Simplification Error Metrics

Richard Zhang

CMPT 464/764: Geometric Modeling in Computer Graphics
Lecture 7
Why measure errors?

- **Evaluation**: to know the quality of your approximation
  - E.g., mesh simplification, compression, remeshing, smoothing, etc.

- **Local guidance**: to control the flow of an algorithm
  - E.g., error measures define cost of a simplification operator and it drives both *outer and inner optimizations*

- **Reconstruction loss**: to train a neural network

- ...
Mesh LOD: inner vs. outer optimization

- Basic paradigms for LOD mesh modeling:
  - Decimation: remove one or more basic primitives at a time and fix the mesh
  - Refinement: the inverse

- Two optimizations:
  - **Outer optimization**: in which order should the primitives be decimated or refined
  - **Inner optimization**: how to fix mesh geometry after the operation
  - Both depend on simplification error metric

Edge Collapse
Vertex Split
Topics to cover

- **Object-space geometric errors**
  - Vertex-to-plane distances, surface-to-surface distances, etc.
  - View-independent

- **Image-space errors** (brief)
  - View-dependent: what really matters but need projection/rendering
  - Example: does image of the approximation cover the right pixels and have the right silhouette as the original?

- **Perception-motivated measures** (brief)
Geometric error basics

- With **point correspondence** given:
  - $L_2$ (Euclidean), $L_1$ (Manhattan distance), $L_p$, $L_\infty$ (max norm)

- Between two point sets (without point correspondence *a priori*) – **symmetric Hausdorff distance**:

  $$H(A, B) = \max (h(A, B), h(B, A)),$$

  where

  $$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

- In general, $h(A, B) \neq h(B, A)$
Problems with Hausdorff distance

- Unpaired or multiply paired points
- Not suitable for comparing attributes, e.g., between two faces — these should be done on corresponding points
- Relatively expensive to compute
Maximum ($L_\infty$) vs. average errors ($L_2$)

- Maximum error: **guaranteed error bounds**

- Focusing on reducing maximum error may result in undesirably large average error [Erikson 00]

- Focusing on minimizing average error only may allow small number of artifacts (large error) — thus often combine with **regularization**, e.g., for ensure smoothness of solution

- Maximum error leads to non-linear optimization

- Average error, if modeled in least squares, leads to linear system
Incremental vs. total error

- Incremental: measure errors of new mesh against the current one along the iterative process
  - Cheaper but errors can accumulate if not controlled properly

- Total error: always measure errors against original (full-res) mesh
  - This is what really matters
  - More expensive to compute, but there are good remedies, e.g., quadric errors presented later
Object-space geometric error

- Vertex-to-vertex (v2v) distances
- Vertex-to-plane (v2p)
- Vertex-to-surface, especially vertex-to-triangle

Distance computations also key in other applications, e.g., collision detection (closest point detection)

[C. Ericson – Real-Time Collision Detection]
Vertex-to-vertex distance (v2v)

- Takes maximum distance between original vertices and new vertice(s) obtained from decimation
- Key: find the (right) vertex correspondence
- Only a sampled view of surfaces: what really matters is distance between surfaces
- For example, cannot always capture geometry change — think about an edge swap
Vertex-to-plane (v2p) distance

- Less expensive to compute than v2v distance
  - Signed distance from point \( v = (x, y, z, 1) \) to plane \( p \), characterized by \((A, B, C, D)\) with **unit normal** \((A, B, C)\) is
    \[
    Ax + By + Cz + D
    \]
  - Distance between two vertices \((x_1, y_1, z_1)\) and \((x_2, y_2, z_2)\) is
    \[
    \sqrt{ (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 }
    \]
  - Take **max or total distance** from vertex to supporting planes
  - The planes are supporting planes of mesh triangles
  - For mesh vertex, store set of **supporting planes** at that vertex
Vertex-to-plane distance: pro and con

- **Max** v2p distance can be inaccurate

- Key advantage: **efficiency**

  For example, **error Quadrics** — store information about one or more planes (QSLIM [Garland and Heckbert 1997]), more next lecture
Vertex-to-surface distance

- What really matters!
- Map original point set to closest points on the simplified surface
The Metro tool

- Distance between **aligned meshes** [Cignoni et al. 98]
- Most frequently used mesh-to-mesh distance tool
- Kind of like Hausdorff distance, **but point-to-surface**
  - Sample one mesh surface and map each sample point to closest face on the other mesh surface
  - Report maximum or average distance
  - For symmetry, swap the two meshes
Sub-problem: vertex-to-$\Delta$ distance (aside)

- Brute-force: If projection $Q$ of $P$ is inside the triangle, then return $|PQ|$; otherwise, test against each edge.

- To be more efficient,
  - Determine whether $P$ lies in a vertex or edge Voronoi region first.
  - If triangle non-obtuse, then sufficient to test sidedness with respect to triangle edges.
Screen-space geometric error

Can be used to select appropriate discrete LOD based on a screen-space error tolerance

Key is how to map object space error $\varepsilon$ to screen-space error $p$. An example:

$$ p = \frac{\varepsilon}{w} = \frac{\varepsilon}{2d \tan(\theta/2)} $$

But possible underestimate of screen-space error, which is bad. Always safer to over-estimate error, but not by too much.
Image-space error driven LOD

- Basic idea: render both original model and simplified model and use their image distance as error measure.
- To encompass multiple views, place, e.g., 20, cameras nicely spaced over a sphere – need then to render 20 images.

[Lindstrom and Turk 00]
Image-space error metric

- **Pros**
  - Image-space error is what really mattered
  - Combined effect of all sources of errors related to image production, e.g., lighting

- **Cons**
  - Too costly
    - Need lots of rendering every step
    - Significantly slower than the slowest geometric error measure algorithm (with 20 – 30 cameras for rendering)
  - Error dependent upon rendering parameters
Perceptual issues for LOD

- Why do we care about visual perception?
  - Our perception is imperfect – so take advantage of this to maximally utilize often limited rendering resources
    - e.g., to minimize popping, let it happen in our visual peripheral
  - Perception-based or “visual” scheme for selecting LOD aims at what and how we see (the how: one step beyond image-space)

- Visual perception is hard and not so well-understood
  - Object-space geometry errors still most common
The human visual system (aside)
Some facts (aside)

- The eye is about 2.4 cm in diameter
- Retina: thin tissue at the back of the eye – this is where light is sensed to produce a visual image …
- Photoreceptors (rods and cones): process light impulses and pass information to the brain through optic nerves
- Our ability to perceive spatial details depends …
On … (aside)

- Background illumination
  - Weber’s Law: minimum detectable difference in luminance increases linearly with background illumination, at daylight levels
- Interference of other senses – cannot count right when hearing beeps
- Age, near/far sightedness, and experience
- Color vision – 10% men RG color blind, but extremely rare in women
- Stereoscopic vision – 10% stereo-blind
- Emotional state 😊
Limits of vision

- **Visual acuity**
  - Retina can resolve detail down to the size of around 0.5 minutes of arc (1 degree = 60 minutes of arc)
  - Rule of thumb: 1 degree of arc ≈ width of thumb at arm’s length

- **Peripheral Vision**
  - Highest sensitivity to spatial detail at fovea (central 4 to 5 degrees of vision)
  - 35-fold reduction from fovea to periphery

- **Motion sensitivity**
  - Eye is less sensitive to details in an object moving across retina
  - Fast moving objects become “blurred”
  - We are sensitive of motion in visual periphery – evolutionary instinct 😊
Some perceptual LOD criteria

- Distance or size of projected area - how far is the object – take advantage of visual acuity
- Velocity – how fast does the object move?
- Eccentricity – is the object in visual periphery
- Depth of field – how far is the object from our focus point
Distance LOD

- LOD based on distance between object and eye, i.e. coarser resolution for distant geometry
Size LOD

- LOD based on projected screen size (or area) of an object
  - Requires 2-D projection
  - Use bounding volumes
Eccentricity LOD

- LOD selected based on degree to which an element exists in the visual periphery
- Humans resolve less detail there
  - More retinal photoreceptors (rods/cones) towards fovea
  - Retinal and cortical cell receptive field sizes increases linearly with eccentricity
  - 80% of cortical cells devoted to central 10 degrees of vision
- Use **eye/head tracking** system
Velocity LOD

- LOD based on angular velocity of a moving object across the visual field
  - Faster moving objects in lower resolution
Under binocular vision, both eyes converge on objects at certain distance in order to focus retinal image.

Objects in front or behind fusion area are unfocused: double images.

Must track both eyes accurately to estimate fusion region.