# Discrete Element Textures

Chongyang Ma — Tsinghua University

Li-Yi Wei — Microsoft Research

Xin Tong — Microsoft Research Asia

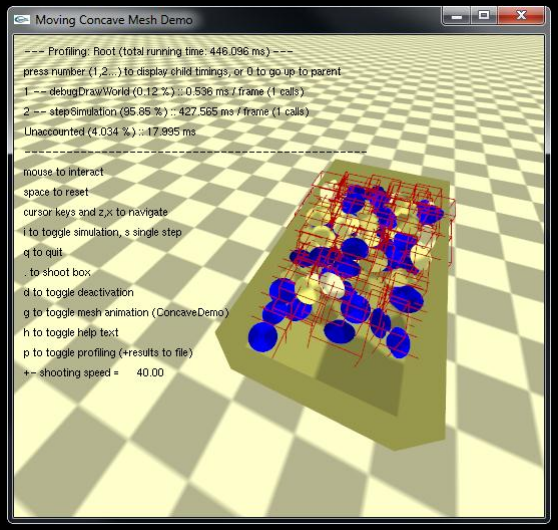
# Introduction

Aggregated elements in daily life

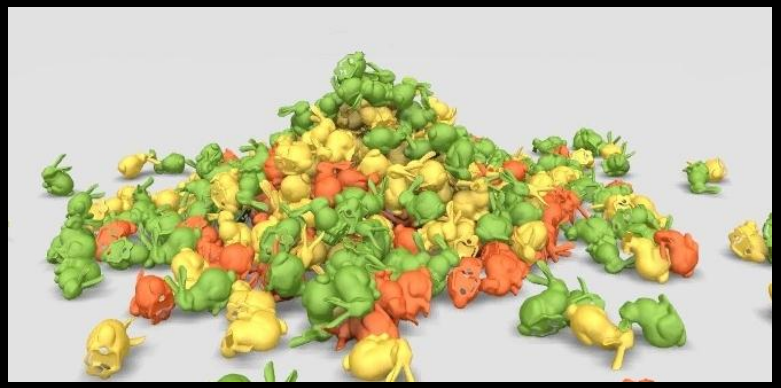
# Introduction

Physics-based simulation

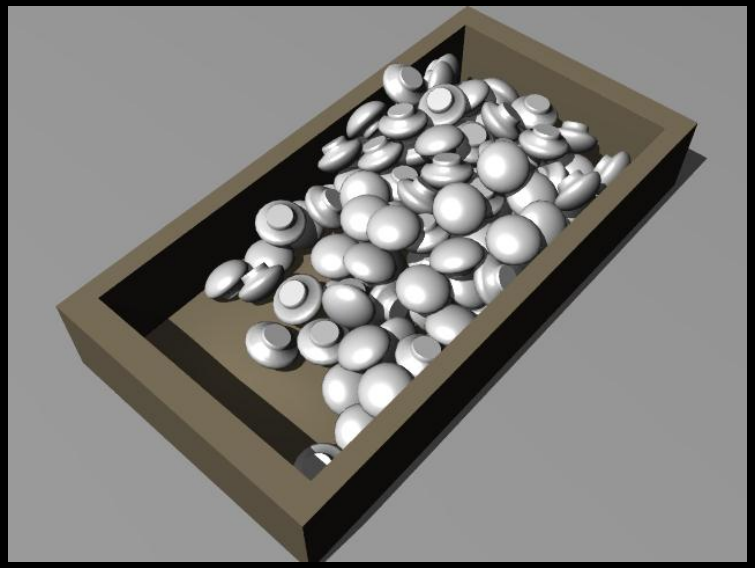
- ☺ Physical and visual reality
- ☹ Hard to control
- ☹ Computationally expensive



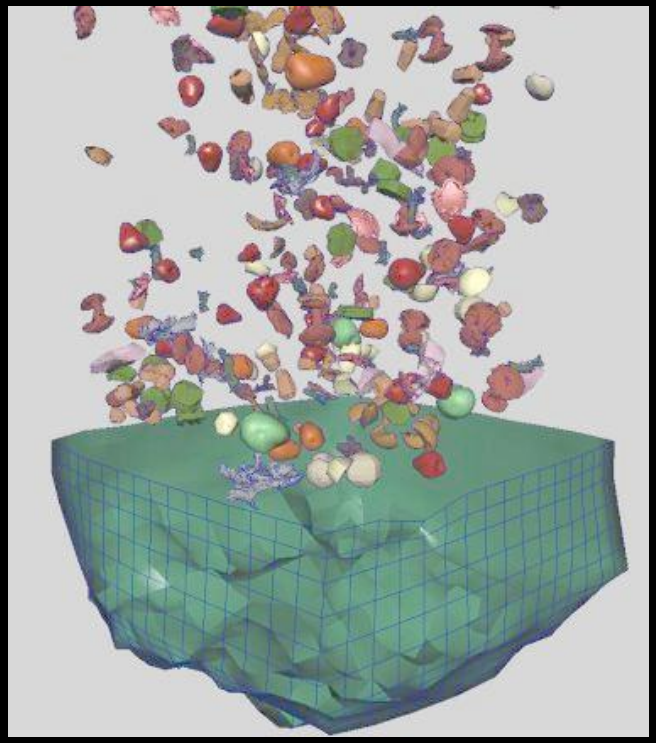
# Non-physically realistic effects



*freezing elements*
*[Hsu & Keyser 2010]*



*unstable*
*[Cho et al. 2007]*



*upside down*
*[Cho et al. 2007]*

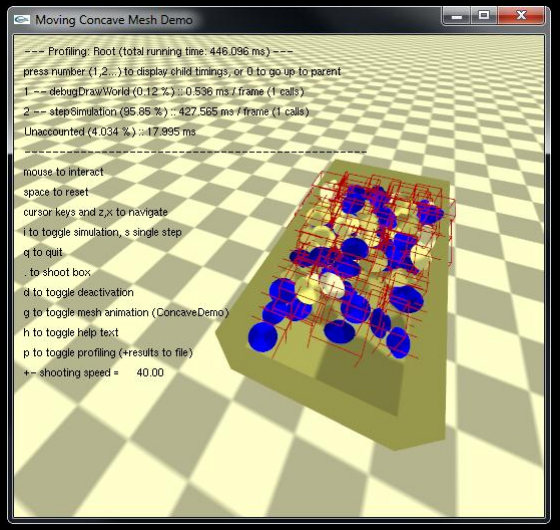
# Introduction

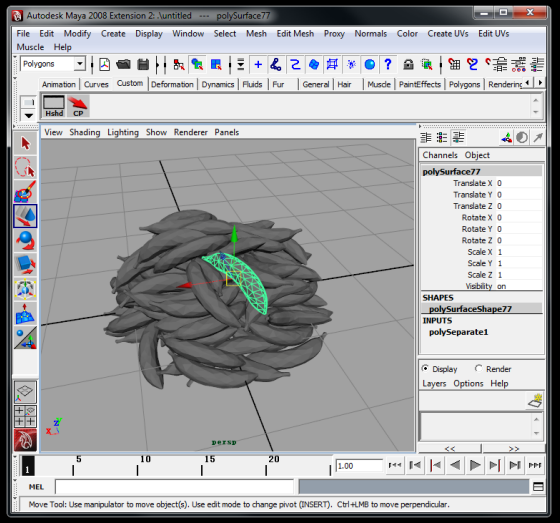
Physics-based simulation

- ☺ Physical and visual reality
- ☹ Hard to control
- ☹ Computationally expensive

Manual placement

- ☺ Intuitive control
- ☹ Tedious



# Our goals

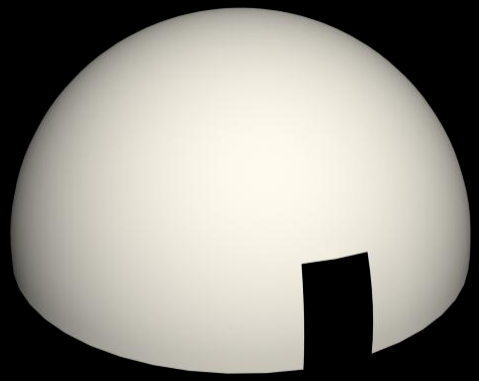
- General & flexible effects
  - Suitable for a variety of phenomena
  - May or may not be physics based
- User friendly
  - Only need input exemplar and large scale control
- Easy computation
  - Fast and stable

# Our approach

## Discrete element textures

*input exemplar*

*control shape*



*synthesis output*

# Why?

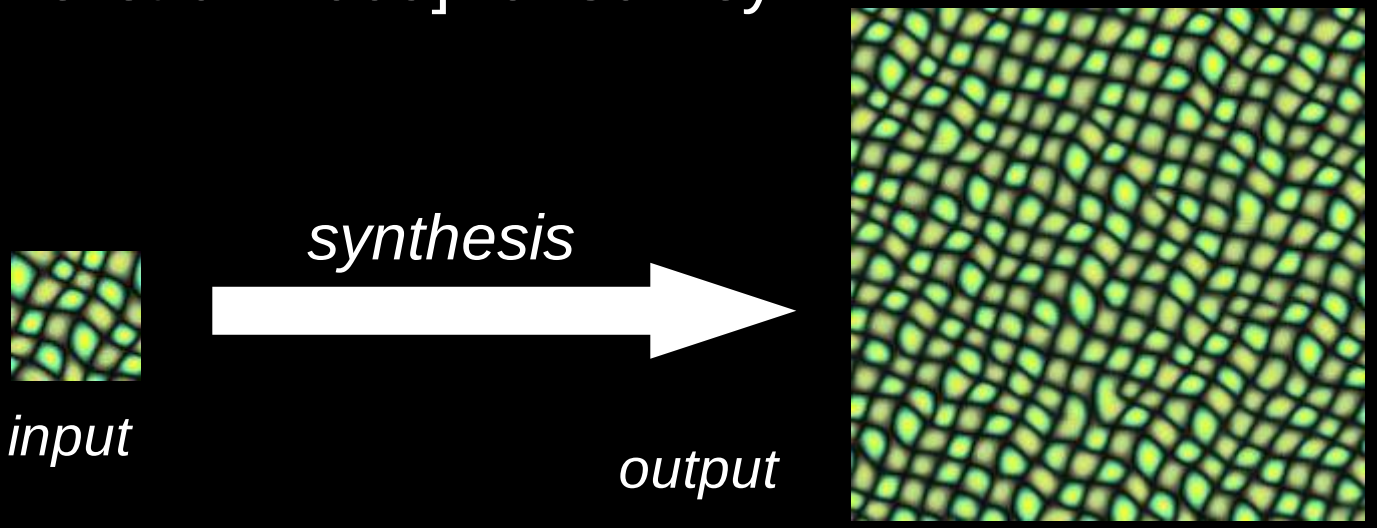


- Natural repetitions
  - Aggregate elements within a large-scale domain
- User friendly
  - Direct only the large-scale domain
  - Avoid tedious manual work
- Generality
  - Example-based vs. procedural

# Example based texturing

Synthesize repetitive phenomena

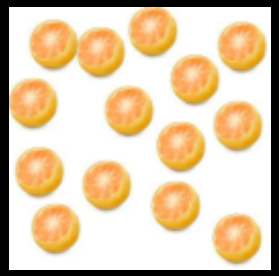
- Pixels, vertices and voxels
- See [Wei et al. 2009] for survey



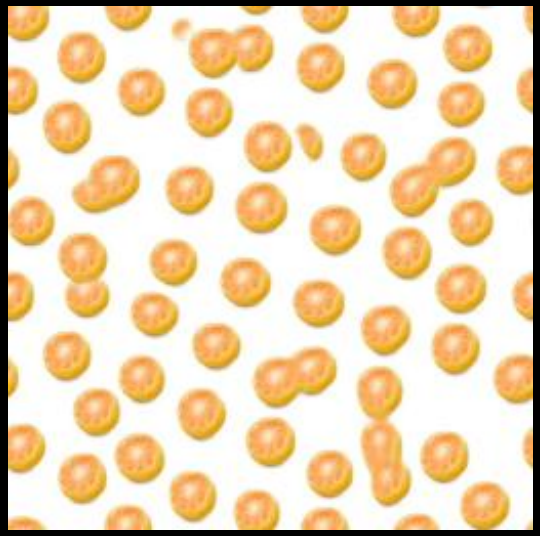
# Pixel based synthesis

No semantic meanings of individual elements

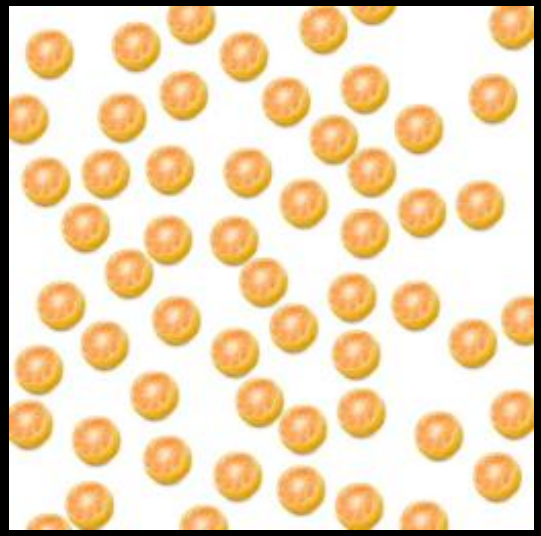
Broken or merged elements in synthesis output



*input*



*[Han et al. 2006]*



*our result*

# Example based element placement

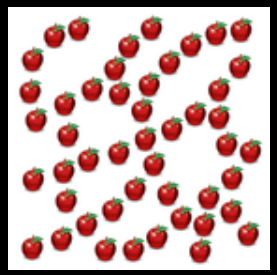
Pioneering works

- 1D strokes [Jodoin et al. 2002]
- 2D elements [Dischler et al. 2002; Ijiri et al. 2008; Hurtut et al. 2009]
- 2D agent motions [Lerner et al. 2007; Ju et al. 2010]
- 2D stipples [Kim et al. 2009; Martín et al. 2010]

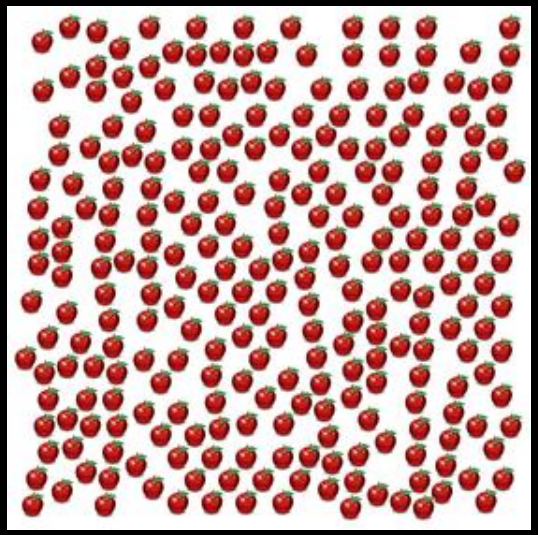
# Example based element placement

No robust neighborhood metric

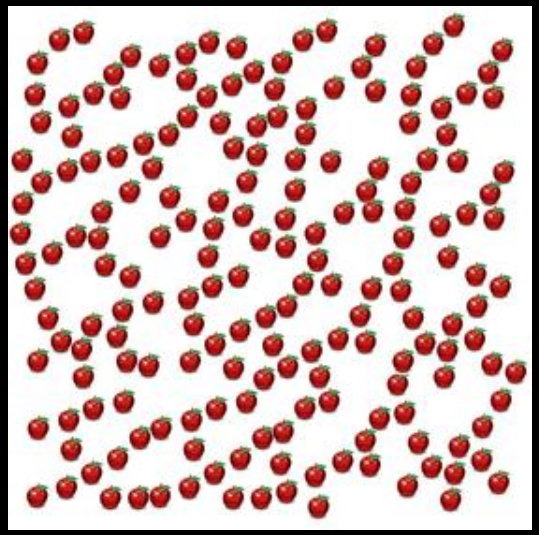
Cannot preserve aggregate distribution



*input*



*[Dischler et al. 2002]*

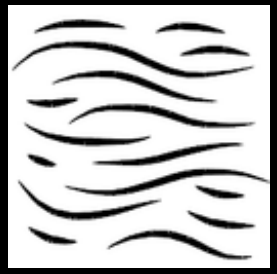


*our result*

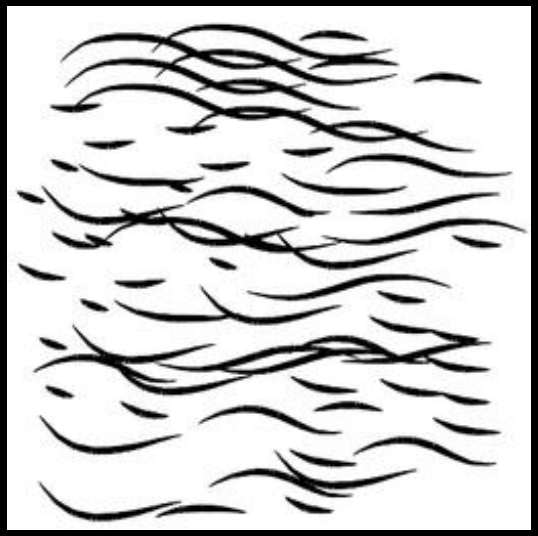
# Example based element placement

Consider center positions only

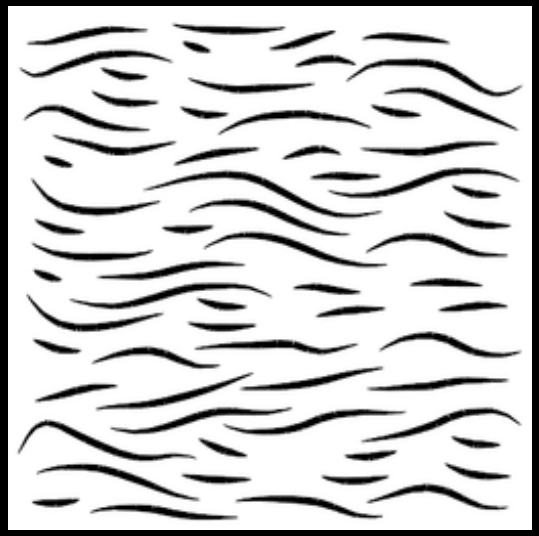
Cannot handle complex element shapes



*input*



*[Ijiri et al. 2002]*

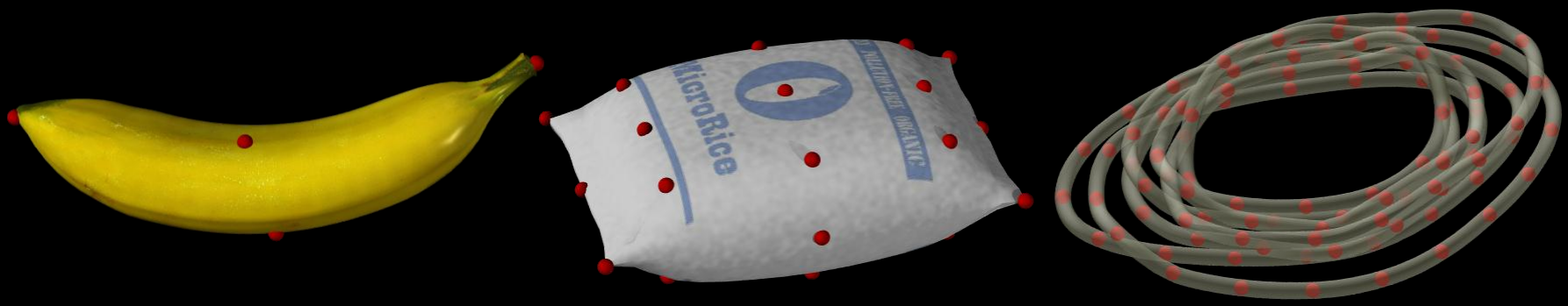


*our result*

# Basic representation

Multiple samples per element

- Simplicity: no need for extra orientation or shape info
- Flexibility: complex and deformable shapes



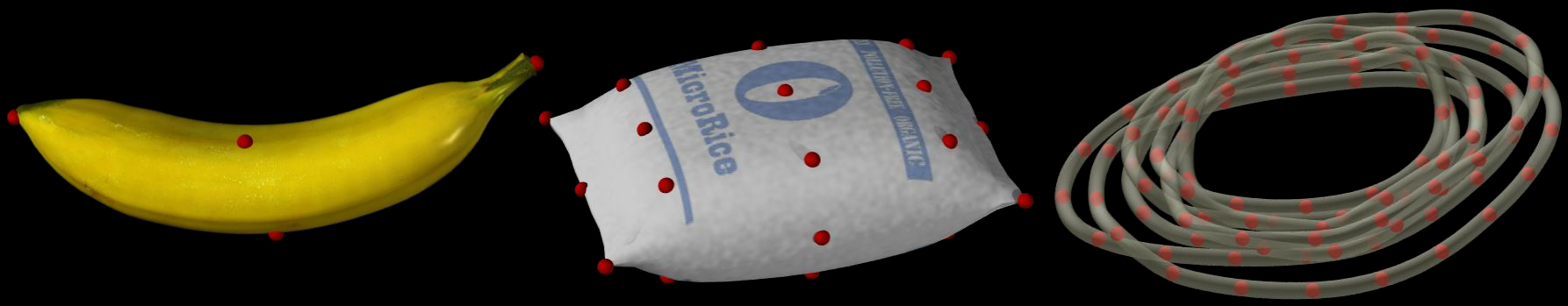
*banana* *bag* *spaghetti*

# Basic representation

Discrete element textures

$\{\mathbf{p}(s)\}$ Sample positions

$\{\mathbf{q}(s)\}$ Sample attributes (color, type, etc.)

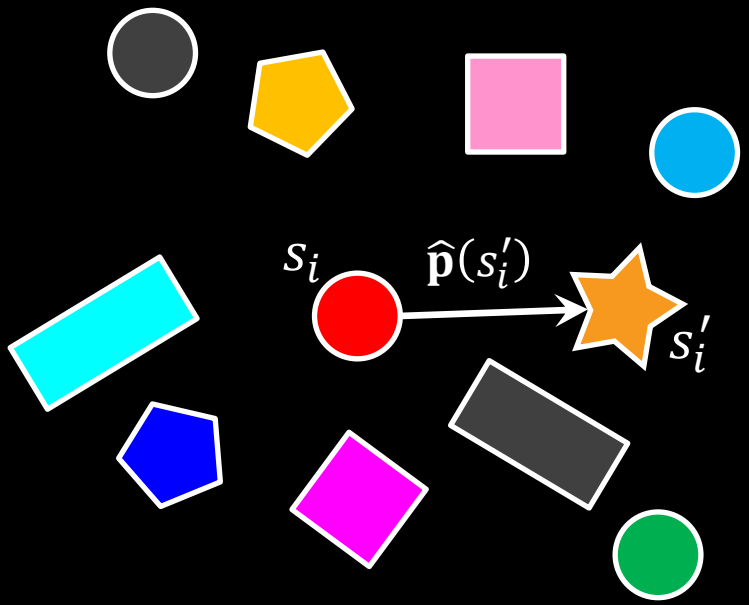



*banana* *bag* *spaghetti*

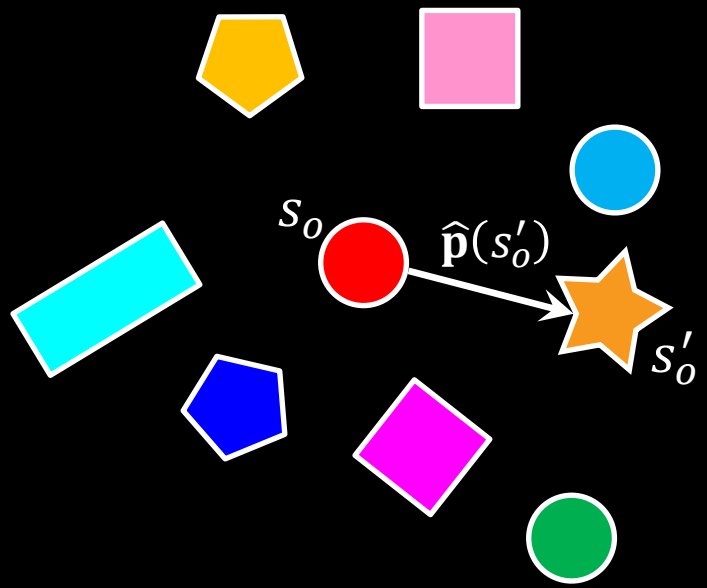
# Neighborhood metric

Align input and output neighborhoods

Match up the samples in pairs $|\hat{\mathbf{p}}(s'_o) - \hat{\mathbf{p}}(s'_i)|^2$




*input neighborhood*

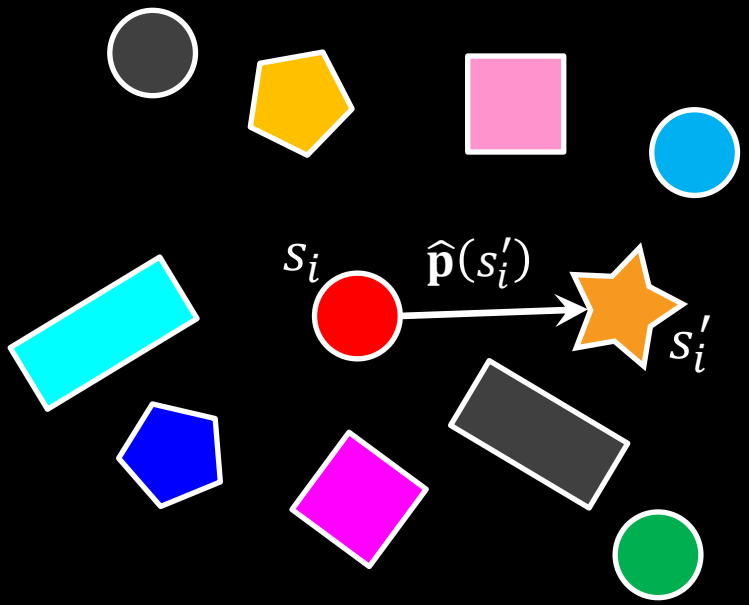



*output neighborhood*

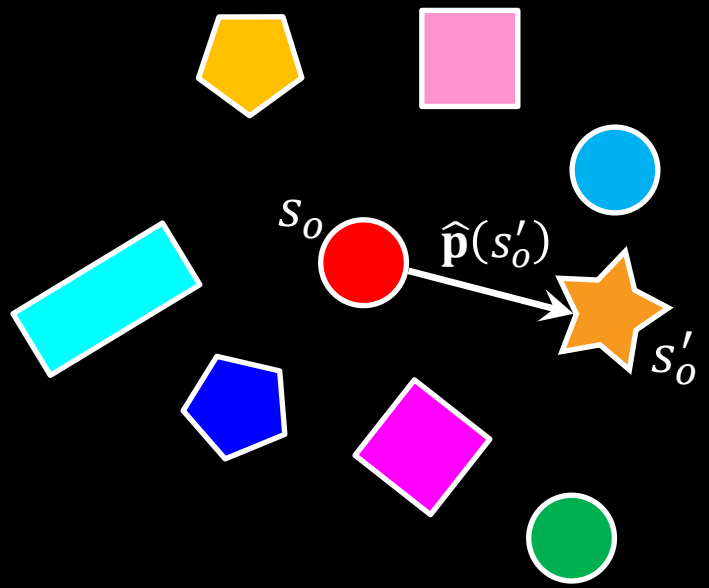
# Neighborhood metric

Sum of squared differences

$$|\mathbf{n}(s_o) - \mathbf{n}(s_i)|^2 = \sum_{s_o' \in \mathbf{n}(s_o)} |\widehat{\mathbf{p}}(s_o') - \widehat{\mathbf{p}}(s_i')|^2 + \alpha |\mathbf{q}(s_o') - \mathbf{q}(s_i')|^2$$



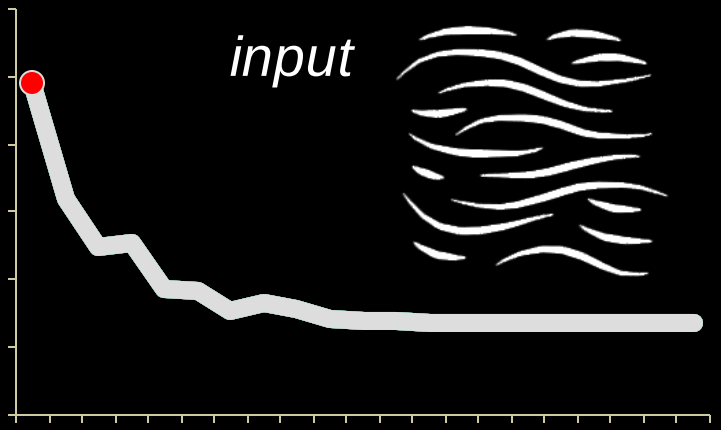

*input neighborhood*

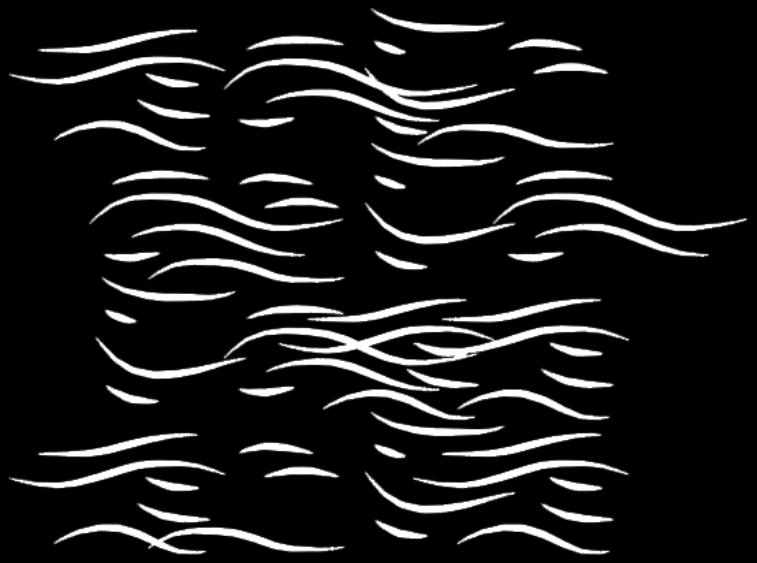



*output neighborhood*

# Basic synthesis

Initialization via patch copy



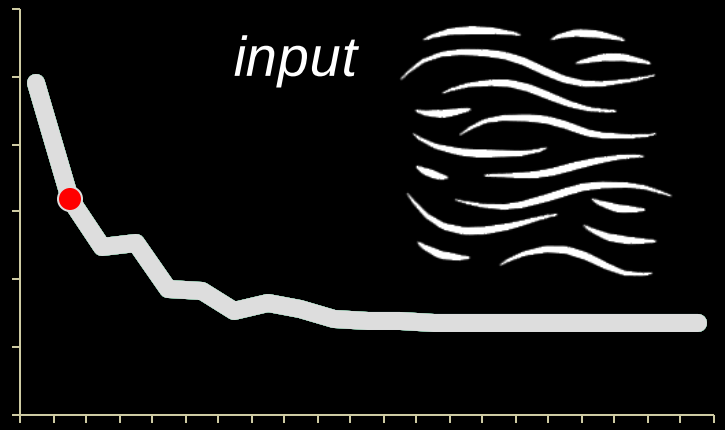
*output (initialization)*

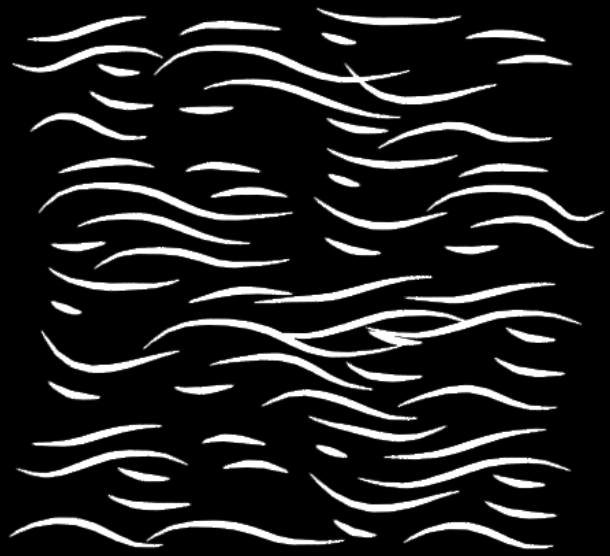
# Basic synthesis

Initialization via patch copy



Optimization based update



*output (iteration 1)*

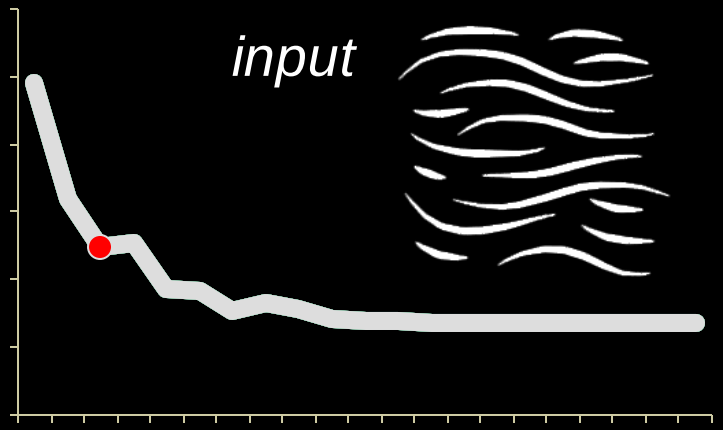
# Basic synthesis

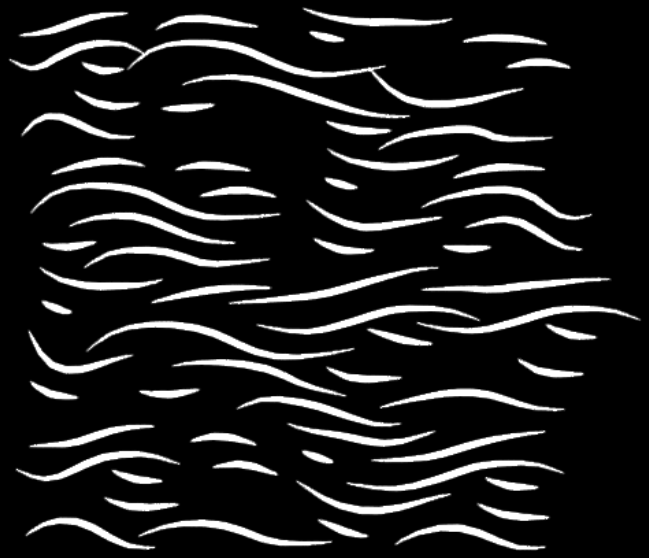
Initialization via patch copy

Optimization based update

*Convergence?*

*No*



*output (iteration 2)*

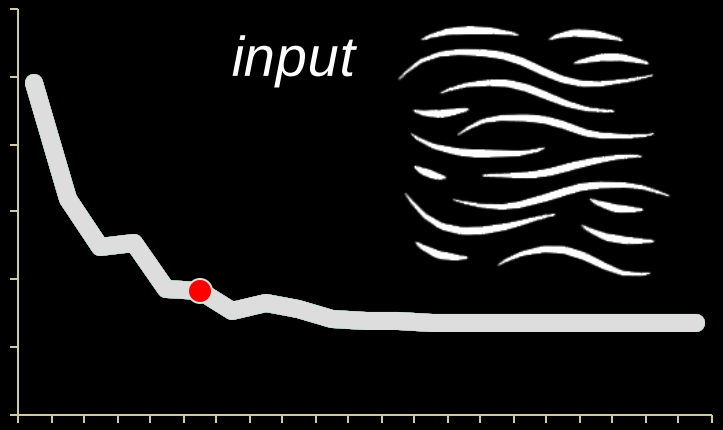
# Basic synthesis

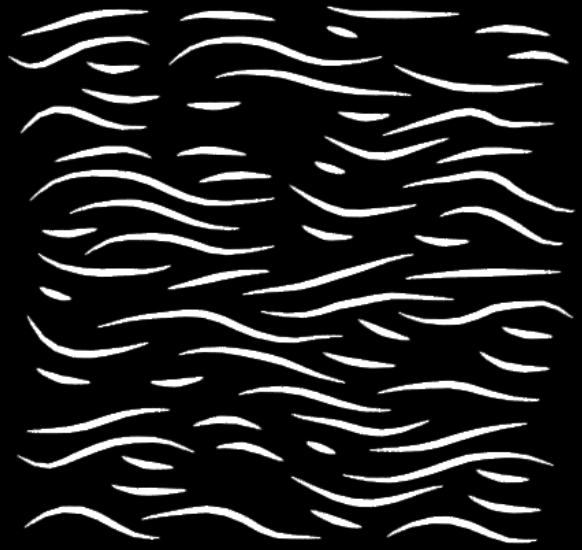
Initialization via patch copy

Optimization based update

Convergence?

No



*output (iteration 5)*

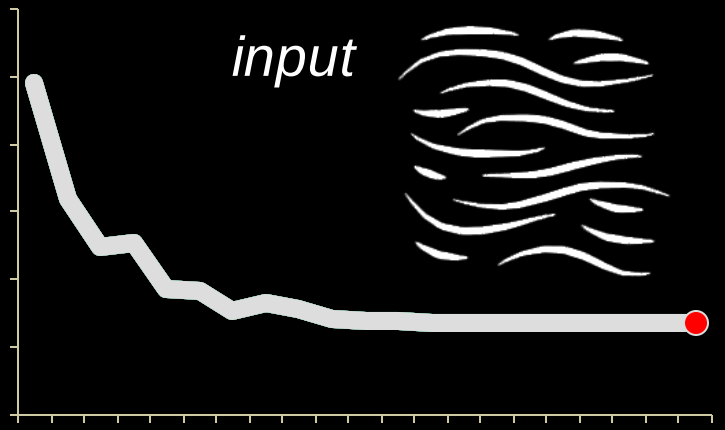
# Basic synthesis

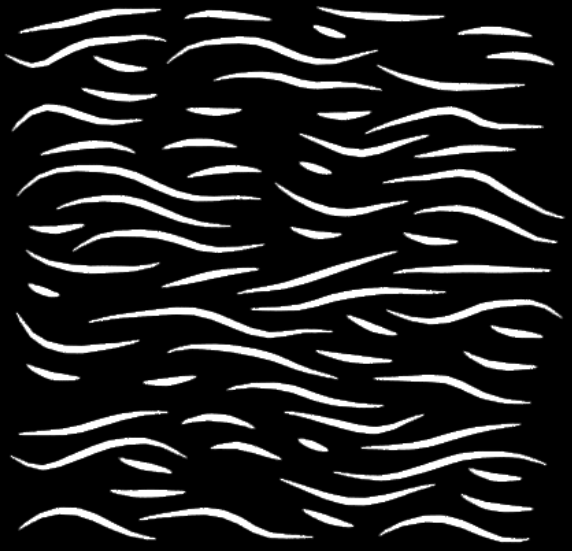
Initialization via patch copy

Optimization based update

Convergence?

No



*output (iteration 20)*

# Basic synthesis

Initialization via patch copy

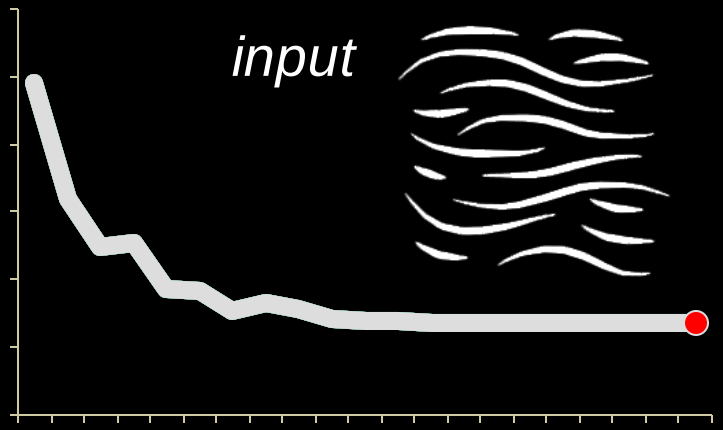
Optimization based update

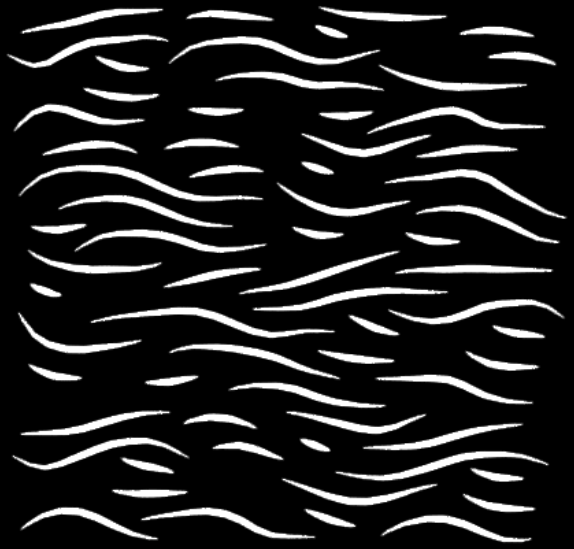
Convergence?

No

Yes

Shape reconstruction



output (iteration 20)

# Basic synthesis

Initialization via patch copy

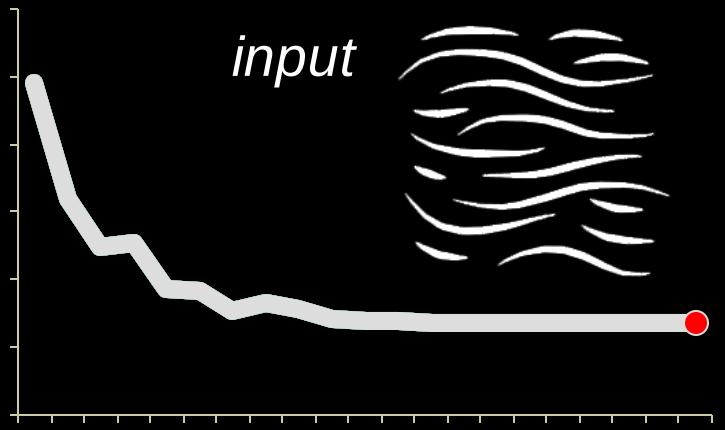
Optimization based update

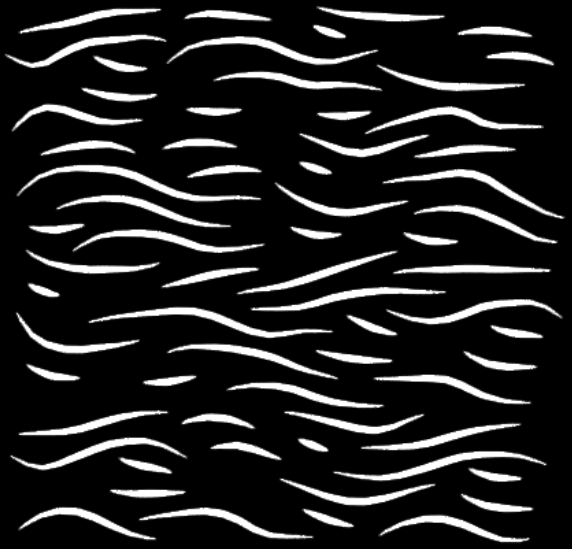
*Convergence?*

*No*

*Yes*

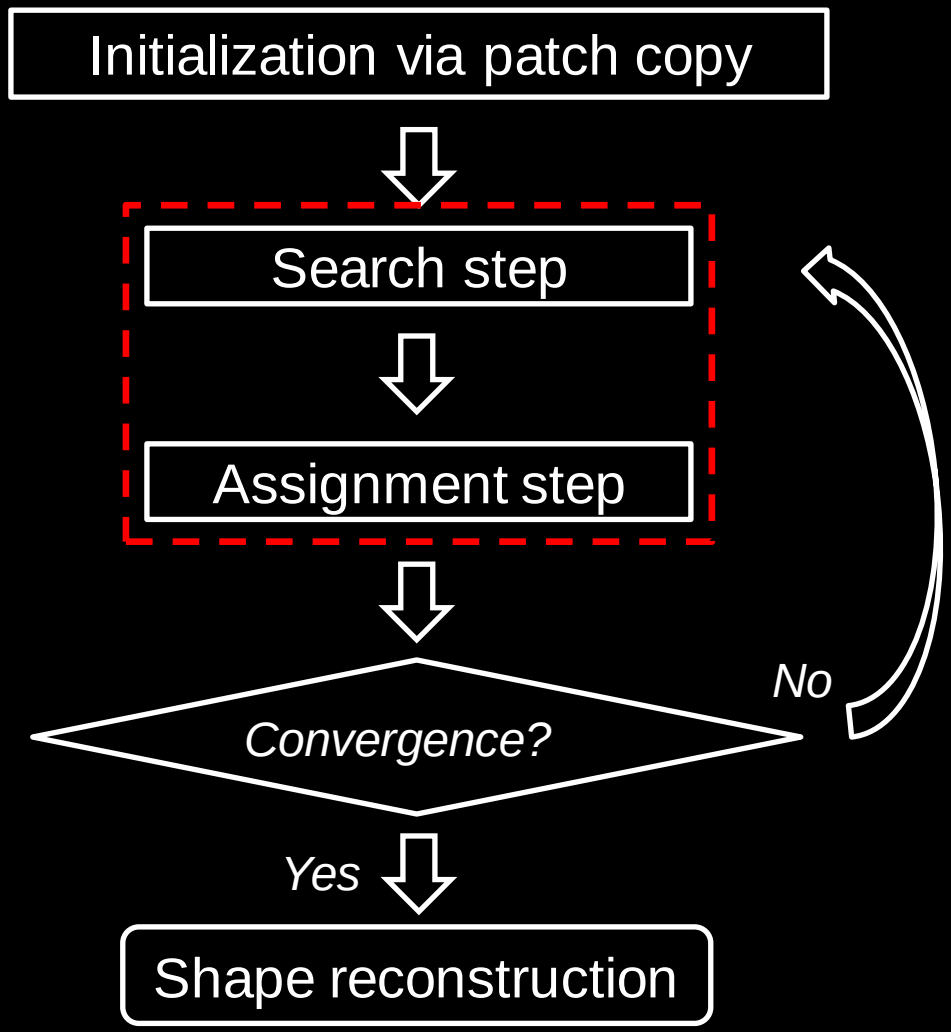
Shape reconstruction

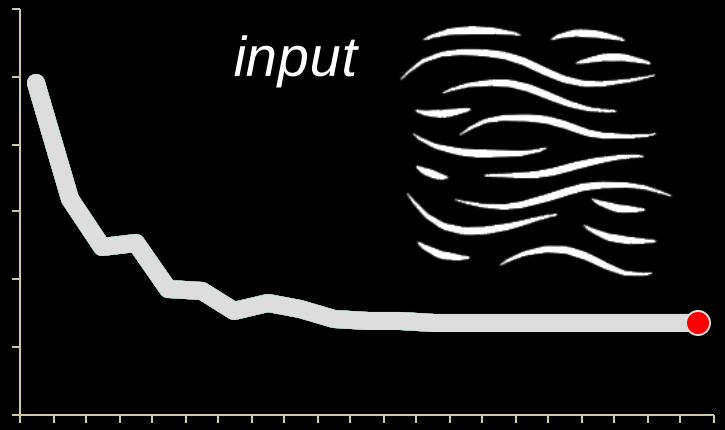


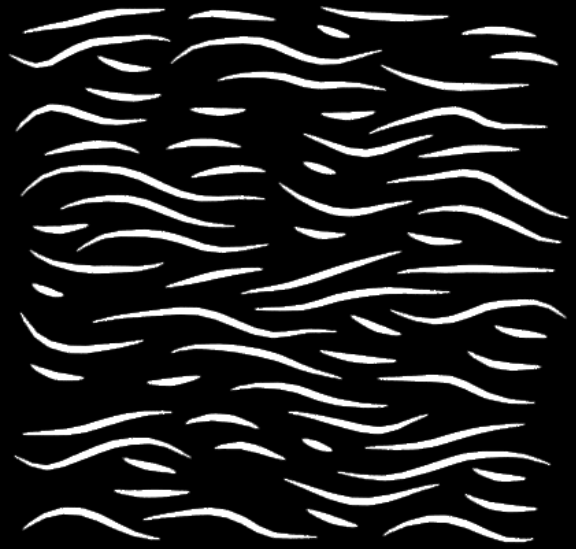



*output (iteration 20)*

# Basic synthesis

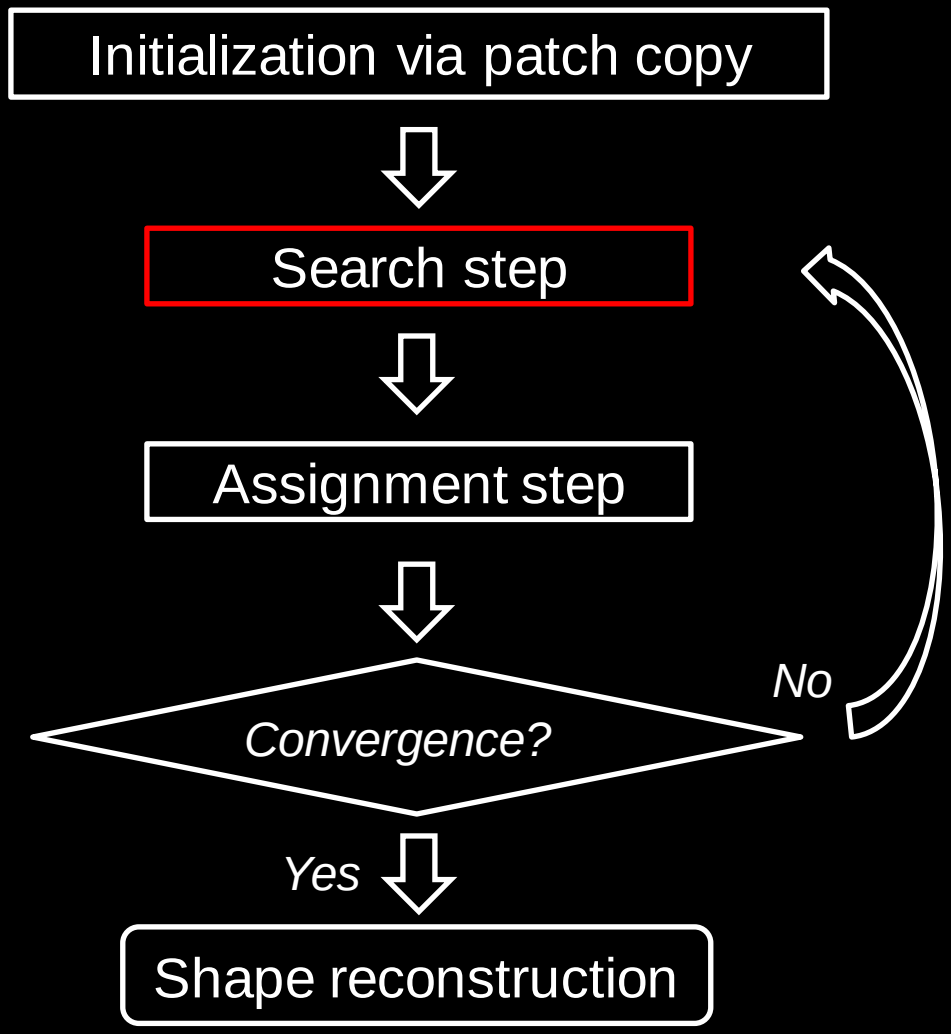


*output (iteration 20)*

# Search step



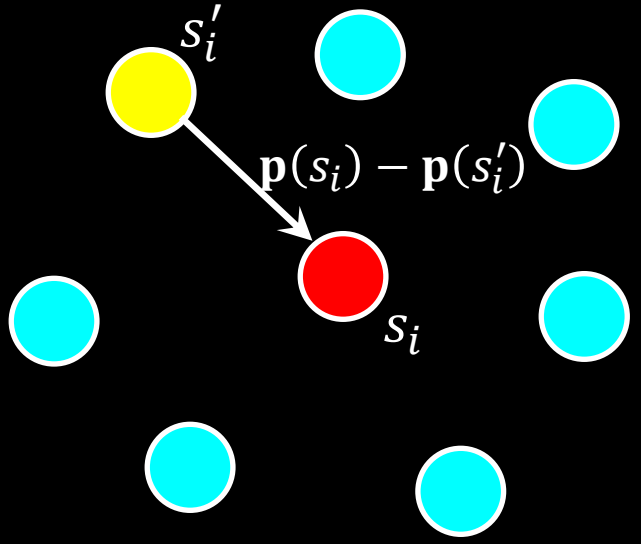

Find the best matching input sample for each output sample

Accelerate by k-coherence search [Tong et al. 2002]

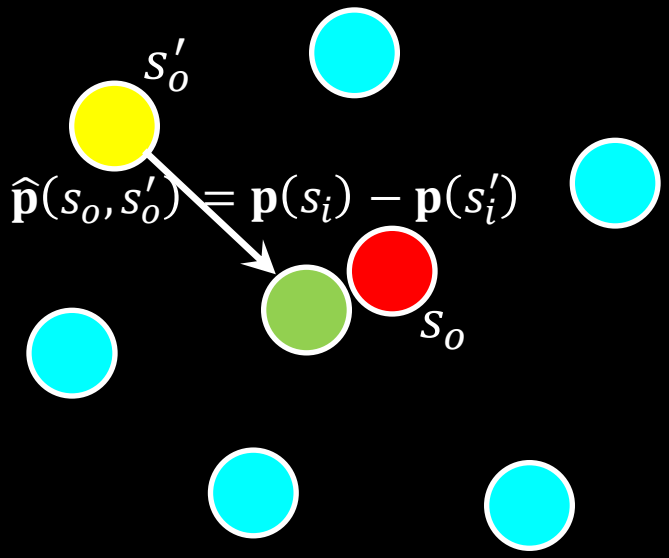
# Position assignment

Least squares from predicted offsets

$$E_{\mathbf{p}}\left(\{\mathbf{p}(s_o)\}_{s_o \in O}\right) = \sum_{s_o \in O} \sum_{s_o' \in \mathbf{n}(s_o)} \left|\left(\mathbf{p}(s_o) - \mathbf{p}(s_o')\right) - \hat{\mathbf{p}}(s_o, s_o')\right|^2$$




*input neighborhood*




*output neighborhood*

# Attribute assignment

For element/sample id, type, color, etc.

Select the vote that minimizes the energy function

$$\mathbf{q}(s_o) = \arg\min_{\mathbf{q}(s_i)} \sum_{s_i' \in \{s_i\}} |\mathbf{q}(s_i) - \mathbf{q}(s_i')|^2$$

Discrete solver vs. least squares

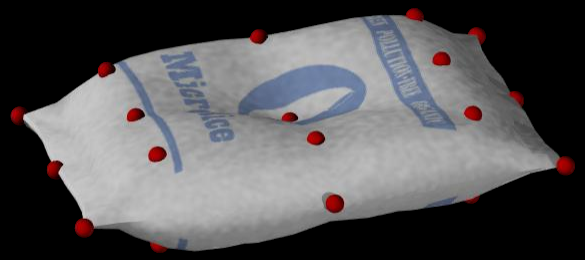
- Some attributes may not be meaningfully blended

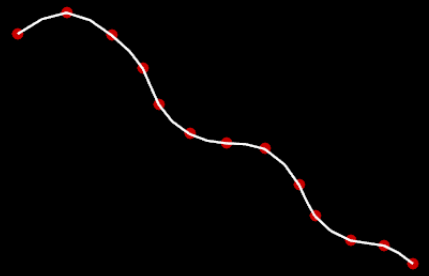
# Shape reconstruction

Consider samples only *during* synthesis

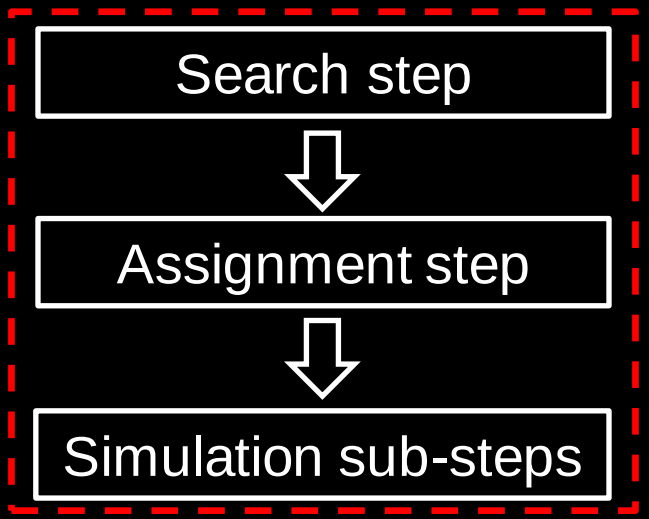
Reconstruct elements *after* synthesis

- Rigid bodies: shape matching
- Deformable volumes: displacement interpolation
- Elongated shapes: fitting with NURBS curves





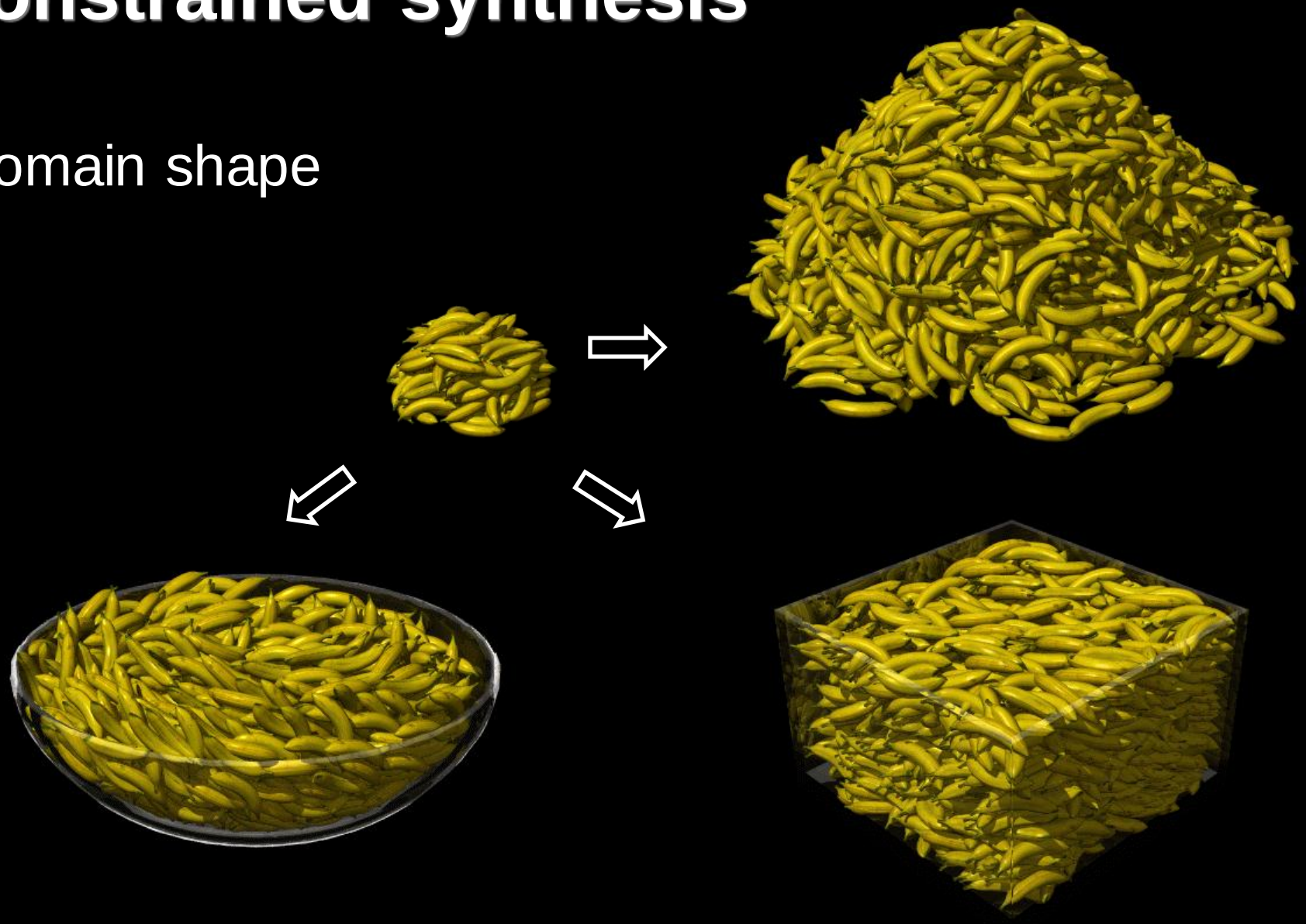
# Interleaved physics solver




- For common physical effects
  - Reduce penetrations
  - Obey gravity
- Simulation sub-steps within each iteration
  - Works as an *implicit* energy term
  - Difficult to formulate *explicitly*

# Constrained synthesis

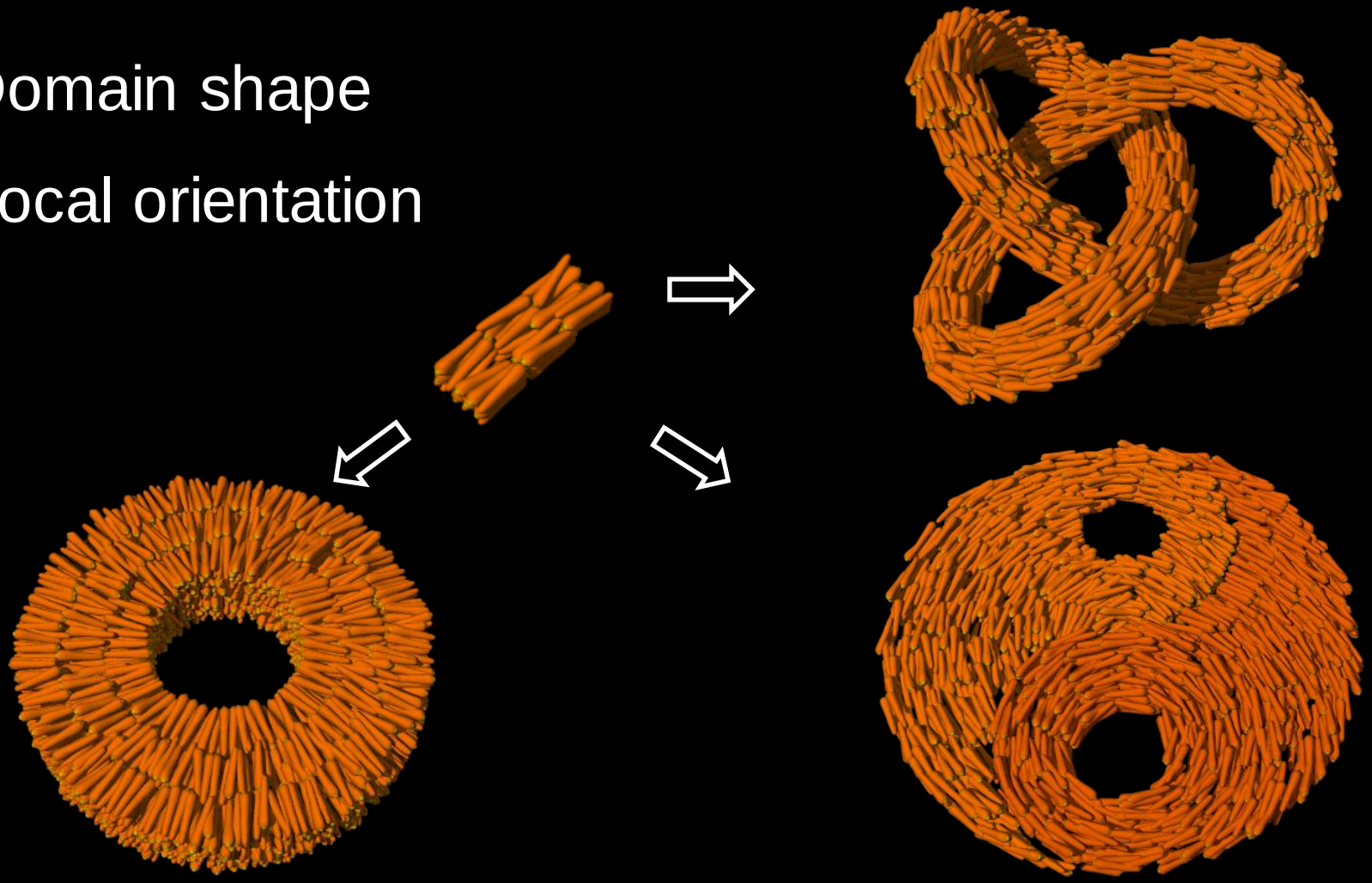
Domain shape



# Constrained synthesis

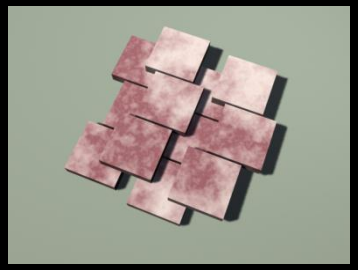
Domain shape

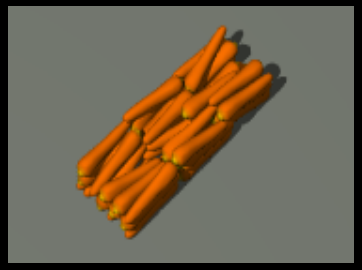
Local orientation



# Implementation

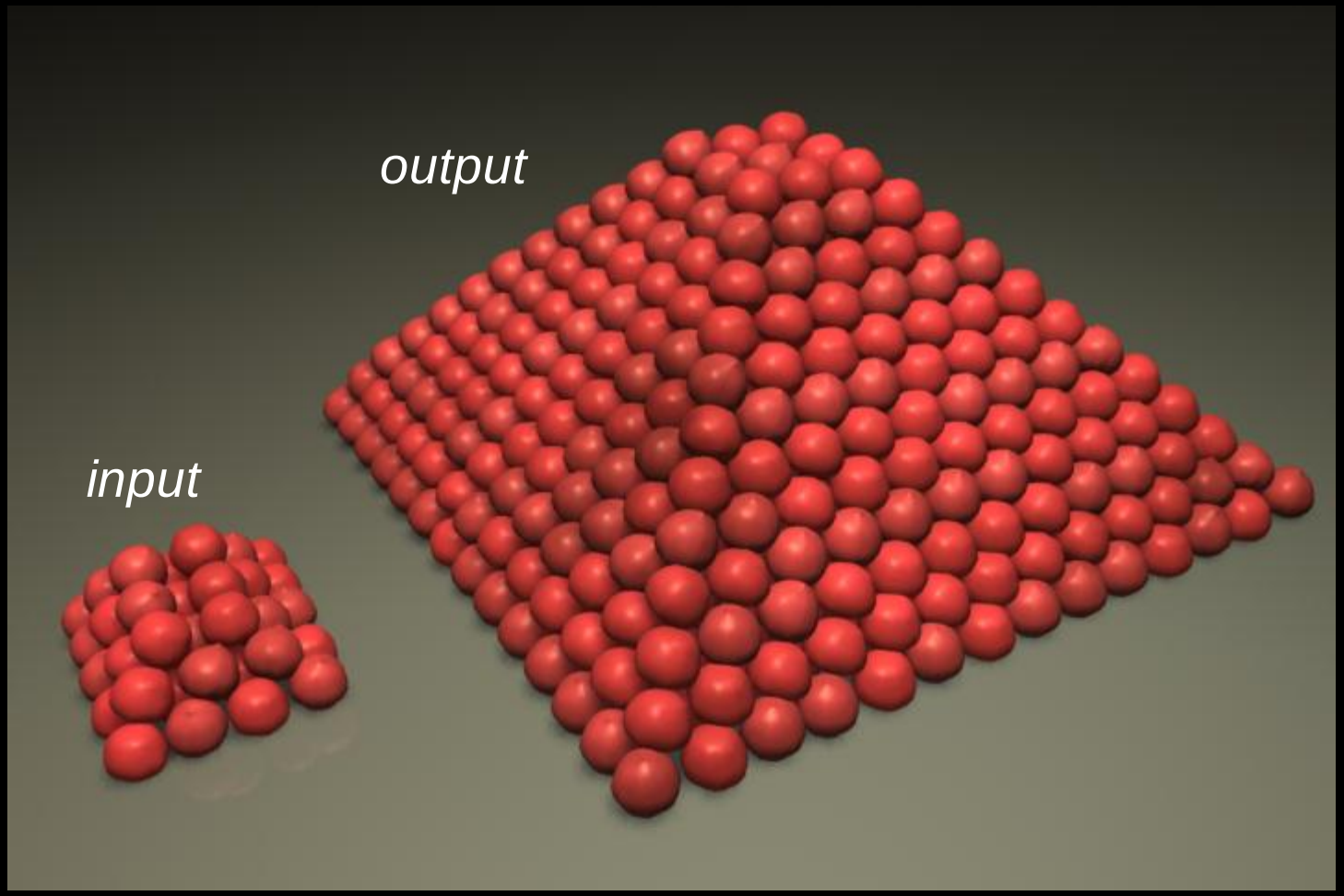
- Input preparation
  - Procedural
  - Simulation
  - Manual work
- Timing
  - One iteration: 5 seconds for 1000 elements
  - Each result: within 10 minutes



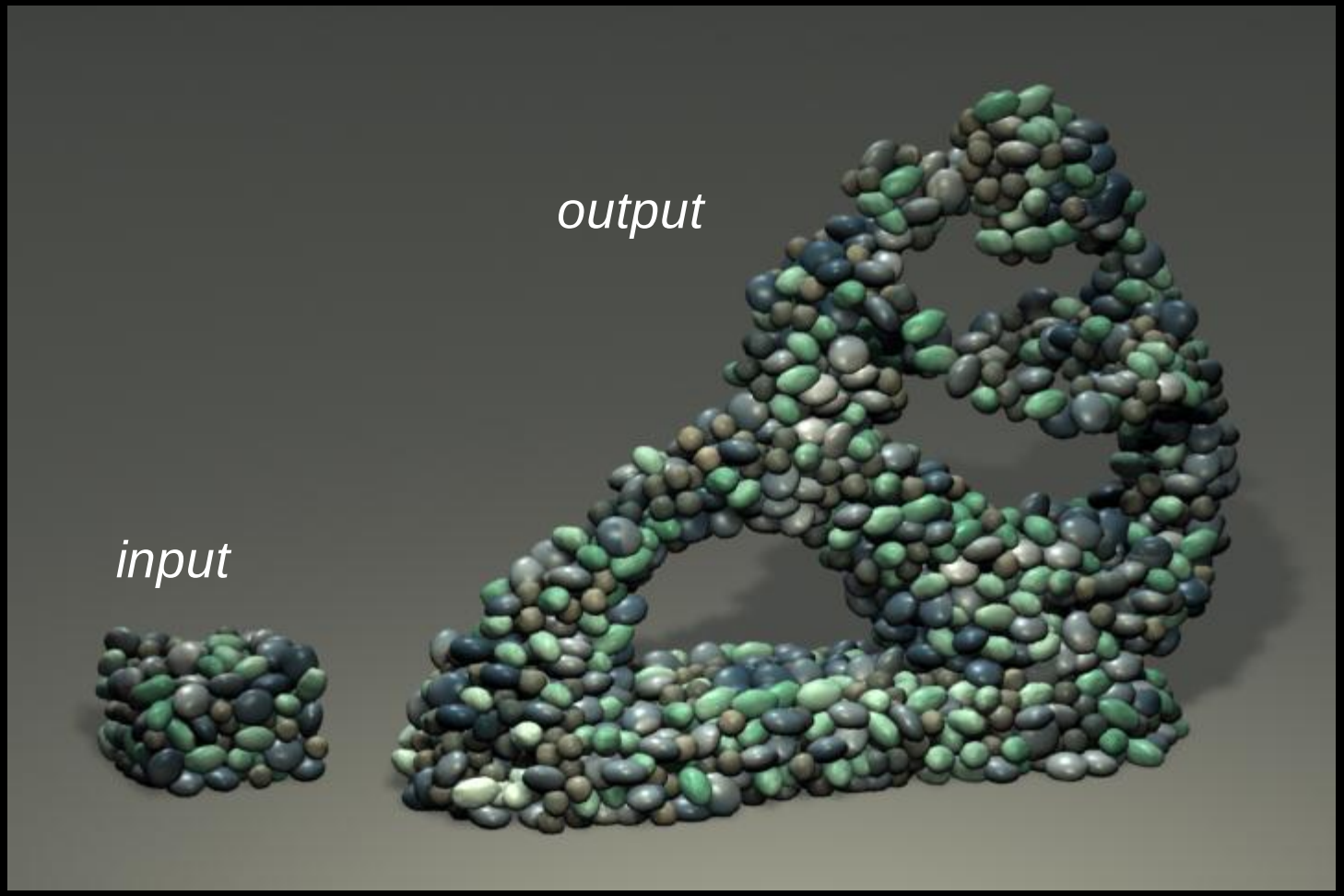



# Results

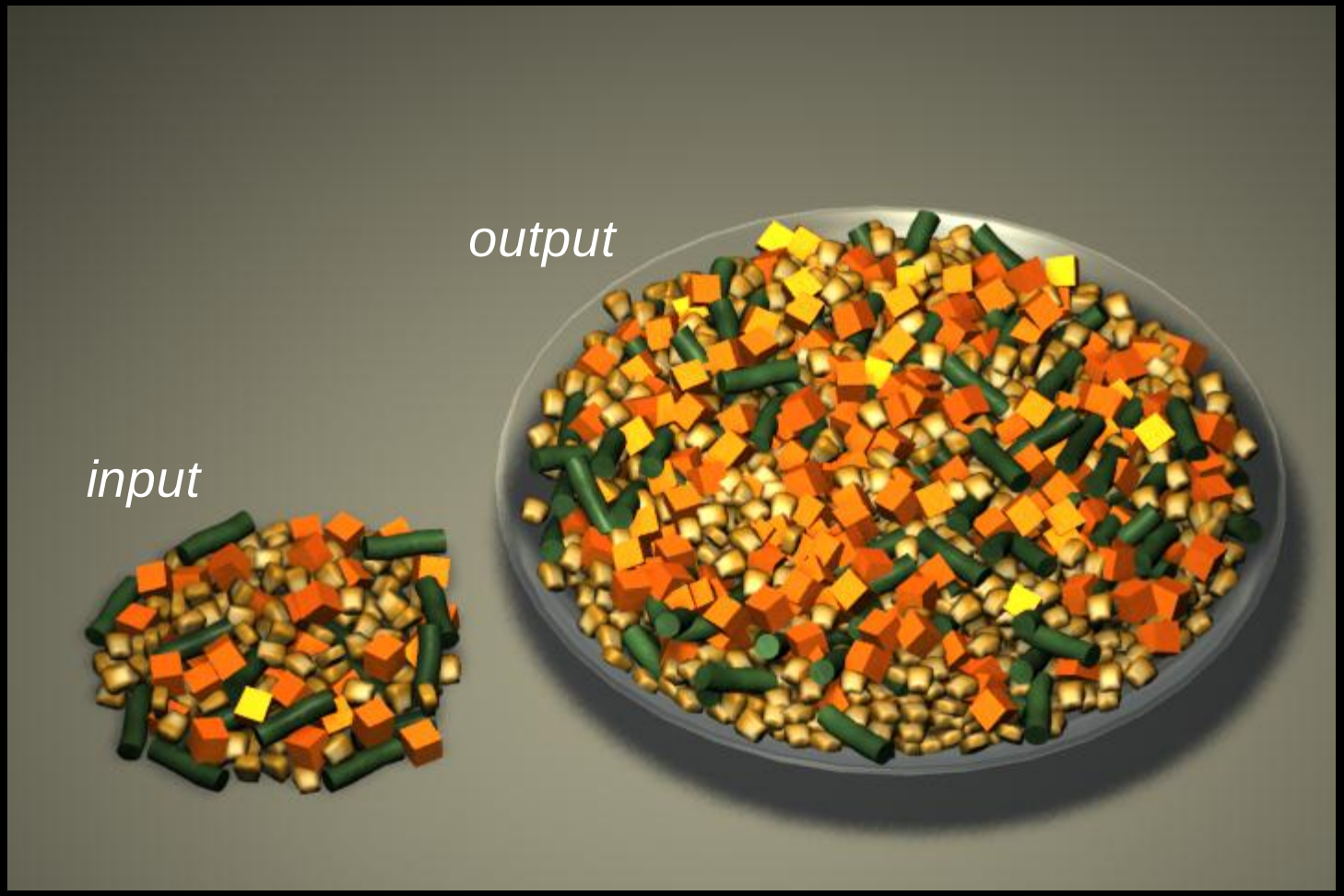
# Plum stack



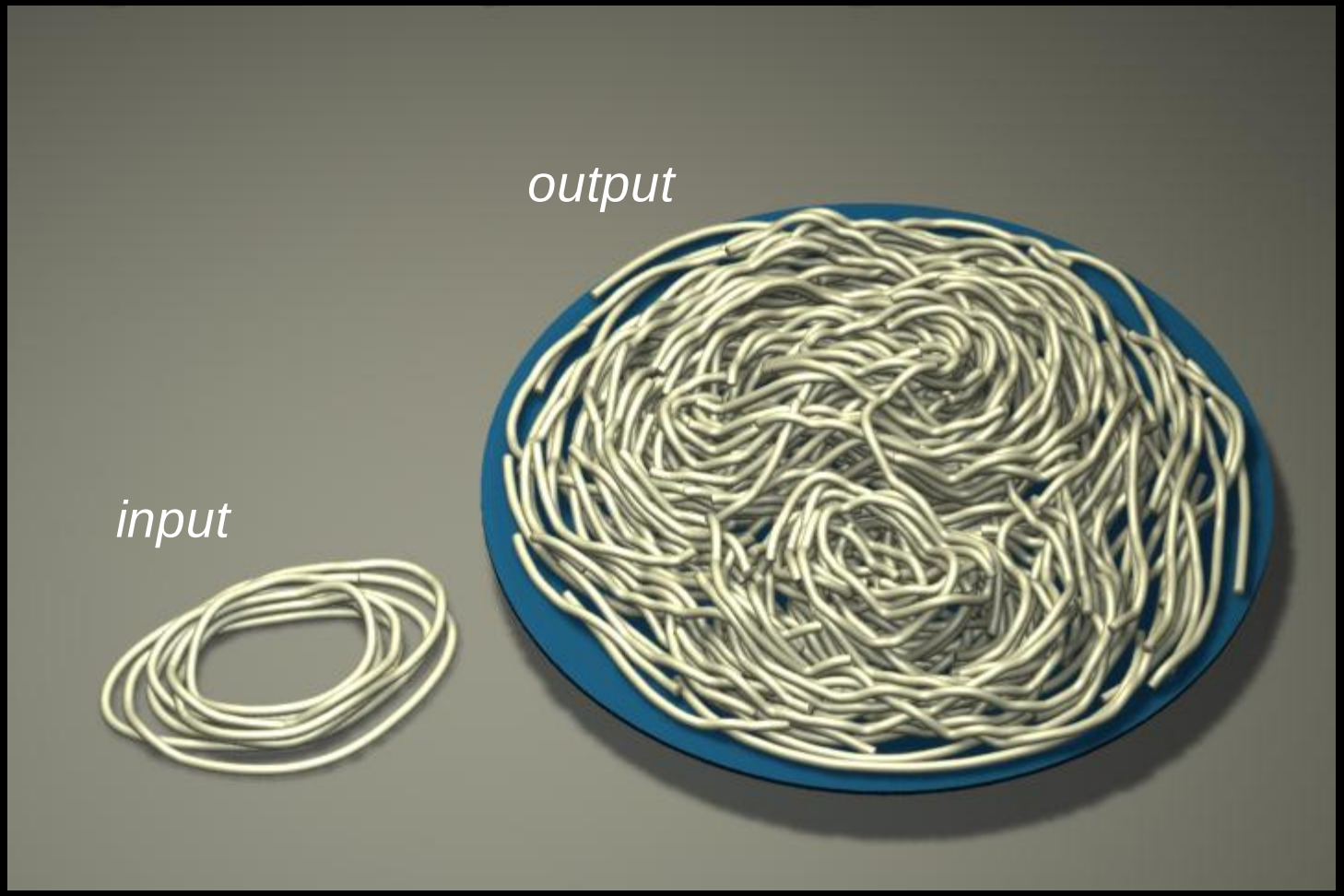
# Pebble sculpture



# Dish of corn kernels, carrots, beans

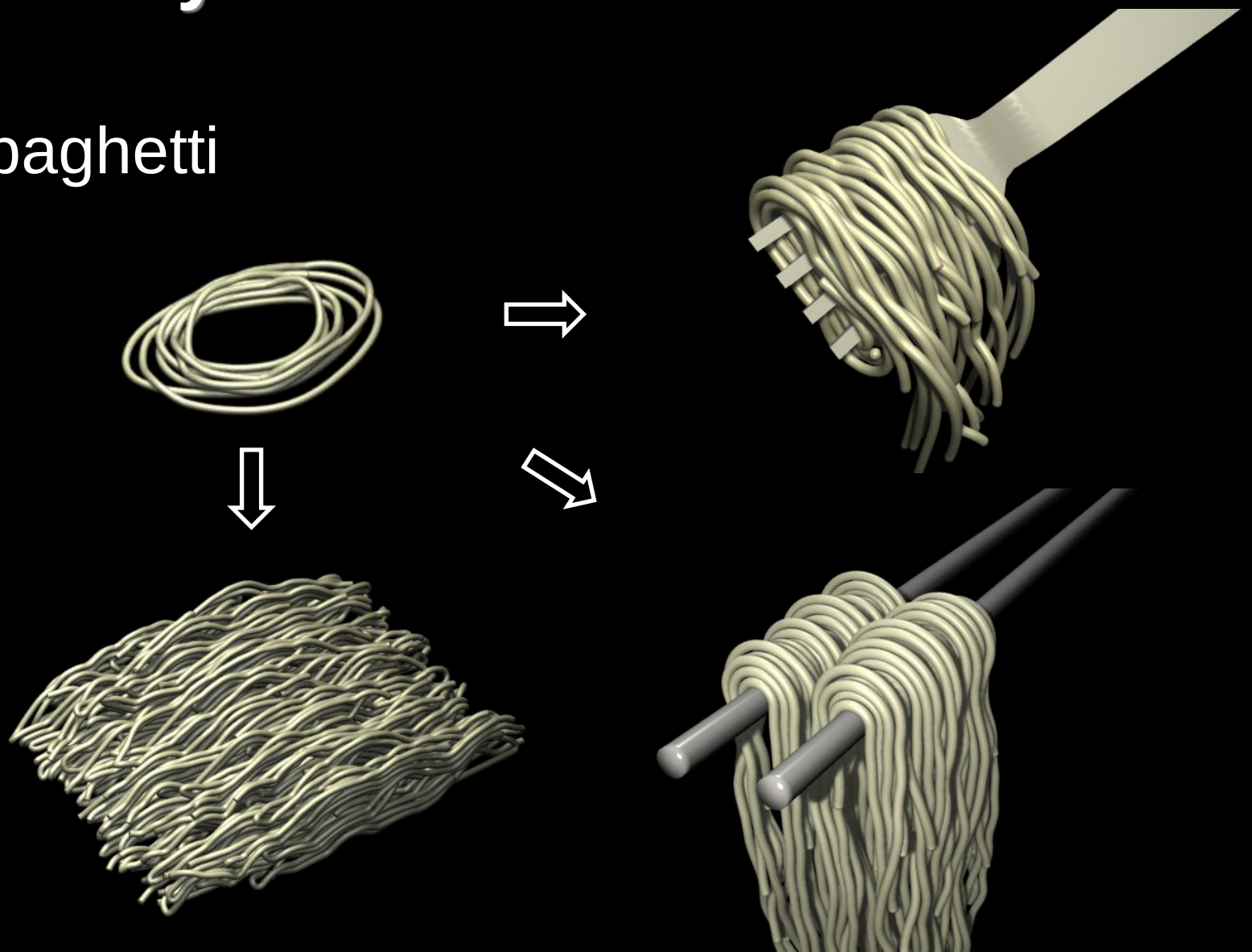


# Spaghetti



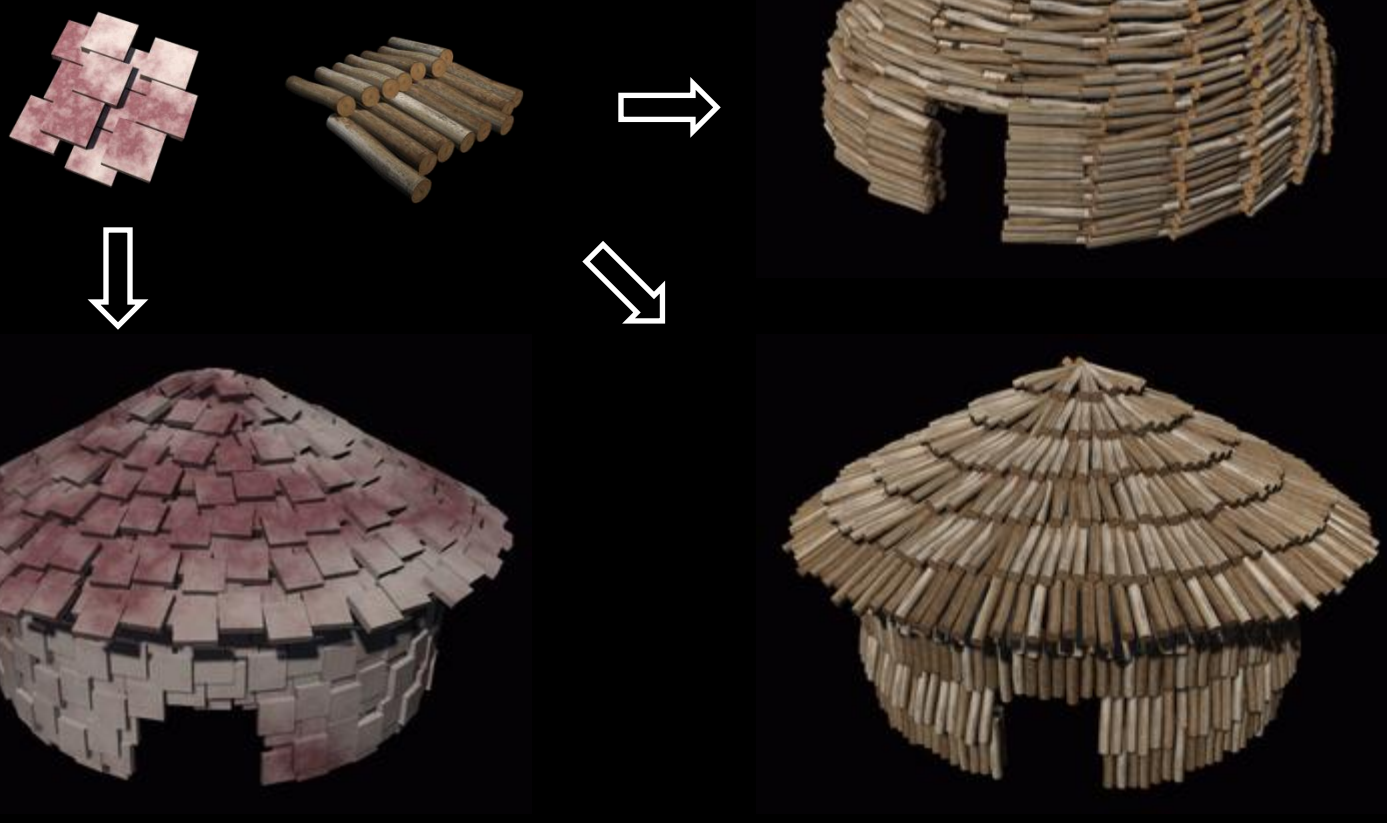
# More synthesis results

Spaghetti



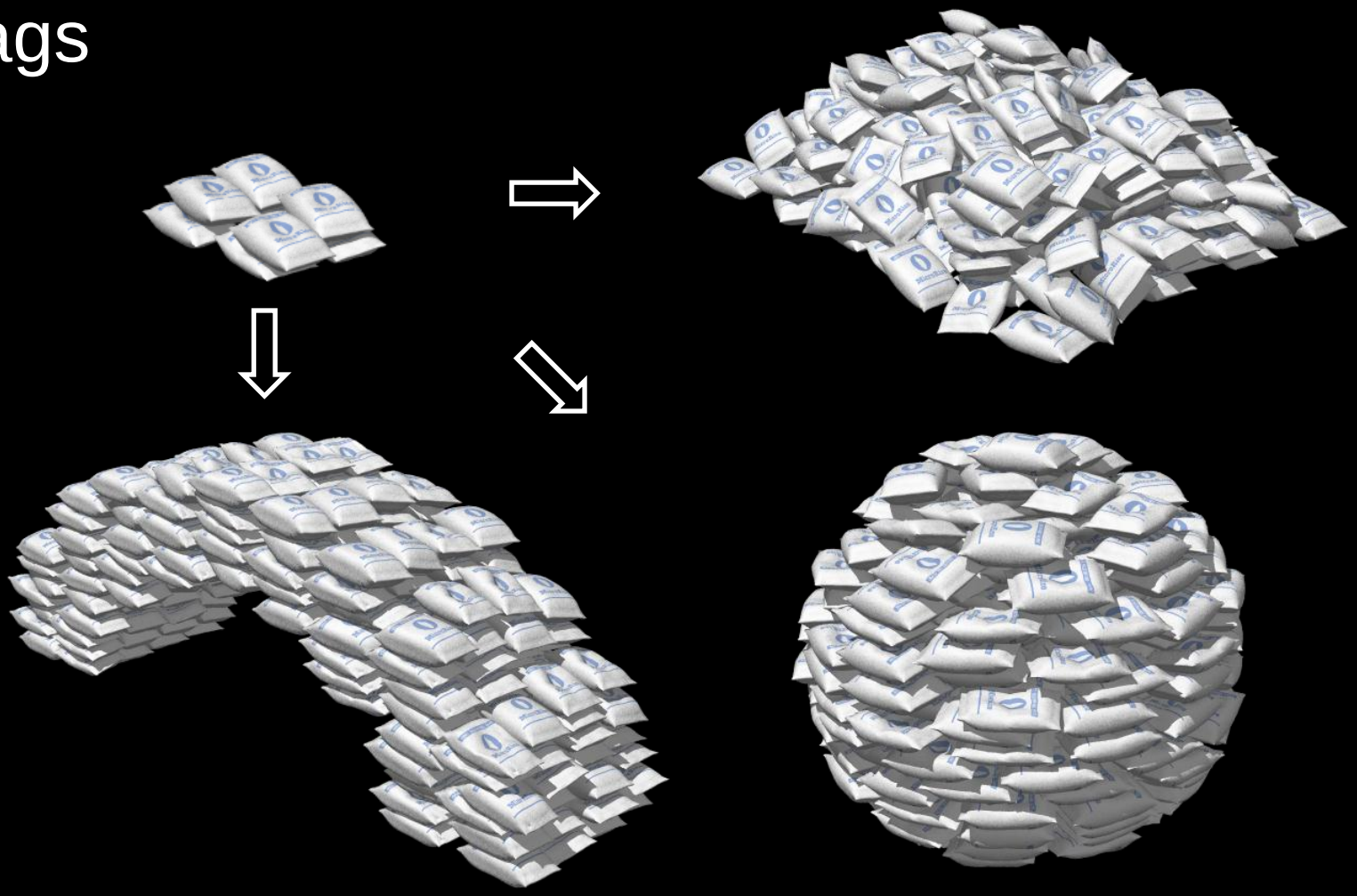
# More synthesis results

Houses



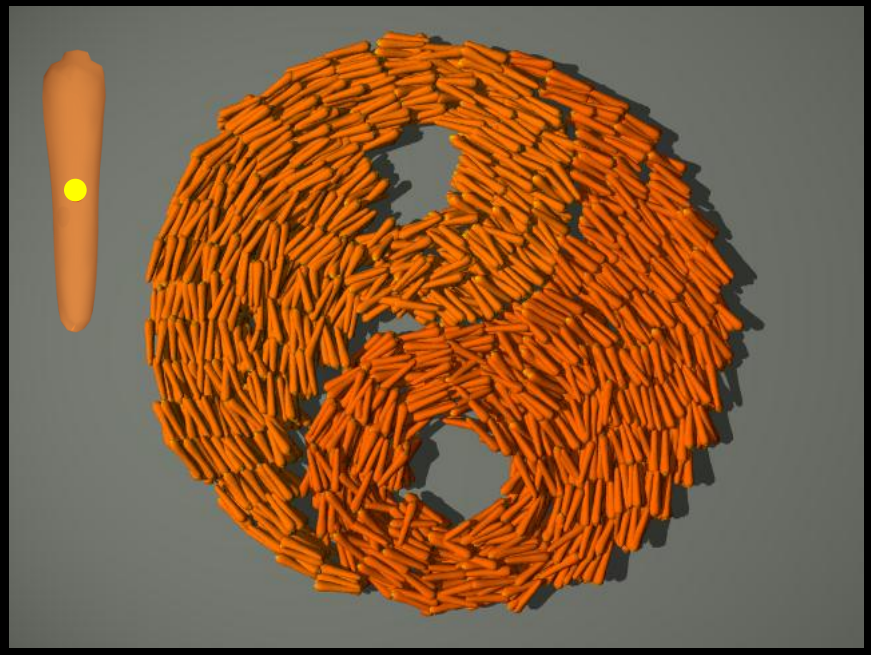
# More synthesis results

Bags



# Comparison

Single-sample vs. multi-sample

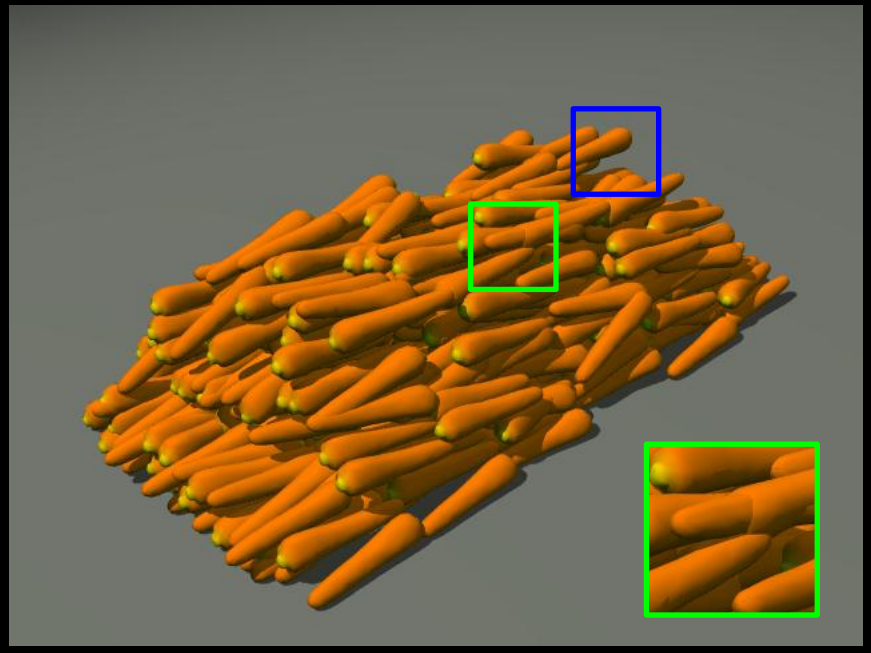


*single sample per element*

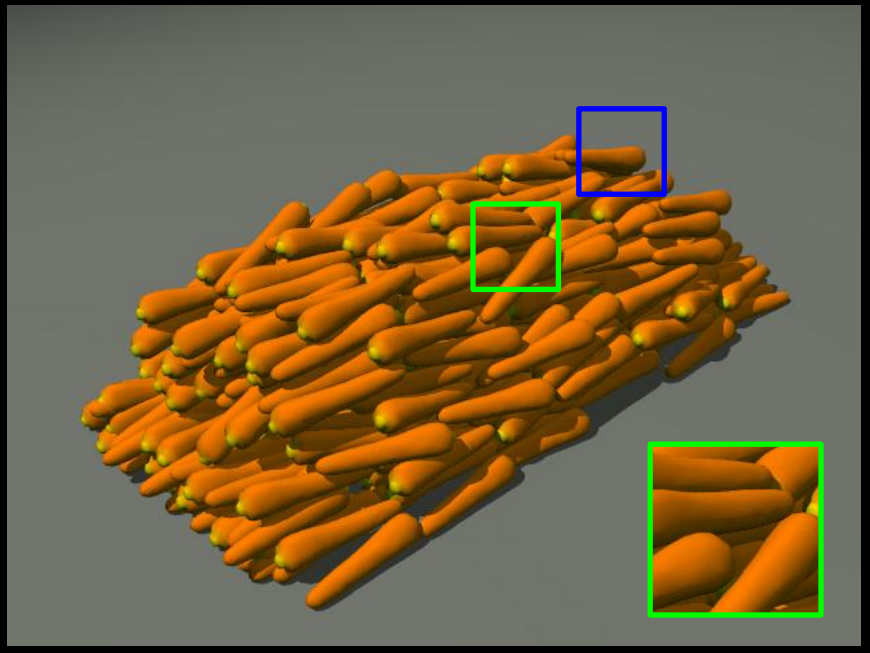
*multiple samples per element*

# Comparison

Without vs. with interleaved physics solver



*without interleaved physics solver*



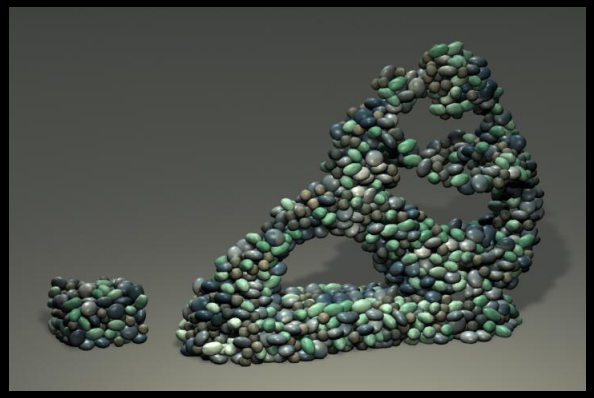
*with interleaved physics solver*

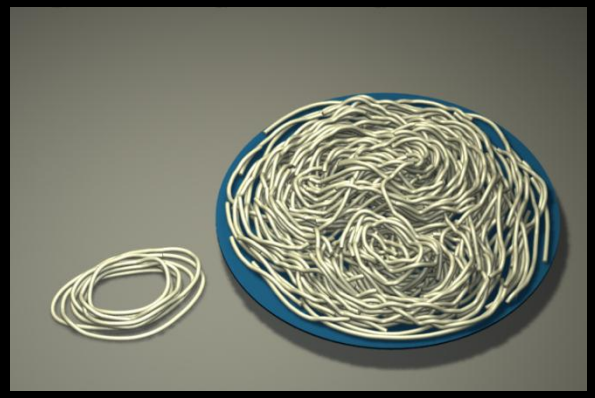
# Conclusion

A sample-based representation

A robust neighborhood metric

Automatic synthesis with flexible control





# Future work

- Automatically obtain individual elements
  - 2D textons [Ahuja and Todorovic 2007]
  - 3D geometry [Pauly et al. 2008]
- Minimum possible input
  - Summarization [Simakov et al. 2008]
  - Inverse synthesis [Wei et al. 2008]
- Dynamic element distributions
  - Crowd animation [Lerner et al. 2007; Ju et al. 2010]

# Acknowledgements

Weiwei Xu

Xin Sun

Stephen Lin

Matt Callcut

Yue Dong

Anonymous reviewers

Shuitian Yan

Luoying Liu

La Tu

Hongwei Li

Laras Anjung Gallery

# Thank you!