Continuous Optimization of Adaptive Quadtree Structures

Jun Wu
TU Delft

Solid and Physical Modeling - SPM 2018, 11-13 June 2018, Bilbao, Spain
Infill in 3D Printing

- Regular pattern
- Denser pattern -> a stronger printer, yet more material

mpi.fs.tum.de

ultimaker.com
Motivation

- To design lightweight and stiff infill structures
Related Work: Structural optimization

- Various parameterizations of the design space

Skin-frame [Wang’13]
Voronoi cells [Lu’14]
Offset surfaces [Musialski’15]
Topology optimization [Wu’16]
The Idea

- Can we do adaptive refinement?
Adaptive Refinements: Potential Advantages

- Semi-regular structures
  - Control over uniform thickness
  - Control over minimum/maximum feature size
  - Robust for unexpected forces
  - Easy to generate toolpath
Adaptive Refinements: Technical Challenges

- Discrete problem: To refine, or not to refine
  - Difficult to solve accurately
- The number of design variables changes
Previous Attempt: A Greedy Approach

- Iteratively refine the most "sensitive" cells

[Wu et al., CAD’2016]
Questions

- Can we re-formulate it as a continuous problem?
- The continuous formulation possibly comes to a more optimal solution?
- To verify how good or bad the greedy approach is.
Method
Multi-level Design Variables $x_{i,j}^k$

- Design variables on the k-th level $x_{i,j}^k \in [0,1]$
Multi-level Design Variables $x^k_{i,j}$

- Design variables on the k-th level $x^k_{i,j} \in [0,1]$

Level 1: $4 \times 2$

Level 3: $16 \times 8$
Finite Elements $\rho$

- Uniform finite elements, mapped from the quadtree grid
  - Density of corresponding finite elements $\rho = x_{i,j}^k \in [0,1]$
Optimization Problem: Minimum Compliance

- \( \min \ c = U^T K(\rho) U \)

- s.t. \( K(\rho) U = F \)
- \( V(\rho) = \sum_{V} \rho_v v_e \leq V^* \)
- \( x_{i,j} \in [0,1] \)
- \( \rho = \sum_{k=0}^{\bar{k}} T^k x^k \)
Result 1
Refinement Filter

- Dependency among two levels $\tilde{x}_{i,j}^k = \min(x_{i,j}^k, x_{i-1,j-1}^{k-1})$
Refinement Filter

- Dependency among two levels \( \tilde{x}_{i,j}^k = \min(x_{i,j}^k, x_{i-1,j-1}^{k-1}) \)
- Among multiple levels \( \tilde{x}_{i,j}^k = \min(x_{i,j}^k, x_{i-1,j-1}^{k-1}, \ldots, x_{i-k+1,j-k+1}^1) \)
Refinement Filter

• Dependency among two levels $\tilde{x}_{i,j}^{k} = \min(x_{i,j}^{k}, x_{i-1,j-1}^{k-1})$

• Among multiple levels $\tilde{x}_{i,j}^{k} = \min(x_{i,j}^{k}, x_{i-1,j-1}^{k-1}, \ldots, x_{i-k+1,j-k+1}^{1})$
Refinement Filter

- Dependency among two levels $\tilde{x}_{i,j}^k = \min(x_{i,j}^k, x_{i-1,j-1}^{k-1})$
- Among multiple levels $\tilde{x}_{i,j}^k = \min(x_{i,j}^k, x_{i-1,j-1}^{k-1}, \ldots, x_{i-k+1,j-k+1}^1)$
- Continuous approximation
  $$\tilde{x}_{i,j}^k \approx \left\| (x_{i,j}^k, x_{i-1,j-1}^{k-1}, \ldots, x_{i-k+1,j-k+1}^1) \right\|_{p_n}$$
Result 2
Refinement Filter: Balanced Quadtree

- Dependency among two levels \( \tilde{x}^{k}_{i,j} = \min(x^{k}_{i,j}, x^{k-1}_{i-1,j-1}) \)
- Dependency among two levels for balanced quadtree
  \[
  \tilde{x}^{k}_{i,j} = \min(x^{k}_{i,j}, x^{k-1}_{i-1,j-1}, x^{k-1}_{i-1+1,j-1+1})
  \]
Refinement Filter: Balanced Quadtree

- Dependency among two levels: \( \tilde{x}_{i,j}^k = \min(x_{i,j}^k, x_{i^{-1},j^{-1}}^{k-1}) \)
- Dependency among two levels for balanced quadtree:
  \[ \tilde{x}_{i,j}^k = \min(x_{i,j}^k, x_{i^{-1},j^{-1}}^{k-1}, x_{i^{-1+1},j^{-1+1}}^{k-1}) \]
Refinement Filter: Balanced Quadtree
Comparison: Designed force

\[ c = 136.1 \]

\[ c = 143.3 \]

\[ c = 165.7 \]
Unexpected force

- Quadtree is robust to unexpected force
Results
Results: Animation
Results: Convergence
Results: Convergence

- Objective $\min_{x} c = U^{T}K(\rho)U$
- Volume $V(\rho) = \sum_{\forall e} \rho_e v_e \leq V^*$
- Sharpness $s = \frac{4}{n} \sum_{e} (\rho_e (1 - \rho_e))$
Results: Feature Size

- Control feature size by allowing different refinement levels
Results: Comparison

- Continuous optimization achieves more optimal solution than the heuristic approach

Greedy approach

\[ v_f = 0.20 \quad c = 1919.9 \]

\[ v_f = 0.30 \quad c = 742.5 \]

Continuous optimization

\[ v_f = 0.20 \quad c = 1266.7 \]

\[ v_f = 0.30 \quad c = 571.6 \]
Results: Fabrication
Results: Fabrication
Results: Fabrication

- Adaptively refined infill is much stiffer than uniform infill
Summary

• Continuous optimization achieves more optimal solutions than the greedy approach
• A *continuous* quadtree for structural optimization
• A refinement filter to encode the dependency of multi-level design variables
Future work

• 2D -> 3D
• Computational performance, e.g., beam elements
• Other subdivisions, e.g., triangles
Thank you for your attention!

Matlab code available at www.jun-wu.net

Jun Wu
TU Delft