Wrinkling Coarse Meshes on the GPU

Jörn Loviscach
Hochschule Bremen
jlovisca@informatik.hs-bremen.de

Objective

- Plausible wrinkles and folds
- Fast: on coarse meshes
- Optional control by artist

Demo

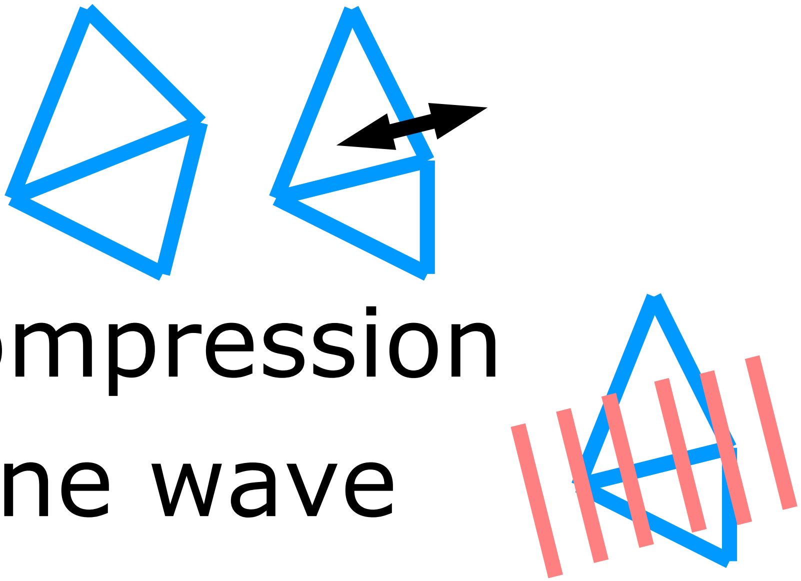
Related Work

Cloth with less than 100% physics:

- Hadap et al., Vis 99: Bump maps invoked according to deformation
- Cutler et al., SCA 05: Wrinkle curves applied according to deformation
- Cordier et al., CGF 05:
  Learn mesh details from physics

... any many more

Basic Idea



- Per vertex:
  - Determine geometric compression
  - Compute plane wave
  - Iteratively align adjacent waves
- Per pixel: blend plane waves; deform normals and tex coords

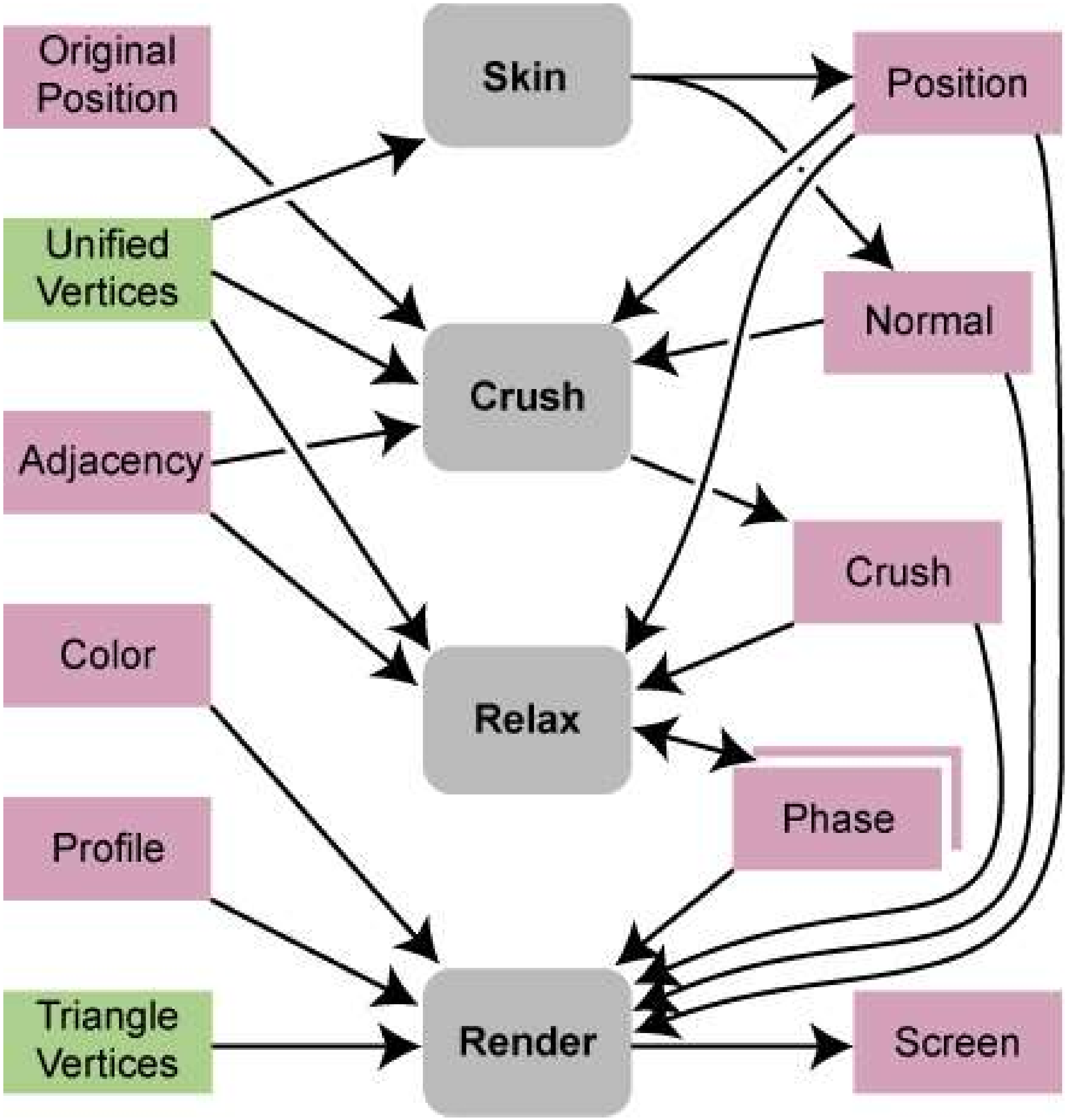
Contributions

- Detailed wrinkles
  from arbitrary deformations;
  fast, robust, controllable,
  no precomputation
- Deformation of normals and texture
  from dynamic height fields
- Optional 3D paint-mode
  for wrinkle density and direction

Outline

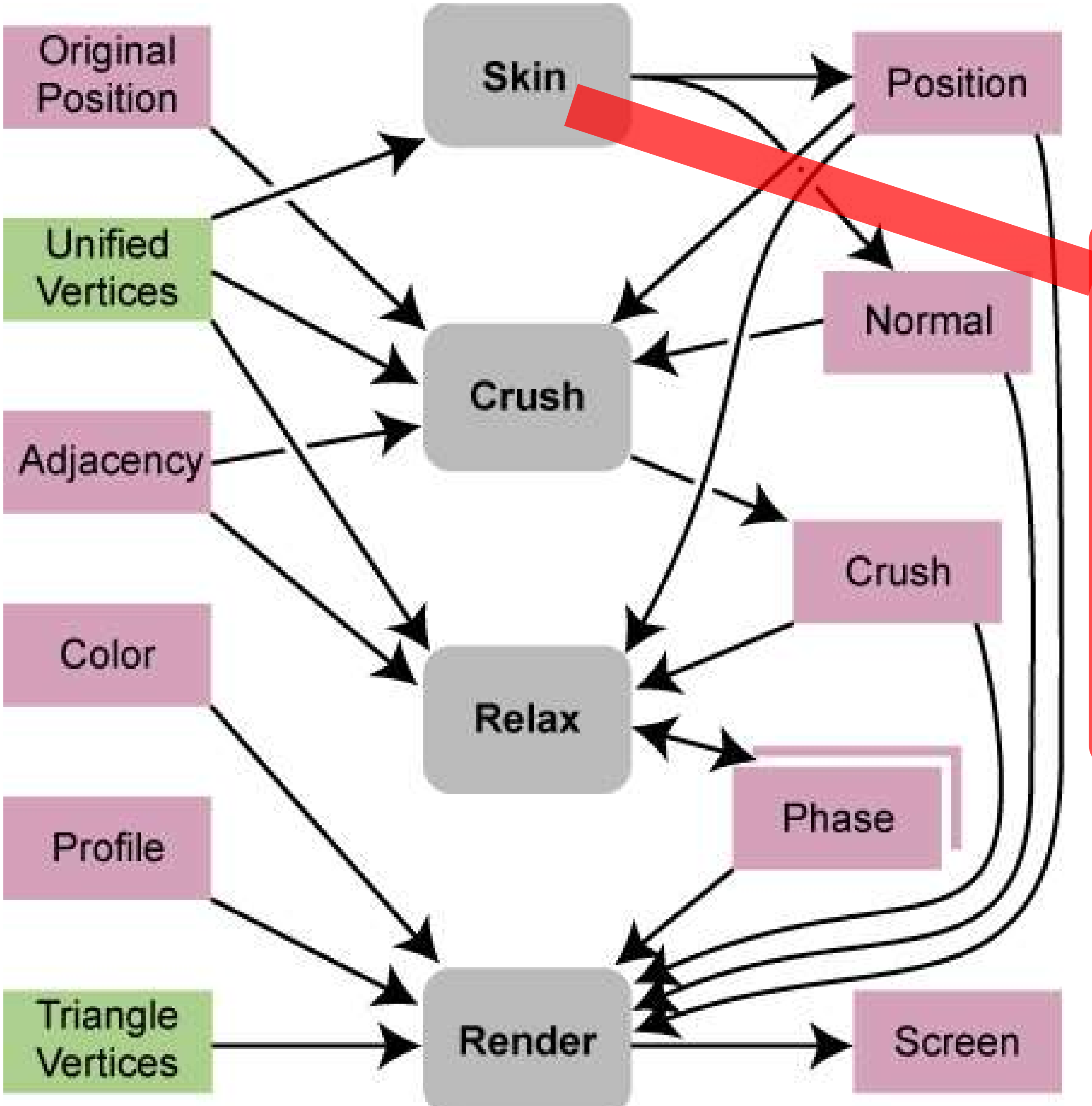
- Overview of the method
- Mesh compression and wrinkle height
- Rendering
- Results
- Outlook

Overview of the Method



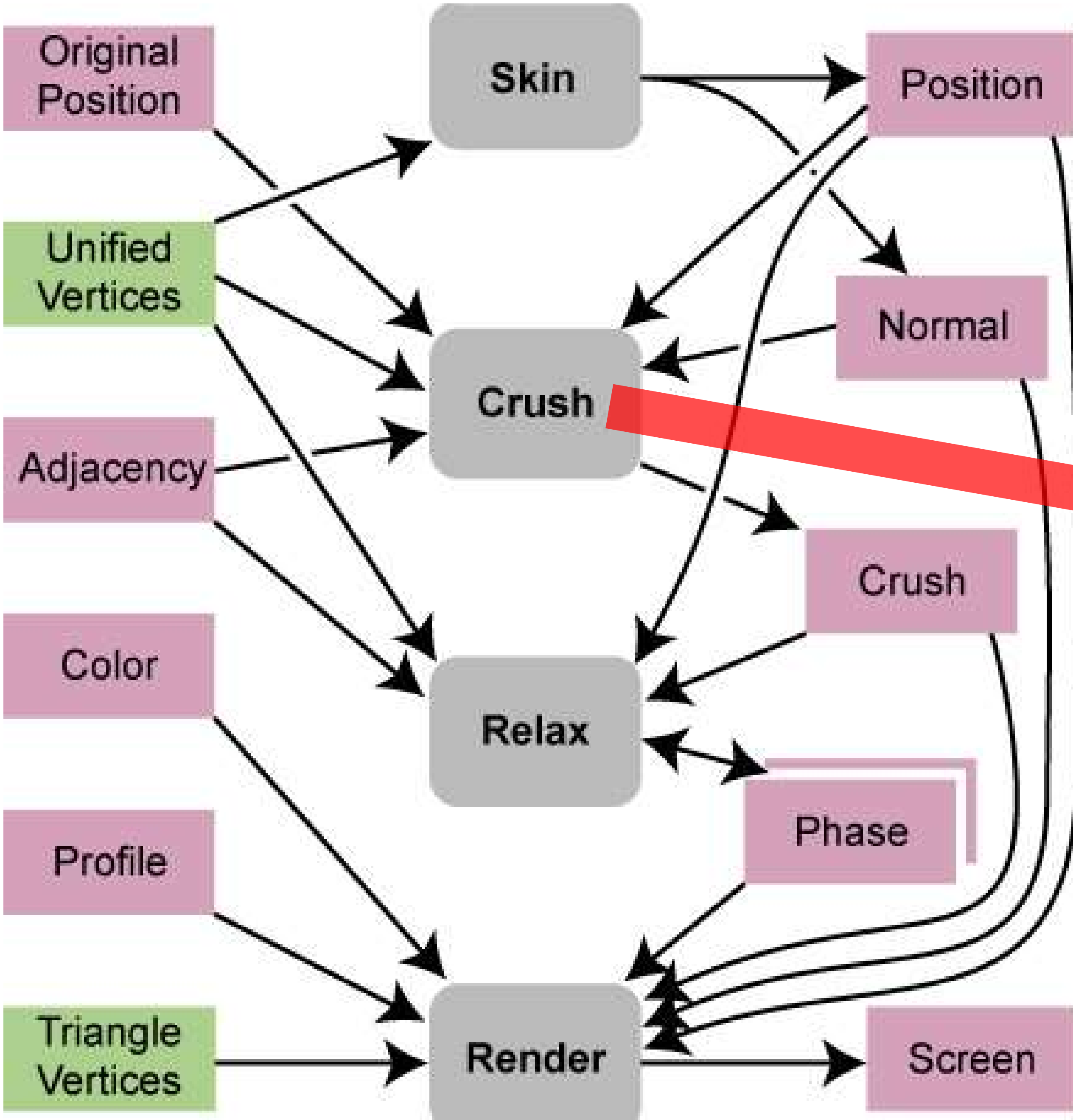
Four rendering passes (= pairs of vertex and pixel shaders)

Overview of the Method



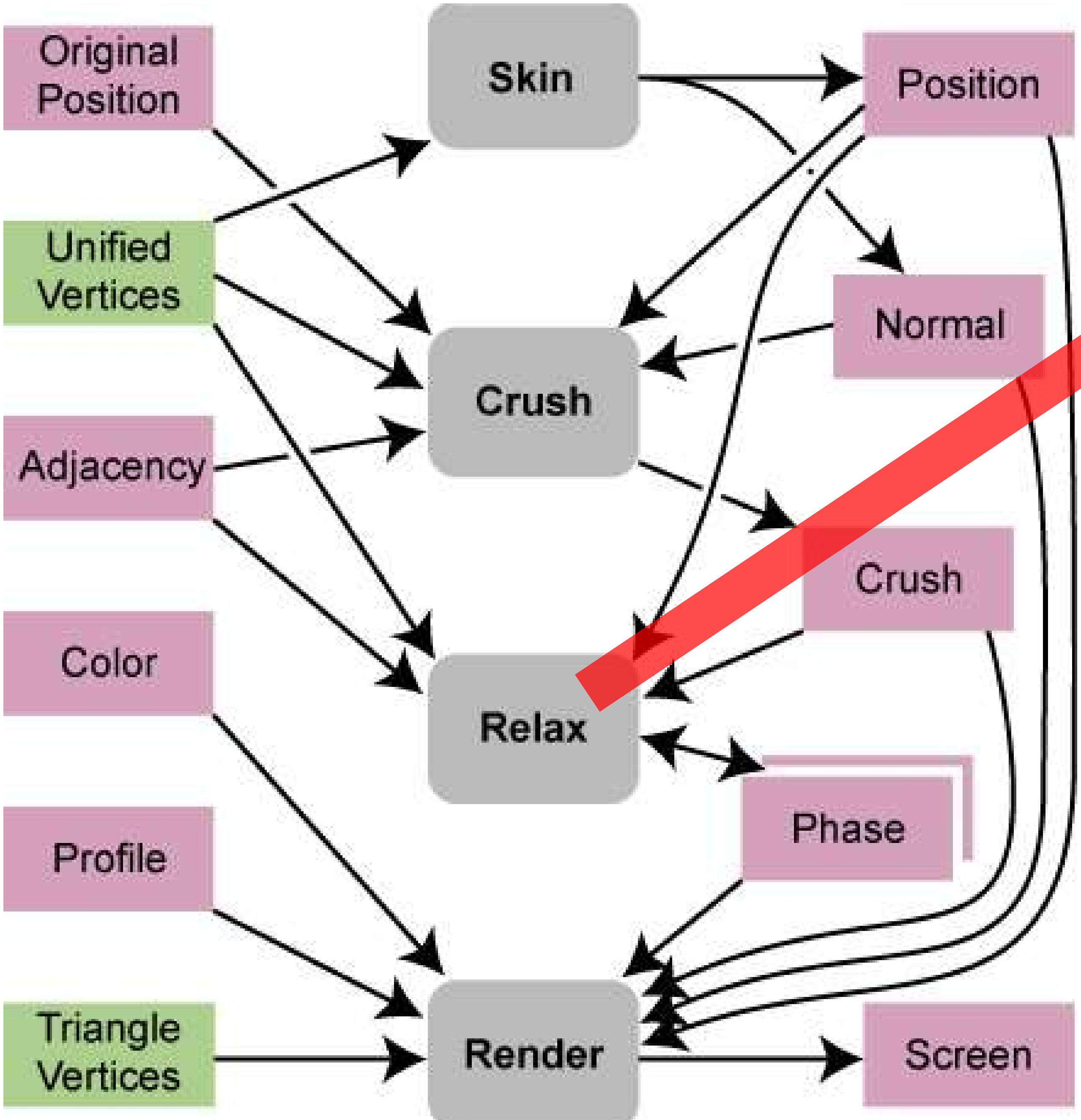
Apply deformation, e.g., skinning. Store new positions and normals.

Overview of the Method



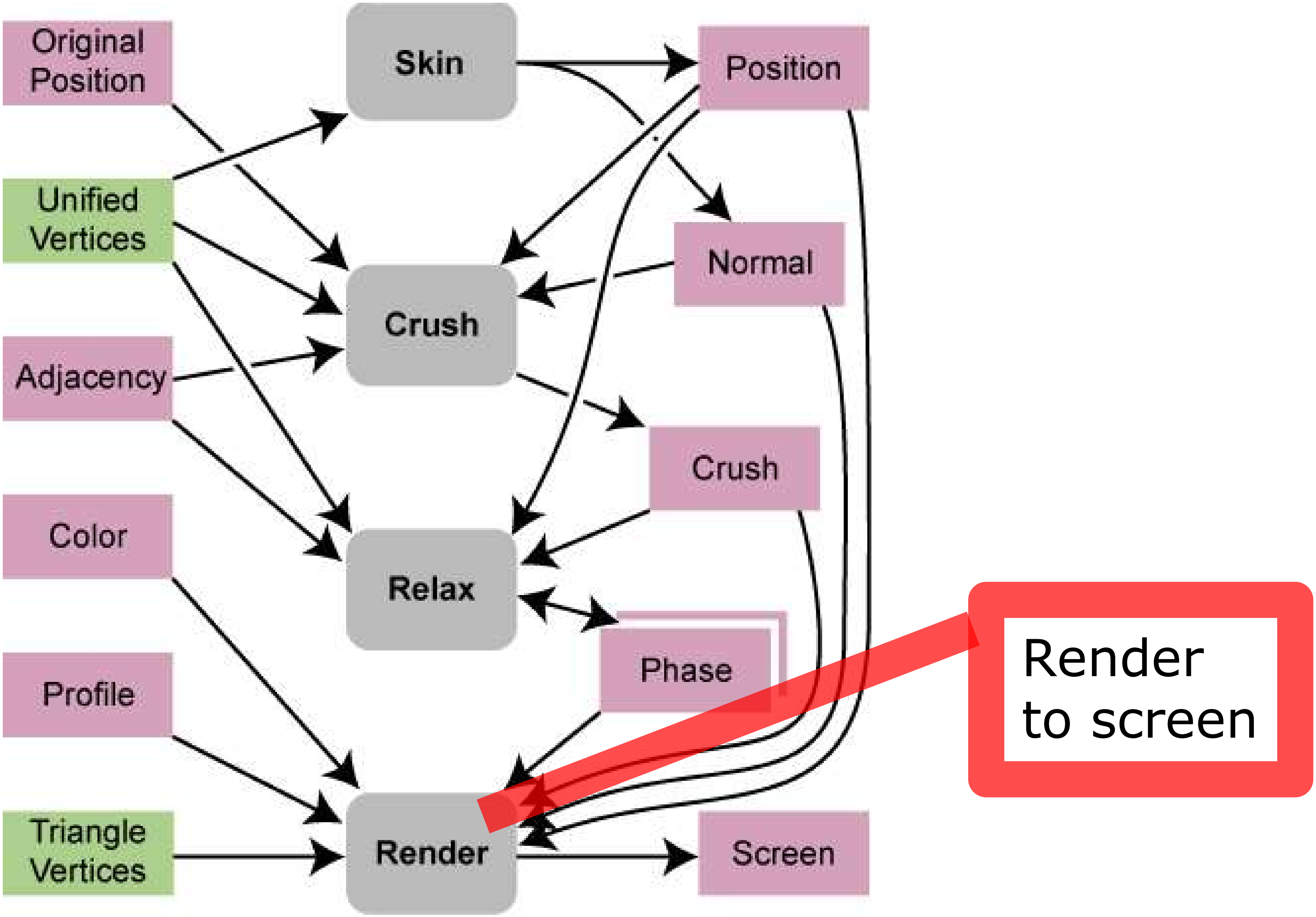
Determine direction and strength of local compression

Overview of the Method



Align phases of local plane waves

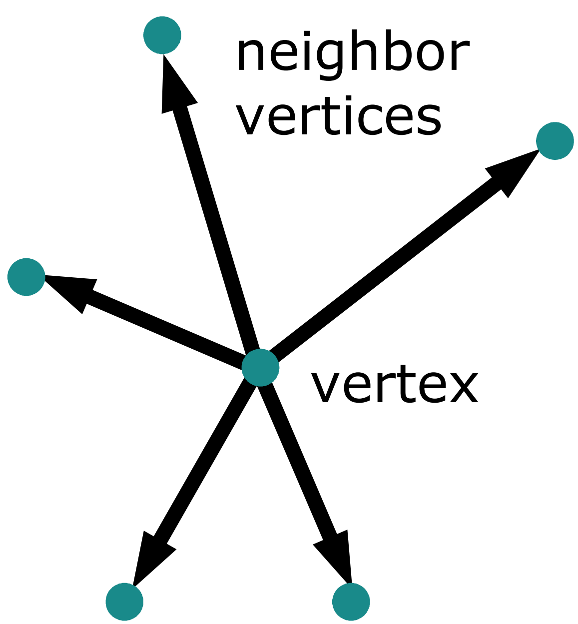
Overview of the Method



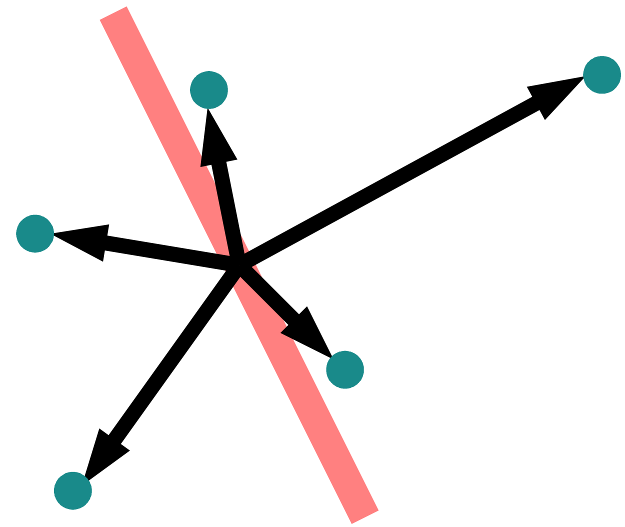
Determining the Compression

For every vertex:

- Linear approx. M of local deformation
- Find direction and amount of strongest compression: eigenanalysis of $M^T M$



before deformation



after deformation

Painting Rest-Pose Wrinkles

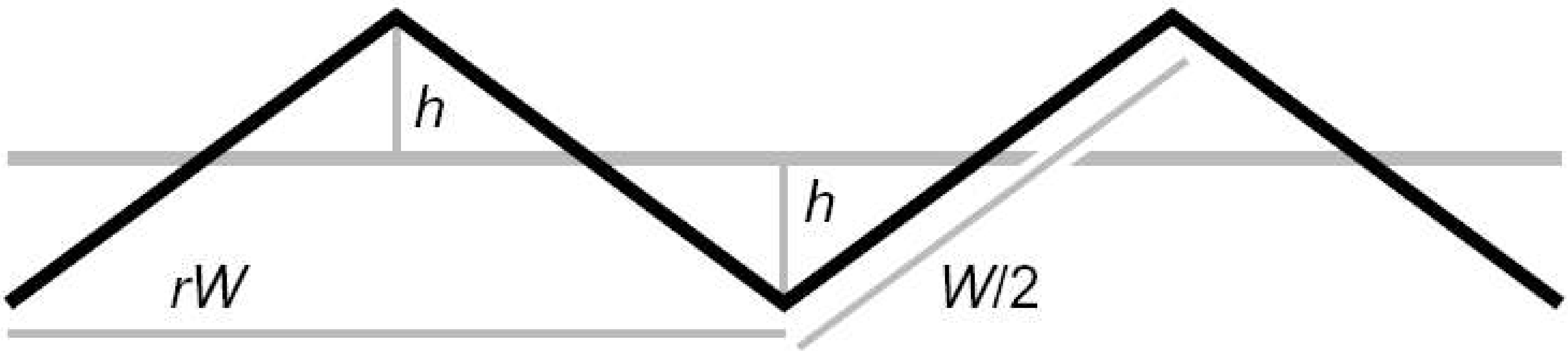
Demo

Integrate user-defined wrinkles:

Bias the computation
of the linear approximation M:
$M \rightarrow M \cdot (1 - \boldsymbol{q} \otimes \boldsymbol{q})$,
$\boldsymbol{q}$ specifies direction and amount,
is defined by 3D painting GUI.

Converting Compression to Height

Simple straight-line approximation:
compression ratio *r*
→ wrinkle amplitude *h*



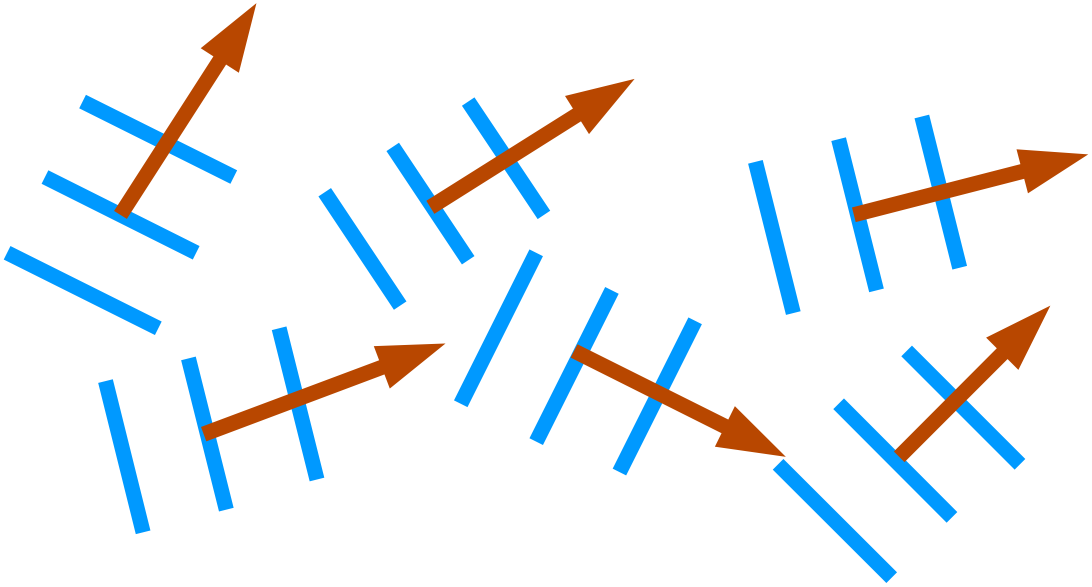
Real-time control on width *W*

Demo

Compression Vector Field (1)

Result at every vertex:

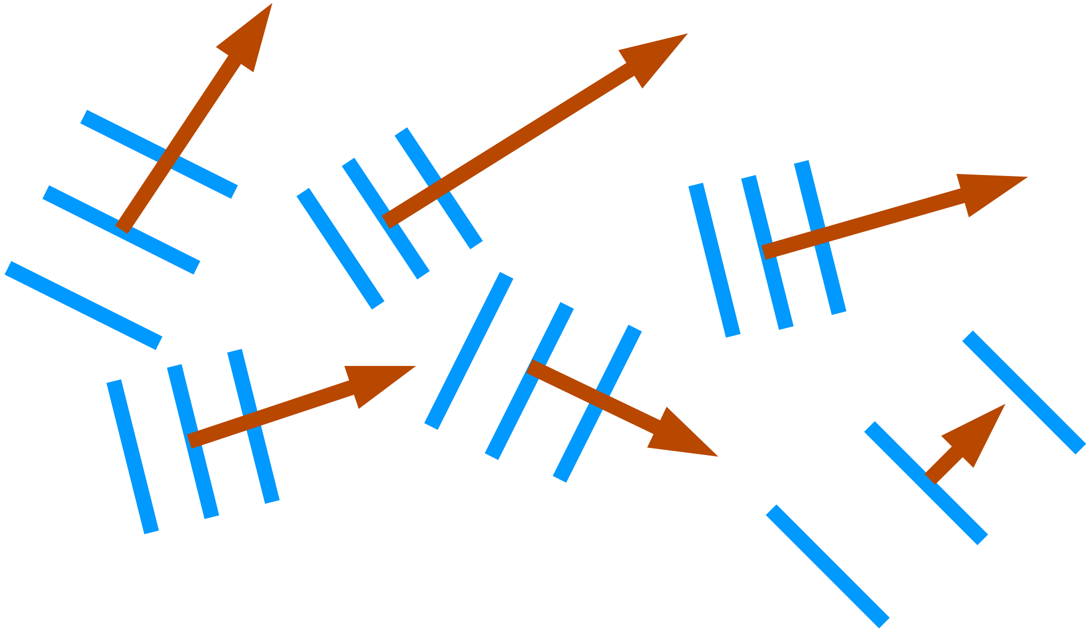
- Tangent unit vector along maximum compression
- Wrinkle amplitude



Local plane waves in rest-pose space

Compression Vector Field (2)

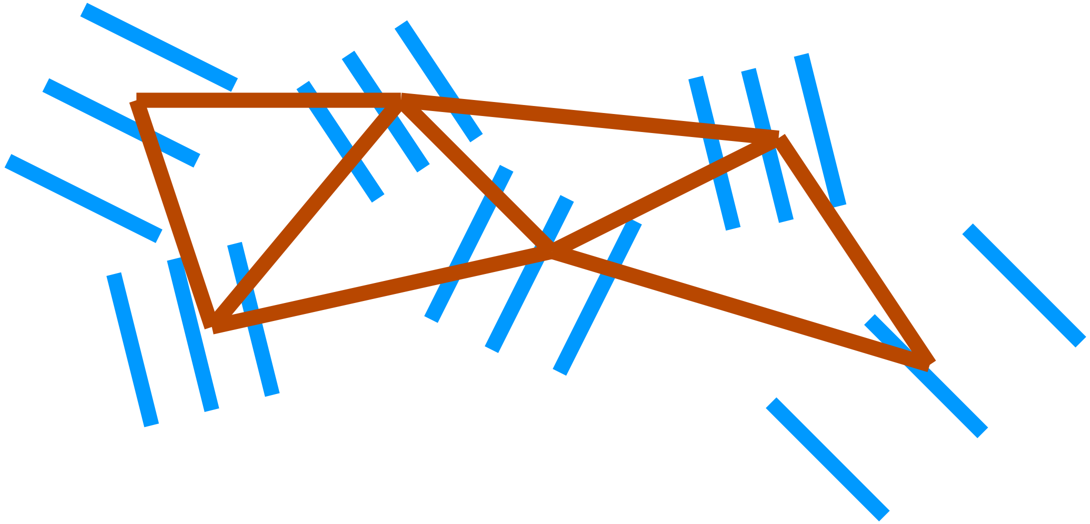
Use $M^{-1\,T}$ to convert
the direction vectors
to wave vectors
in post-deformation space.



→ Waves are compressed with the mesh.

Generating the Height Field (1)

Idea:
Blend the linear waves across every triangle.

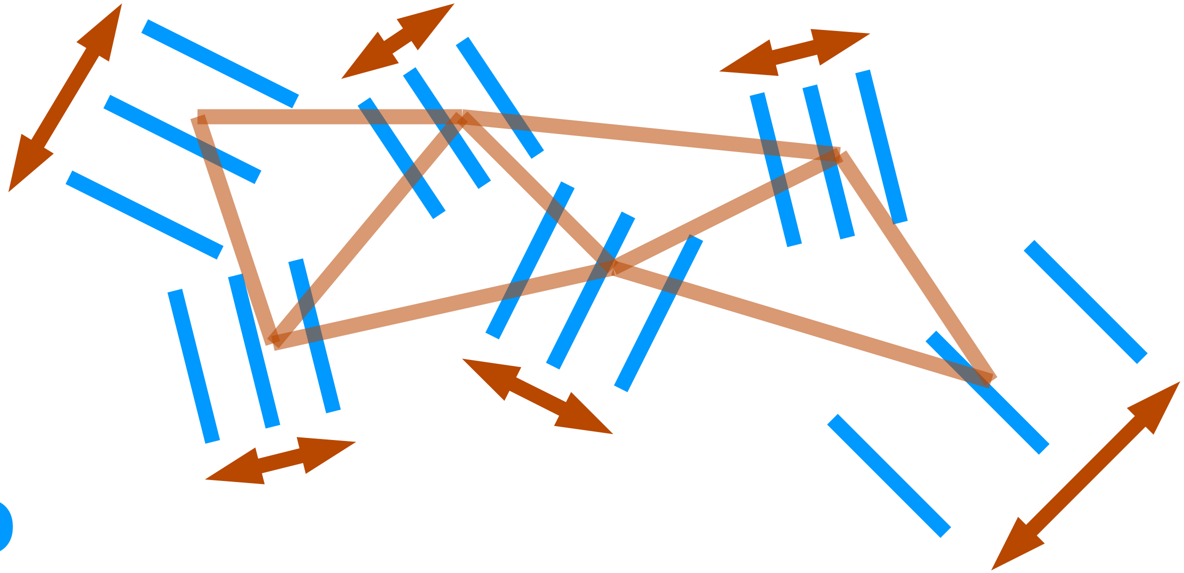


Ooops: Waves aren't aligned.

Generating the Height Field (2)

Problem: The phases of the local plane waves are not yet determined.

Solution: Relax the phases gradually to diminish local misfit.



Demo

Rendering (1)

Render coarse polygons,
fake fine-scale deformation.

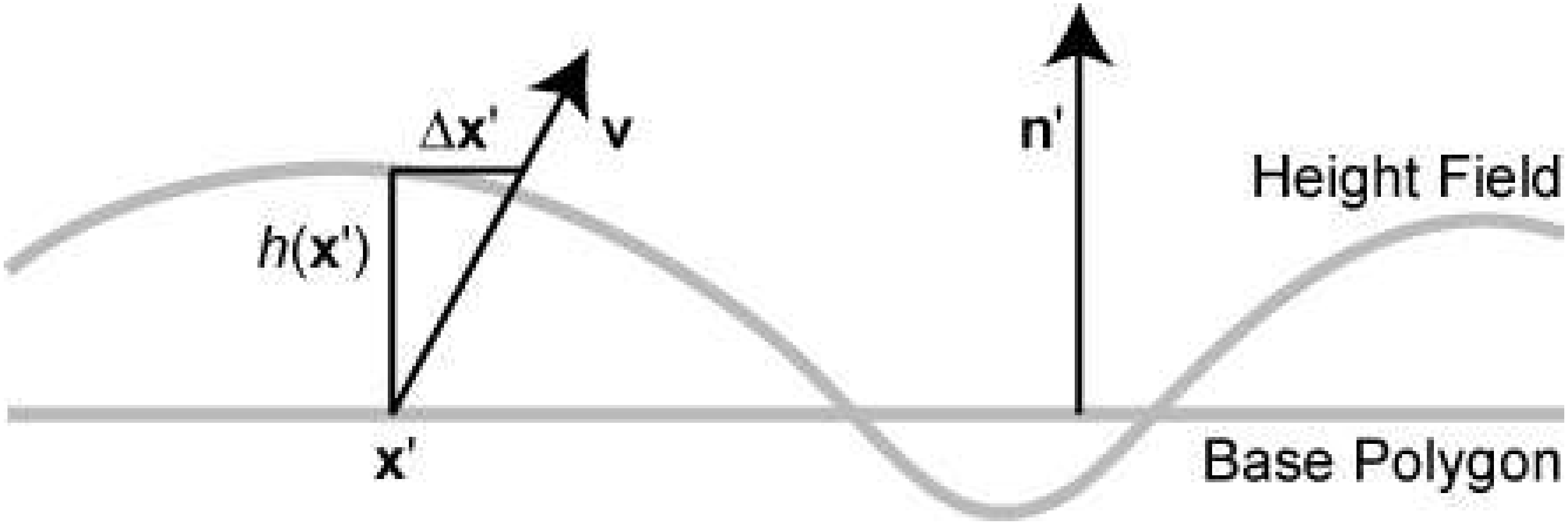
Two issues to address:

- Deform texture
- Adjust normal vector

Demo

Rendering (2)

Texture deformation
similar to Parallax Mapping

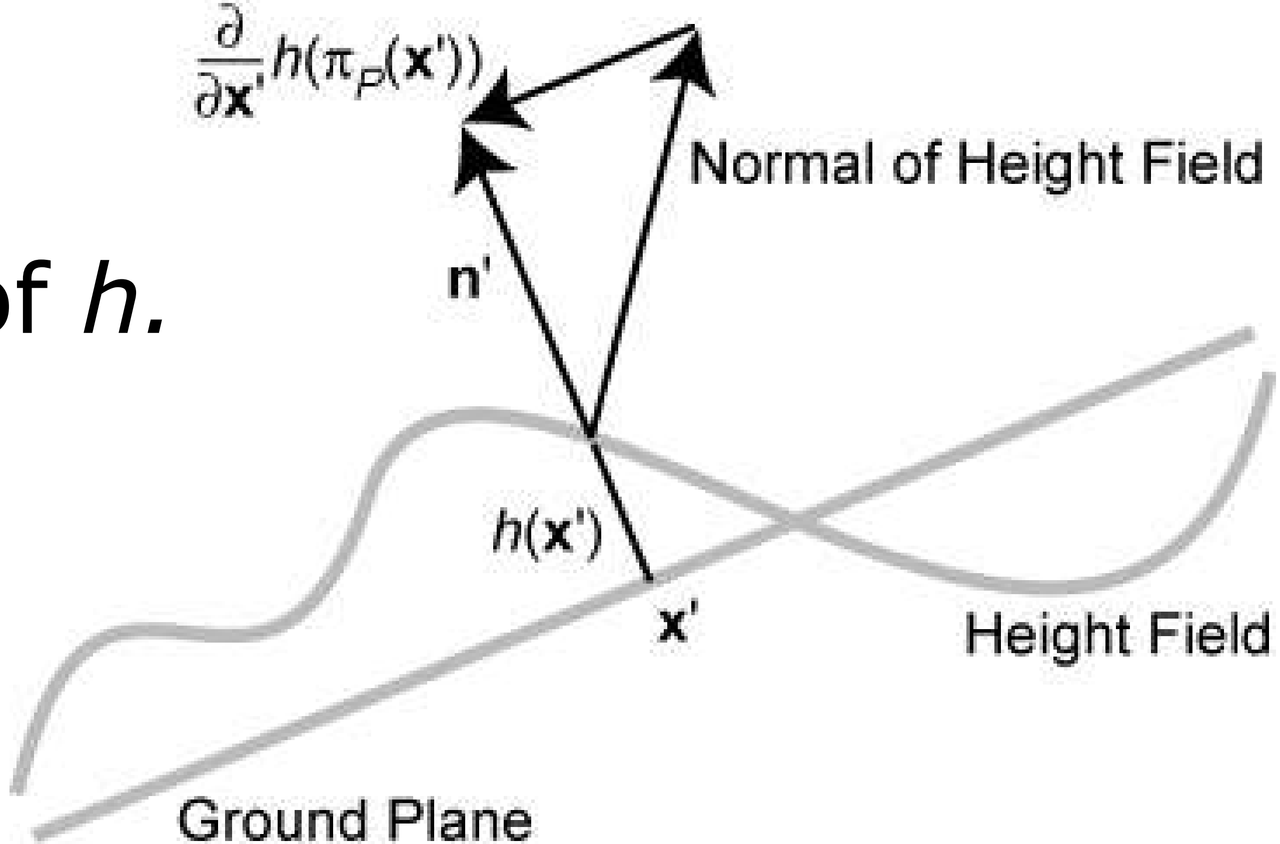


$$\Delta \mathbf{x}' = \left(\frac{\mathbf{v}}{\mathbf{v} \cdot \mathbf{n}'} - \mathbf{n}' \right) h(\mathbf{x}')$$

Rendering (3)

Illumination: Compute normals of dynamic height field

Need the gradient of *h*.



Rendering (4)

Wrinkle profile determined by:

- height: cosine
- gradient: −sine

Replace each
with a 1D texture lookup:
arbitrary profiles

Demo

Results

Name	# Vertices	# Pixels (average)	fps
Shirt	455	≈ 330.000	328
Zeppelin	508	≈ 260.000	540
Curtain	92	≈ 505.000	537

One pixel per vertex

Stage	# Shader instructions		Contribution	
	Vertex	Pixel	A	B
Skin	$13M+20$	2	0.06 ms	0.25 ms
Crush	8	$28N+102$	0.08 ms	1.83 ms
Relax	7	$33N+23$	0.11 ms	1.75 ms
Render	67	47	3.11 ms	5.01 ms
Total time incl. non-shader part			3.45 ms	8.90 ms

M = #bones used per vertex; N = #neighbors

A: 1 Mpix, 100 verts; B: 55 kPix, 50 kVerts

Outlook

- Create curvature-aligned hatching with analytical filtering **Demo**
- Fewer pseudo-textures with D3D 10: stream VS output to memory; access neighbor vertices in GS
- Diamond buckling (cf. this EG: “Virtual Garments”), tension wrinkles
- Real-time cracks (cf. Iben, SCA 06)

Thanks for your attention.

Questions?