SIGGRAPH2012

Symmetry-Guided Texture Synthesis and Manipulation

Vladimir G. Kim

Princeton University

Yaron Lipman

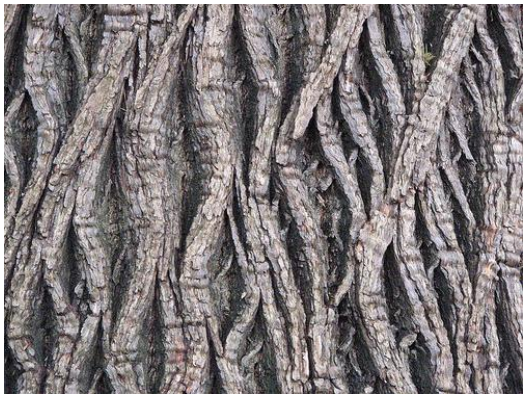
Weizmann Institute

Thomas Funkhouser

Princeton University

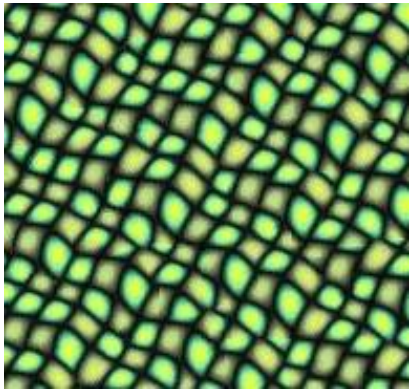
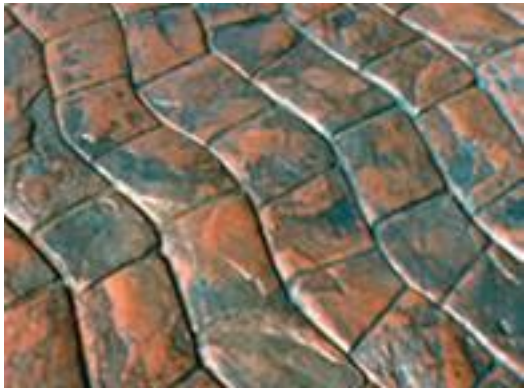
Motivation



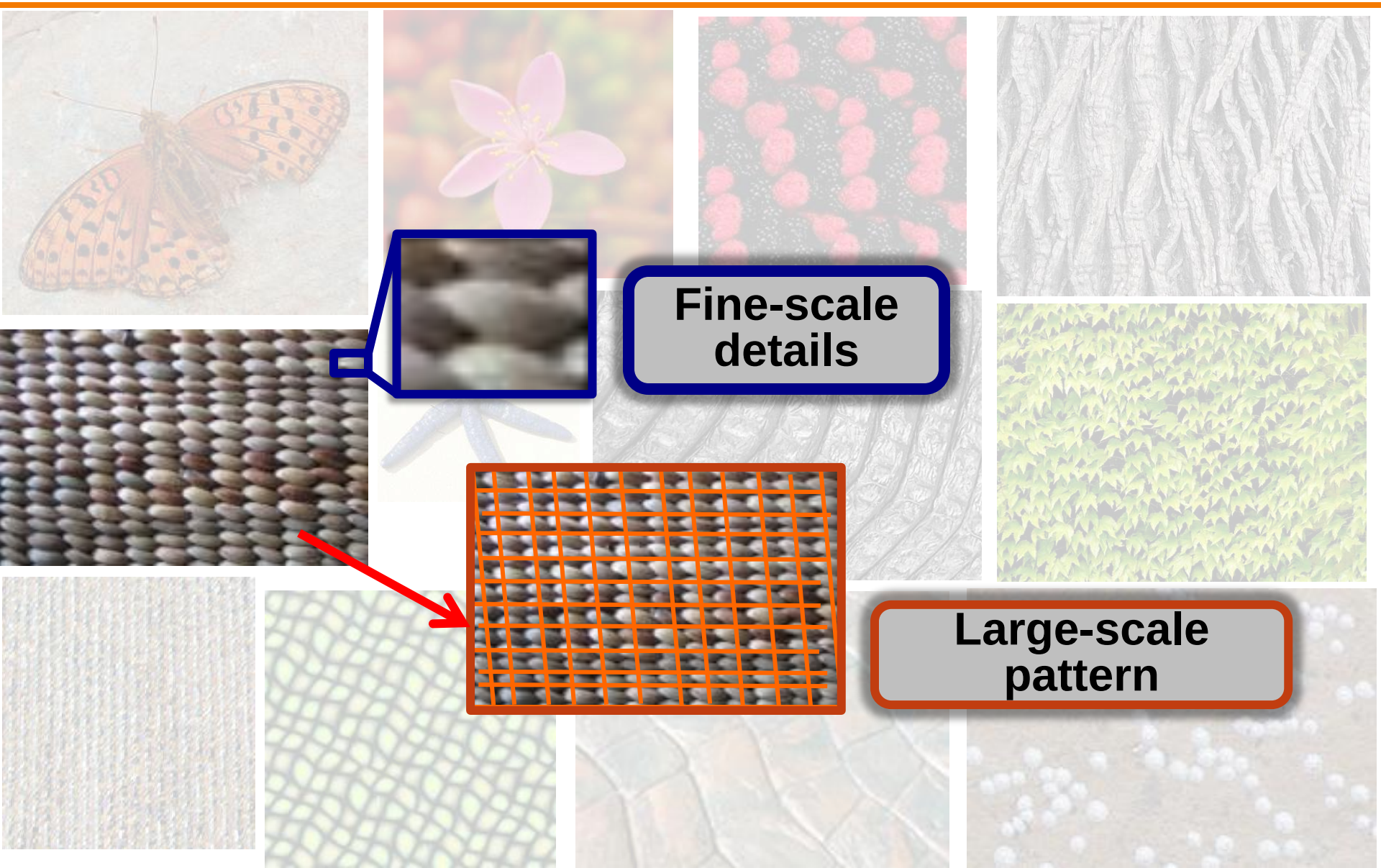





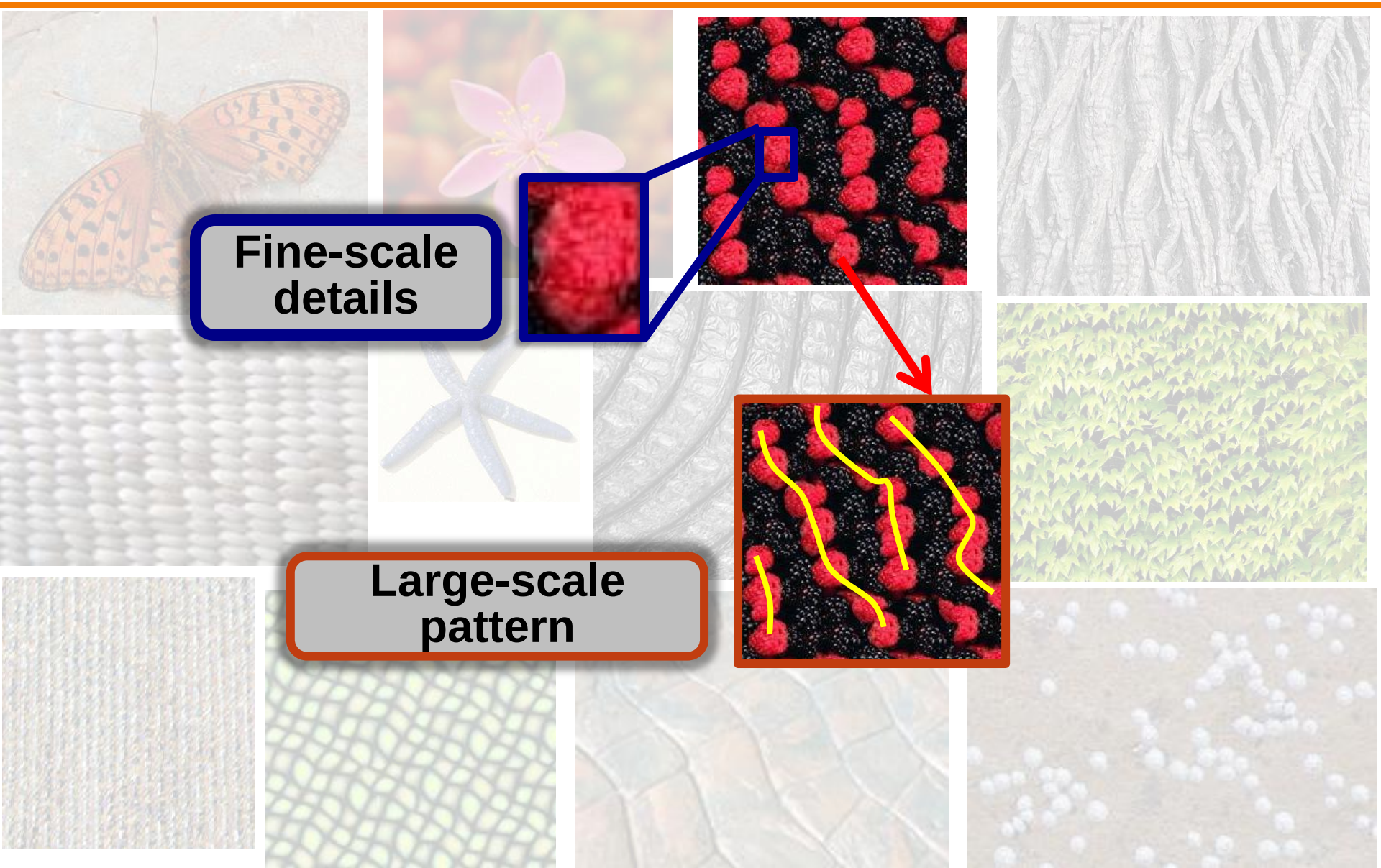


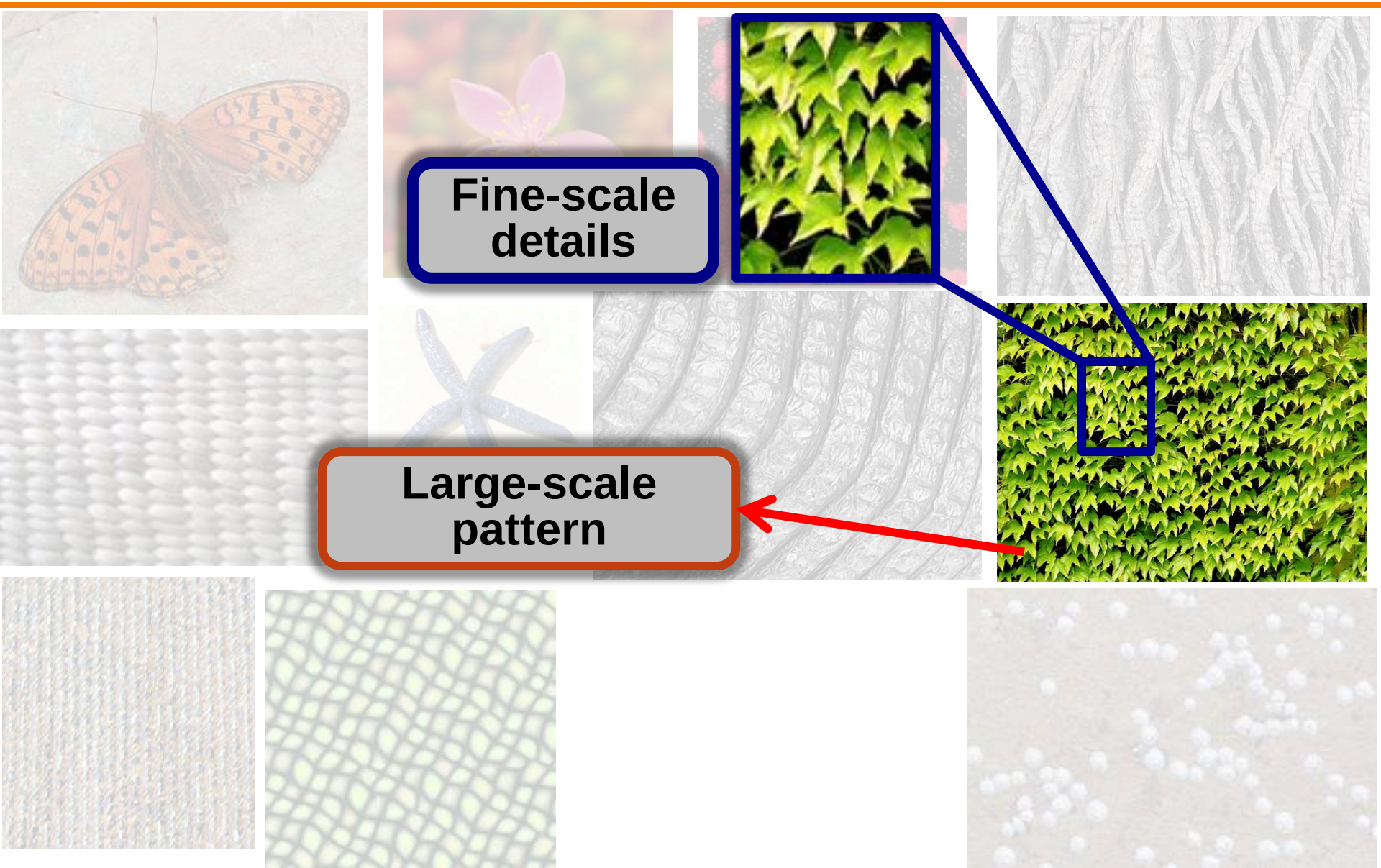

Motivation



Motivation



Motivation



Goal

Control large-scale patterns while preserving fine-scale details

- Synthesize textures with desirable patterns
- Manipulate patterns in textures and images

Related Work

Analyzing, manipulating, and synthesizing textures

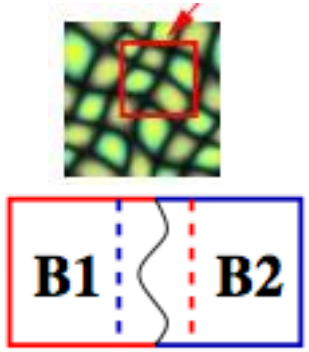
- Texture Synthesis
- Near-Regular Textures
- Symmetry Detection

Related Work

Analyzing, manipulating, and synthesizing textures

- Texture Synthesis
  - Near-Regular Textures
  - Symmetry Detection

Patch-based synthesis



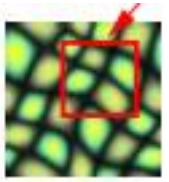
Efros and Lleung ‘99
Efros and Freeman ’01,
Kwatra et al. ‘03,
Kwatra et al. ’05,
Lefebvre et al. ‘10

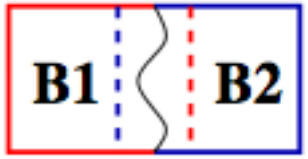
Related Work

Analyzing, manipulating, and synthesizing textures

- Texture Synthesis
  - Near-Regular Textures
  - Symmetry Detection

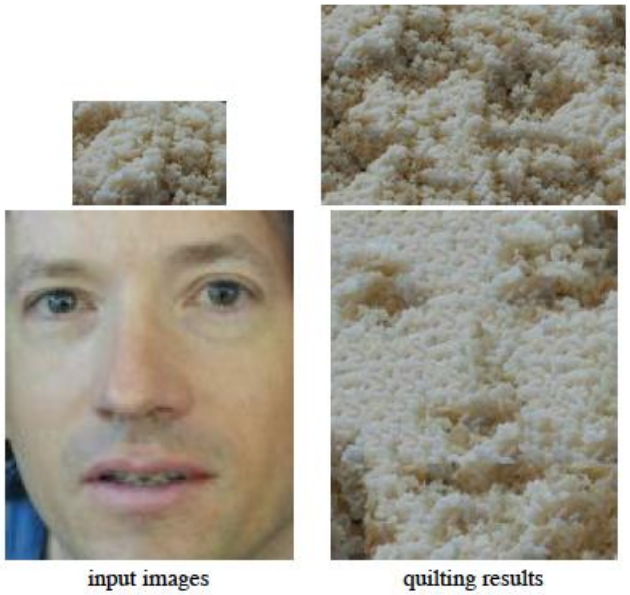
Patch-based synthesis



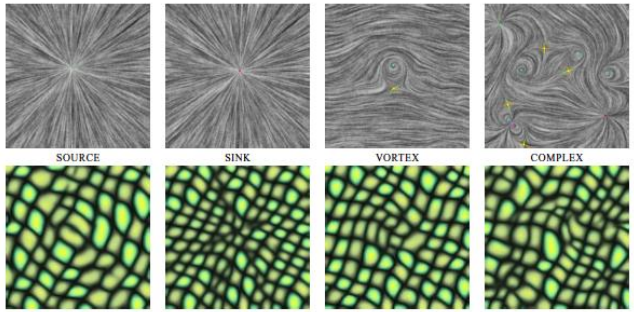


Efros and Lleung '99
Efros and Freeman '01,
Kwatra et al. '03,
Kwatra et al. '05,
Lefebvre et al. '10

Additional Objectives



Texture Transfer



Flow-field Guided

Interactive Merging

Related Work

Analyzing, manipulating, and synthesizing textures

- Texture Synthesis
- ➢ Near-Regular Textures
- Symmetry Detection

Near-Regular Texture Analysis and Manipulation. *Liu et al. '04*

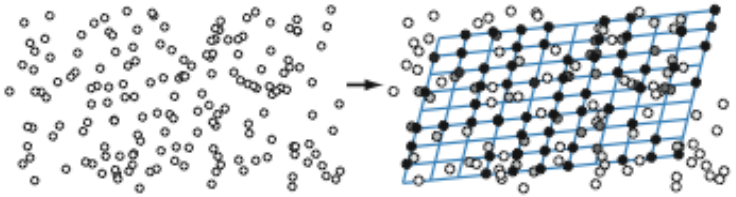
Image De-Fencing. *Liu et al. '08, Park et al. '10*

Related Work

Analyzing, manipulating, and synthesizing textures

- Texture Synthesis
- Near-Regular Textures
- ➢ Symmetry Detection

Symmetry Detection from Real World Images. *Ingmar Rauschert, Kyle Brocklehurst, Somesh Kashyap, Jingchen Liu, Yanxi Liu, '12*



Partial and approximate symmetry detection…

Mitra et al. '06

Discovering Structural Regularity…

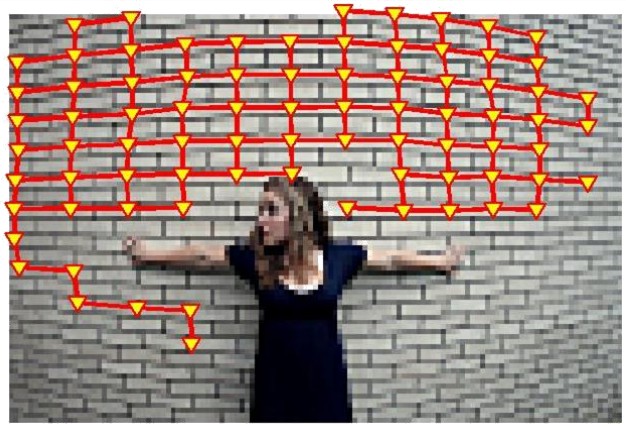
Pauly et al. '08

Related Work

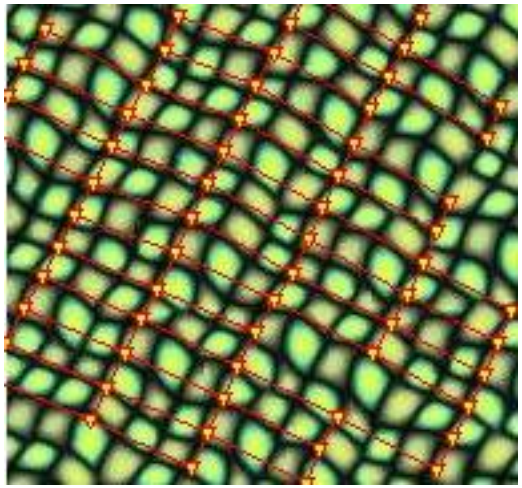
Analyzing, manipulating, and synthesizing textures

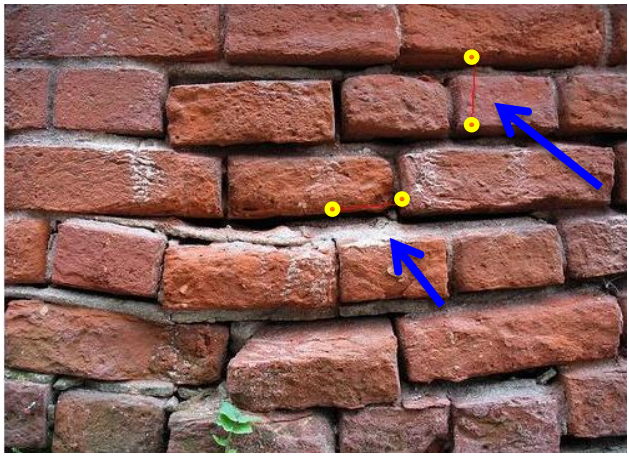
- Texture Synthesis
- Near-Regular Textures
- ➢ Symmetry Detection

Lattice detection is hard…



© cwazymandy





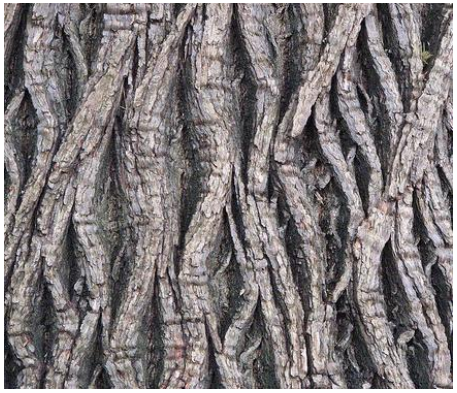
Deformed Lattice Detection… *Park et al. '09*,
Discovering Texture Regularity… *Hays et al. '06*

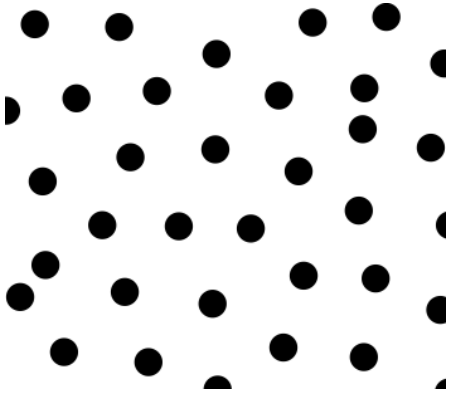
Related Work

Analyzing, manipulating, and synthesizing textures

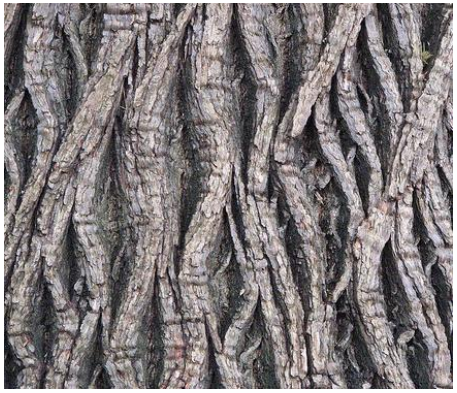
- Texture Synthesis
- Near-Regular Textures
- ➢ Symmetry Detection

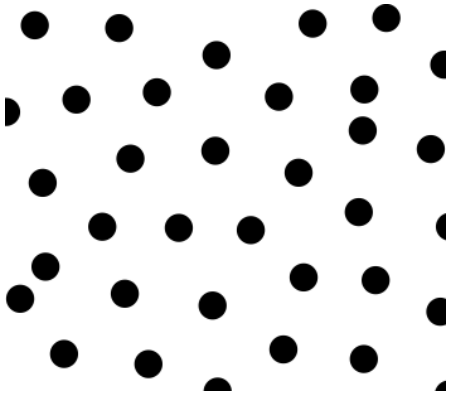
Lattice detection is hard…
especially if there is none!



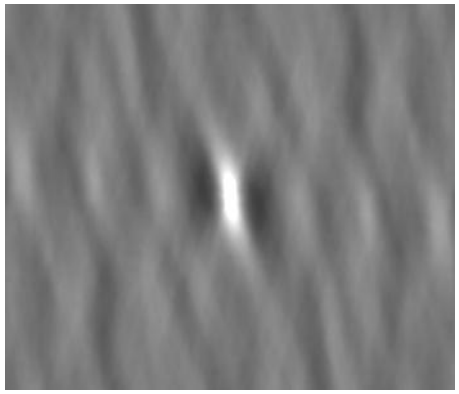


Our Approach

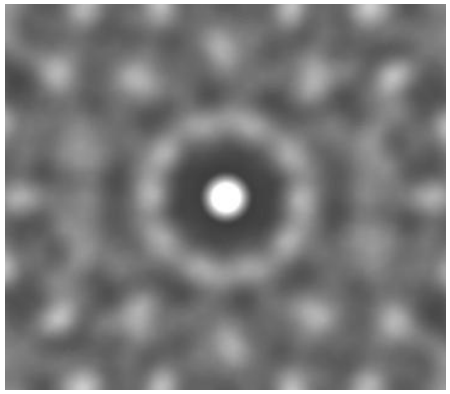




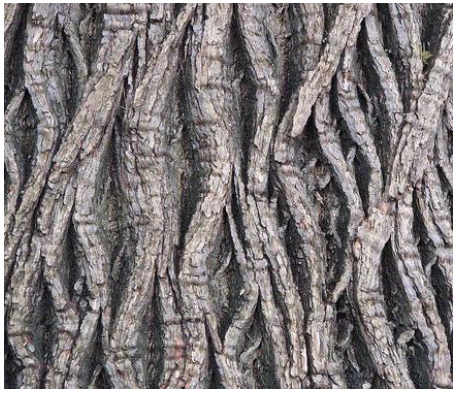
Approximate Symmetry Representation:

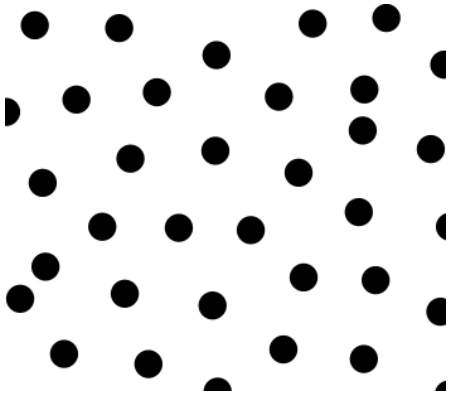


Approximate Symmetry Representation:

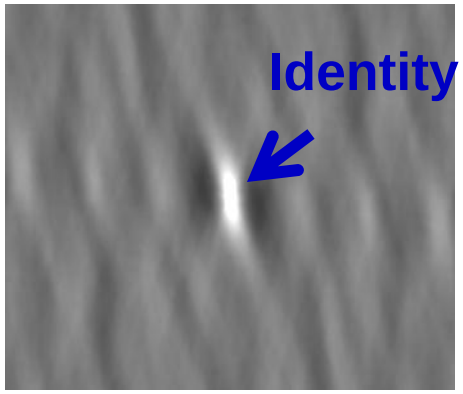


Our Approach

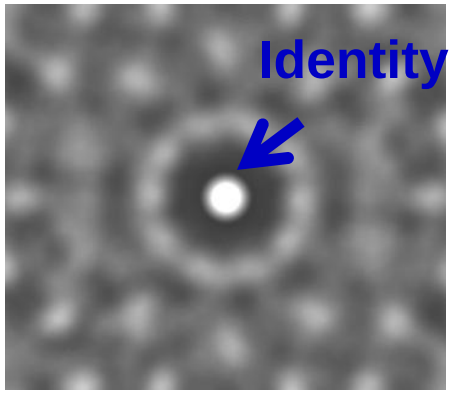




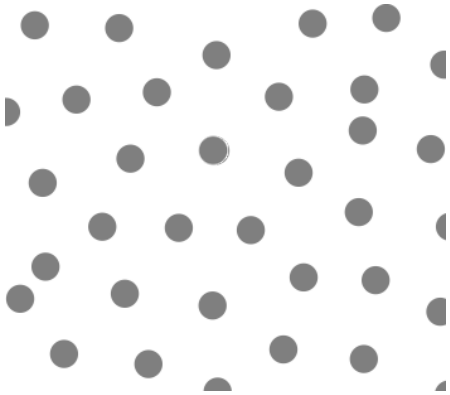
Approximate Symmetry Representation:



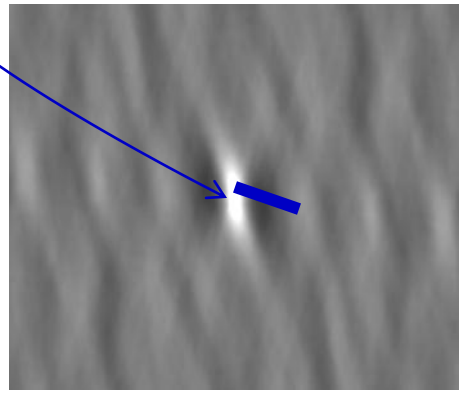
Approximate Symmetry Representation:



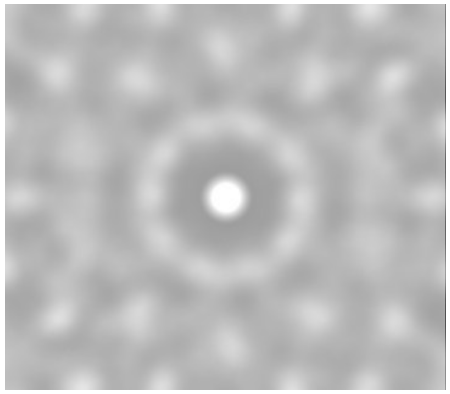
Our Approach



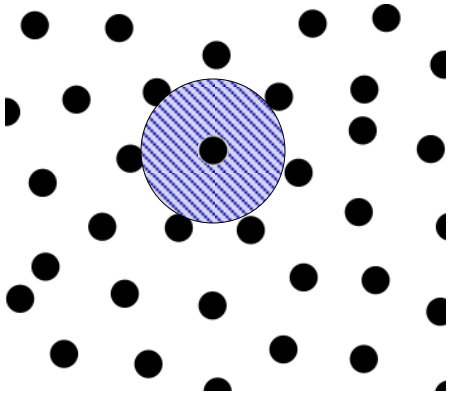
Approximate Symmetry Representation:



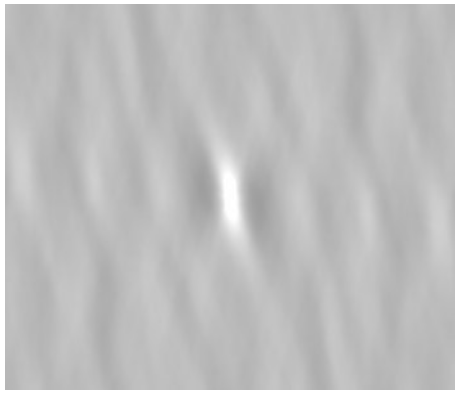
Approximate Symmetry Representation:



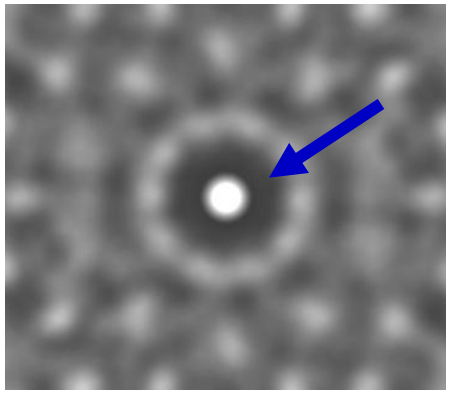
Our Approach



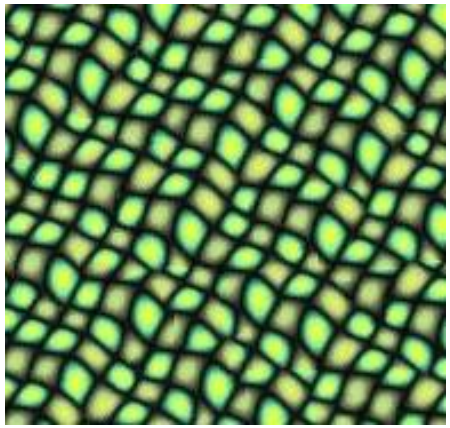
Approximate Symmetry Representation:



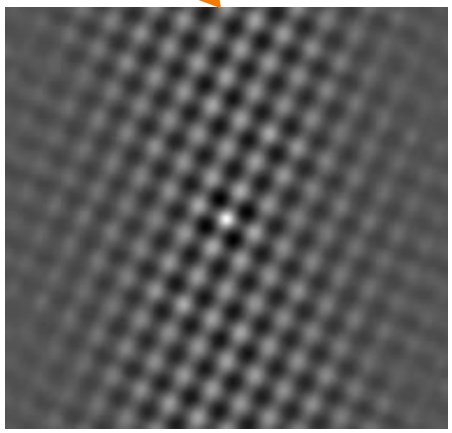
Approximate Symmetry Representation:



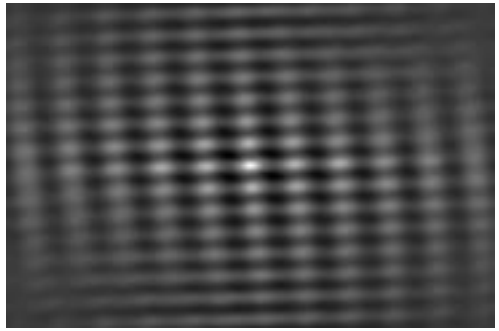
Our Approach



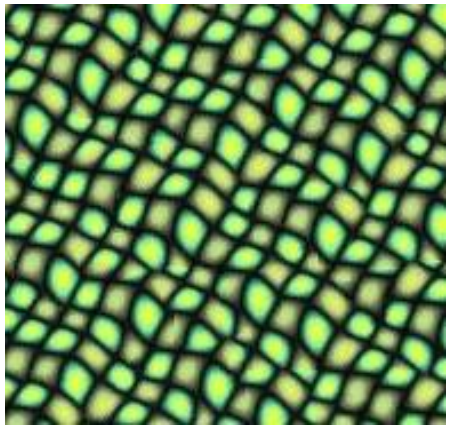
Approximate Symmetry Representation:



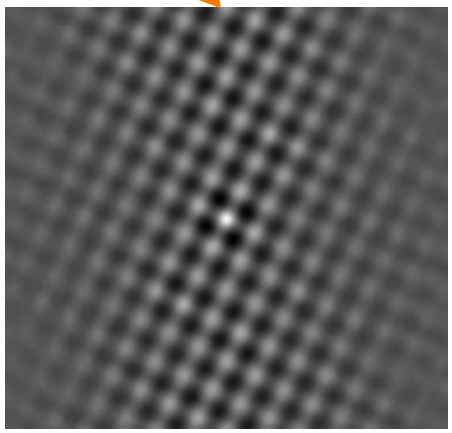
Approximate Symmetry Representation:



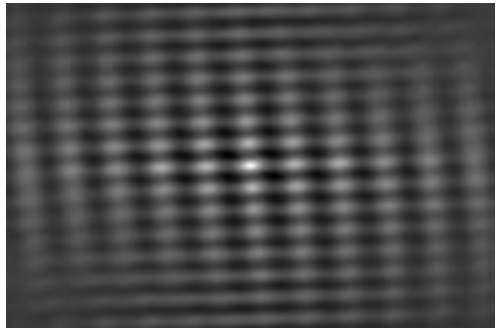
Our Approach



Approximate Symmetry Representation:



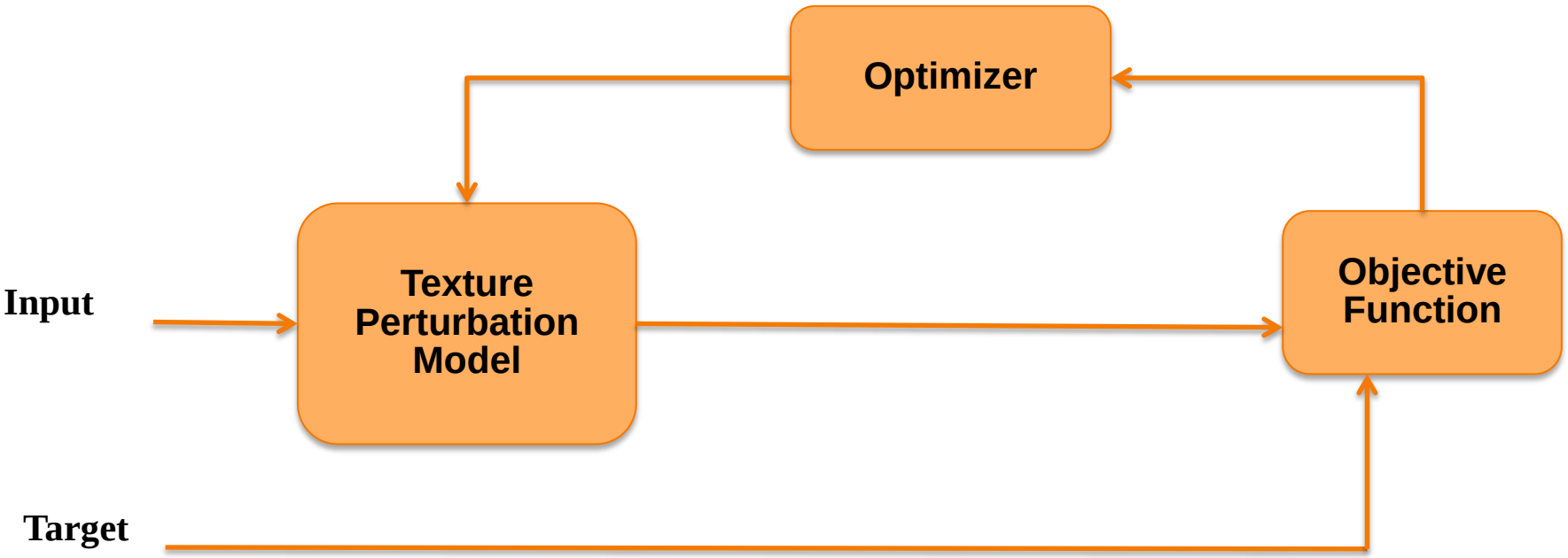
Approximate Symmetry Representation:



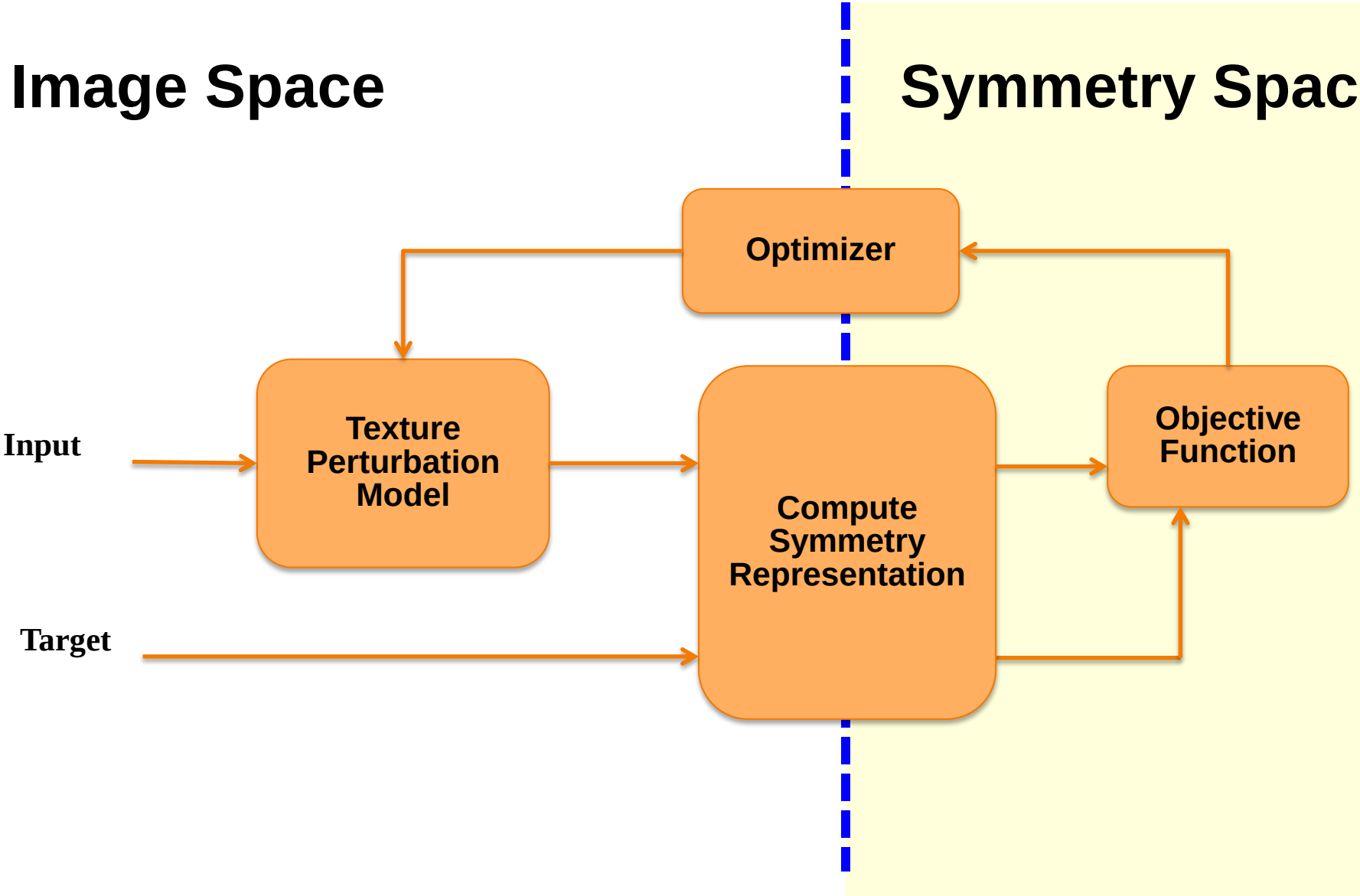
"Reflective symmetry descriptor", *Kazhdan et al. 2003*,
"Planar reflective symmetry transform", *Podolak et al. 2006, …*

Typical Texture Synthesis Pipeline

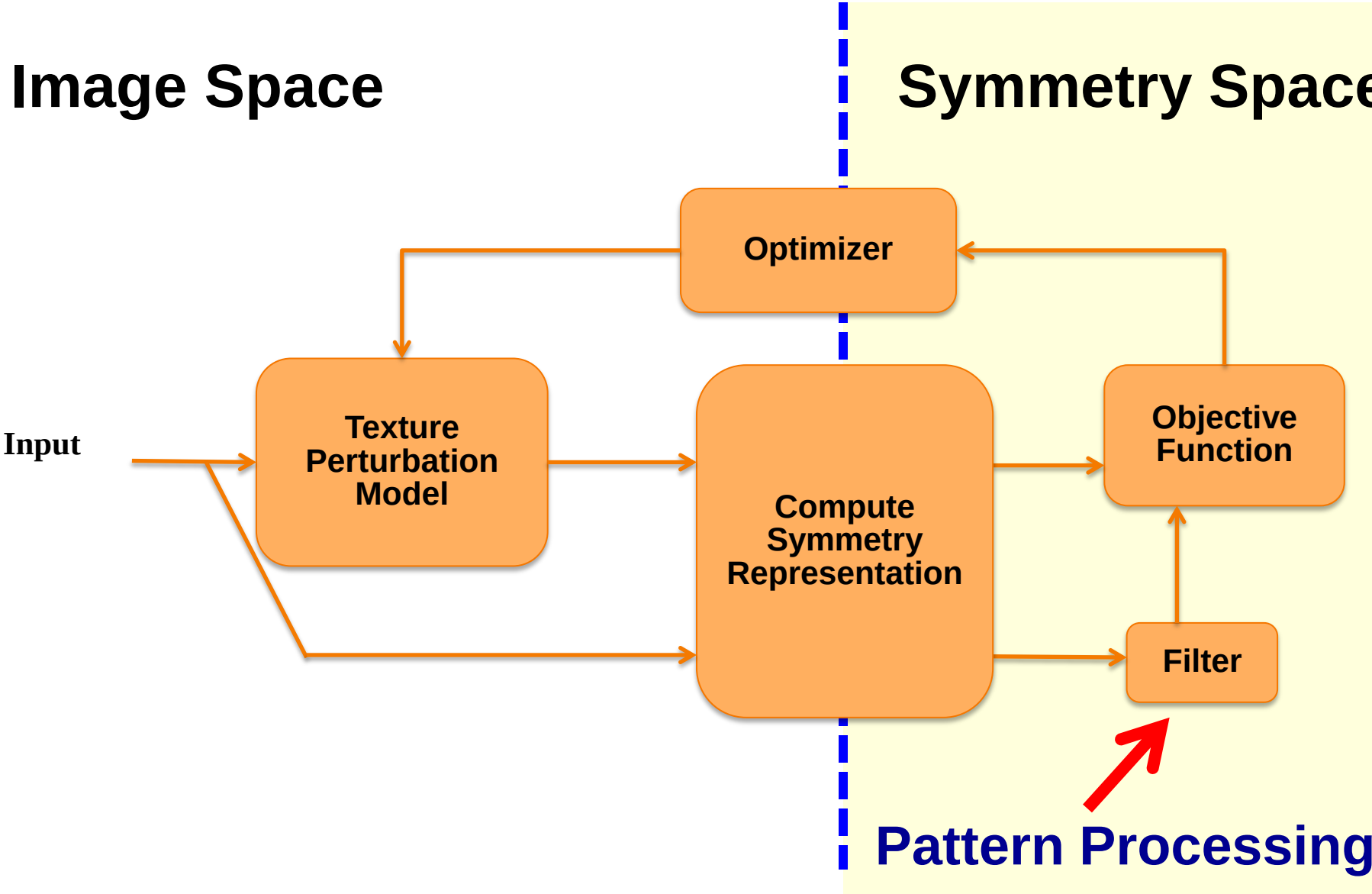
Image Space



Our Framework Overview



Pattern Processing



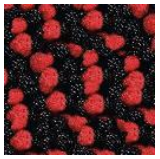
Pattern Processing: Goal

Filter the symmetry representation: e.g.

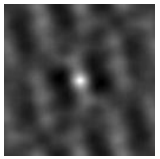
- Blur = make less symmetric
- Sharpen = make more symmetric
- Scale = scale pattern (same scale of texture details)
- …

Pattern Processing: Scale

Input

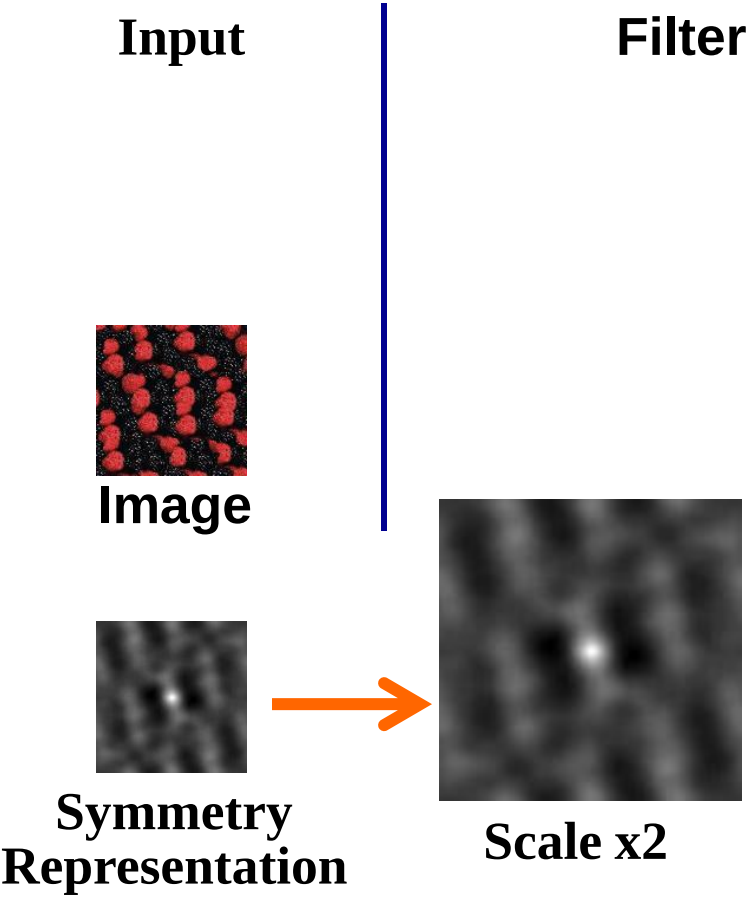


Image

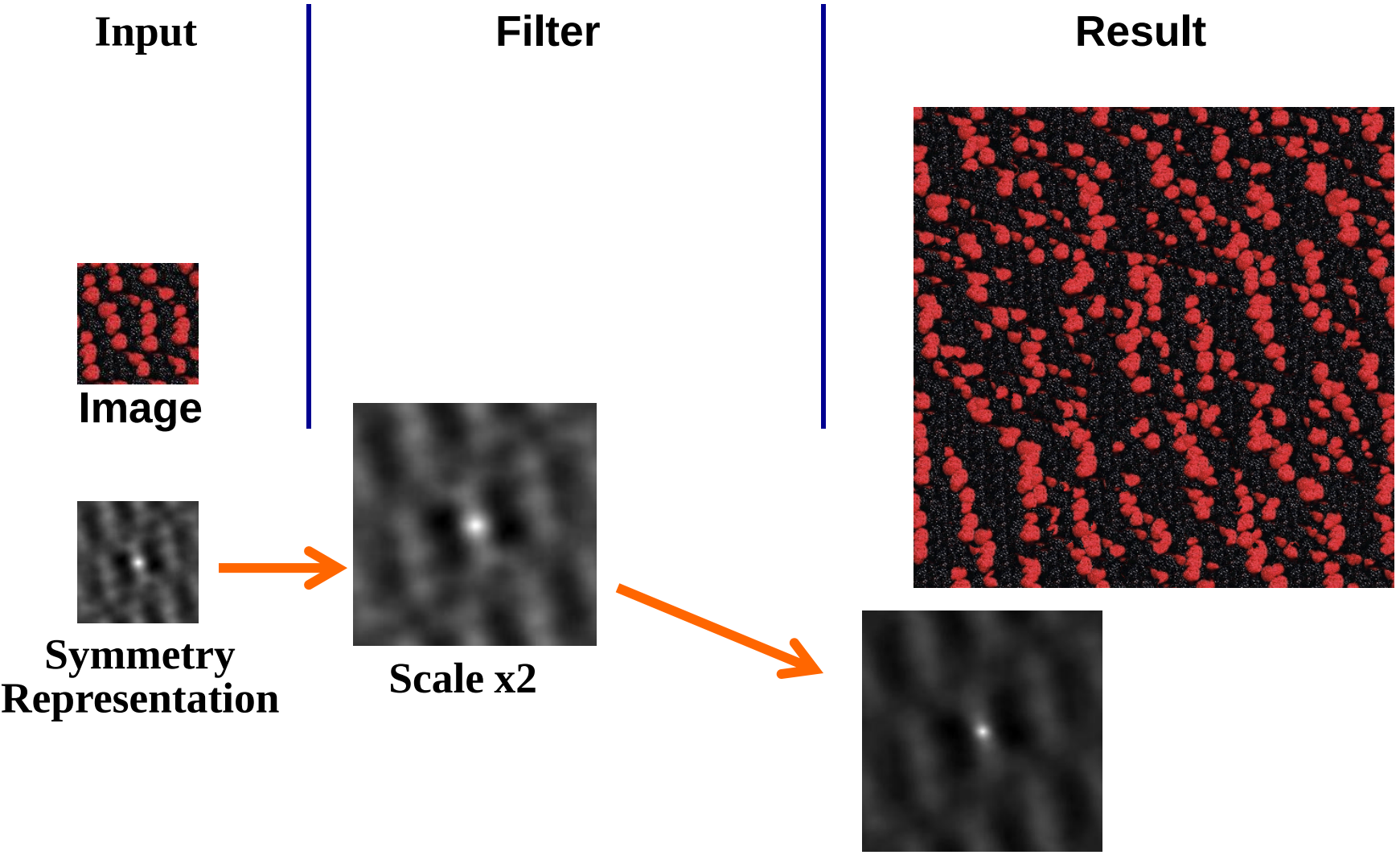


Symmetry Representation

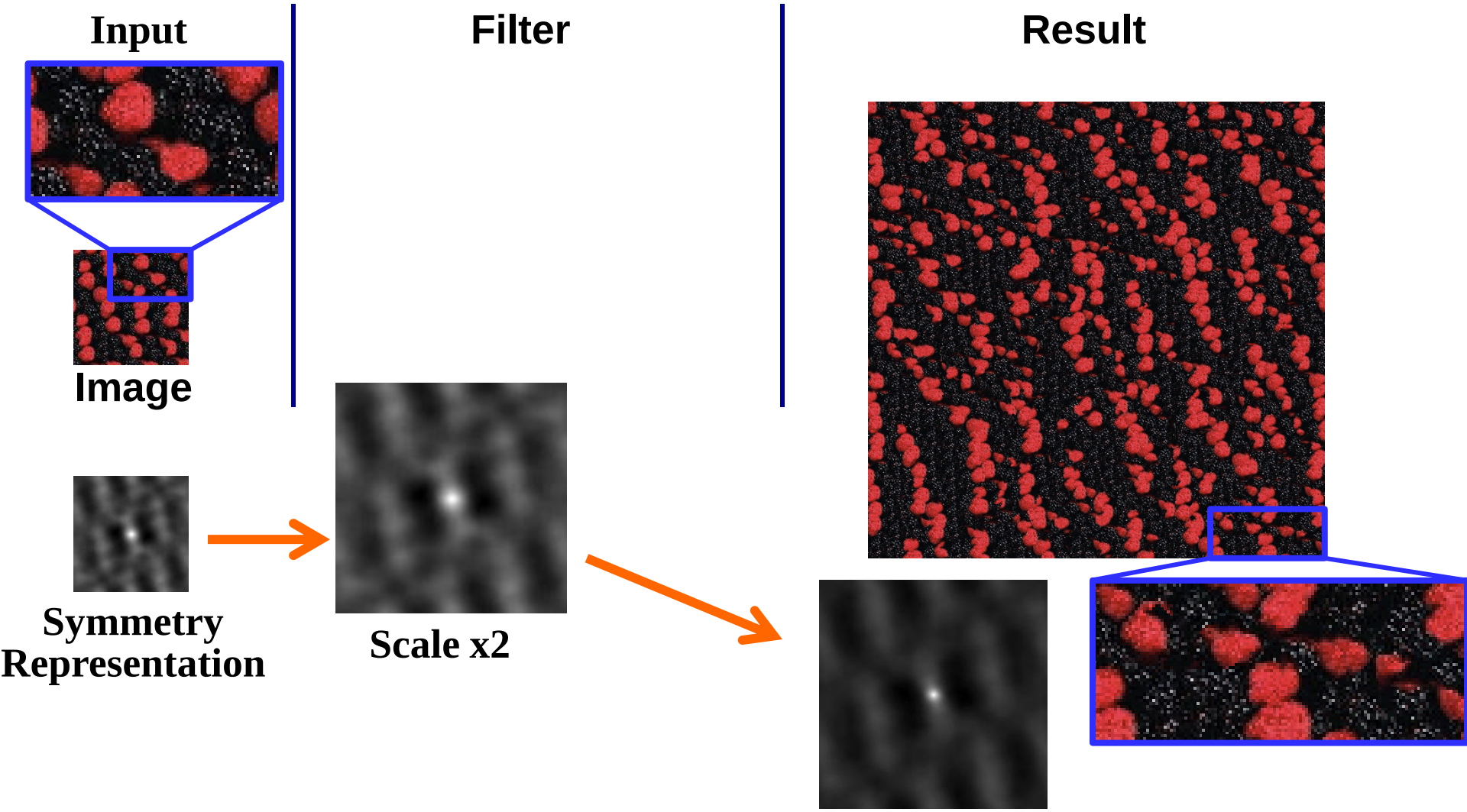
Pattern Processing: Scale



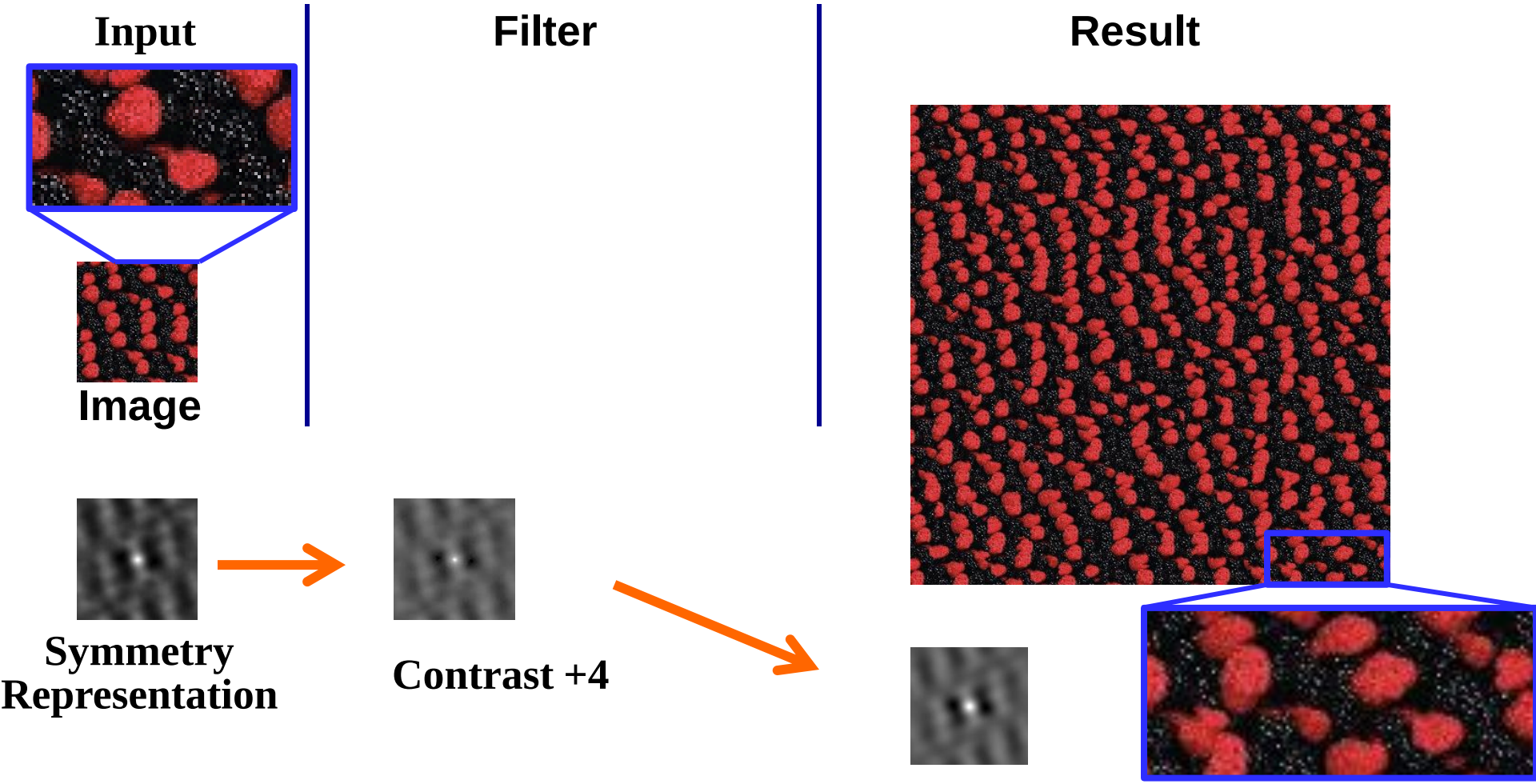
Pattern Processing: Scale



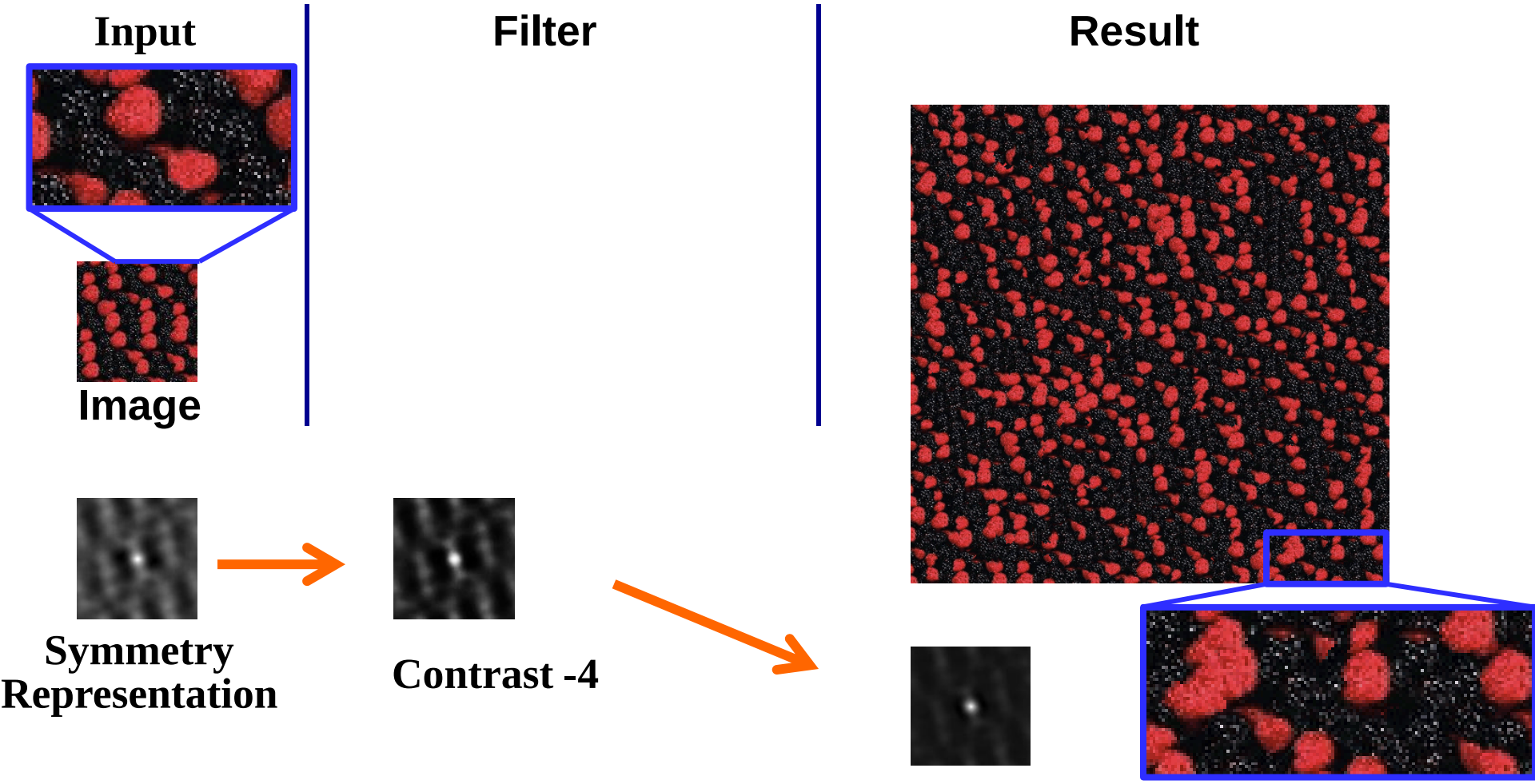
Pattern Processing: Scale



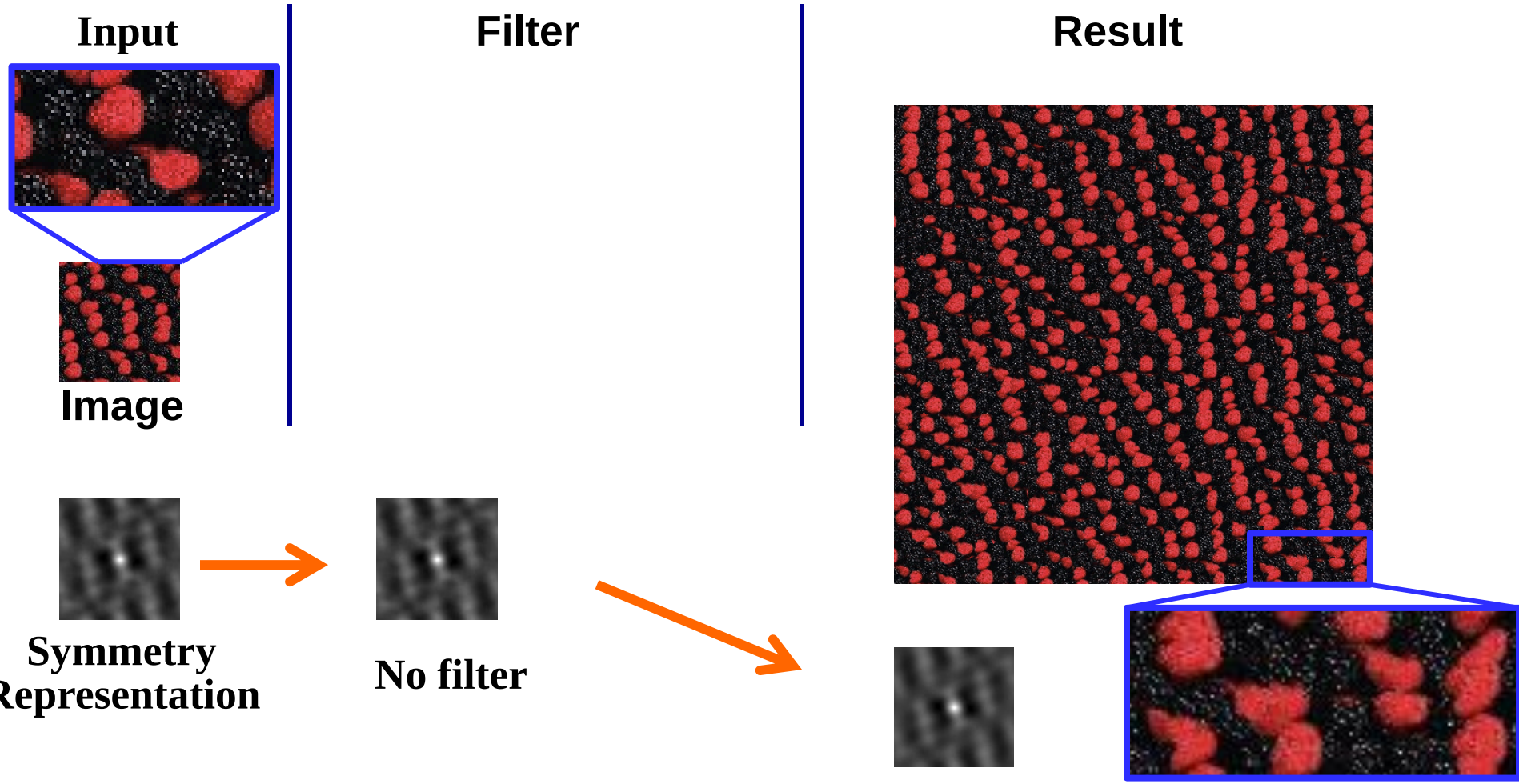
Pattern Processing: Reduce Contrast



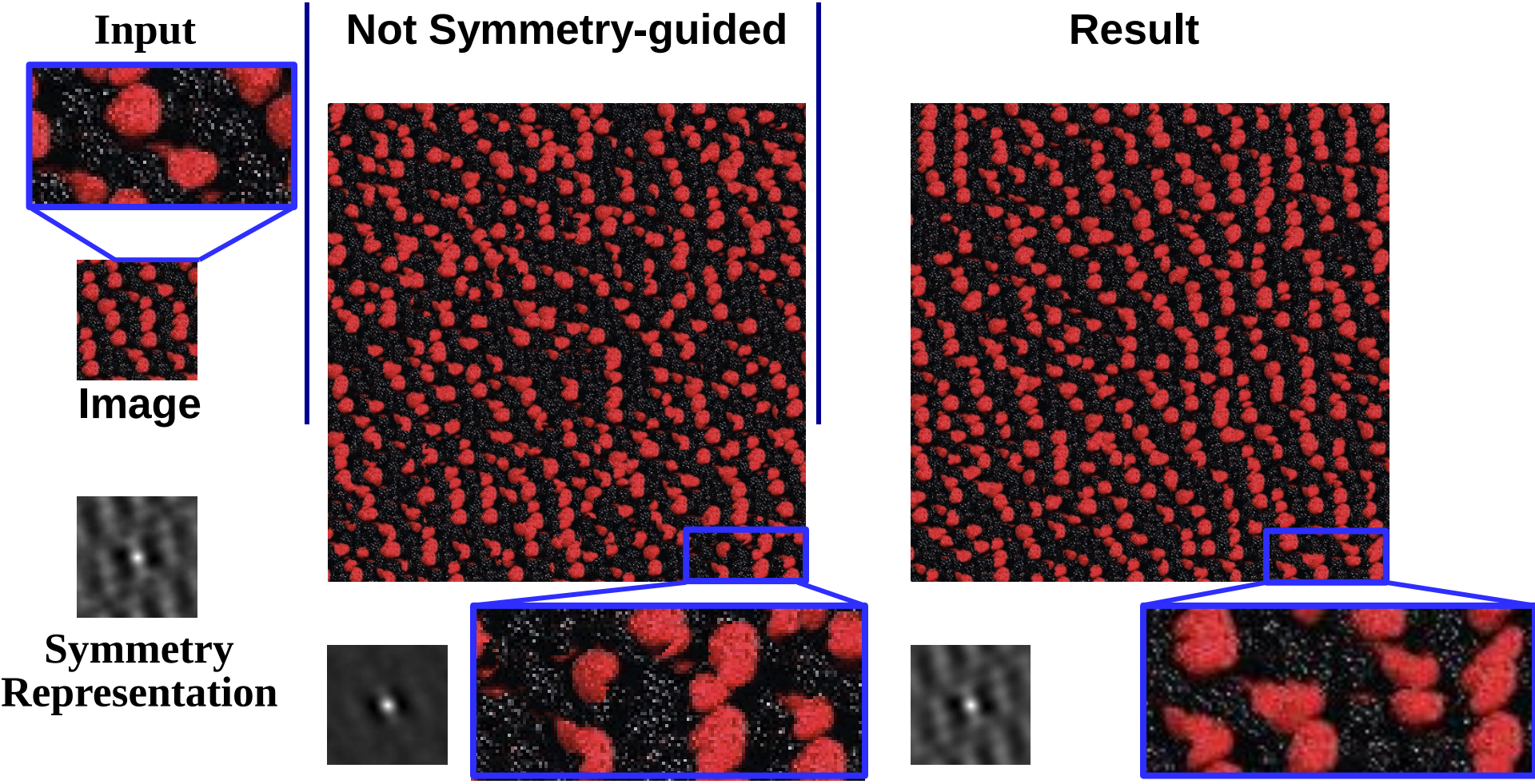
Pattern Processing: Increase Contrast



Pattern Processing: Identity

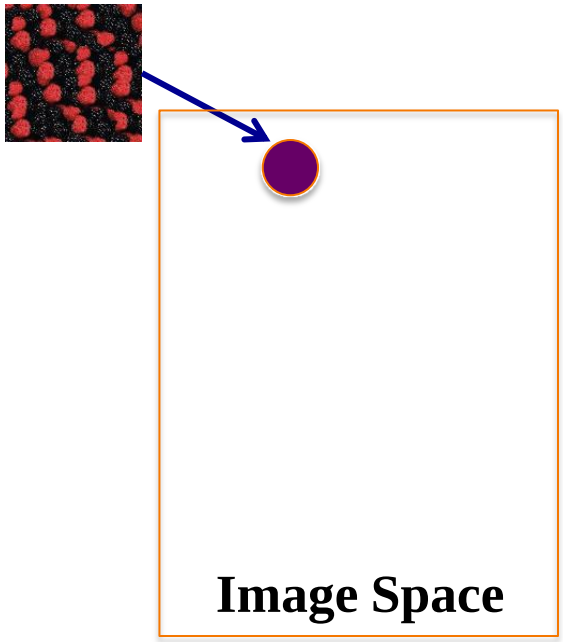


Pattern Processing: Identity



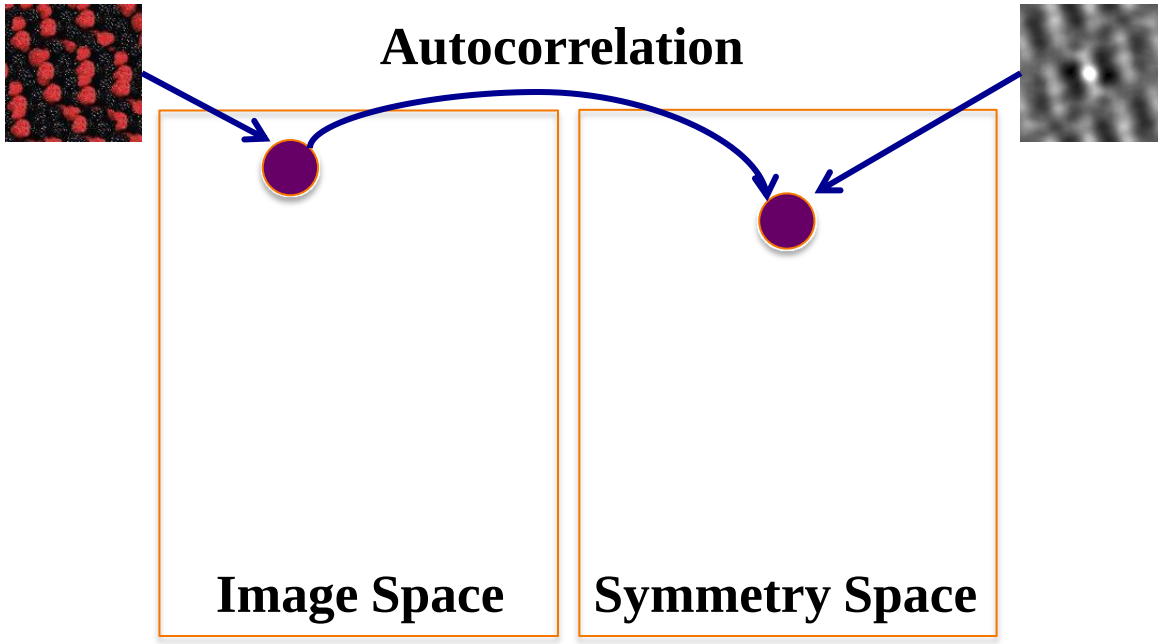
Optimization Example

Represent large-scale patterns with approximate symmetry representations



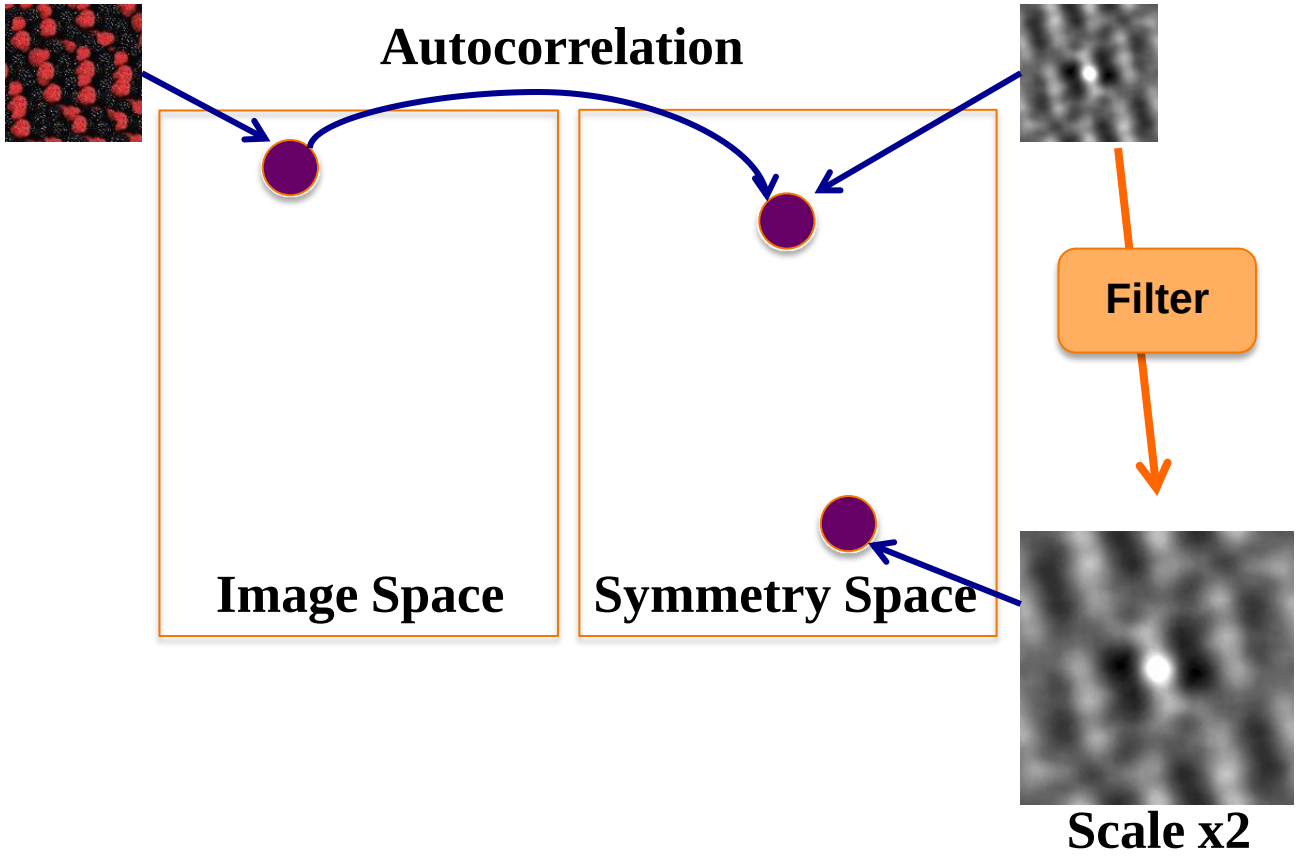
Optimization Example

Represent large-scale patterns with approximate symmetry representations



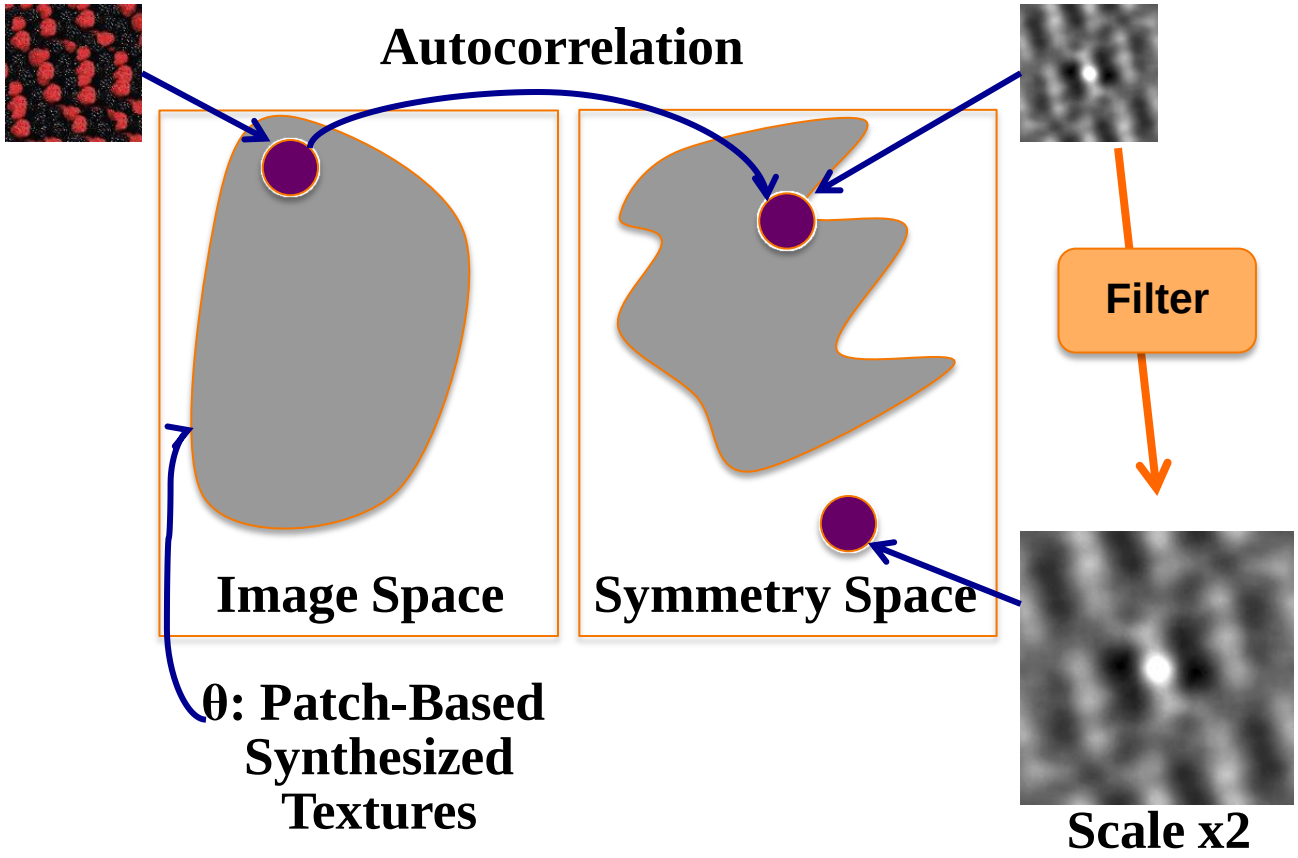
Optimization Example

Represent large-scale patterns with approximate symmetry representations



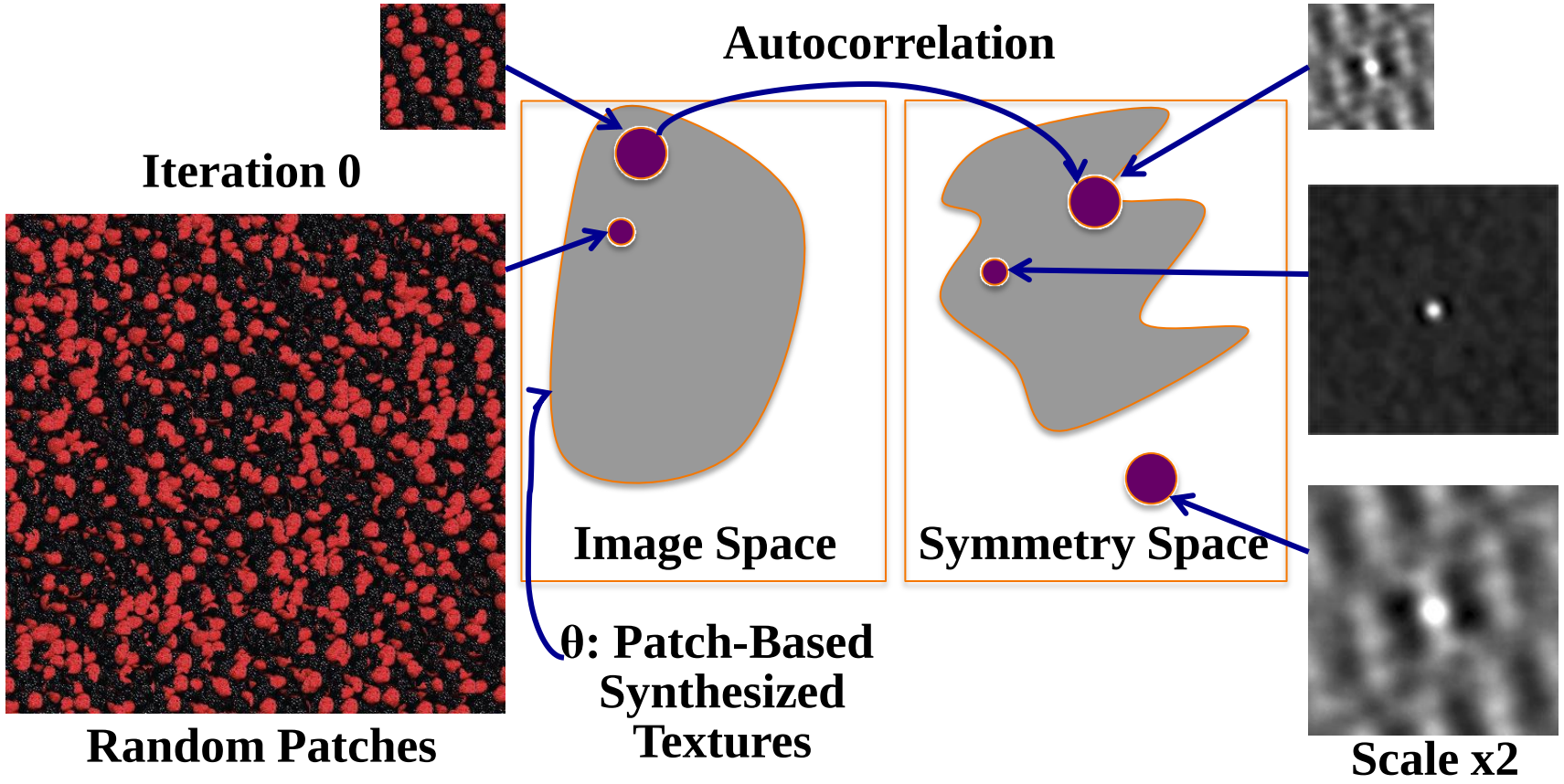
Optimization Example

Represent large-scale patterns with approximate symmetry representations



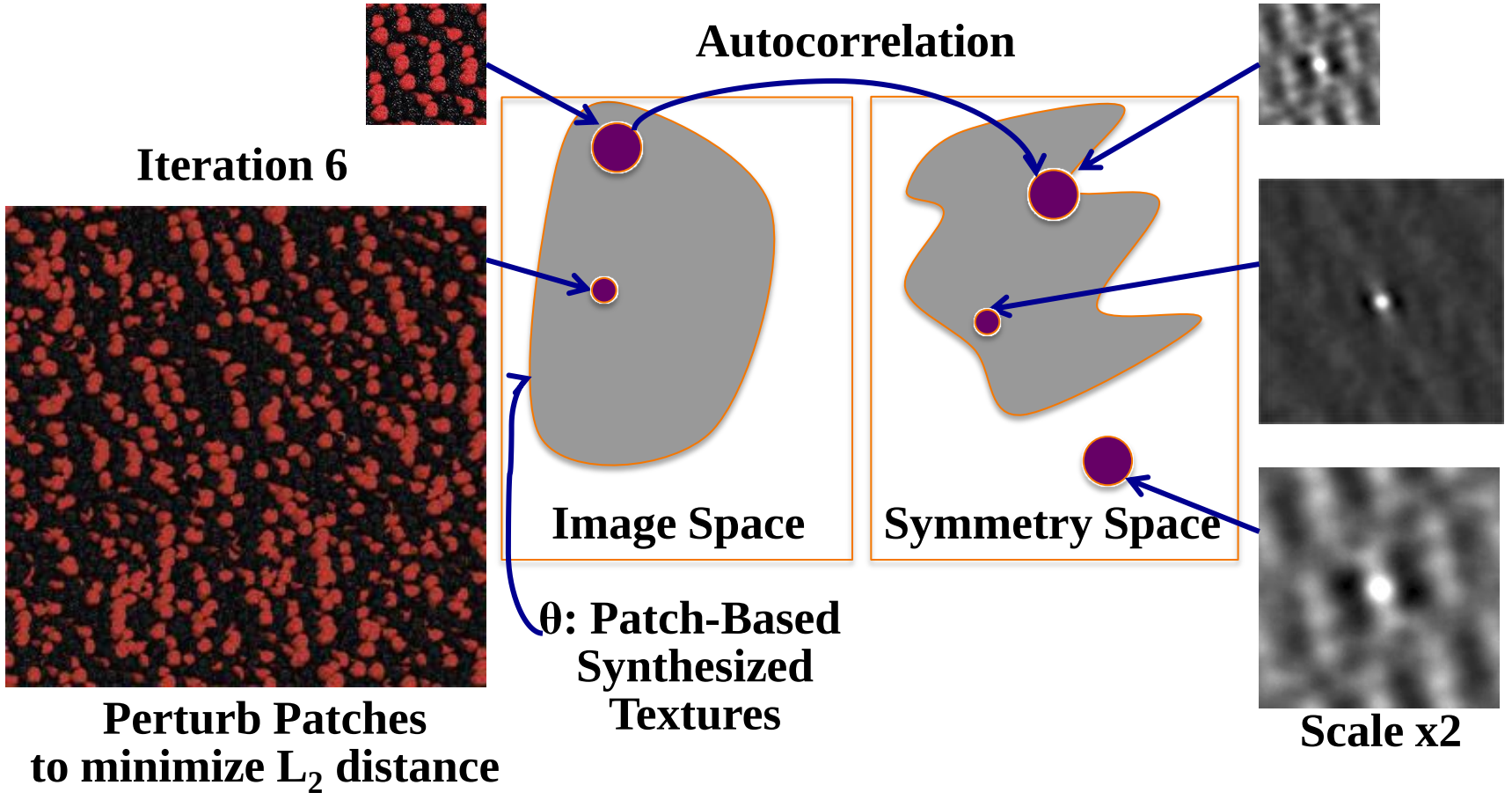
Optimization Example

Represent large-scale patterns with approximate symmetry representations



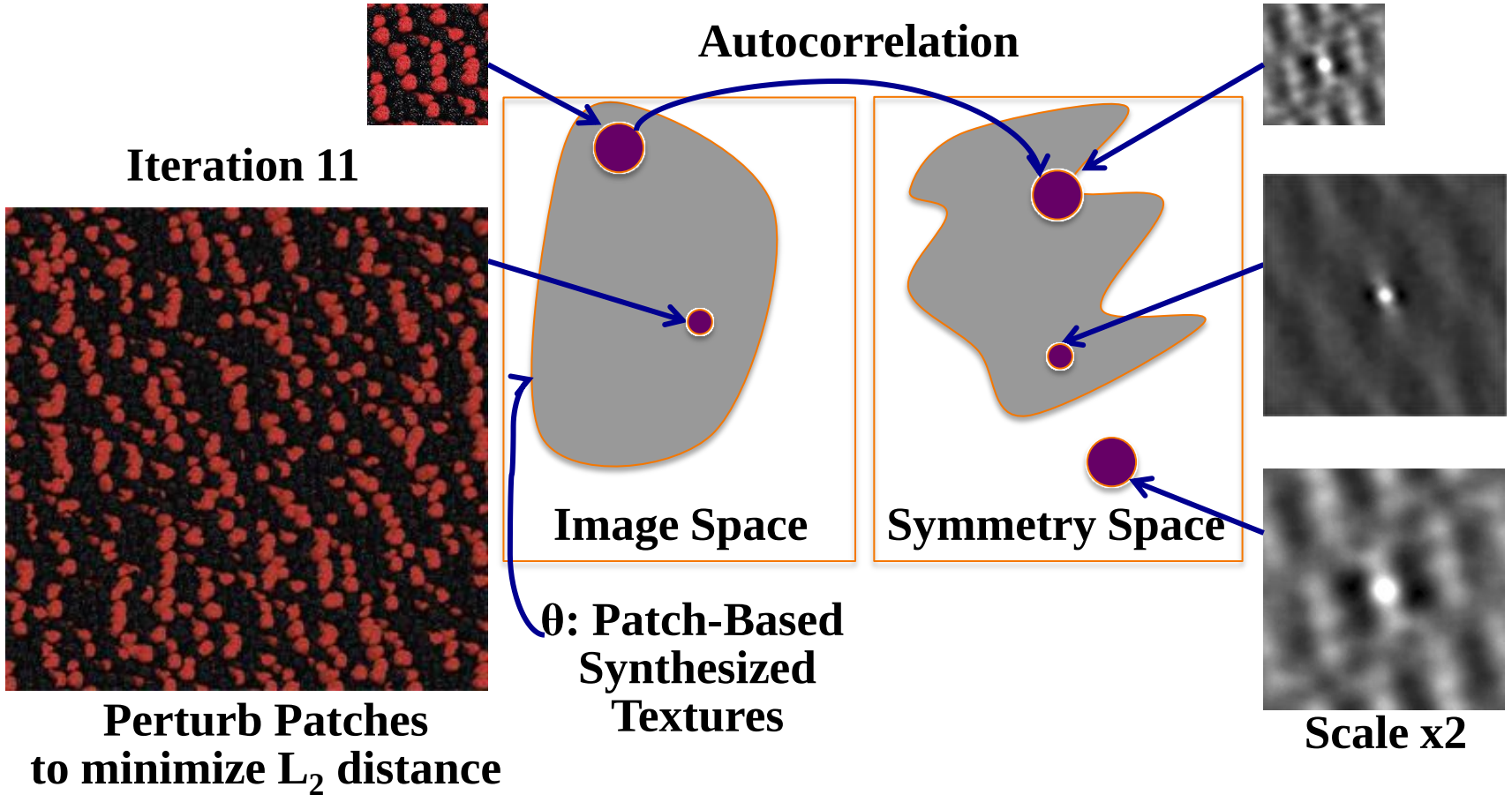
Optimization Example

Represent large-scale patterns with approximate symmetry representations



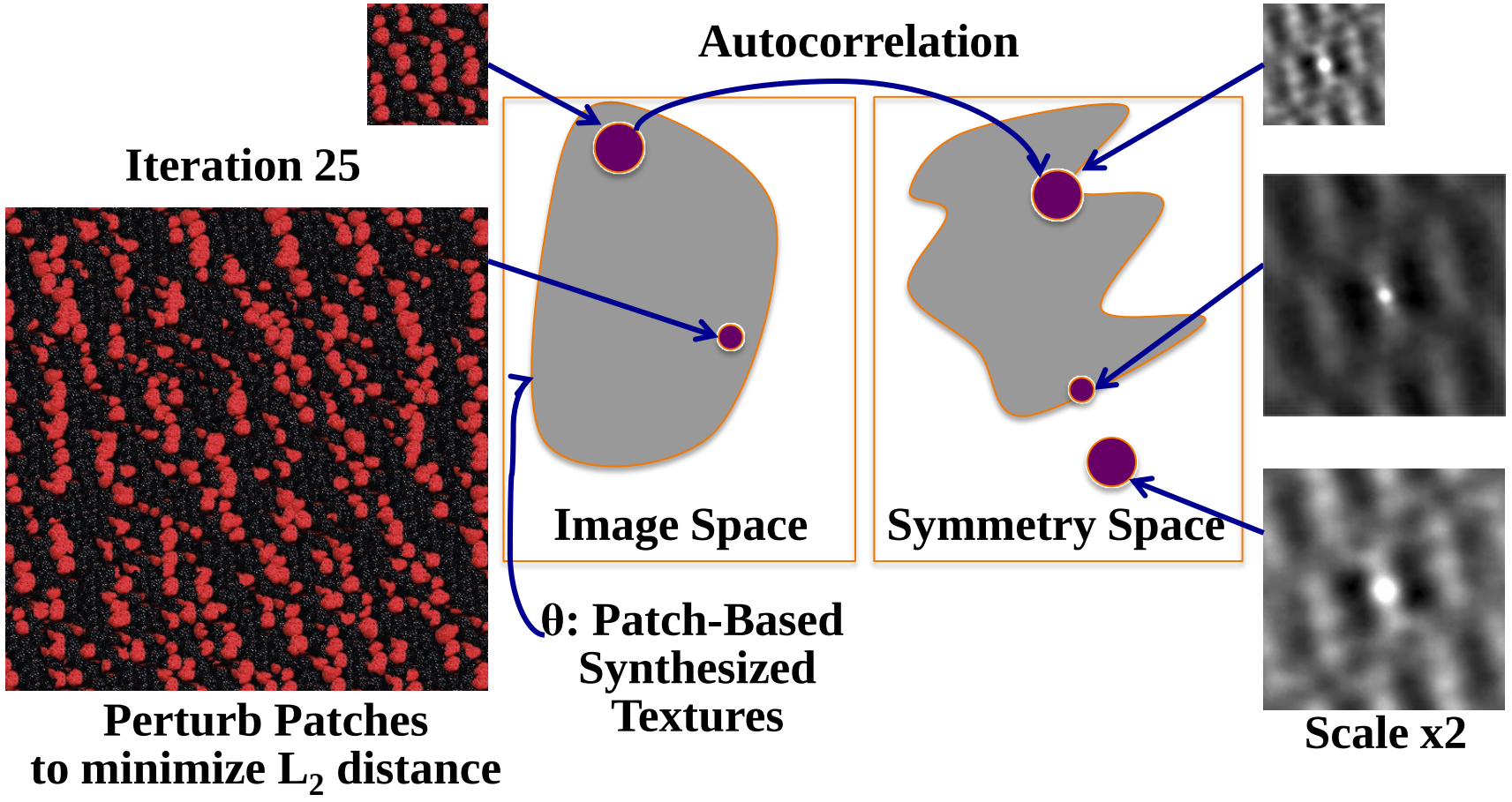
Optimization Example

Represent large-scale patterns with approximate symmetry representations

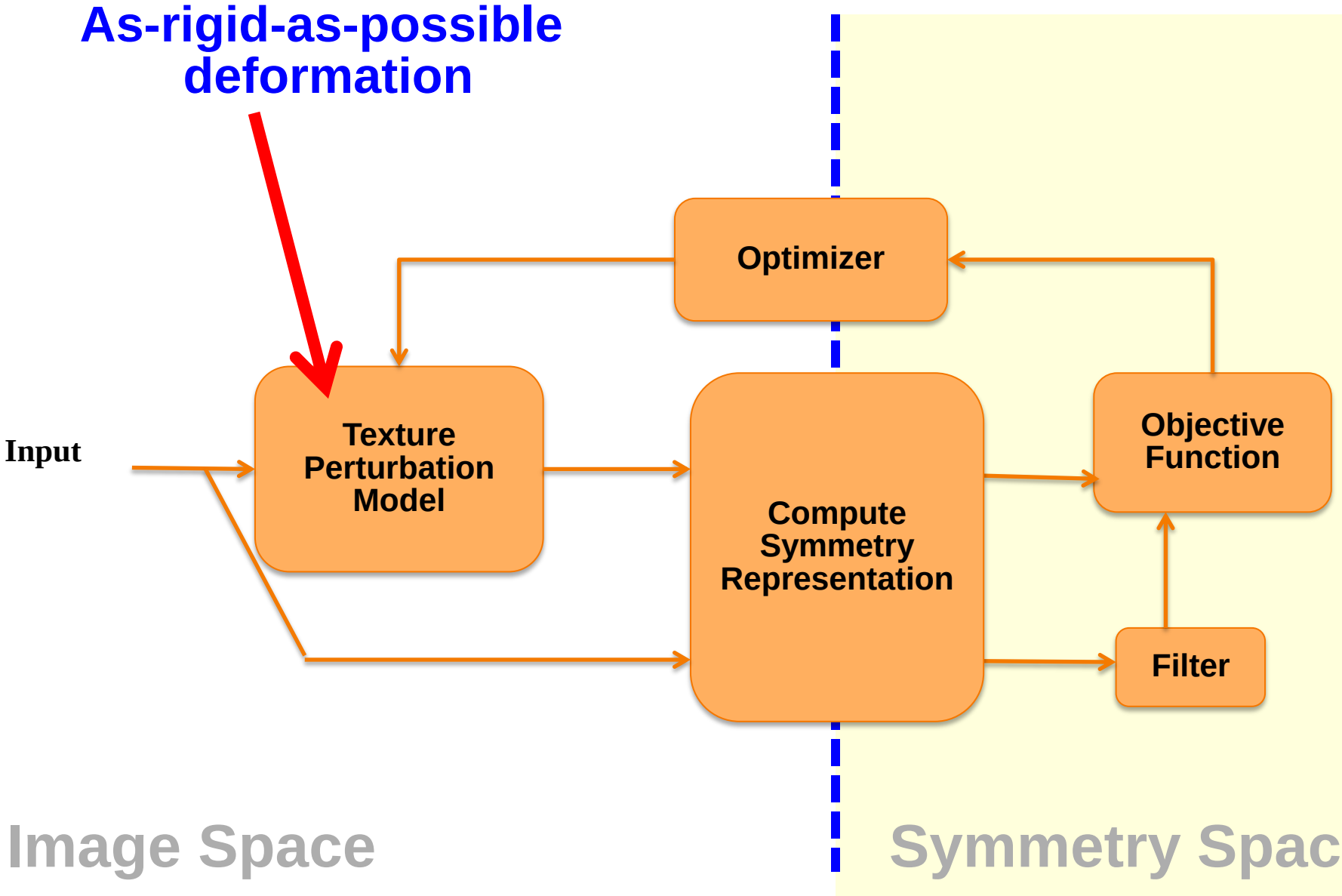


Optimization Example

Represent large-scale patterns with approximate symmetry representations



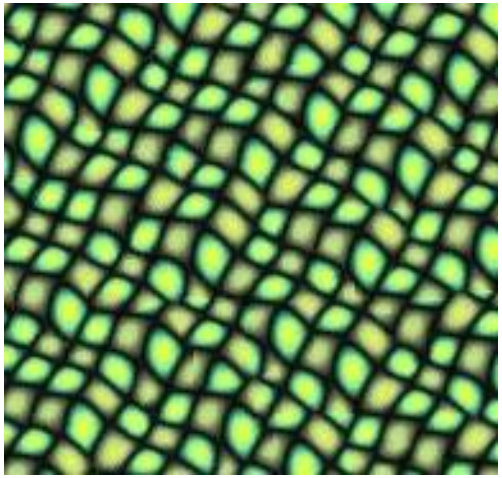
Pattern Processing: Warping Texture

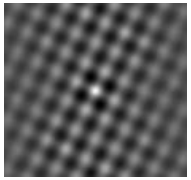


Pattern Processing: Sharpen

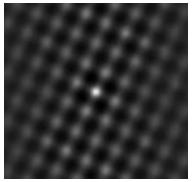
Input

Filter





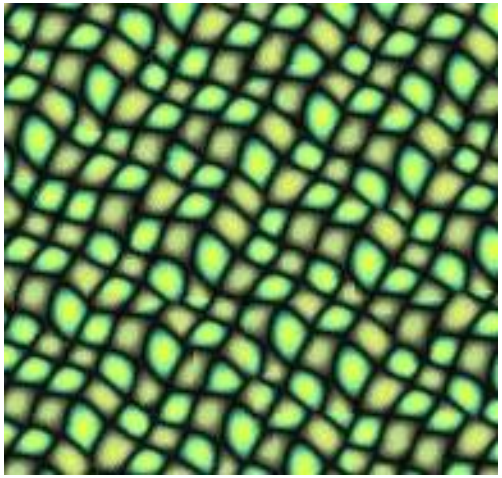
Symmetry Representation



Sharpen

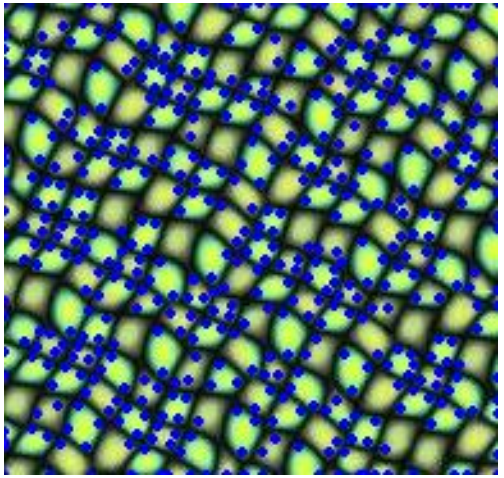
Pattern Processing: Sharpen

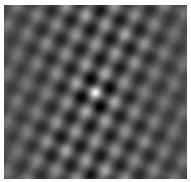
Input



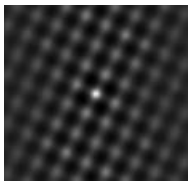
Filter

Texture Perturbation

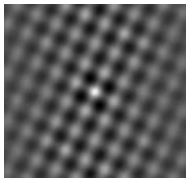




Symmetry Representation



Sharpen

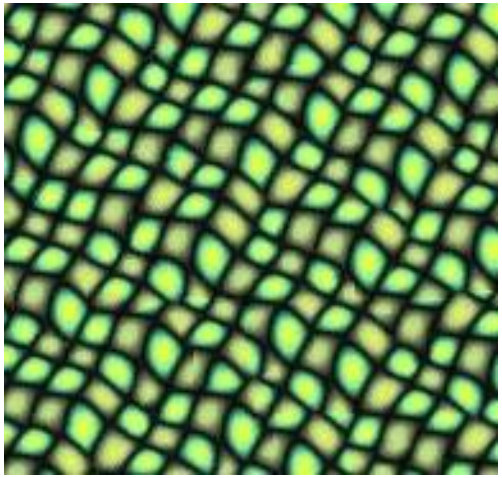


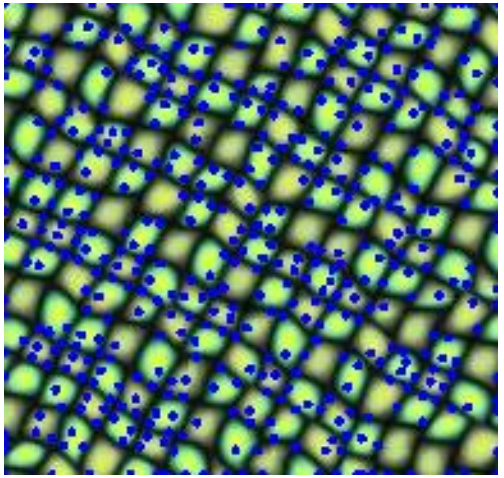
Pattern Processing: Sharpen

Input

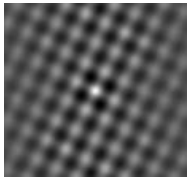
Filter

Texture Perturbation

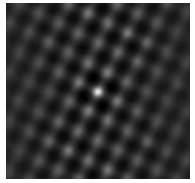




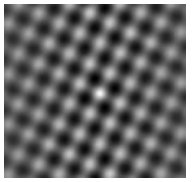
Iteration 2



Symmetry Representation



Sharpen

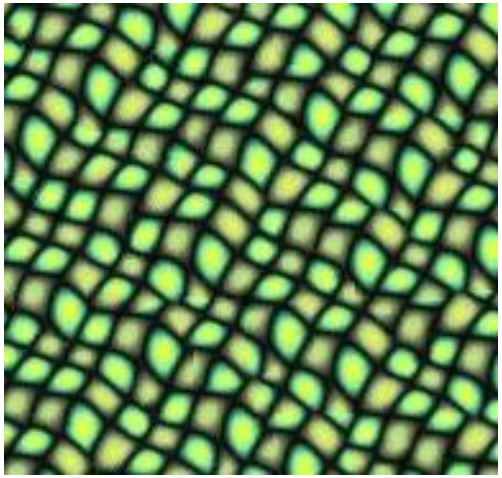


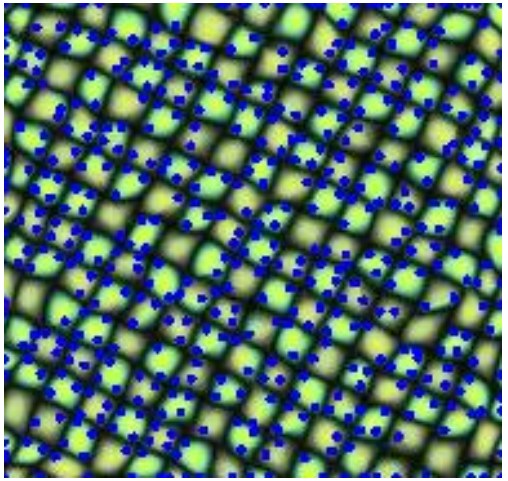
Pattern Processing: Sharpen

Input

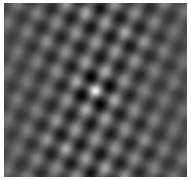
Filter

Texture Perturbation

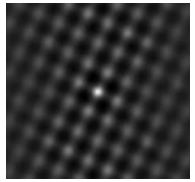




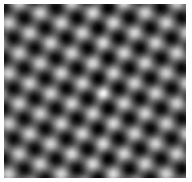
Iteration 5



Symmetry Representation



Sharpen

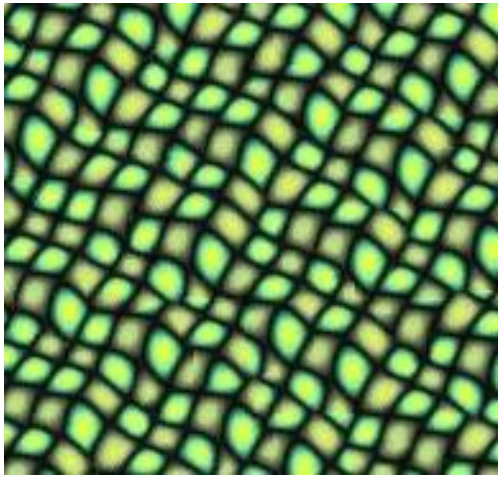


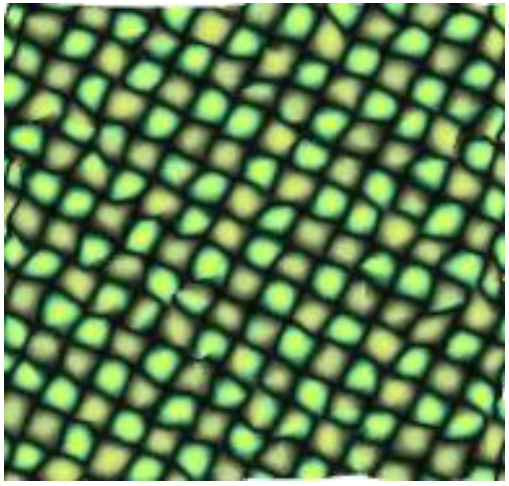
Pattern Processing: Sharpen

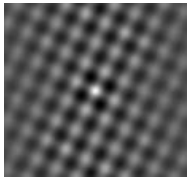
Input

Filter

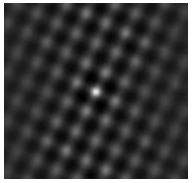
Result



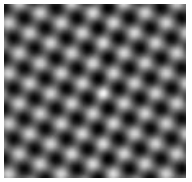




Symmetry Representation



Sharpen

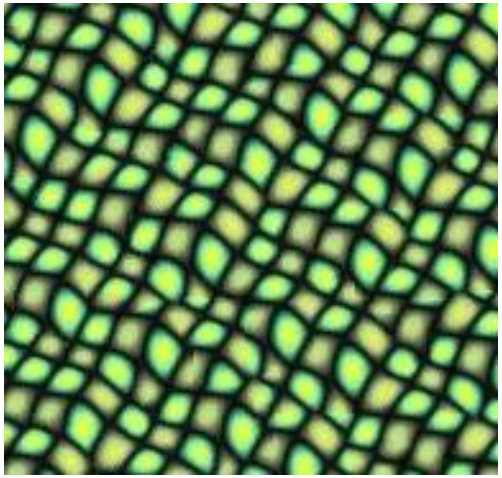


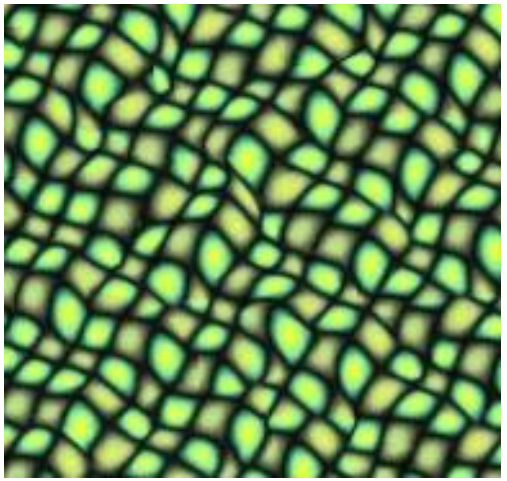
Pattern Processing: Blur G(2)

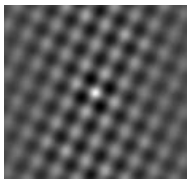
Input

Filter

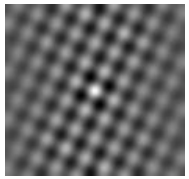
Result



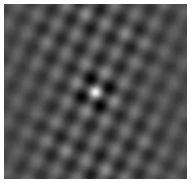




Symmetry Representation



Blur G(2)

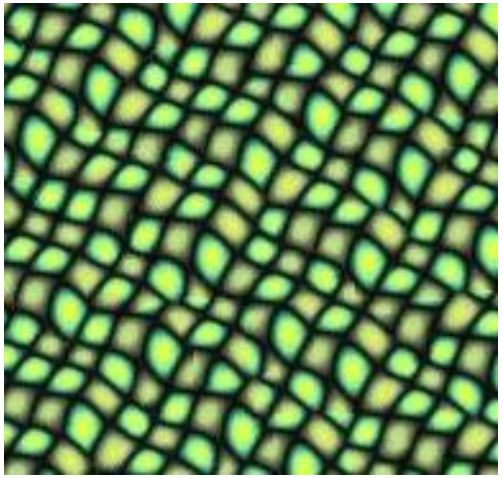


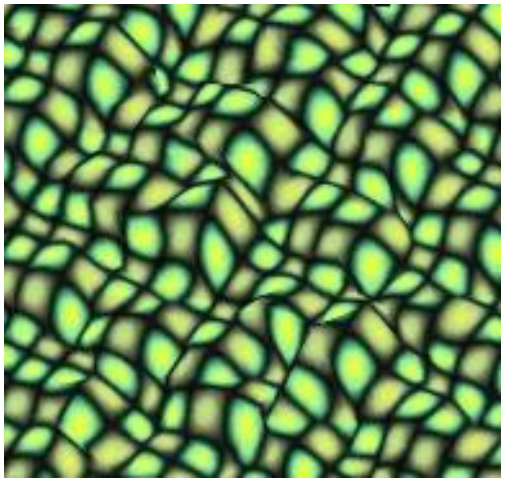
Pattern Processing: Blur G(4)

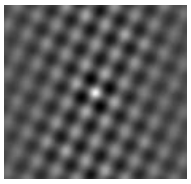
Input

Filter

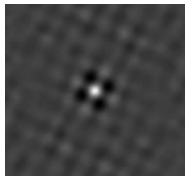
Result



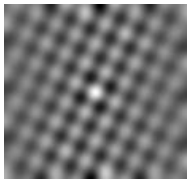




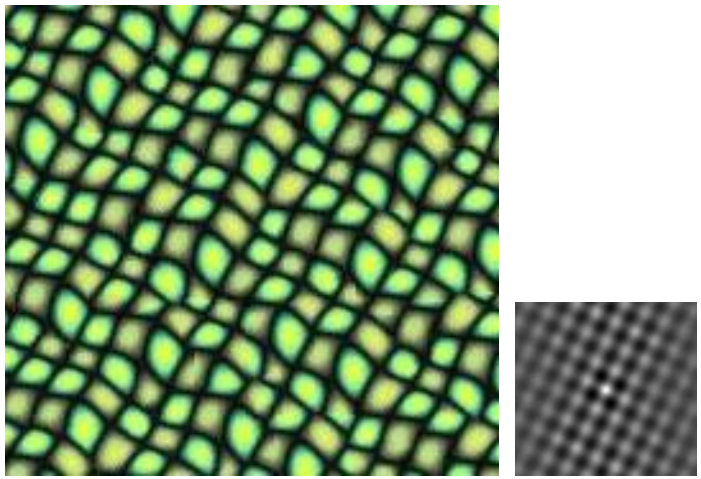
Symmetry Representation



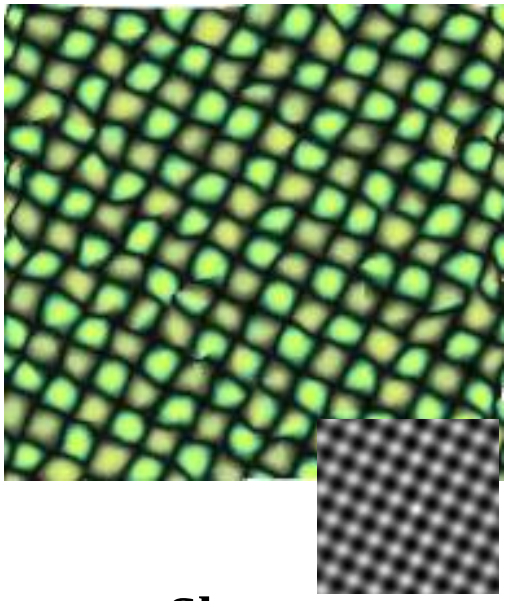
Blur G(4)



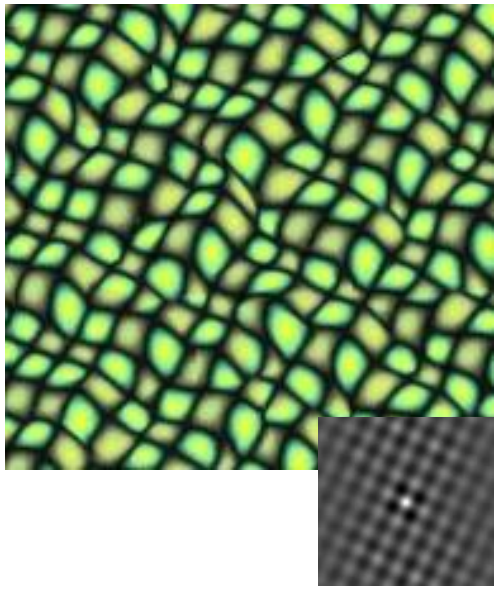
Pattern Processing: Sharpen / Blur



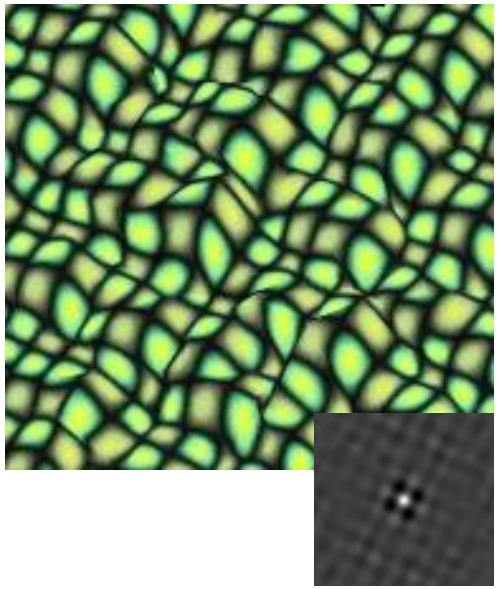
Input



Sharpen



Blur G(2)



Blur G(4)

Applications

Pattern Processing

Pattern Transfer

Pattern Optimization

Pattern Transfer: Goal

Transfer pattern of texture *g* to texture *f*

© shallowend24401

Texture: f

Pattern: g

Result: f’

Pattern Transfer

Input

Optimizer

Texture Perturbation Model

Compute Symmetry Representation

Objective Function

Target

Image Space

Symmetry Space

Pattern Transfer From a Sketch

Target Pattern

Source Texture:

© shallowend24401

© euart

Result

Result

Pattern Transfer From a Sketch

Target Pattern

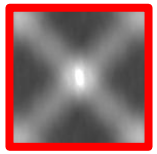
Source Texture:

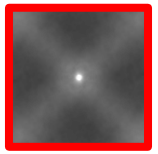
© shallowend24401

© euart

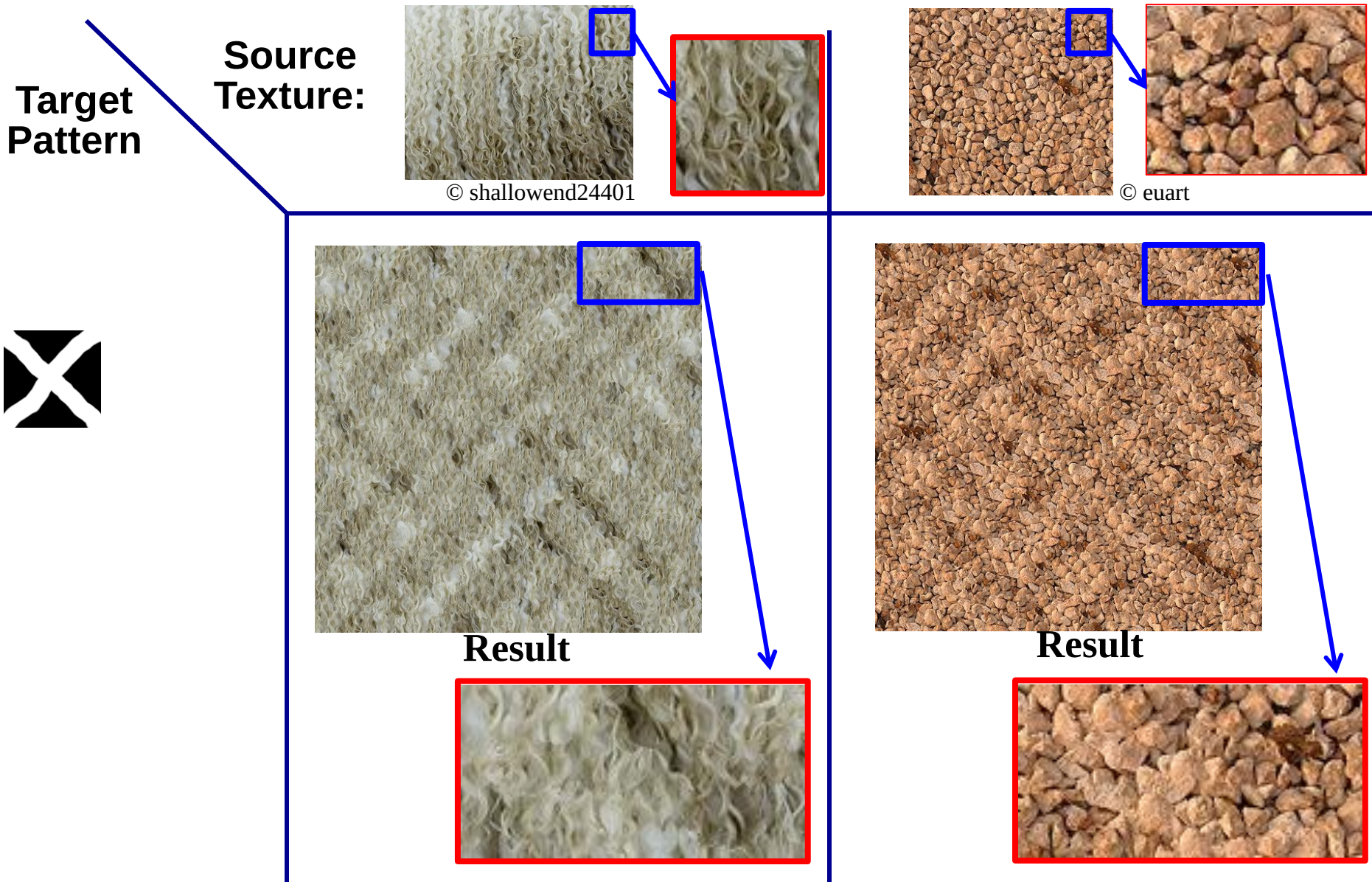
Result

Result





Pattern Transfer From a Sketch



Pattern Transfer From an Image

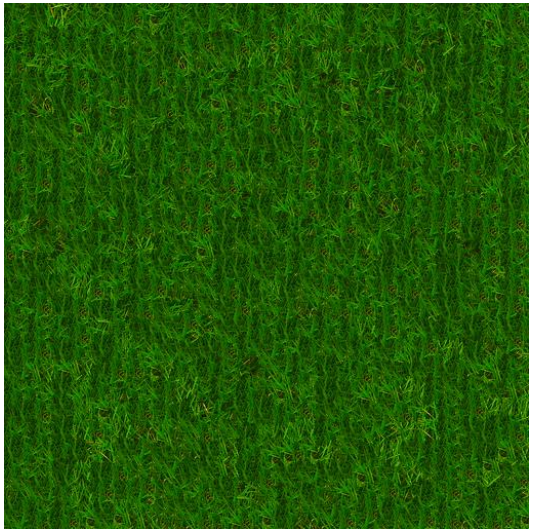
Target Pattern

Source Texture:

© shallowend24401

© 100kr

Result



Result

Pattern Transfer From an Image

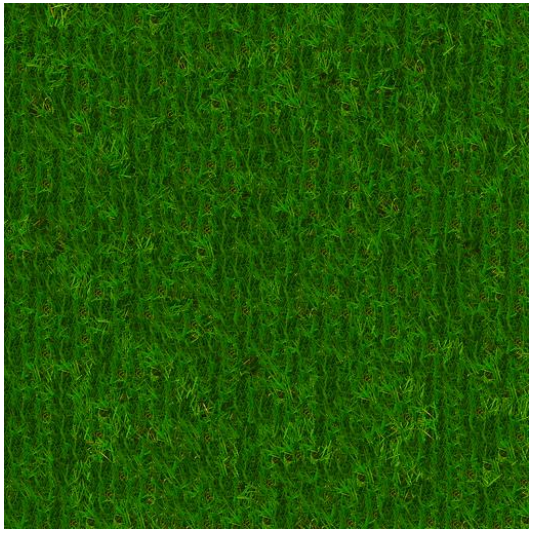
Target Pattern

Source Texture:

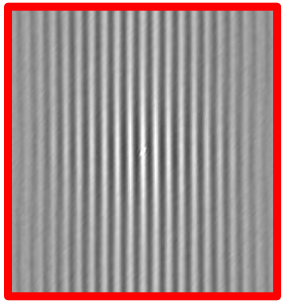
© shallowend24401

© 100kr

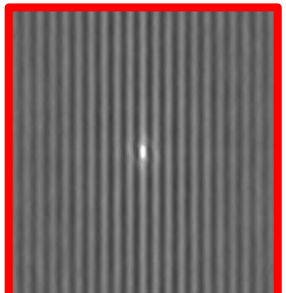
Result

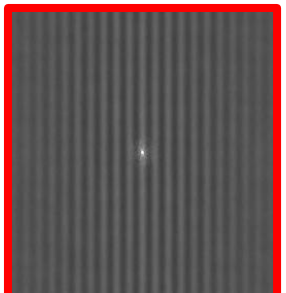


Result



Symmetry





Pattern Transfer From an Image

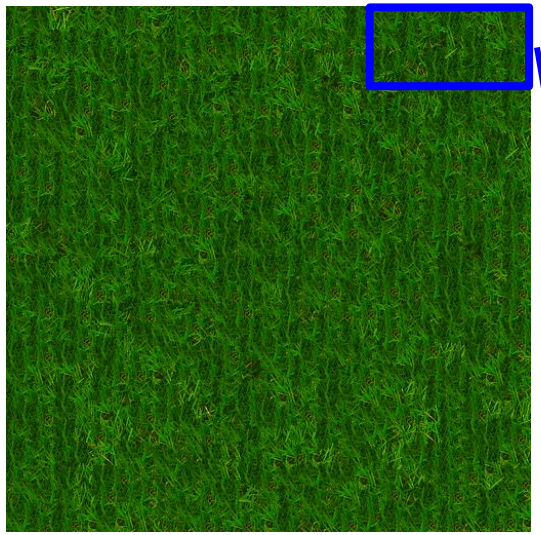
Target Pattern

Source Texture:

© shallowend24401

© 100kr

Result



Result

Applications

Pattern Processing

Pattern Transfer

Pattern Optimization

Pattern Optimization: Goal

Optimize function defined over symmetry representation, e.g.

- Symmetrize

Input

Symmetrized

Pattern Optimization

Input

Texture Perturbation Model

Compute Symmetry Representation

Objective Function

Optimizer

Maximize standard Deviation of symmetry representation

Image Space

Symmetry Space

Pattern Optimization: Tileable Texture

Blend edges + tile

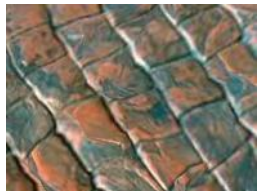
Pattern Optimization: Tileable Texture

Symmetrize

Blend edges + tile

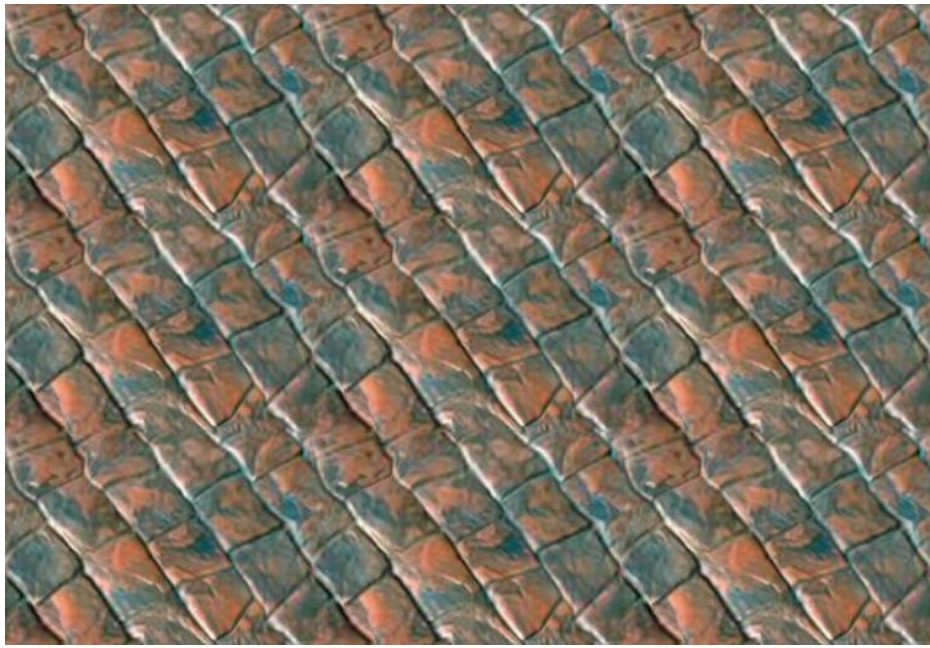
Pattern Optimization: Tileable Texture

Symmetrize



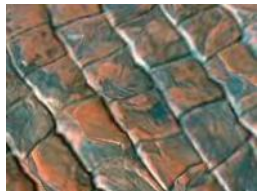
Blend edges + tile

Blend edges + tile



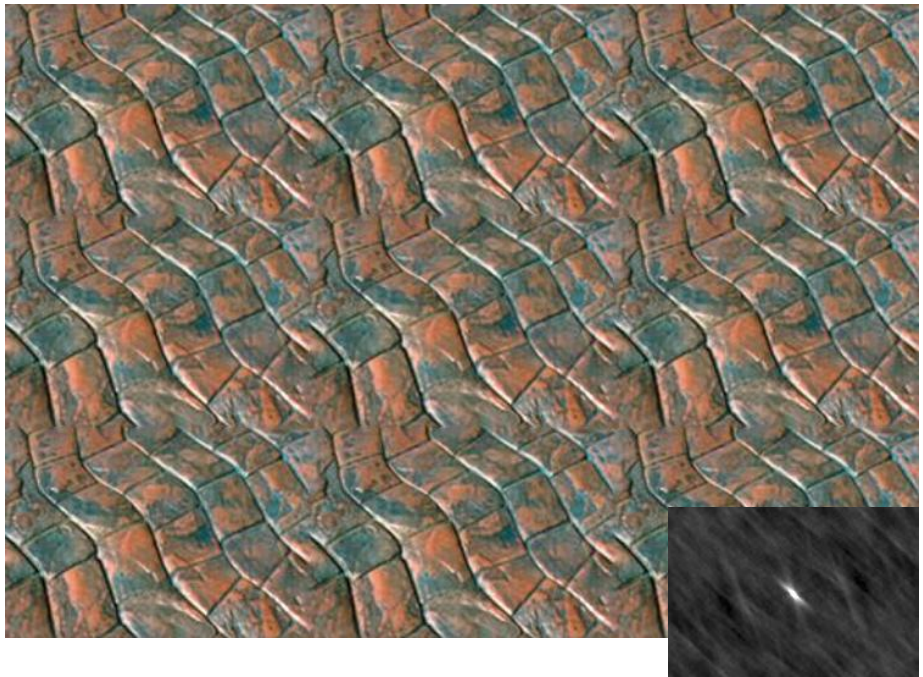
Pattern Optimization: Tileable Texture

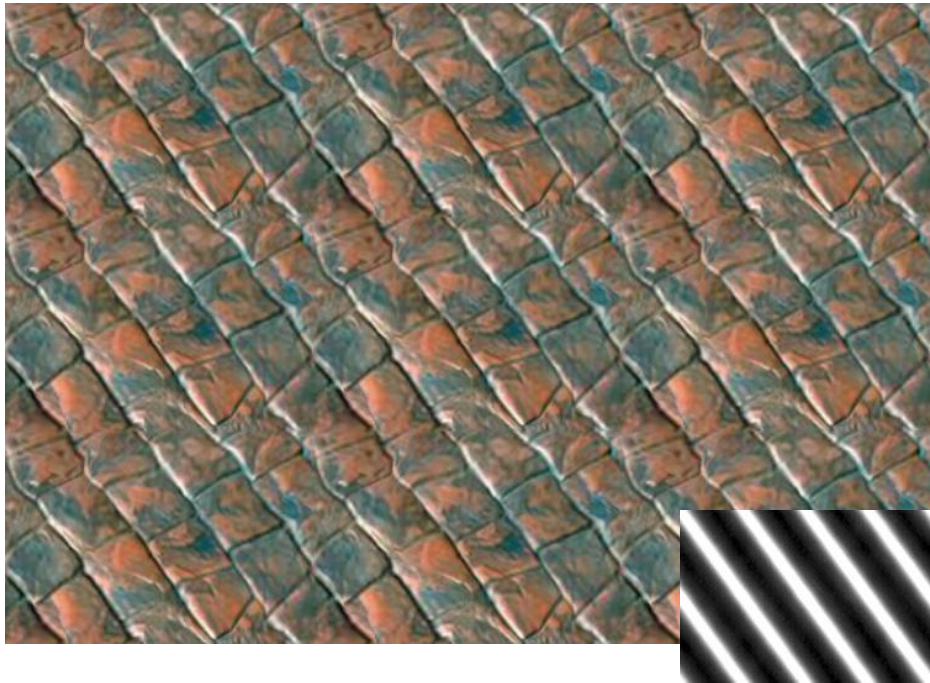
Symmetrize



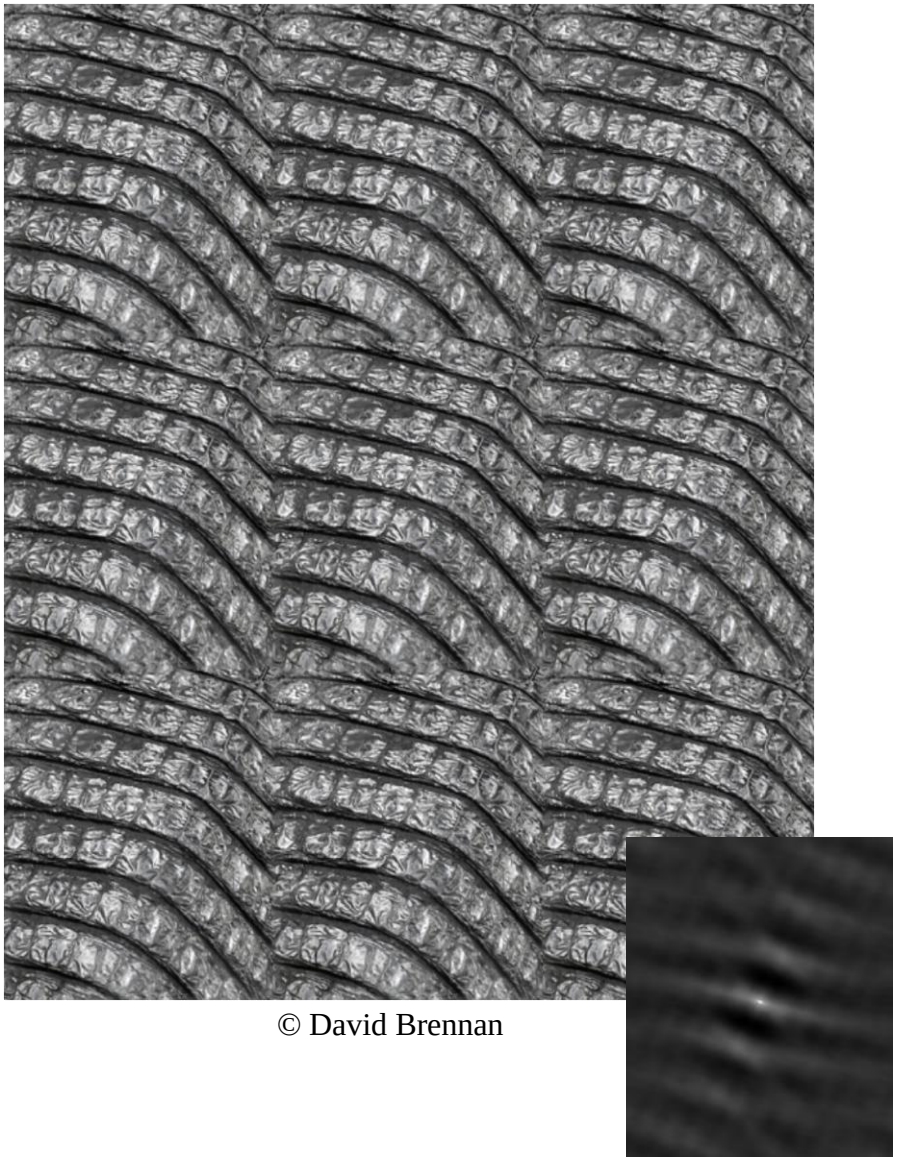
Blend edges + tile

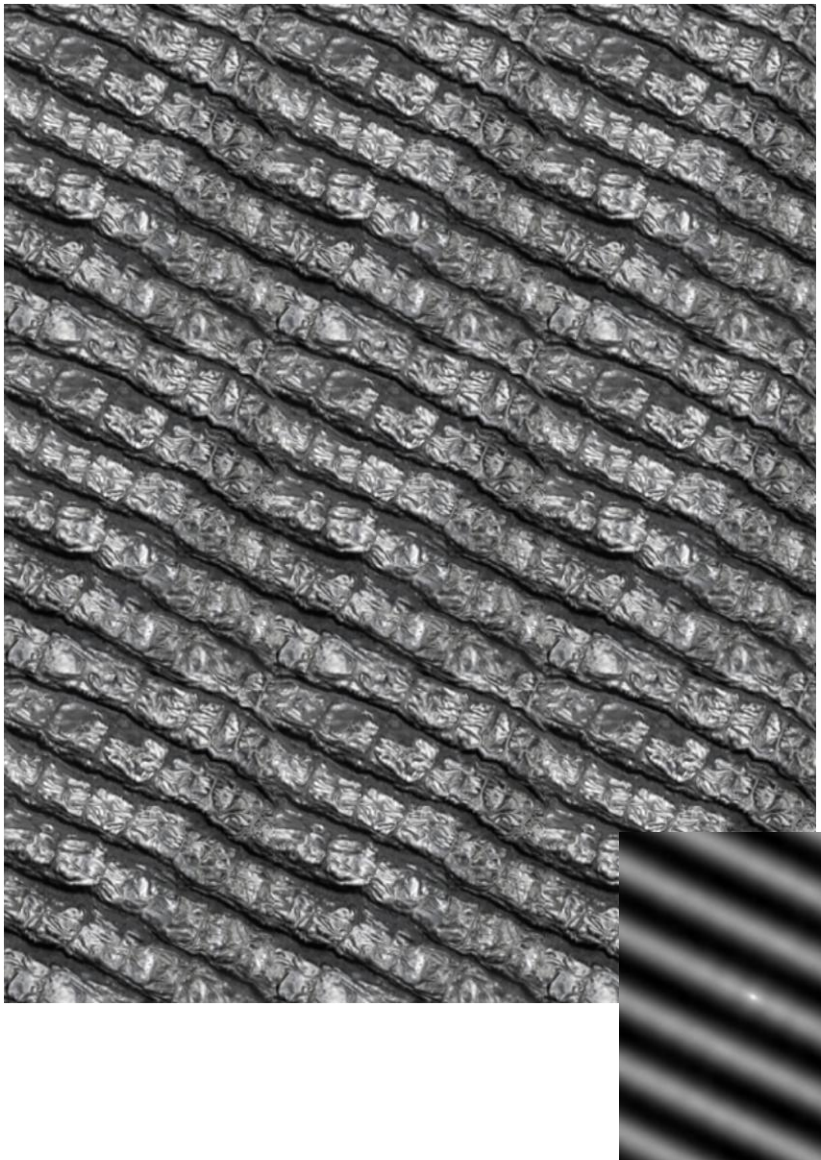
Blend edges + tile



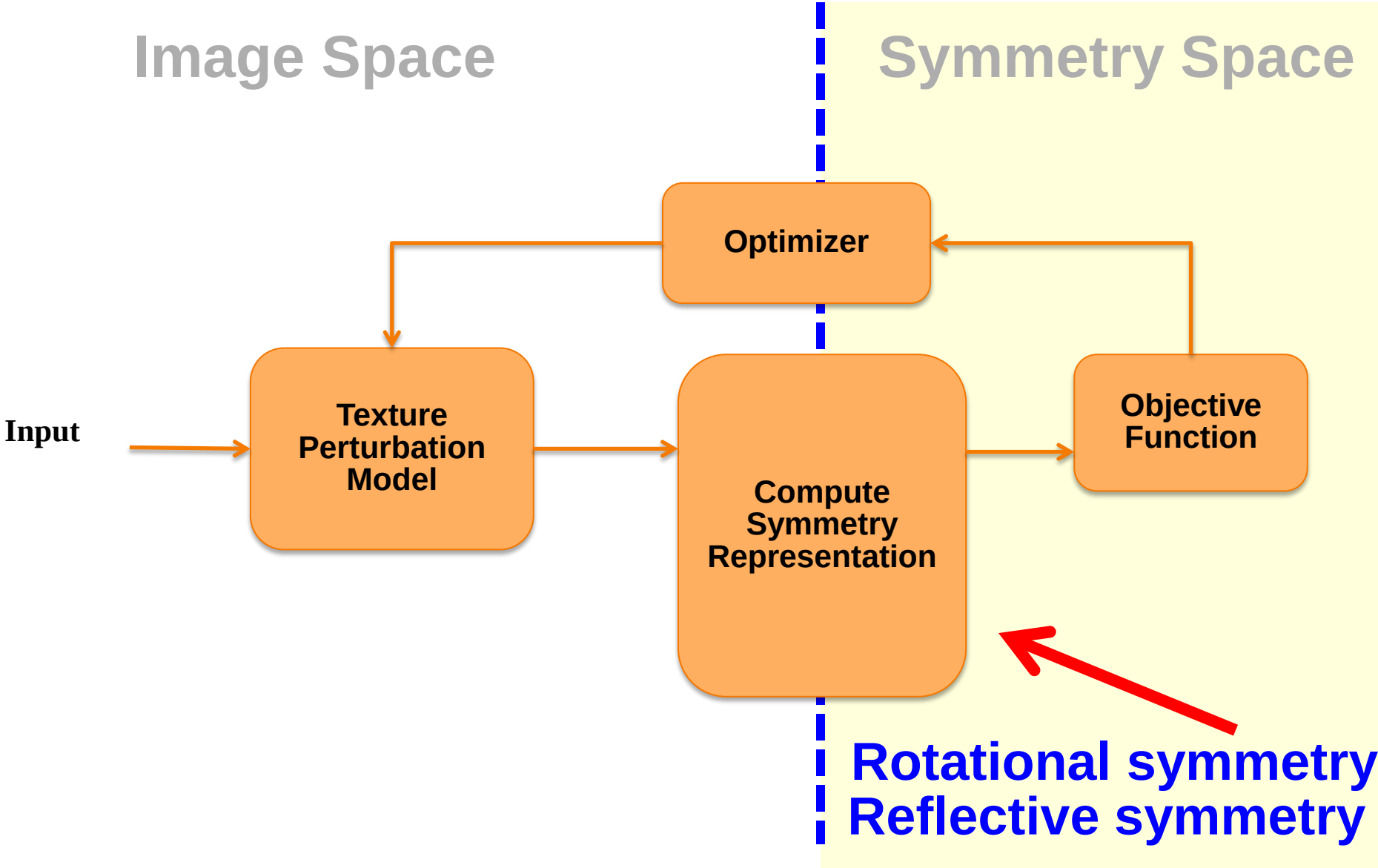


Pattern Optimization: Tileable Texture





Pattern Optimization



Pattern Optimization: Rotational

Input Image

Pattern Optimization: Rotational

Symmetrized

Pattern Optimization: Reflectional

Input Image

Pattern Optimization: Reflectional

Symmetrized

Timing

Most of the time is spent on iterative computation of a symmetry transform

Total Time ≈ 1hr

Total Time ≈ 30min

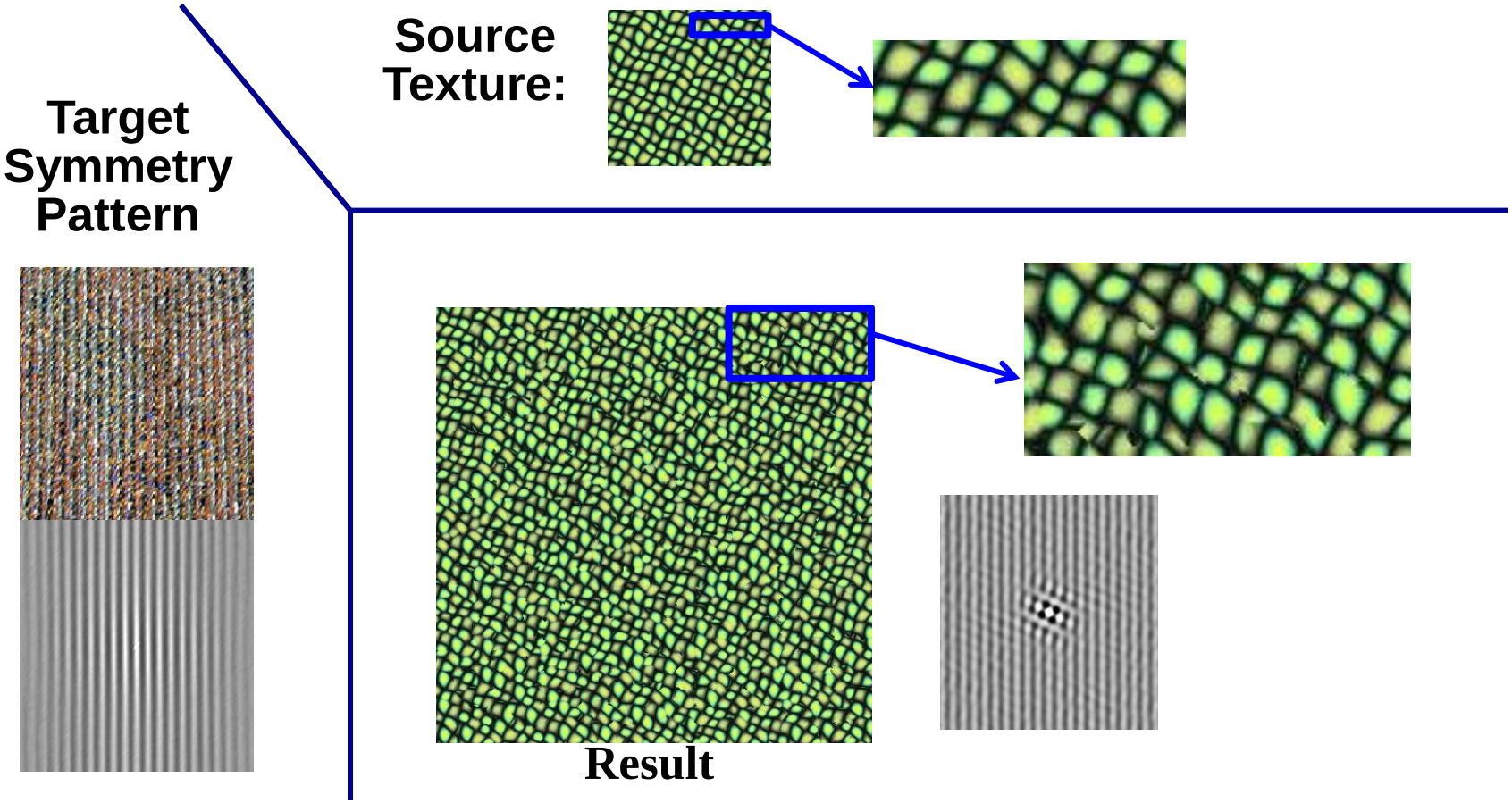
Summary

General framework for operating on textures in symmetry space

Investigated applications: pattern transfer, pattern filtering and pattern optimization

Limitations

Must be gap in size between fine-scale details and large-scale pattern



Future work

Other applications

- Symmetry-guided manipulation of 3D geometries

Acknowledgements + Project Website

Code and Data:

- Flickr: shallowend, dolescum, C. Lewis (cloois), 100kr, snappa2006, cwazymandy, D. Brennan (davidbrennan), J. Perivolaris (dr_john2005), M. Heigan (martin_heigan)
- CMU NRT database
- Code (for comparison): J. Hays, M. Park, Y. Liu

Funding:

- Google, Intel, Adobe, NSERC, NSF

Project Website (code, data, and examples):

http://www.cs.princeton.edu/~vk/SymmetryTexture

Thank You!