CS 498 VR

Lecture 19 - 4/9/18

go.illinois.edu/VRlect19

Review from previous lectures

- Image-order Rendering and Object-order Rendering

  Image-order Rendering:
  - Process: Ray Generation, Ray Intersection, Assign Value
  - Shading: Blinn-Phong Specular model plus Ambient, Diffuse

  Object-order Rendering:
  - Process: Rasterization, Depth Order, Assign Value
  - Concept: Painter’s Algorithm, Z-buffer, Interpolation, Barycentric Coordinates

Today:

- Mappings
  - Texture mapping
  - Bump mapping
  - Normal mapping

  Challenges and (some) Solutions for Graphics in VR
  - Aliasing & Solutions
  - Mapping related Problems & Mipmapping
  - Optical Distortion
  - Latency

  Graphics Topics Summary

Texture Mapping:

Understand: “Paint” Texture/Repeating Patterns on Geometries (i.e. Triangle)

Texture Mapping:

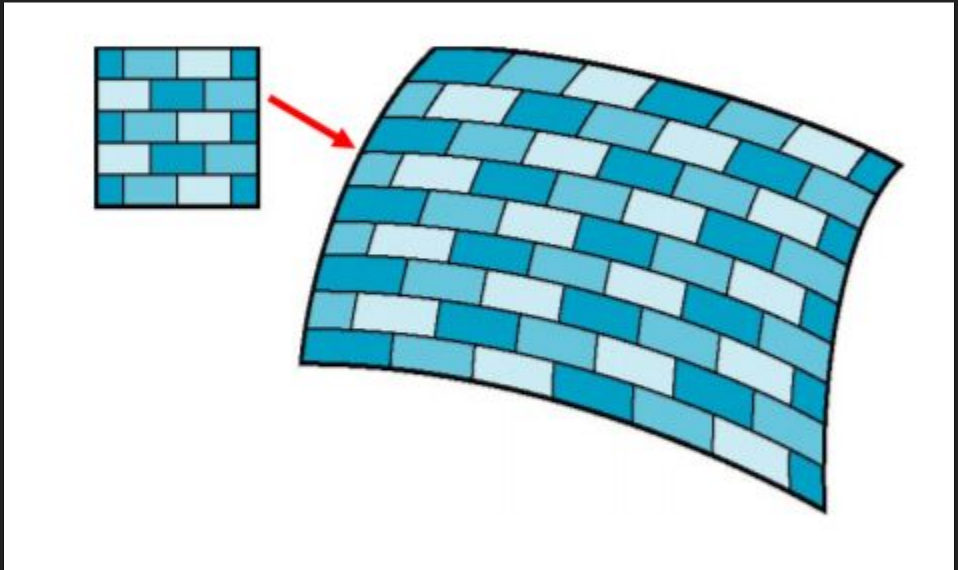
Our goal: map an image(texture) to a surface
Each pixel of the surface comes from the corresponding part of the image.

How many dimensions does the texture have?

____

What about the surface?

____

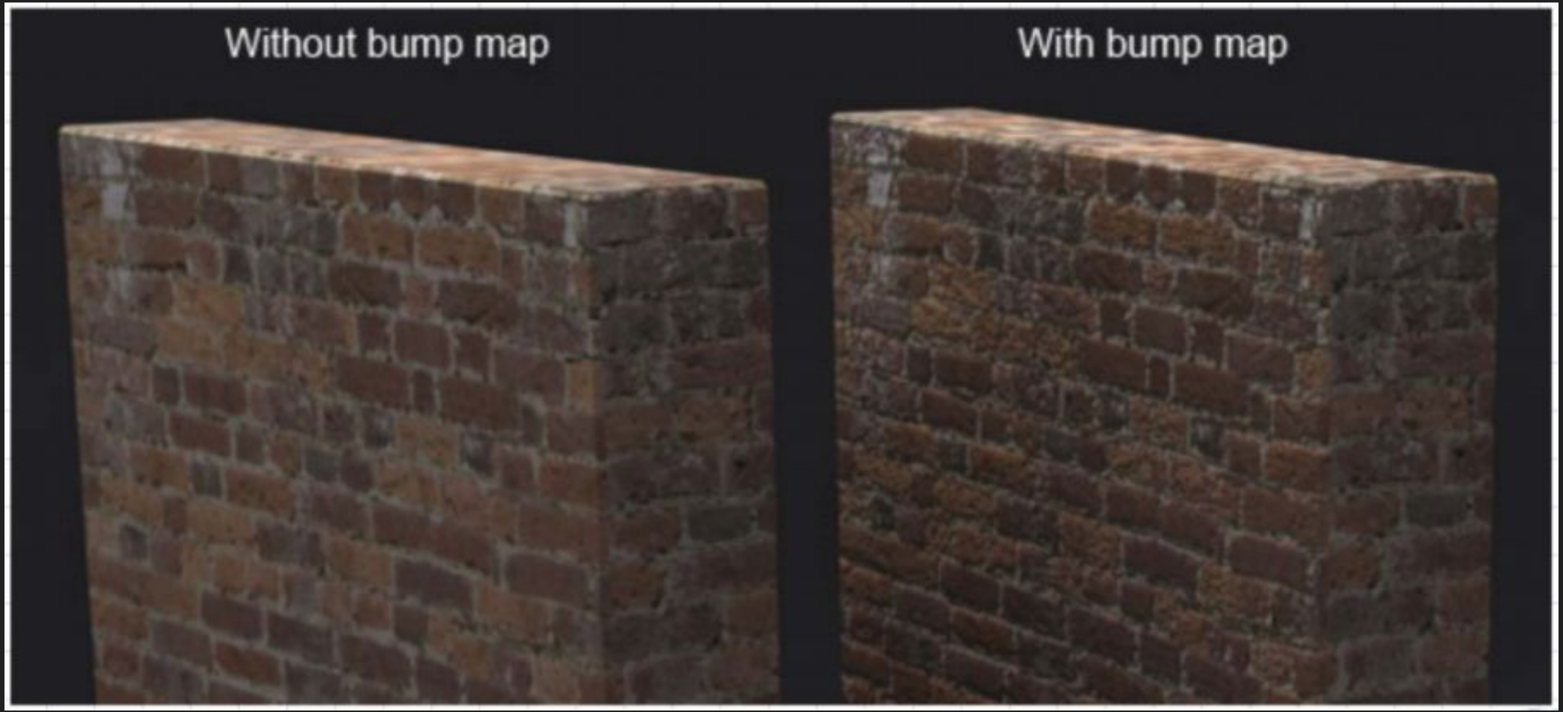


Texture Mapping requires:

____

Bump Mapping

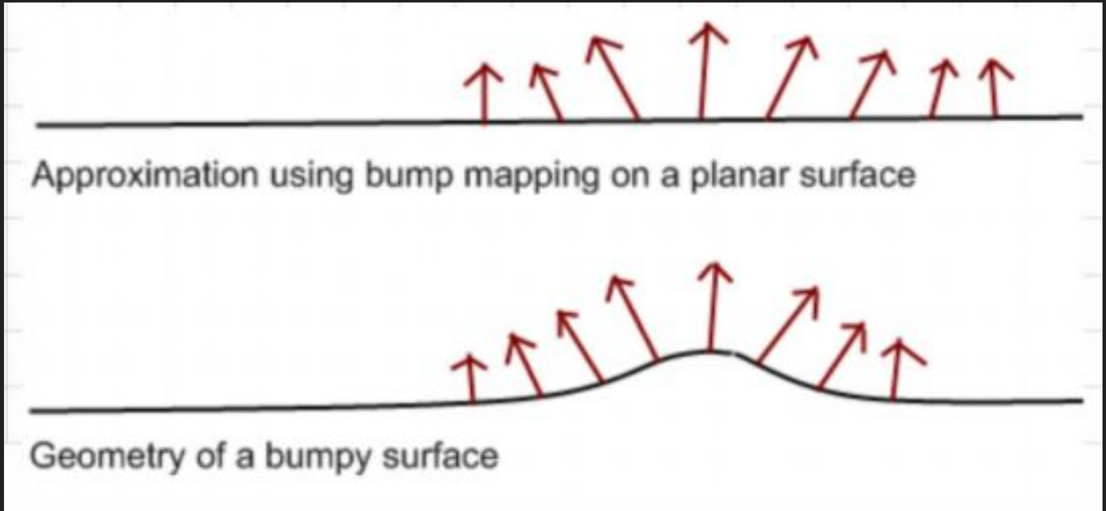
Applicable in shading models taking reflections into account (e.g. Blinn Phong)



Bump Mapping

Recall our lighting model with diffuse, Blinn specular and ambient components:

L = ______________________________________



When applied to the model above, Bump mapping changes the ___ vector, without changing the shape.

Normal Mapping

Using RGB values as a 3-vector to encode the surface normal of the object.

Left: A real 3D-shaped surface;
Middle: A normal map generated from this surface;
Right: Apply the normal map on a completely flat surface.

Challenges and (some) Solutions for Graphics in VR

Part 1. Issues of shading/graphics pipeline

- Texture mapping issues
- Bump & Normal mapping, Shading Model

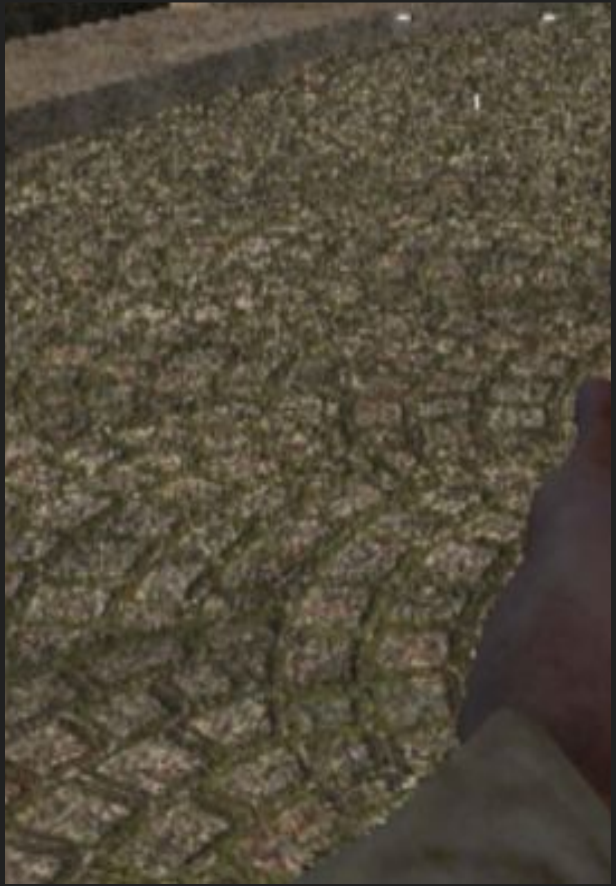
Part 2. Issues in rendering of the contents

- Aliasing & solution
- Optical Distortion & solution

Texture Mapping Issues

Suppose we have the texture:

We attach the texture to the floor. Observer looks from a common perspective. For each pixel on the screen, we fetch one pixel from the texture...



Problems?

Texture Mapping Issues and Solution

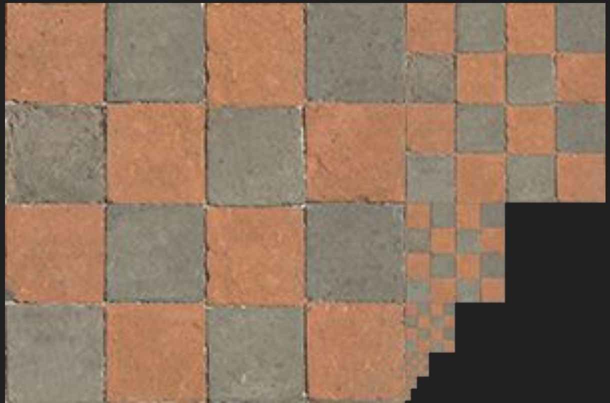
(If we directly choose one pixel from the texture image as the value of the result), there will be problem if:

- Resolution of the texture is far higher than the surface.
- Resolution of the texture is far lower than the surface.

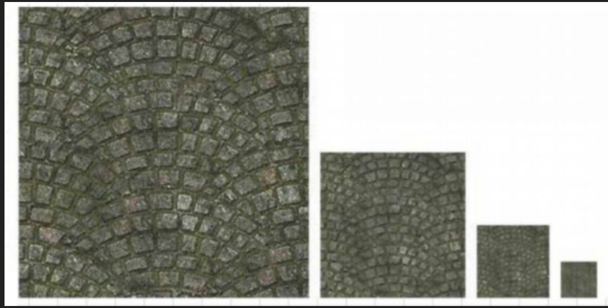
Solution 1: Mipmapping

Idea: pre-create series of textures that have different resolutions (typically in powers of 2)

e.g. Average n*n pixels into a smaller image.



Texture Mapping - Mipmapping



Shader can pick the best texture image at different positions.

Texture Mapping - Modify Sampling

Solution 2: Change sampling way.

Idea: Instead of picking one expected pixel from the texture image, pick several and take the average.

Especially efficient for case #2 when Resolution of the texture is far lower than the surface.

Also an good way to solve aliasing problem…

Bump & Normal Mapping, Blinn Phong Issues

Too many simplifications on shading model. Result looks fake!

Cannot deal with many objects:

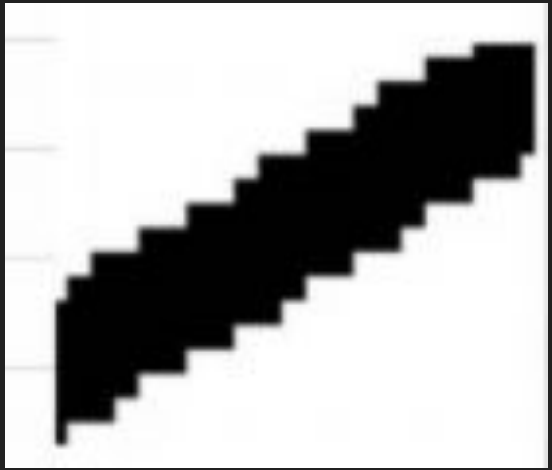
1. _______

   Changing only the normal vectors of a surface cannot support blendings & transparency.

2. _______

   Reflecting objects are also light source. Blinn Phong model cuts them out.

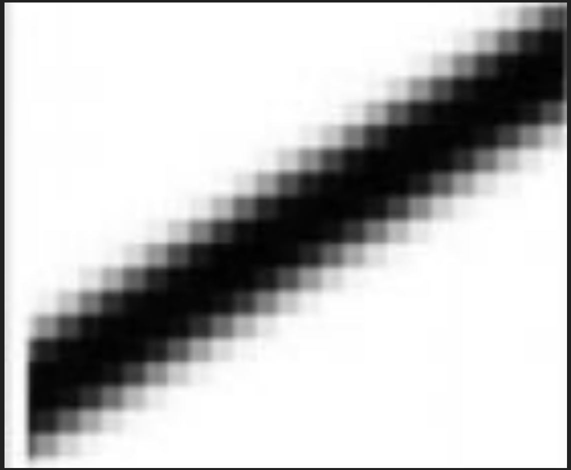
Aliasing & anti-aliasing



Remember that this is a result of rasterization, each pixel is either edge/non-edge.

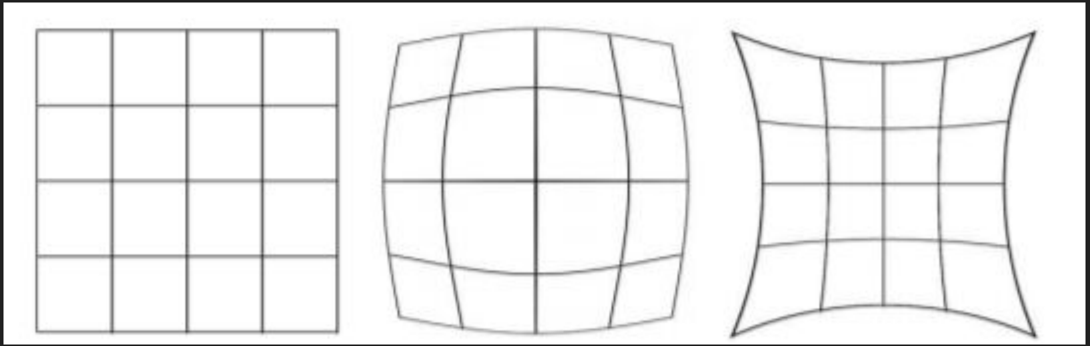
Solve/reduce aliasing of line by multi-sampling.

Example: Mixing the pixels at the border of edge/non-edge region



Optical Distortion

Recall optical distortion by our lens



https://www.youtube.com/watch?v=B7qrgrrHry0

Correcting Optical Distortion

- By Hardware: DLP(Digital Light Process) projects light directly into eyes using MEMS (micro-electromechanical systems).
- By Software: Assume Distortion is always radially symmetric.

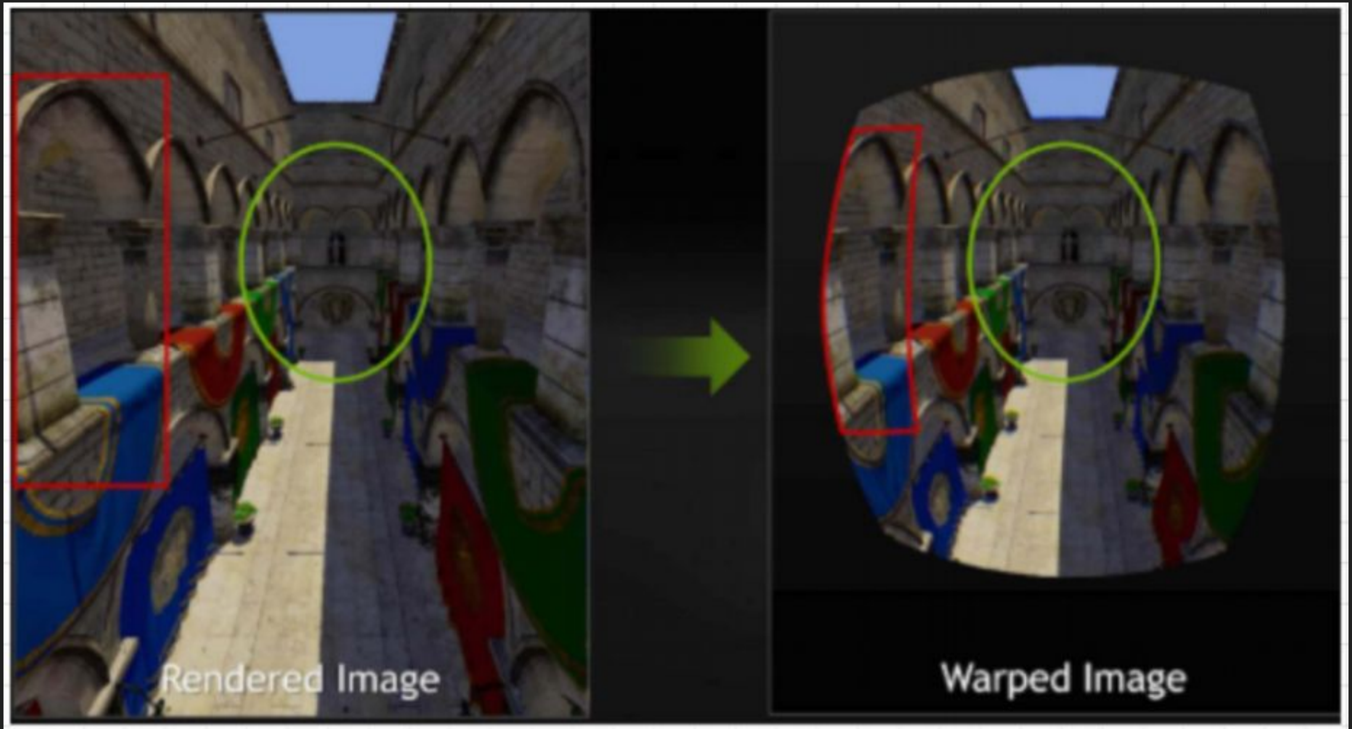
Transform (x, y) coordinate of the image into polar coordinate (r, θ).

In general function f(r, θ), but for our immediate needs no θ dependence.

$r_{distort} = P(r)$, where P(r) is approximate polynomial e.g. $r_{distort} = r + c_1 r^3 + c_2 r^5$

Correct Optical Distortion

Current Approach. Optimization: Multi-resolution shading



Challenge in VR: Latency

The amount of time it takes to update the display in response to head orientation and position.

GPU pipeline has been optimized for triangle outputs, not latency.

Some current methods for latency reduction:

1. Lower the complexity of the virtual world
2. Improve rendering pipeline performance
3. Remove delays along the path from the rendered images to switching pixels
4. Use prediction to estimate future viewpoints and world states
5. Shift or distort the rendered image to compensate for last-moment viewpoint error

Summary of Graphics topics for VR

1. Graphics pipelines: Object-ordered Rendering and Image-ordered Rendering
2. Techniques in Shading models, Depth orders, Interpolation
3. Mappings: Texture, Bump and Normal Mappings
4. Shortcomings of all concepts above and possible solutions

Upcoming Lectures: Practical suggestions for optimizing and designing virtual environments

More physiology and head mounted display tracking

Announcements

- May your future be visualized.
- LaValle, Chapter 7