CSE 167:
Introduction to Computer Graphics
Lecture #18: Bump Mapping

Jürgen P. Schulze, Ph.D.
University of California, San Diego
Fall Quarter 2019

Announcements

- No discussion on Monday
  - But TA and tutor hours in computer labs as usual
- Instructor's office hour as usual Tuesday 1-2pm
- Blog entry #3 due next Wednesday night
- Final project due Thursday at 3pm:
  - Upload your source code to Canvas (one person per team)
  - Link your video to the Youtube playlist
  - Video screening in CSE conference room 1242
  - Project grading in basement labs in two shifts:
    - 4-5pm and 5-6pm
    - Shift assignments to be announced on Monday
- Final project grades to be posted on Canvas next Friday afternoon

Bump Mapping with Normal Maps

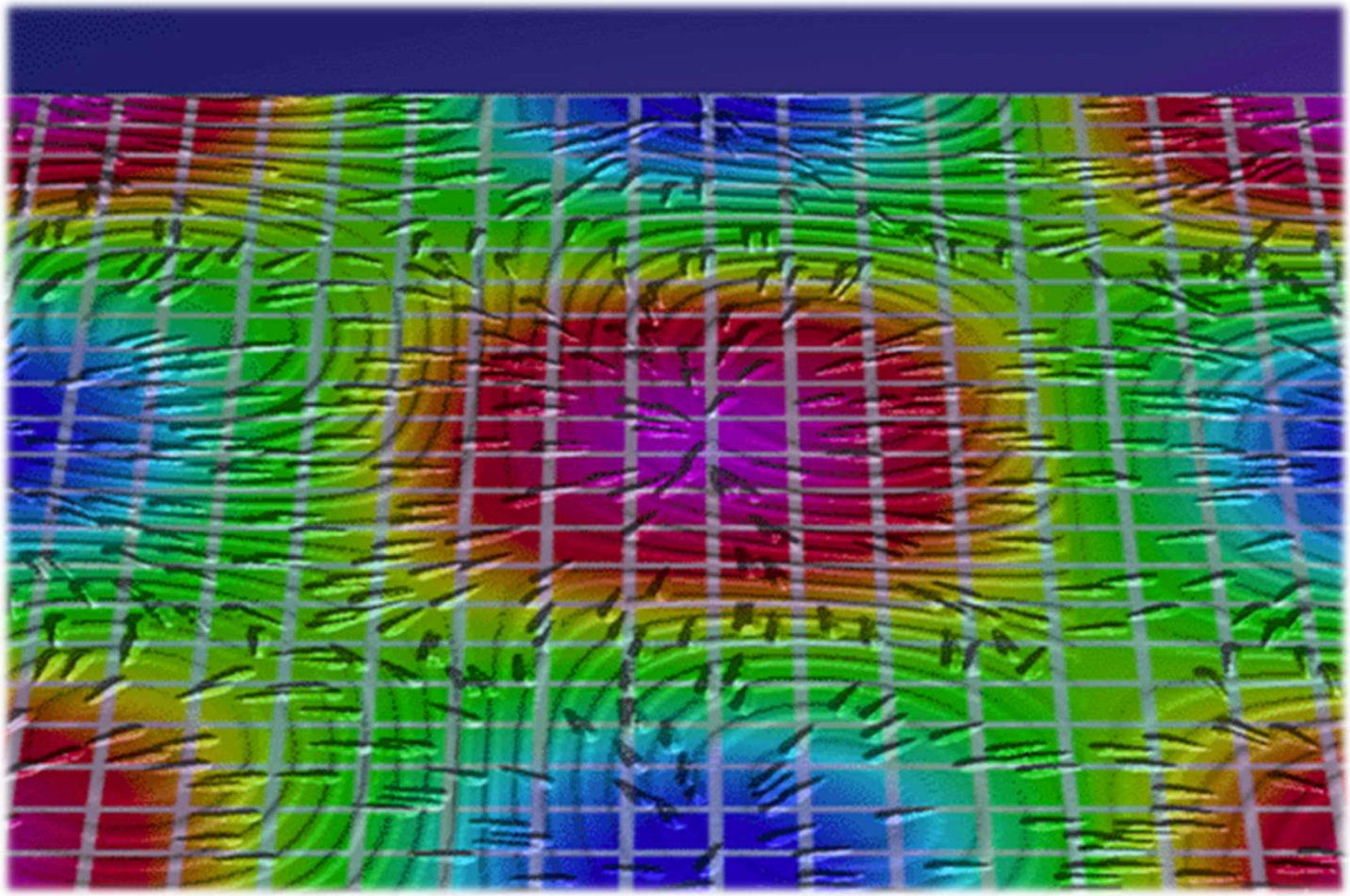
Bump Mapping

- Many textures are the result of small perturbations in the surface geometry
- Modeling these changes would result in an explosion in the number of geometric primitives.
- Bump mapping attempts to alter the lighting across a polygon to provide the illusion of texture.

[This chapter includes slides by Roger Crawfis]

UCSD

Bump Mapping Example

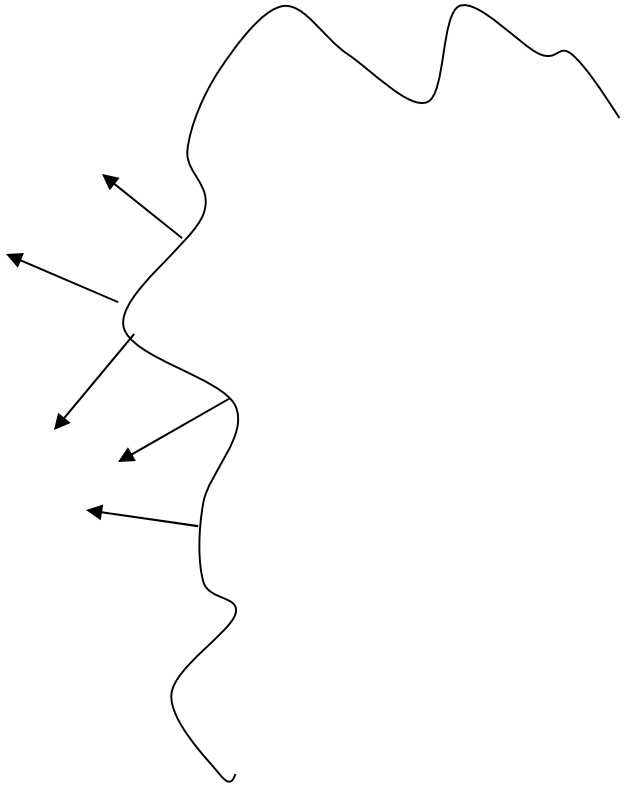


Crawfis 1991

UCSD

Surface Shading

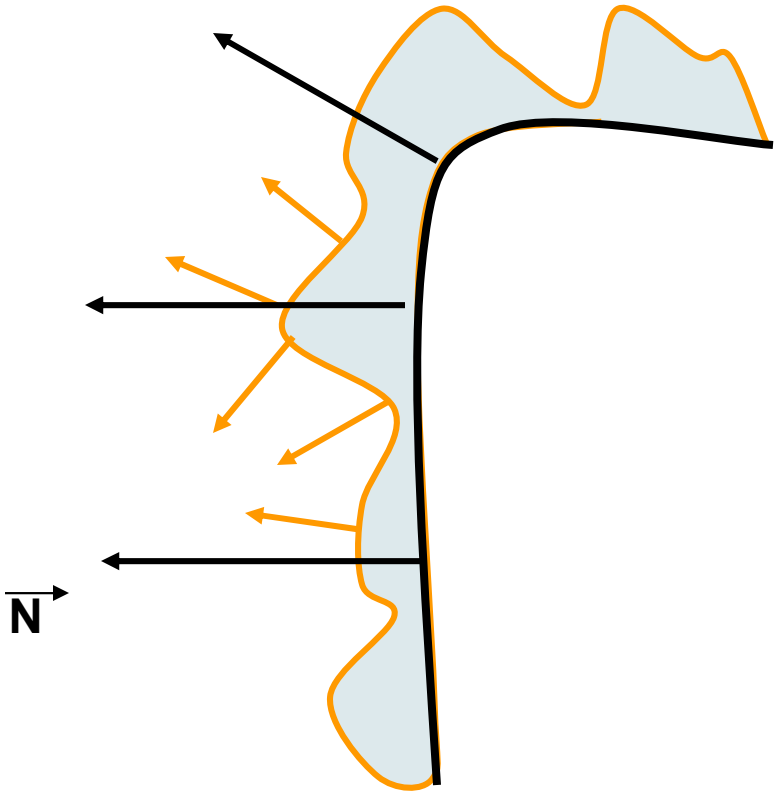
- Consider the lighting for a modeled surface.



UCSD

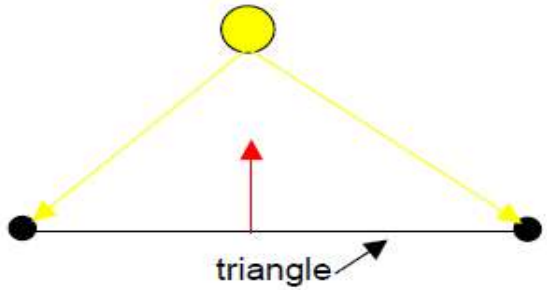
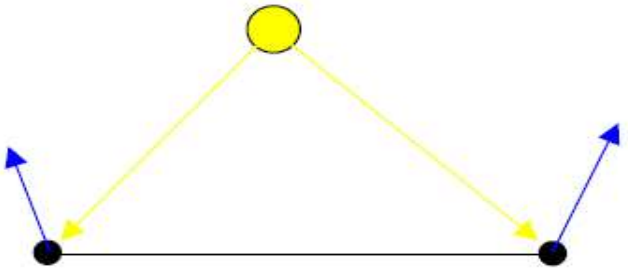
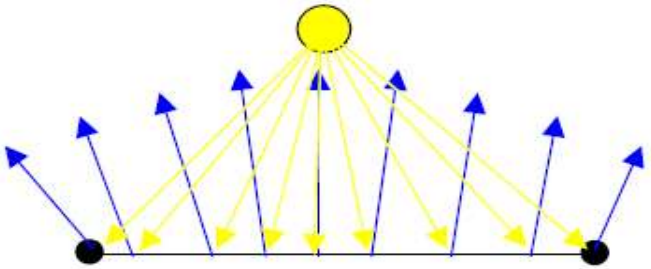
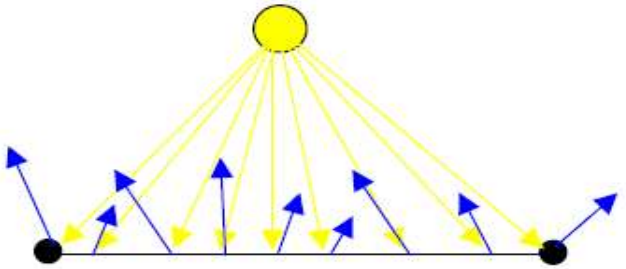
Surface Shading

- We can model this as deviations from some base surface.
- The question is then how these deviations change the lighting.



UCSD

Bump Mapping

Flat shading	Goraud shading
	
Only the first normal of the triangle is used to compute lighting in the entire triangle.	The light intensity is computed at each vertex and interpolated across the surface.
Phong shading	**Bump mapping**
	
Normals are interpolated across the surface, and the light is computed at each fragment.	Normals are stored in a bumpmap texture, and used instead of Phong normals.

UCSD

Bump Mapping with Normal Maps

Just texture mapped

Texture and normal maps

Notice: The geometry is unchanged. There's the same number of vertices and triangles. This effect is entirely from the normal map.

UCSD

Normal Maps

Diffuse Color Texture Map

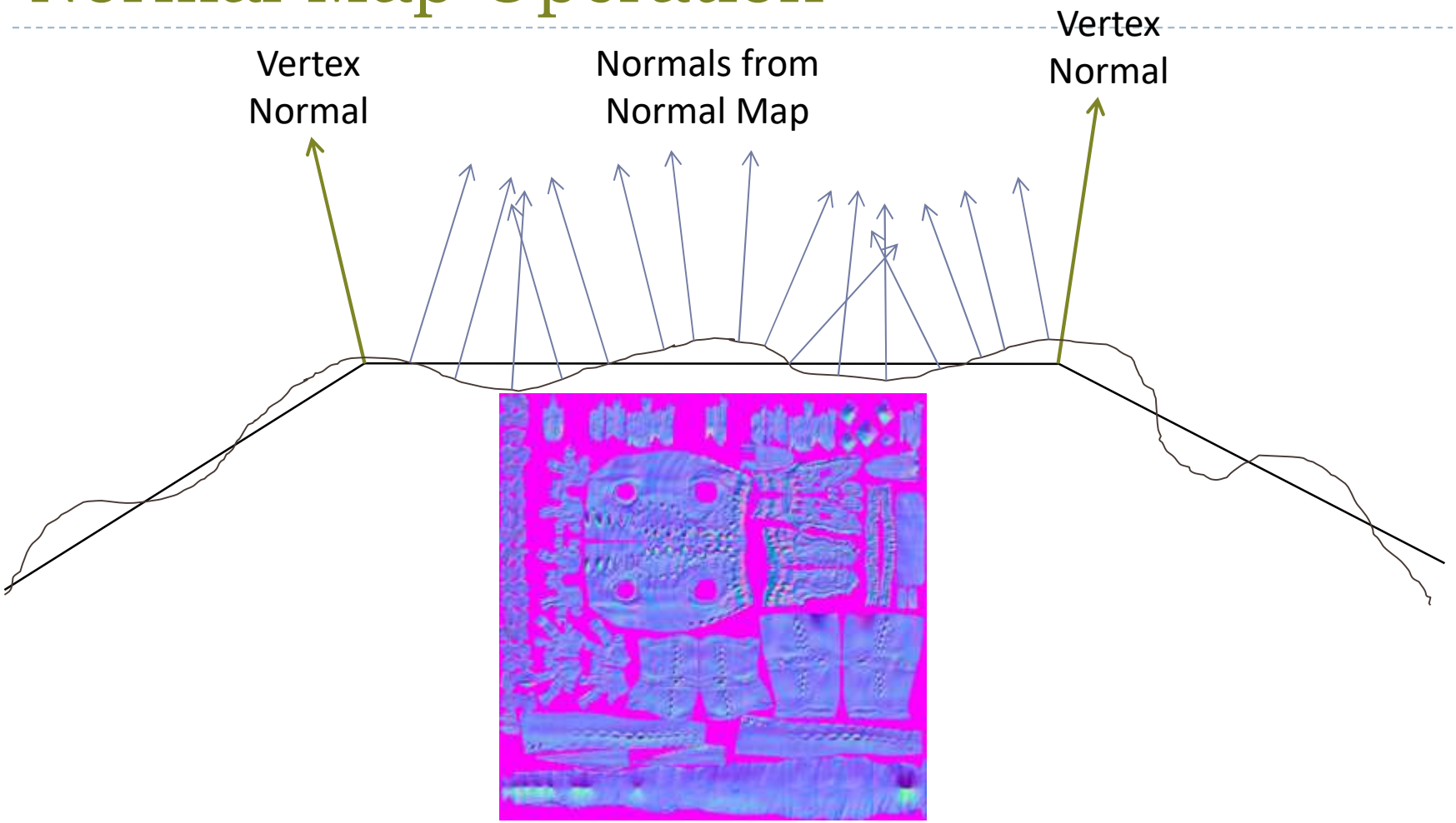
Normal Map

Each pixel represents a normal vector relative to the surface at that point. -1 to 1 range is mapped to 0 to 1 for the texture so normals become colors.

→ Inverse of Normal Coloring

UCSD

Normal Map Operation



For each pixel, determine the normal from a texture image. Use that to compute the color.

UCSD

What's Missing?

- There are no bumps on
  the silhouette of a
  bump or normal-mapped object

→ Displacement Mapping can model that

UCSD

Wrapping it up

Computer Graphics and Related Courses in CSE Department

- CSE 167: Computer Graphics (Fall and Winter)
- CSE 168: Computer Graphics II: Rendering (Ramamoorthi in Winter)
- CSE 169: Computer Animation (Spring?)
- CSE 190: Virtual Reality Technologies (Schulze in Spring)
- CSE 165: 3D User Interfaces (Schulze, not in 2020)
- CSE 152: Intro to Computer Vision (Fall and Winter)
- CSE 166: Image Processing (when?)
- CSE 170: Interaction Design (Klemmer in Winter)

Computer Graphics: State of The Art

- ACM SIGGRAPH Los Angeles 2019 Technical Papers Preview (3'16)
  - https://www.youtube.com/watch?v=EhDr3Rs5fTU
- GDC 2019 Unity Features Reel (3'13)
  - https://www.youtube.com/watch?v=DCq0cndrYDQ
- SIGGRAPH 2019 Computer Animation Festival: Electronic Theater (1'32)
  - https://www.youtube.com/watch?v=UoosuGaBsbs&list=PL7wOdnrlw7uKyART_MfPL6Ur4BHdYcNW6&index=2

Good luck with your final projects!