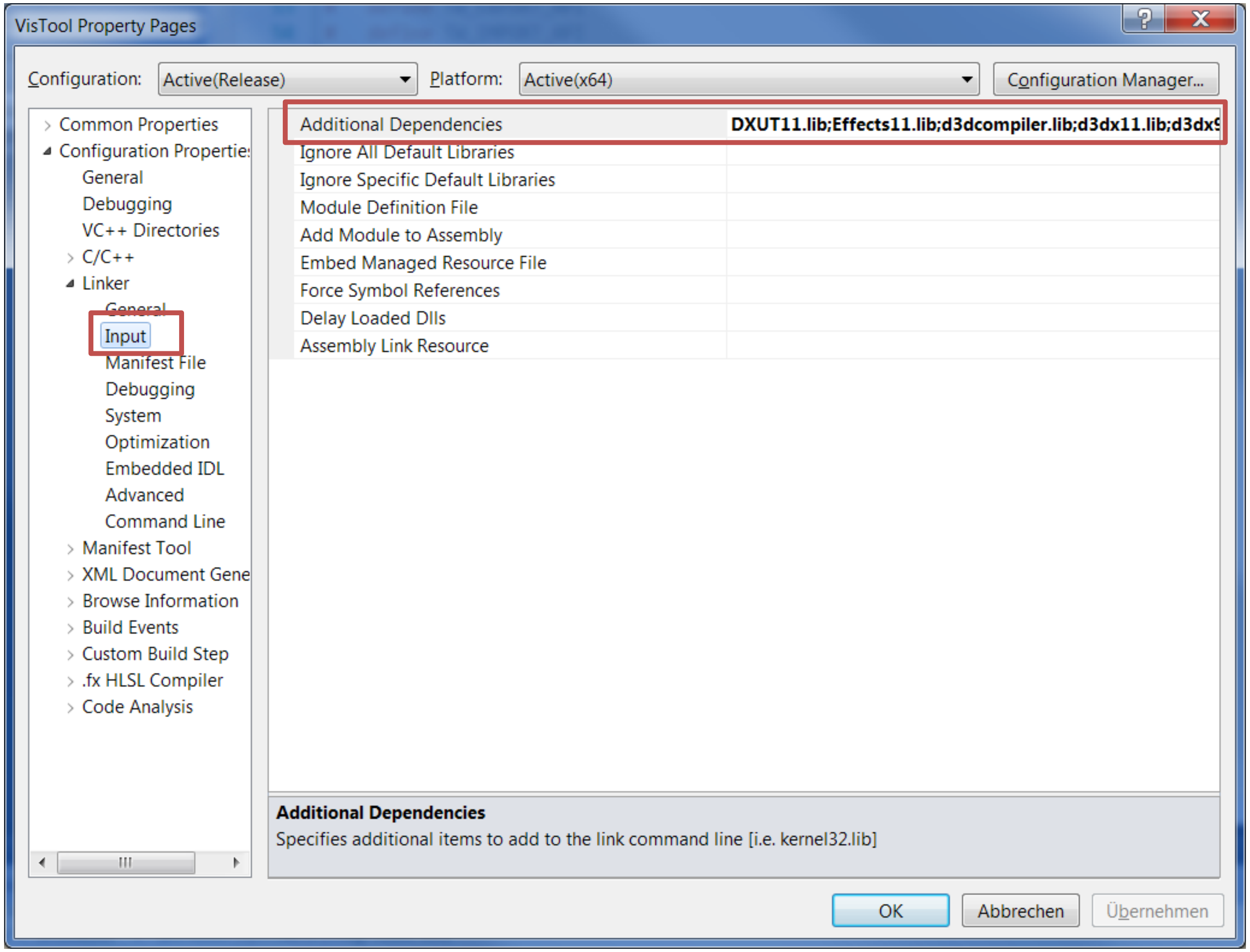
Configuration: Active(Release) Platform: Active(x64) Configuration Manager...

Output File	$(OutDir)$(TargetName)$(TargetExt)
Show Progress	Not Set
Version	
Enable Incremental Linking	
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	**on)\;$(SolutionDir)..\..\external\AntTweakBar_115\lib\**
Link Library Dependencies	Yes
Use Library Dependency Inputs	No
Link Status	
Prevent Dll Binding	
Treat Linker Warning As Errors	
Force File Output	
Create Hot Patchable Image	
Specify Section Attributes	

Additional Library Directories
Allows the user to override the environmental library path (/LIBPATH:folder)

Adding External Libraries in VS

- Add library name for linker



VisTool Property Pages

Configuration: Active(Release) Platform: Active(x64) Configuration Manager...

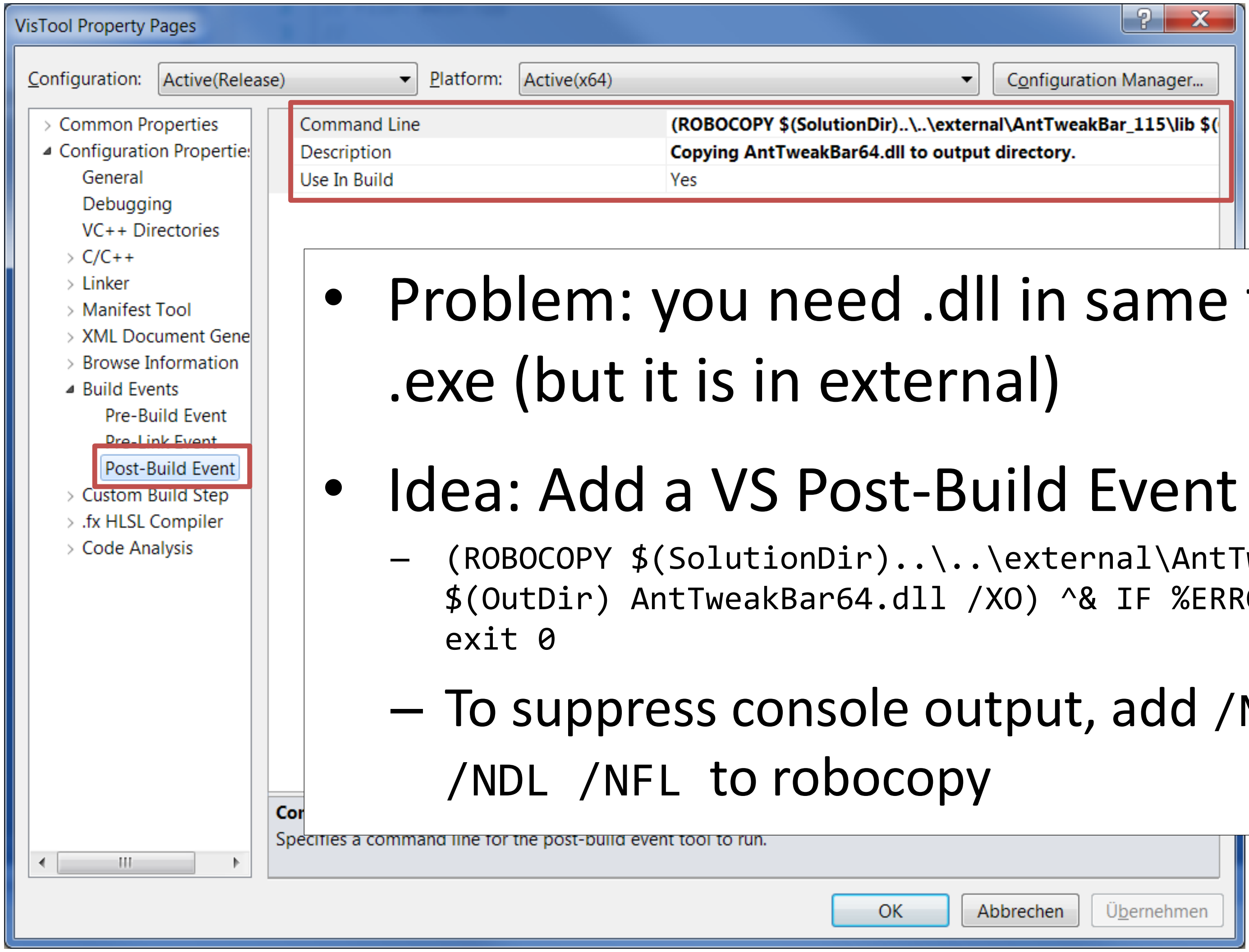
- Common Properties
- Configuration Properties
  - General
  - Debugging
  - VC++ Directories
  - C/C++
  - Linker
    - General
    - Input
    - Manifest File
    - Debugging
    - System
    - Optimization
    - Embedded IDL
    - Advanced
    - Command Line
  - Manifest Tool
  - XML Document Gene
  - Browse Information
  - Build Events
  - Custom Build Step
  - .fx HLSL Compiler
  - Code Analysis

Additional Dependencies	**DXUT11.lib;Effects11.lib;d3dcompiler.lib;d3dx11.lib;d3dx9**
Ignore All Default Libraries	
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to the link command line [i.e. kernel32.lib]

OK Abbrechen Übernehmen

Adding DLL to Project

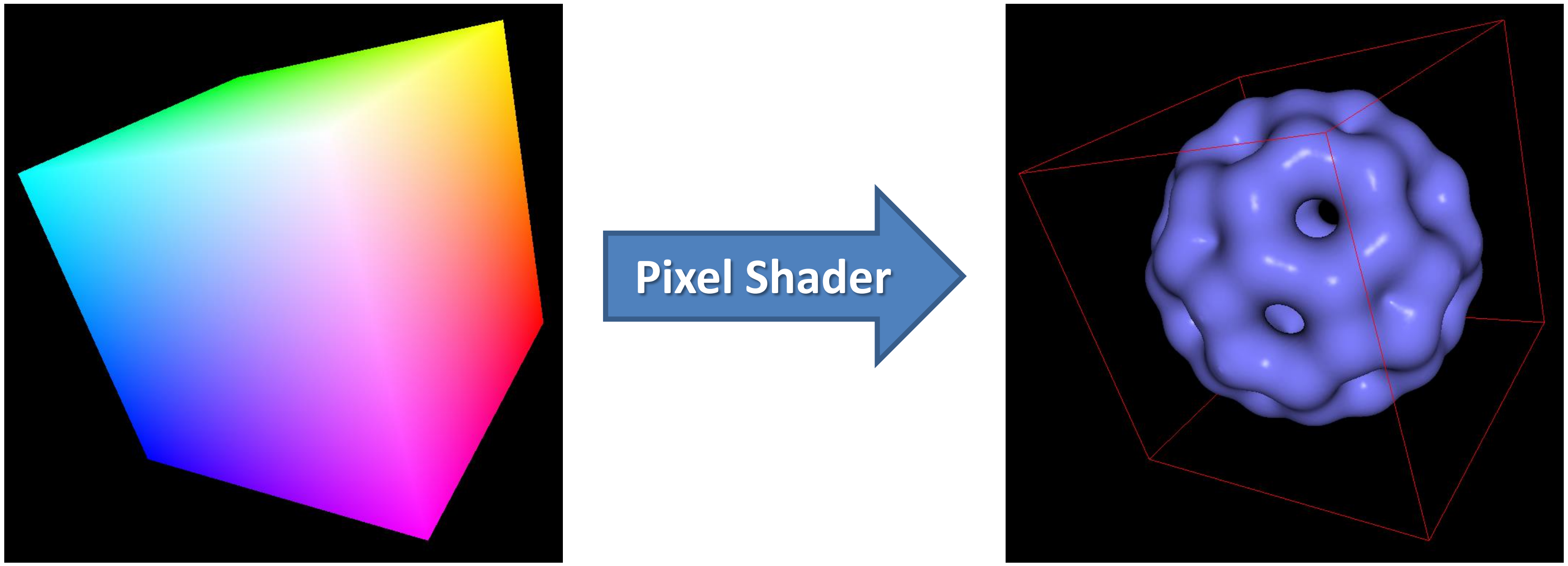


Command Line	(ROBOCOPY $(SolutionDir)..\..\external\AntTweakBar_115\lib $(
Description	Copying AntTweakBar64.dll to output directory.
Use In Build	Yes

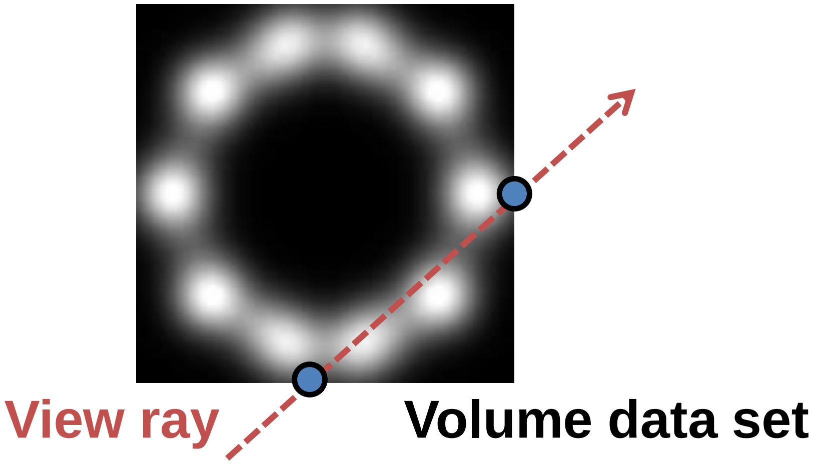
Specifies a command line for the post-build event tool to run.

- Problem: you need .dll in same folder as .exe (but it is in external)
- Idea: Add a VS Post-Build Event
  - `(ROBOCOPY $(SolutionDir)..\..\external\AntTweakBar_115\lib $(OutDir) AntTweakBar64.dll /XO) ^& IF %ERRORLEVEL% LEQ 1 exit 0`
  - To suppress console output, add `/NJH /NJS /NDL /NFL` to robocopy

Volume Rendering

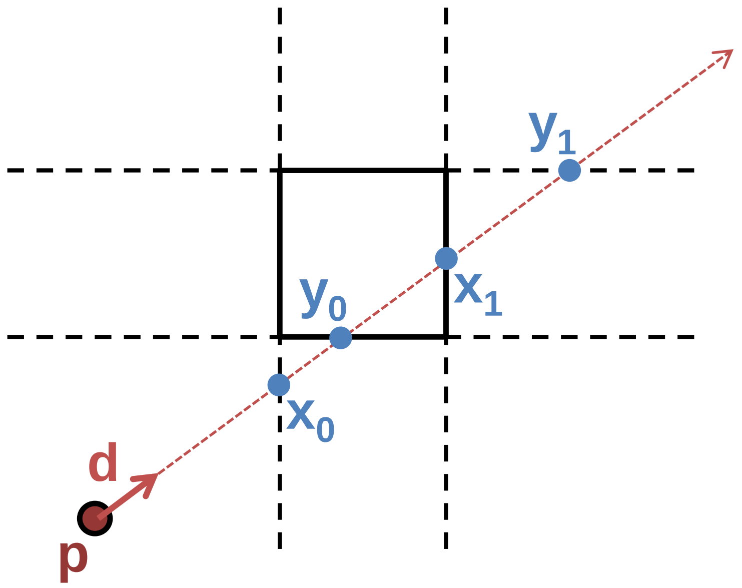


- We‘ll need the entry and exit point!



Ray-AABox Intersection

- Do in object space (box = $[0;1]^3$) → box is axis-aligned!
- Reduce to ray-plane intersections



- t = max(min(t_x0, t_x1), min(t_y0, t_y1))
- More info: http://people.csail.mit.edu/amy/papers/box-jgt.pdf

Volume Data

- `.dat/.raw` file pairs
- `.dat` contains metadata:

`ObjectFileName:`	`ppmlow_t000z.raw`	`.raw` file name
`Resolution:`	`128 128 128`	Volume size x, y, z
`Format:`	`BYTE`	Data type
`SliceThickness:`	`1 1 1`	Grid spacing

- `.raw` contains binary volume data
  – Use either `std::ifstream` (`#include <fstream>`)
  – or `fopen/fread/fclose` (`#include <cstdio>`).
  – In either case, make sure you open the file in binary mode! (`std::ios_base::binary` or „`rb`“)

3D Texture

- `ID3D11Device::CreateTexture3D` and `ID3D11Device::CreateShaderResourceView`
  - Check MSDN!
  - Hint: DXGI_FORMAT_R8_UNORM

„Unsigned normalized“: Mapped to a float in [0,1]

- Usage in HLSL:

```
Texture3D<float> g_texVolume; // declare texture as global var
```

Optional: Return type; not necessarily the same as the underlying data type!

```
SamplerState samLinear { // define how to access it
      Filter = MIN_MAG_MIP_LINEAR;
}
```

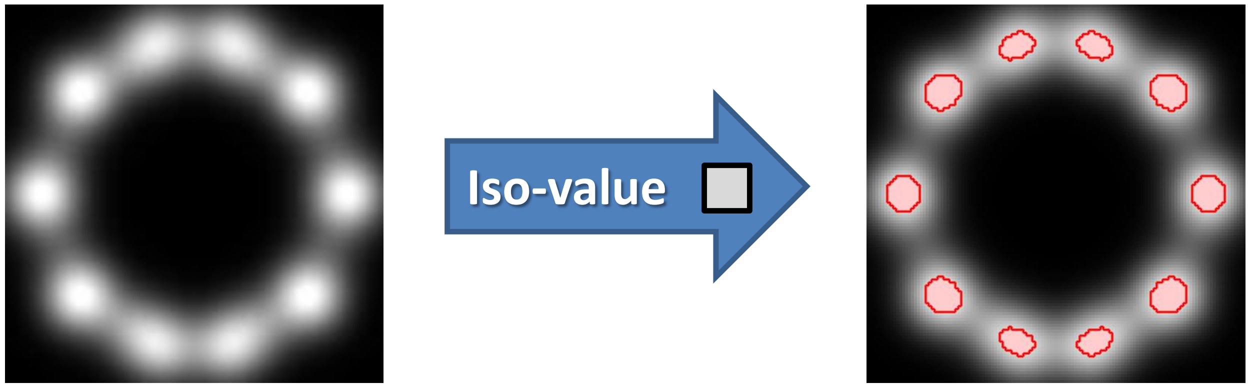
Linear interpolation for minification, magnification and mipmapping

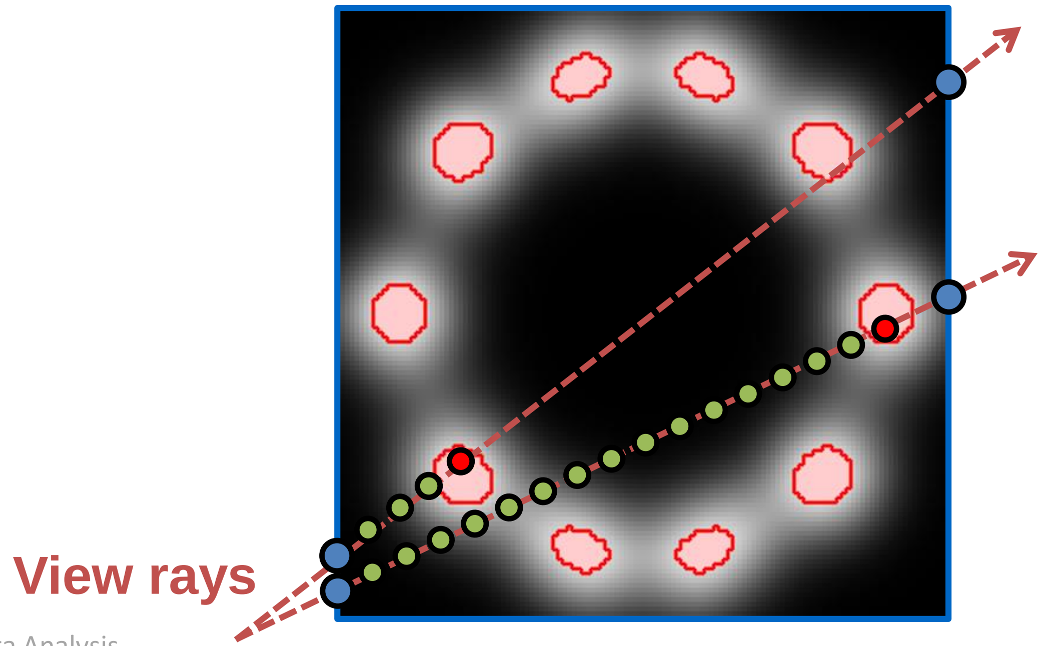
```
// sample (i.e. read) from the texture
float val = g_texVolume.SampleLevel(samLinear, texCoord, 0.0);
```

Coordinates in $[0,1]^3$ Mip level

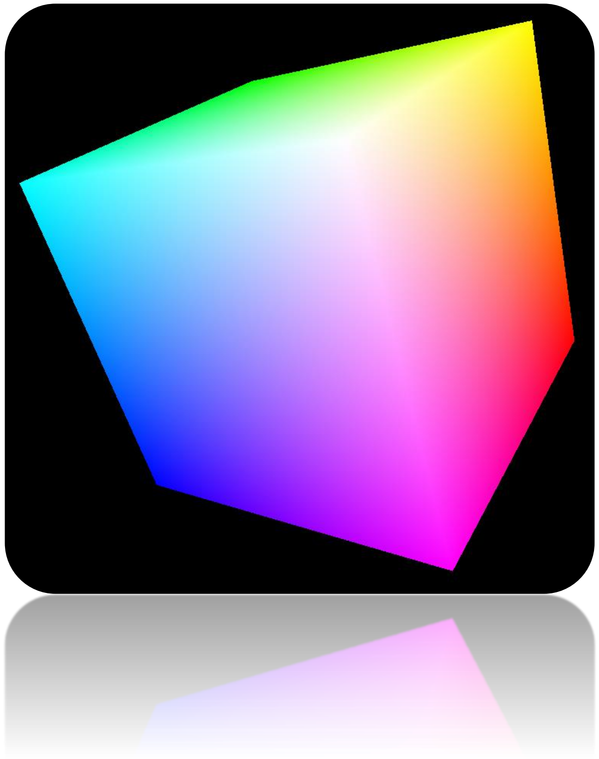
Volume Ray Casting: Iso-surfaces

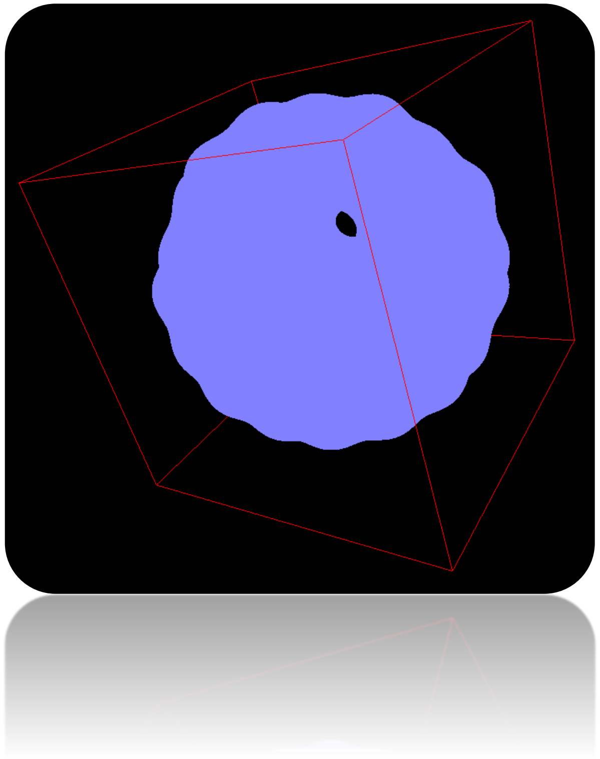
Iso-surface: all points with a value equal to the iso-value





Result so far

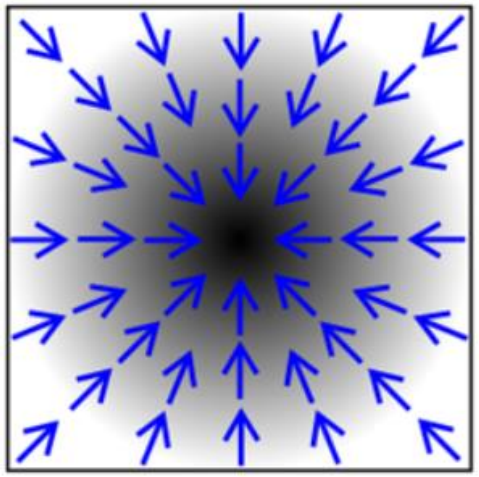


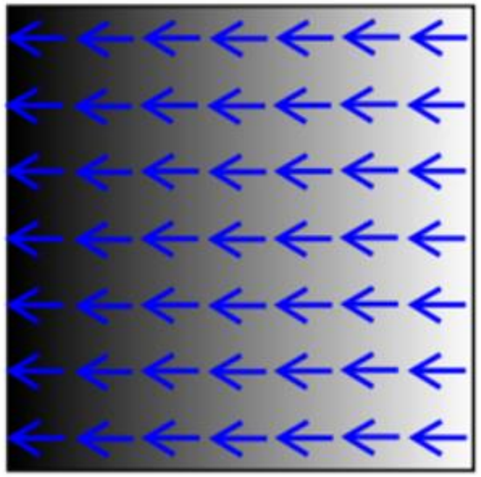


We need lighting!

Where do we get a normal vector?

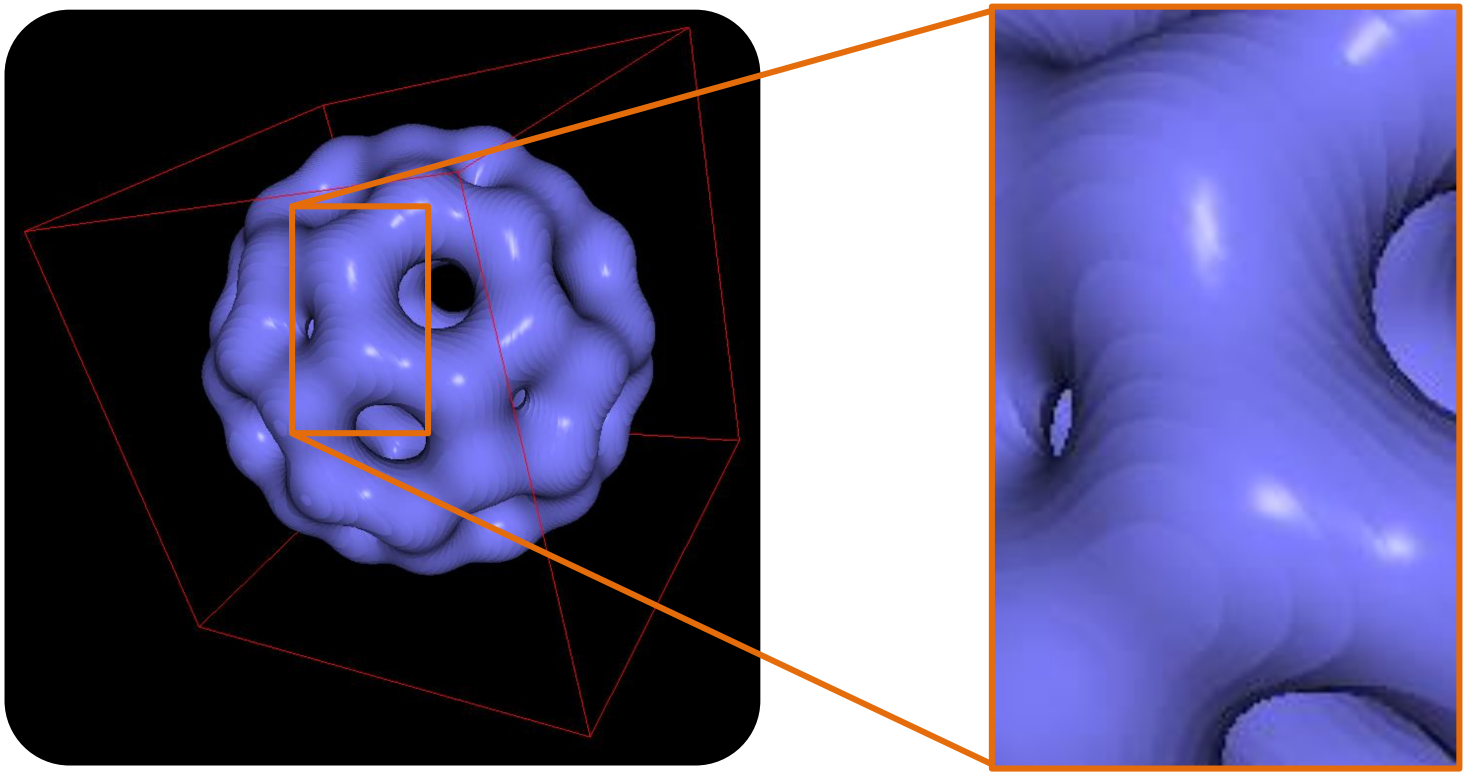
- Use negative (normalized) gradient as normal vector

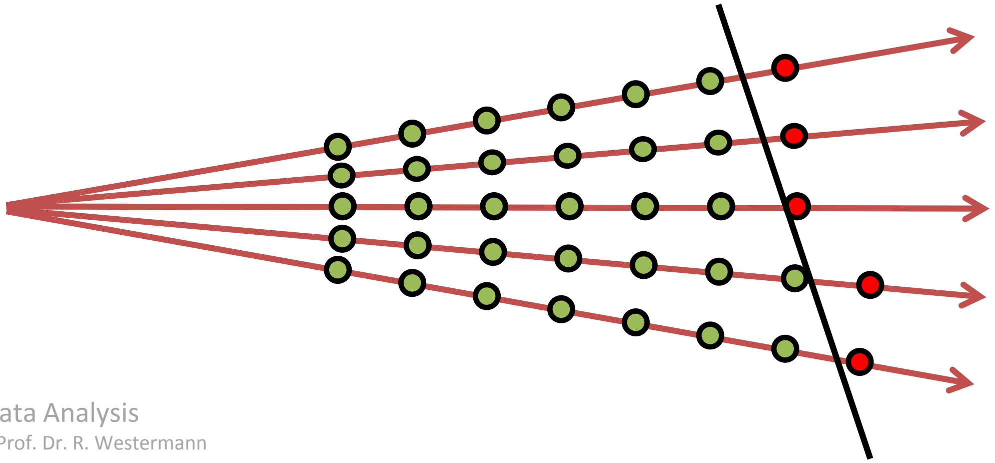




- Central differences: $f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}$
- h should be the size of one voxel/texel
  - Hint: Use optional `Offset` parameter to `SampleLevel`!

Banding Artifacts





→ Binary Search!

References

- AntTweakBar
  - Official website (documentation, tutorial, …) http://www.antisphere.com/Wiki/tools:anttweakbar
  - Sample code in `external`
- Ray-AABox Intersection
  - http://people.csail.mit.edu/amy/papers/box-jgt.pdf

Q&A

computer graphics & visualization

Questions ?