Simplification of Meshes with Digitized Radiance

Kenneth Vanhoey [1,2] Basile Sauvage [1] Pierre Kraemer [1]
Frédéric Larue [1] Jean-Michel Dischler [1]

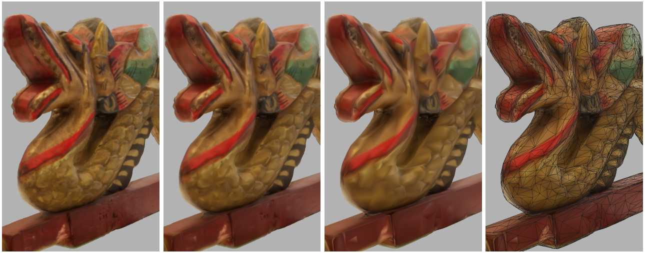
Computer Graphics International
June 24-26, 2015, Strasbourg

1 ICUBE CNRS UNIVERSITÉ DE STRASBOURG

2 Inria informatiques mathématiques

Teaser

Full resolution (921k vertices)



58k 29k 14k +wireframe

Table of contents

Table of contents

Archiving, remote visualization, restoration, . . .

The added value of view-dependent colors

Diffuse color

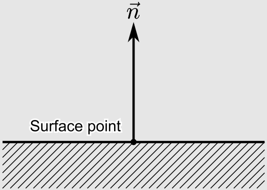
View-dependent color (radiance)

Acquisition & Representation

Geometry

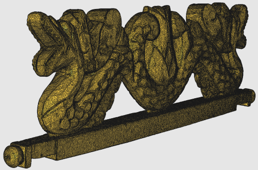
e.g.: $1M$ vertices

Radiance



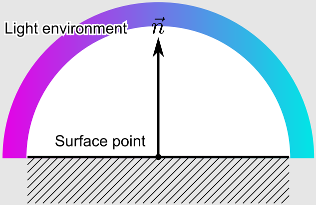
Acquisition & Representation

Geometry



e.g.: $1M$ vertices

Radiance

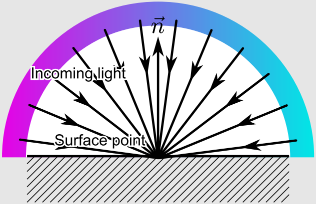


Acquisition & Representation

Geometry

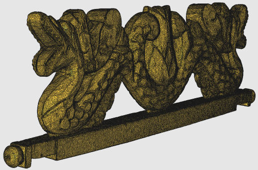
e.g.: 1*M* vertices

Radiance



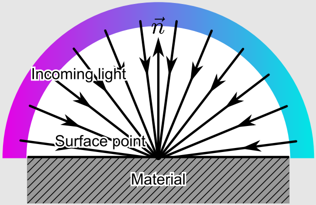
Acquisition & Representation

Geometry



e.g.: 1M vertices

Radiance

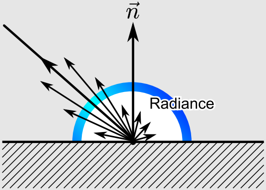


Acquisition & Representation

Geometry

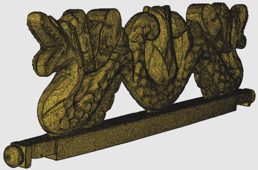
e.g.: 1*M* vertices

Radiance



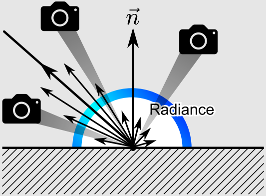
Acquisition & Representation

Geometry



e.g.: $1M$ vertices

Radiance

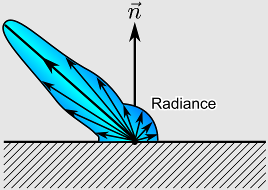


Acquisition & Representation

Geometry

e.g.: $1M$ vertices

Radiance

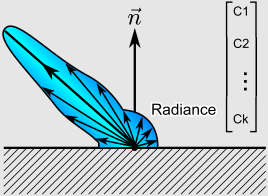


Acquisition & Representation

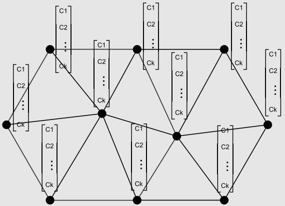
Geometry

e.g.: $1M$ vertices

Radiance



Dense data



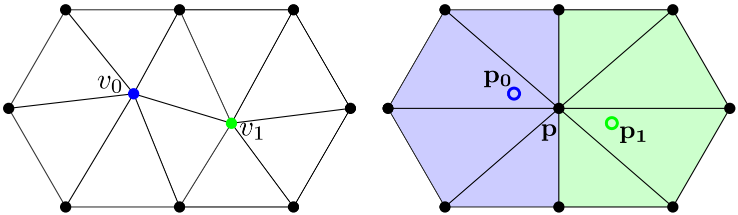
Simplification by:

- Global compression (PCA, quantization, ...)

  [Nishino et al., 2001, Coombe et al., 2005]

- Iterative simplification

Mesh simplification: edge collapse [Hoppe, 1996]



Features

- Ease of implementation
- Topology control
- Local control of the damage

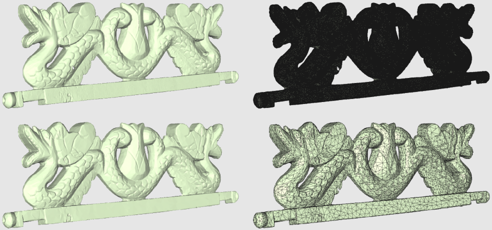
Needs

1. Priority criterion (error metric)
2. Embedding strategy

Mesh simplification: metrics

Many metrics

- **Geometry**: quadric error metric is standard
  [Garland and Heckbert, 1997]
- Vectorial attributes: normals, textures, colors
  [Garland and Heckbert, 1998, González et al., 2007, Kim et al., 2008]
- Radiance is *a function*



Our goal

Design a metric that captures the change in rendered appearance (involving geometry and radiance)

Mesh simplification: metrics

Many metrics

- Geometry: quadric error metric is standard

  [Garland and Heckbert, 1997]

- **Vectorial attributes**: normals, textures, colors

  [Garland and Heckbert, 1998, González et al., 2007, Kim et al., 2008]

- Radiance is *a function*

Our goal

Design a metric that captures the change in rendered appearance (involving geometry and radiance)

Mesh simplification: metrics

Many metrics

- Geometry: quadric error metric is standard

  [Garland and Heckbert, 1997]

- Vectorial attributes: normals, textures, colors

  [Garland and Heckbert, 1998, González et al., 2007, Kim et al., 2008]

- **Radiance** is *a function*

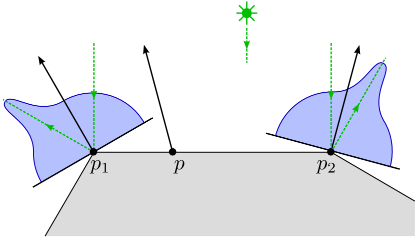
?

Our goal

Design a metric that captures the change in rendered appearance (involving geometry and radiance)

Table of contents

Radiance functions interpolation

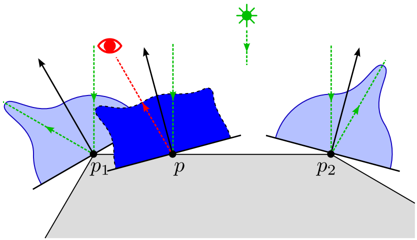


Toy example

Acquired radiance induced from:

- Directionnal light source
- Phong material

RADIANCE FUNCTIONS INTERPOLATION



$$L_{\mathbf{t}}(\mathbf{p},\omega) = \alpha L(\mathbf{p_1},\omega)+(1-\alpha)L(\mathbf{p_2},\omega)$$

TOY EXAMPLE

Acquired radiance induced from:

- Directionnal light source
- Phong material

NAIVE INTERPOLATION

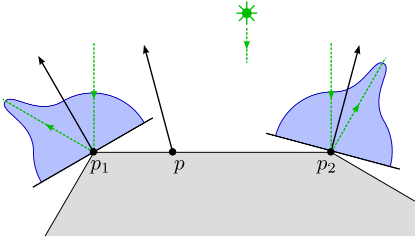
Linear interpolation within face

▶ **Highlights fade out**

Radiance functions interpolation

Demo

Radiance functions interpolation



Toy example

Acquired radiance induced from:

- Directionnal light source
- Phong material

Naive interpolation

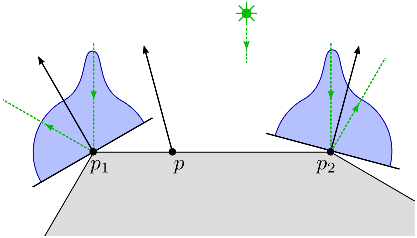
Linear interpolation within face

▶ **Highlights fade out**

Improved interpolation

1. Reflection around normals
2. Linear interpolation within face
3. Reflection around interpolated normal

Radiance functions interpolation



Toy example

Acquired radiance induced from:

- Directionnal light source
- Phong material

Naive interpolation

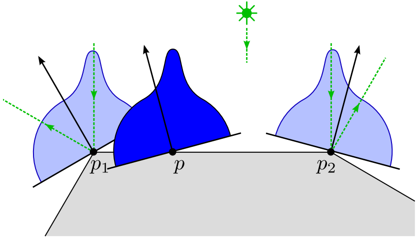
Linear interpolation within face

▶ **Highlights fade out**

Improved interpolation

1. Reflection around normals
2. Linear interpolation within face
3. Reflection around interpolated normal

Radiance functions interpolation



$$\widetilde{L}_{\mathbf{t}}(\mathbf{p},\omega) = \alpha\widetilde{L}(\mathbf{p_1},\omega)+(1-\alpha)\widetilde{L}(\mathbf{p_2},\omega)$$

Toy example

Acquired radiance induced from:

- Directionnal light source
- Phong material

Naive interpolation

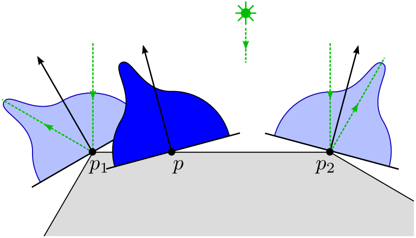
Linear interpolation within face

▶ **Highlights fade out**

Improved interpolation

1. Reflection around normals
2. Linear interpolation within face
3. Reflection around interpolated normal

Radiance functions interpolation



$$\widetilde{L}_{\mathbf{t}}(\mathbf{p},\omega) = \alpha\widetilde{L}(\mathbf{p_1},\omega)+(1-\alpha)\widetilde{L}(\mathbf{p_2},\omega)$$

Toy example

Acquired radiance induced from:

- Directionnal light source
- Phong material

Naive interpolation

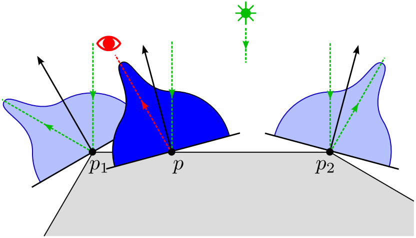
Linear interpolation within face

▶ **Highlights fade out**

Improved interpolation

1. Reflection around normals
2. Linear interpolation within face
3. Reflection around interpolated normal

Radiance functions interpolation



$$\widetilde{L}_{\mathbf{t}}(\mathbf{p},\omega) = \alpha\widetilde{L}(\mathbf{p_1},\omega)+(1-\alpha)\widetilde{L}(\mathbf{p_2},\omega)$$

Toy example

Acquired radiance induced from:

- Directionnal light source
- Phong material

Naive interpolation

Linear interpolation within face

▶ **Highlights fade out**

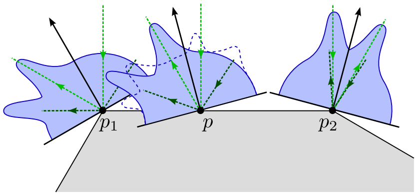
Improved interpolation

1. Reflection around normals
2. Linear interpolation within face
3. Reflection around interpolated normal

Radiance functions interpolation

Demo

Radiance functions interpolation



Improved interpolation

- Reflection around normal
- Linear interpolation within face
- Reflection around interpolated normal

Generalization

- Any distant lighting environment
- Limited to materials reflecting in the mirror direction

Table of contents

Spatially continuous radiance

Based on interpolation, but now at any point p in space:

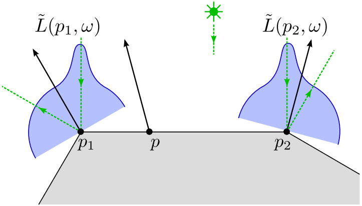
$$\widetilde{L}_{\mathbf{t}}(\mathbf{p},\omega) = \widetilde{L}(\mathbf{p_1},\omega) + \nabla_p \widetilde{L}_t(\omega) \cdot (\mathbf{p} - \mathbf{p_1})$$

where $\nabla_p \widetilde{L}_t(\omega)$ is the gradient of $\widetilde{L}$ w.r.t. triangle t

Distance between radiance functions

$$d\Big(L(\mathbf{p_1},\cdot), L(\mathbf{p_2},\cdot)\Big) = \left\|\widetilde{L}(\mathbf{p_1},\cdot) - \widetilde{L}(\mathbf{p_2},\cdot)\right\|_{L_2(\Omega)}$$

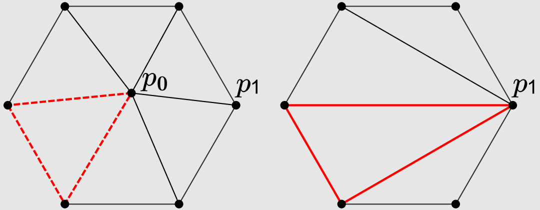
where $\left\|f\right\|_{L_2(\Omega)} = \sqrt{\frac{1}{2\pi}\int_\Omega f^2}$



Tricky implementation: functions in non-aligned local frames

Error metric: measure the *visual* difference

Examples of configurations



Available tools

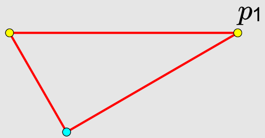
- Extrapolation
  $L_t(p, w) = f(\nabla_p \widetilde{L_t})$
- Distance metric
  $d(L_1, L_2)$

Collapse error

$$E = \sum_{\mathbf{t}} Area(\mathbf{t}) \quad \times \quad ?$$

Error metric: measure the *visual* difference

Examples of configurations



Available tools

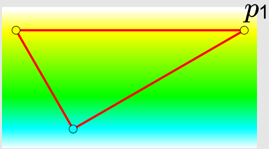
- Extrapolation
  $L_t(p, w) = f(\nabla_p \widetilde{L_t})$
- Distance metric
  $d(L_1, L_2)$

Collapse error

$$E = \sum_{\mathbf{t}} Area(\mathbf{t}) \quad \times \quad ?$$

Error metric: measure the *visual* difference

Examples of configurations



Available tools

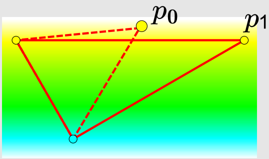
- Extrapolation
  $L_t(p, w) = f(\nabla_p \widetilde{L_t})$
- Distance metric
  $d(L_1, L_2)$

Collapse error

$$E = \sum_{\mathbf{t}} Area(\mathbf{t}) \quad \times \quad ?$$

Error metric: measure the *visual* difference

Examples of configurations



Available tools

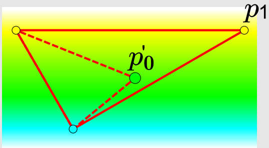
- Extrapolation
  $L_t(p, w) = f(\nabla_p \widetilde{L_t})$
- Distance metric
  $d(L_1, L_2)$

Collapse error

$$E = \sum_{\mathbf{t}} Area(\mathbf{t}) \quad \times \quad ?$$

Error metric: measure the *visual* difference

Examples of configurations



Available tools

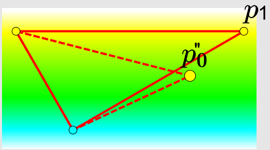
- Extrapolation
  $L_t(p, w) = f(\nabla_p \widetilde{L_t})$
- Distance metric
  $d(L_1, L_2)$

Collapse error

$$E = \sum_{\mathbf{t}} Area(\mathbf{t}) \quad \times \quad ?$$

Error metric: measure the *visual* difference

Examples of configurations



Available tools

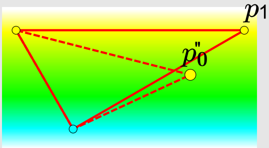
- Extrapolation
  $L_t(p, w) = f(\nabla_p \widetilde{L_t})$
- Distance metric
  $d(L_1, L_2)$

Collapse error

$$E = \sum_{\mathbf{t}} Area(\mathbf{t}) \quad \times \quad ?$$

Error metric: measure the *visual* difference

Examples of configurations



Available tools

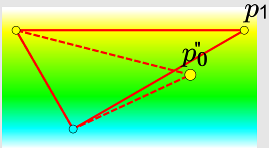
- Extrapolation
  $L_t(p, w) = f(\nabla_p \widetilde{L_t})$
- Distance metric
  $d(L_1, L_2)$

Collapse error

$$E = \sum_{\mathbf{t}} Area(\mathbf{t}) \quad d(L(\mathbf{p_0}, \omega), L_{\mathbf{t}}(\mathbf{p_0}, \omega))$$

Error metric: measure the *visual* difference

Examples of configurations



Available tools

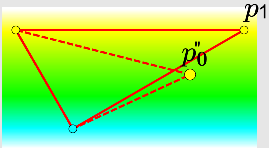
- Extrapolation
  $L_t(p, w) = f(\nabla_p \widetilde{L_t})$
- Distance metric
  $d(L_1, L_2)$

Collapse error

$$E = \sum_{\mathbf{t}} Area(\mathbf{t}) \quad d(L(\mathbf{p_0}, \omega), L_{\mathbf{t}}(\mathbf{p_0}, \omega)) \quad + \quad QEM$$

Error metric: measure the *visual* difference

Examples of configurations



Available tools

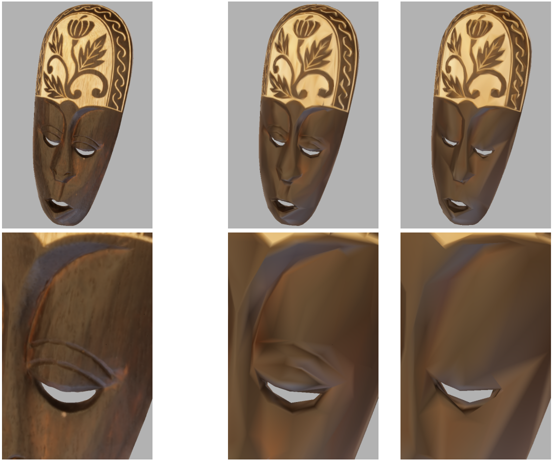
- Extrapolation
  $L_t(p, w) = f(\nabla_p \widetilde{L_t})$
- Distance metric
  $d(L_1, L_2)$

Collapse error

$$E = \sum_{\mathbf{t}} Area(\mathbf{t}) \quad d(L(\mathbf{p_0}, \omega), L_{\mathbf{t}}(\mathbf{p_0}, \omega)) \quad + \quad QEM$$

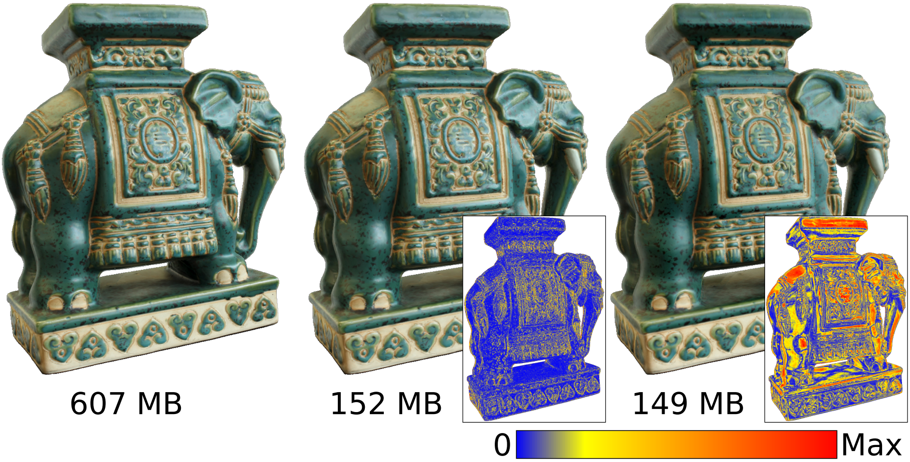
Tricky implementation: no closed form for some bases

Comparison to color metrics



193k 3k vertices

Spatial versus directional simplification



Application example: Progressive Meshes

Demo

Table of contents

Wrap-up

Contributions

- Simplification metric that respects the visual appearance
- Improved rendering (interpolation)
- Based on formulas on radiance functions: gradient, distance

Results

- On colors: compete with state-of-the-art
- On radiance: higher quality than directional reduction
- Nice applications: *e.g.*, interactive navigation

Feature: robustness

- Mesh scale (*e.g.*, for animation)
- Basis functions (*e.g.*, spherical harmonics)
- Color space (*e.g.*, Lab)
- ...

Future Work

Compression

- Numerical evaluation against global compression methods (*e.g.*, PSNR)
- Global compression methods (lossy or lossless) can be added upon our simplification

Textures

- Storage
- Filtering
- Mip-mapping

Simplification of Meshes with Digitized Radiance

Kenneth Vanhoey [1,2] Basile Sauvage [1] Pierre Kraemer [1]
Frédéric Larue [1] Jean-Michel Dischler [1]

Computer Graphics International
June 24-26, 2015, Strasbourg

1 ICUBE CNRS UNIVERSITÉ DE STRASBOURG

2 Inria informatiques mathématiques

Coombe, G., Hantak, C., Lastra, A., and Grzeszczuk, R. (2005).
Online construction of surface light fields.
In *Proceedings of the Sixteenth Eurographics conference on Rendering*, EGSR'05, pages 83–90.

Garland, M. and Heckbert, P. S. (1997).
Surface simplification using quadric error metrics.
In *Proceedings of SIGGRAPH '97*, pages 209–216. ACM Press.

Garland, M. and Heckbert, P. S. (1998).
Simplifying surfaces with color and texture using quadric error metrics.
In *Proceedings of the conference on Visualization '98*, VIS '98, pages 263–269, Los Alamitos, CA, USA. IEEE Computer Society Press.

González, C., Castelló, P., and Chover, M. (2007).
A texture-based metric extension for simplification methods.
In *Proceedings of the Second International Conference on Computer Graphics Theory and Applications (GRAPP)*, pages 69–76.

Hoppe, H. (1996).
Progressive meshes.
In *Proceedings of SIGGRAPH '96*, pages 99–108. ACM Press.

Kim, H. S., Choi, H. K., and Lee, K. H. (2008).
Mesh simplification with vertex color.
In *Proceedings of the 5th international conference on Advances in geometric modeling and processing*, GMP'08, pages 258–271, Berlin, Heidelberg. Springer-Verlag.

Nishino, K., Sato, Y., and Ikeuchi, K. (2001).
Eigen-texture method: Appearance compression and synthesis based on a 3D model.
IEEE Trans. Pattern Anal. Mach. Intell., 23(11):1257–1265.