A Future of Polygon Reduction

Changkun Ou
changkun.de/s/polyred4us
@changkun

IDC 2020 Autumn
Venice, Italy
Oct 7, 2020
Previously on Polygon Reduction (Polyred)...

Previously on Polygon Reduction (Polyred)...

A Glimpse into Advances of Mesh Representation Learning

Changkun Ou
https://changkun.de/s/polyred1step
IDC 2019 Spring
Bernried, Germany
Apr 3, 2019

chankun.de/s/polyred1step (2019a)
How?
Previously on Polygon Reduction (Polyred)...

A Glimpse into Advances of Mesh Representation Learning

Changkun Ou
https://changkun.de/s/polyred1step

IDC 2019 Spring
Bernried, Germany
Apr 3, 2019

changkun.de/s/polyred1step (2019a)
L3 → L2 → L1 → L2
Tri2simNet
Tri2quadNet
Quad2simNet

L3  L2  L1
Tri2DelauneyNet

L3  L2  L1
Tri2gridNet
Previously on Polygon Reduction (Polyred)...

A Glimpse into Polygon Mesh Representation

Changkun Ou
https://changkun.de/s/polyred1step
IDC 2019 Spring
Bernried, Germany
Apr 3, 2019

Simplicity is Complicated:
On the Balance of Performance and Knobs

Changkun Ou
https://changkun.de/s/polyred2what
IDC 2019 Autumn
Vienna, Austria
Oct 9, 2019

changkun.de/s/polyred2what (2019b)
No Free Lunch: Simplicity is complicated

\[ \sum_{\text{models}} \text{cost}(\text{models, methods, target}) = \text{constant} \]

Nothing Works As Expected
Doesn't Work At Scale

Impractical

VS

The Ultimate Polyred Program
Target \#Poly: 

Reduce!
"Polyred is solved!!"

640k#poly
Ground truth

29k#poly
Handcraft
40 hours+

119k#poly
My approach
~30s

126k#poly
Blender ([Hoppe96]+[Garland97])
~4min
"Polyred is solved!!"
"Polyred is solved at scale!!"

Blood Circulation System
"Polyred is solved at scale!!"
Previously on Polygon Reduction (Polyred)...
Previously on Polygon Reduction (Polyred)...
Worker Multiplexing 🔥🔥🔥

Naïve Polyred

// Polyred reduces number of polygons // while preserving local topologies.
func Polyred(m *Mesh, c *Criteria) {
    for !m.Eval(c) {
        local := m.Pick()
        local.Simplify()
    }
}

// SGD implements mini-batch // stochastic gradient descent.
func SGD(m *Model, d *Dataset) {
    for !m.Converge() {
        miniB := d.Batch()
        m.GradientDescent(miniB)
    }
}

Inspiration

func Polyred(m *Mesh, c *Criteria) {
    // SPEEDUP1: build workQ concurrently
    for local := m.Pick(); local != nil; {
        sched.Submit(func() {
            quality, ok := local.Eval(c)
            if ok {
                workQ.Push(quality, local)
            }
        })
        local = m.Pick()
    }
    sched.Wait() // sync barrier

    // SPEEDUP2: run workQ concurrently
    for w := workQ.Pop(); w != nil; {
        sched.Submit(w.Simplify)
        w = workQ.Pop()
    }
    sched.Wait() // sync barrier
}

var sched Sched // M:N work-steal scheduling
func (s *Sched) Submit(f func()) { ... }
Under the Hood
Under the Hood: Mesh Energy Minimization

$$\min_{\psi} \sum_{e_{ij} \in S} |\psi_j - \psi_i|^2$$
Under the Hood: Mesh Energy Minimization

$$\min_{\psi} \sum_{e_{ij} \in S} |\psi_j - \psi_i|^2$$

$$\min_{\psi} \int |\nabla \psi|^2 \quad \text{(Dirichlet Energy)}$$
Under the Hood: Mesh Energy Minimization

$$\Delta (\psi_i - \psi_j) = 0$$
Under the Hood: Mesh Energy Minimization
Previously on Polygon Reduction (Polyred)...

A Glimpse into Mesh Representation

Simplicity is Complexity On the Balance of Performance

404 NOT FOUND PANDEMIC
A Future of Polygon Reduction
"Polyred is solved at scale!!"

- 640k#poly
  Ground truth
  Handcraft
  40 hours+

- 29k#poly
  My approach
  ~30s

- 119k#poly
  Blender ([Hoppe96]+[Garland97])
  ~4min

- 126k#poly
"Polyred is solved at scale!!"

Ground truth

Handcraft
40 hours+

My approach
~30s

Blender ([Hoppe96]+[Garland97])
~4min

640k#poly

29k#poly

119k#poly

126k#poly
Automation, Speed, Quality Tradeoff

- **Obvious Stupid**
- **Algorithmic Research**
- **Current Practice**
- **Maybe Interesting**
- **Impractical**

- Manual

- Automatic

- High Comp. Complexity

- Low Comp. Complexity
A First Thought on Human Cost

\[ T = \sum_{a \in A} \left( DT(x) + P(m) \right) \]

---

**Interface**

```go
package Polyred

import "time"

// Polyred is an interface that provides access to the Polyred API.

// Polyred defines methods for interacting with the Polyred API.

type Polyred interface {
    Upload(m *Model) (sessID string)
    Run(sessID string, n int) (opIDs []OpID) // 1m poly ≈ n minutes → n models
    Eval(opIDs []OpID, scores []int)
    Download(OpID string) (m *Model)
}
```

*Talk to me for web APIs access at [https://poly.red](https://poly.red)*
A First Thought on Human Cost

\[ T = \sum_{a \in A} (DT(x) + P(m)) \]

**interface Polyred**

```go
interface Polyred {
    Upload(m *Model) (sessID string)
    Run(sessID string, n int) (opIDs []OpID) // 1m #poly ≈ n minutes → n models
    Eval(opIDs []OpID, scores []int)
    Download(OpID string) (m *Model)
}
```

*Talk to me for web APIs access at [https://poly.red](https://poly.red)*
Yet Another Ultimate Polyred Program?

Rate from 1~5 for each model
(1=worst, 5=best)
Yet Another Ultimate Polyred Program?

(1/5) Rate from 1~5 for each model
(1=worst, 5=best)

-99.5%  -90.0%  -50.0%  -20.0%

Evaluate
Yet Another Ultimate Polyred Program?

(1/5) Rate from 1~5 for each model
(1=worst, 5=best)

-99.5%
-90.0%
-50.0%
-20.0%
Yet Another Ultimate Polyred Program?

(2/5) Rate from 1~5 for each model
(1=worst, 5=best)

-99.0%
-95.0%
-80.0%
-60.0%
Yet Another Ultimate Polyred Program?

(5/5) Optimal Reduced Model:

-89.4%

Done
How Good Is the Design? Will it Success at Scale?

Looks like a similar and successful idea proven in XYZ, will it also success (at scale) for polyred?
How Good Is the Design? Will it Success at Scale?

- Initially it indeed draws a lot of attention from the industrial partner
- Later proven this is a really just a look-fancy but horrible design
- Reasons:
  - 
  - 
**Human Cost: What did I miss??**

\[ T = \sum_{a \in A} (DT(x) + P(m)) \]

- **Preference Elicitation**
- **UI Complexity**
- **Model Complexity**

**Flowchart:**
- Start
  - Run
  - Satisfied?
    - Yes
      - Evaluate
    - No
      - Stop

Human Cost: What did I miss??

**Table:**
- **Time Spent**
- **#Attempts**
- **Decision**
- **Polyred**

<table>
<thead>
<tr>
<th>Time Spent</th>
<th>#Attempts</th>
<th>Decision</th>
<th>Polyred</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rethinking Human Cost In Reality

\[ T = p \sum_{\alpha \in A} (DT(\mathbf{x}) + P(\mathbf{m})) + (1 - p)H(\mathbf{m}) \]

- **Time Spent**
- **#Attempts**
- **Decision**
- **Polyred**
- **Abandon Probability**
- **Manual Retopology**
- **Iteration Complexity**
- **UI Complexity**
- **Model Complexity**
- **Algorithm Boundedness & Robustness**

### Flowchart

- **Start**
  - **Run**
  - **Evaluate**
    - **Satisfied?**
      - **Yes**
      - **Try Again?**
        - **Yes**
        - **No**
        - **Hell No!!**
      - **Stop**
    - **No**
  - **Manual**
Crowd Wisdom: Can we “bring insights from XAI research”?
Crowd Wisdom: Can we “bring insights from XAI research”?

“Maybe you just need address the trustworthy issue, that let human believe this is a good result”
“Maybe you just need address the trustworthy issue, that let human believe this is a good result”

Revisit the modeling pipeline:
“Maybe you just need address the trustworthy issue, that let human believe this is a good result”

Revisit the modeling pipeline:

- Tons of work after the initial modeling phase
  - Subdivision
  - Surface parameterization
  - Tuning materials
  - Stretching for animation
A good approximation of positions?
Example: Meshes with Equal Hausdorff Distance to A Reference

\[ \mathcal{H}(M, M') = \sqrt{\frac{1}{S} \int \int_{v \in S} d(v, S')^2 dS} \]
What We Talk About When We Talk About *Good* Meshing

- Approximation of position is not enough
  - Can still have wrong appearance, normals, curvatures, ...

- Rule of thumb today
  - Polygon shape, e.g. Delaunay
  - Vertex degree (Tri6, Quad4)
  - Adaptive sampling
  - Conformality

- Sounds trivial but harder to realize
- Even when you talk with domain experts
How much control knobs is enough?
Mesh Energy Minimization (Revised)

\[ \Delta ( \quad ) = 0 \]

Subject to “pen strokes” as boundary condition
A Beautiful Mesh
Do we really want care about meshing?
“If everything looks correct, then it is correct.”
"If everything looks correct, then it is correct."
Even More Beautiful
Render Settings

Settings that constitute the final rendered object:

- Rendering algorithm: Rasterization, path tracing, photon mapping, ...
- BRDF parameters
- Light sources
- Camera parameters
- Procedural parameters
- ...

Here should add good visualization but I just don’t have the time to render it.
Sorry.
“Differential” Rendering: The Objective

\[ L(\text{Model}) = \left\| \text{Renderer} \right\|^2 \]

minimize \[ L(M, r(s)) \] for \( s \in S \)

Objective \quad Model \quad Renderer \quad Render Settings
Representation Learning Approach

Classic Approach (well-studied)

Human-in-the-loop Approach

?
Challenges
The Cancer in 3D Modeling Pipeline

• **Reusability**
  • More than 90% of the art works are served as one time use purpose

• **User education**
  • Artists are trained strictly, and quite comfortable with their current modeling workflow, whereas
  • General users won’t even notice the low quality meshes (analogy as in “color science”)
  • **Poly counts is only an issue with rasterization, but not an issue with ray tracing family**

• ...
3D artists are “educated” and “trained” to produce adaptive sampled parallel flow lines with fewer singularities

Solving edge cases is much harder than the problem itself, and utterly hard to maintain consistency

Fancy algorithms are slow when N is small, and N is usually small

Always rethink the problem whether it is a “bug” that worth a fix

Respect design work and technical details

Robust preference elicitation replaces mesh manipulation

Intermediate representation replaces polygon meshes eventually

Professional end users produce unique understanding the original purpose of a feature

Don’t trust domain experts when you start touching their area, just become one of them

...
What was missed?
What else can be simplified?
What else can be done better?
What works what doesn’t?
Report your experience...

A Future of Polygon Reduction
Changkun Ou
changkun.de/s/polyred4us
IDC 2020 Autumn
Venice, Italy
Oct 7, 2020
What was missed?
What else can be simplified?
What else can be done better?
What works what doesn’t?
Report your experience

The Death
A Future of Polygon Reduction

Changkun Ou
changkun.de/s/polyred4us

IDC 2020 Autumn
Venice, Italy
Oct 7, 2020
The Death
A Future of Polygon Reduction

Changkun Ou
changkun.de/s/polyred4us

IDC 2020 Autumn
Venice, Italy
Oct 7, 2020

What was missed?
What else can be simplified?
What else can be done better?
What works what doesn’t?
Report your experience