Scalable Manufacturing of Shaped Polymers

Adarsh Krishnamurthy[1], Dan Stoecklein[1], Dino Di Carlo[2], Baskar Ganapathysubramanian[1]

[1] Iowa State University

[2] University of California, Los Angeles

Introduction

Aim

- To develop a scalable manufacturing method to create designer, 3D, shaped polymer particles for engineering applications
- Combine masked photo-polymerization and flow sculpting in orthogonal directions

Motivation and Applications

- Particle shape is critical in defining its physical & chemical function
- Applications include tuned optical properties, plasmon resonances, tunable catalysis, cell sorting, and drug delivery
- Widespread access to complex shaped particles will enable new optical, mechanical, and chemical catalysis approaches with unique advantages, transforming industries into the future

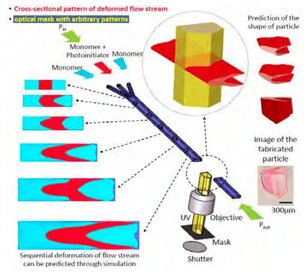


Figure 1: Schematic of scalable manufacturing system. Precursors flow in the microfluidic channel. Pillar sequences sculpt the monomer to desired shape. A shaped laser pulse (shaped with a mask) polymerizes the stream into desired particle shape. This can be multiplexed to create large quantities of shaped particles and fibers. Cost is similar to other lab-on-chip productions (~ < $1/chip).

Flow Sculpting

- Sculpt fluid streams in a micro-channel using a set of pillars that deform the flow in a predictable manner
  - Individual pillars are placed in a defined sequence to yield the composition of the individual flow deformations
- Create complex user-defined flow shapes
  - Discretization of single pillar operations followed by their programmed superposition allows for the hierarchical assembly of complex flow programs

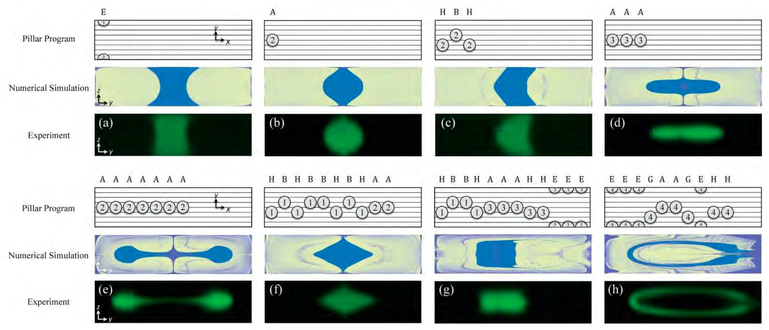
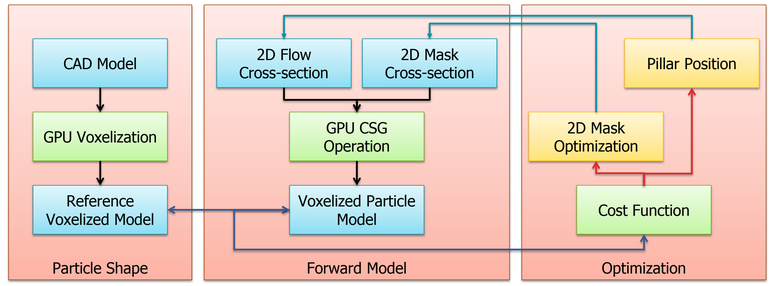


Figure 2: Different cross-sections of flow that can be achieved by positioning pillars using flow sculpting. The middle row represents the numerical solution while the bottom row represents the experimental cross-sections.

Particle Shape Modeling and Optimization

Modeling Flowchart



Particle Shape

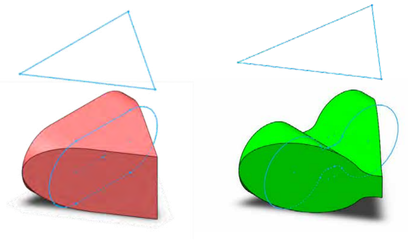


Figure 3: Shaped particle resulting from intersection of two mutually independent profiles. One of the profiles is controlled by flow sculpting and the other profile is a laser polymerization mask.

GPU Voxelization

- Voxelized representation used for constructing the 3D shape as an intersection of orthogonal 2D shapes
- GPUs accelerate constructive solid modeling operations and create voxelized representation of the final particle shape
- Allows for fast computation of the final particle shape
  - Can be easily used for shape optimization
  - Cost-function computed using the GPU as the positive sum of the difference between the required and estimated voxelized shapes

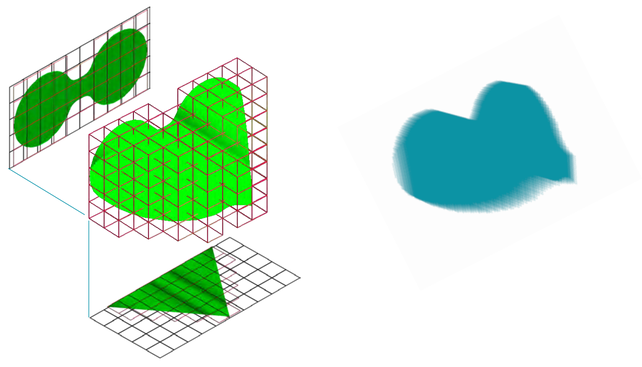


Figure 4: GPU Voxelization used to construct the final 3D shape of the particle. The individual 2D profiles can be used for optimization. Volume rendering of the voxelized particle is shown on the right.

GPU-based Optimization

Flow Optimization

- Graphical tool for predicting flow cross-section resulting from pillar geometries that deform the laminar flow field
- Computed by superposition of pre-computed advection maps to enable fast, real-time feedback of flow deformation without requiring a full Navier-Stokes solve

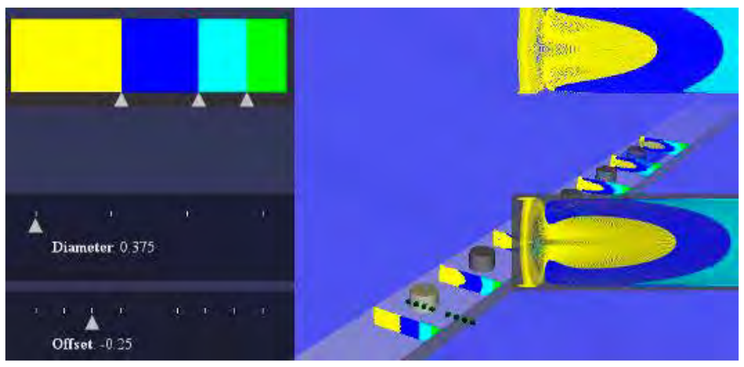


Figure 5: Virtual flow sculpting program to optimize flow and pillar characteristics to obtain the required flow cross-section.

Results and Conclusions

Sample Particles

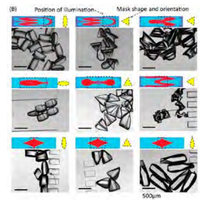


Figure 6: Shaped particles created using different flow cross-sections and photo-polymerization masks.

Conclusions

- GPUs enable testing of several possible cross-sections in parallel as part of the optimization process
- Voxelized representation instead of a boundary representation allows for binary representation of particle shapes
  - Allows for user-defined resolution and error quantification
- A general strategy and modular framework to program fluid streams and masks to design arbitrarily shaped particles can have a transformative impact on biological, chemical and materials automation in the same way that abstraction of semiconductor physics from computer programmers enabled a revolution in computation

Acknowledgments

AK acknowledges Iowa State University for computing support; DD, BG, and DS gratefully acknowledge NSF CBET 1306866 for partial support