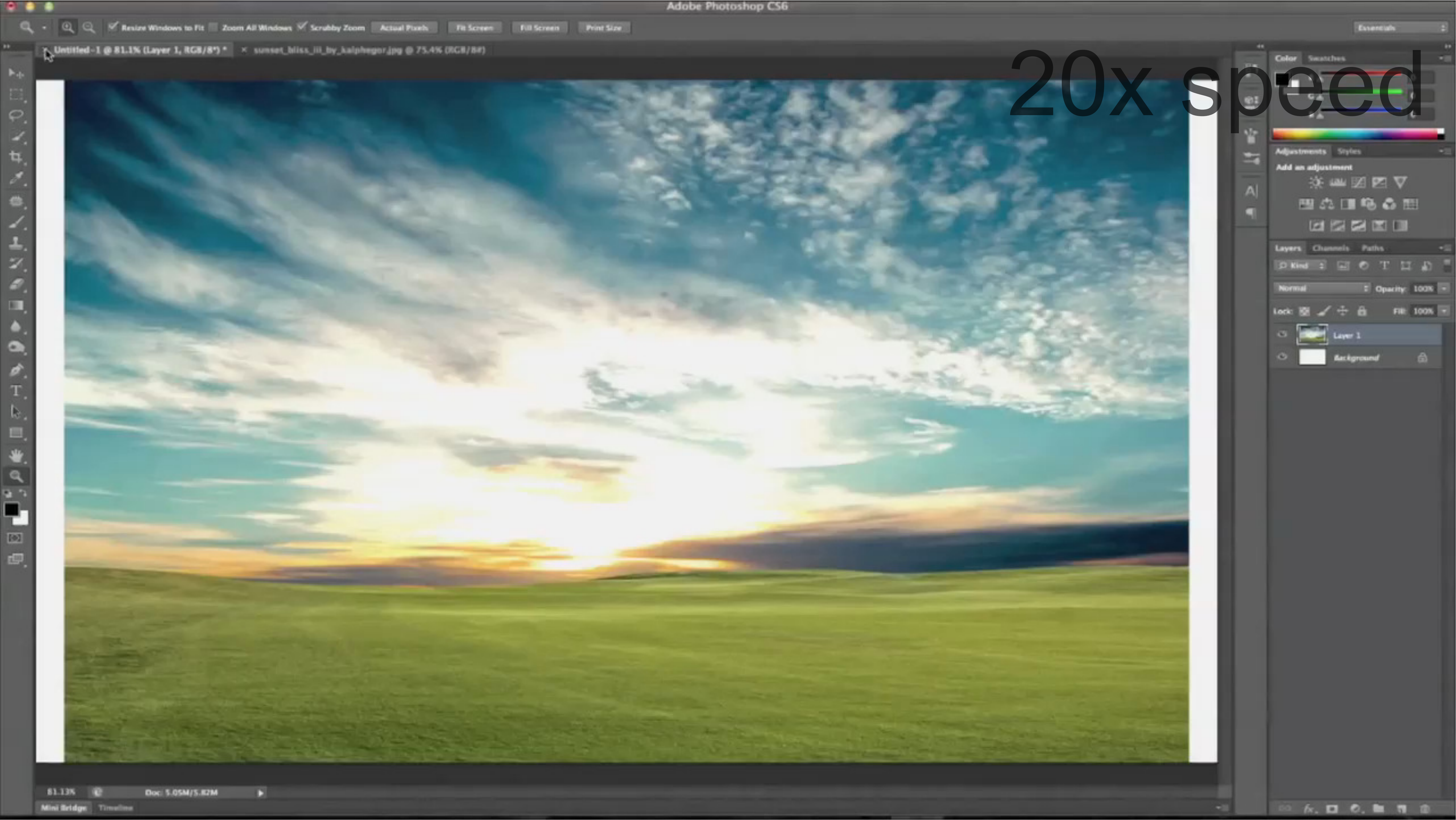
Mid-level Image Editing with PatchMatch

Connelly Barnes

University of Virginia

Painting by Rado Javor



☑ Resize Windows to Fit □ Zoom All Windows ☑ Scrubby Zoom

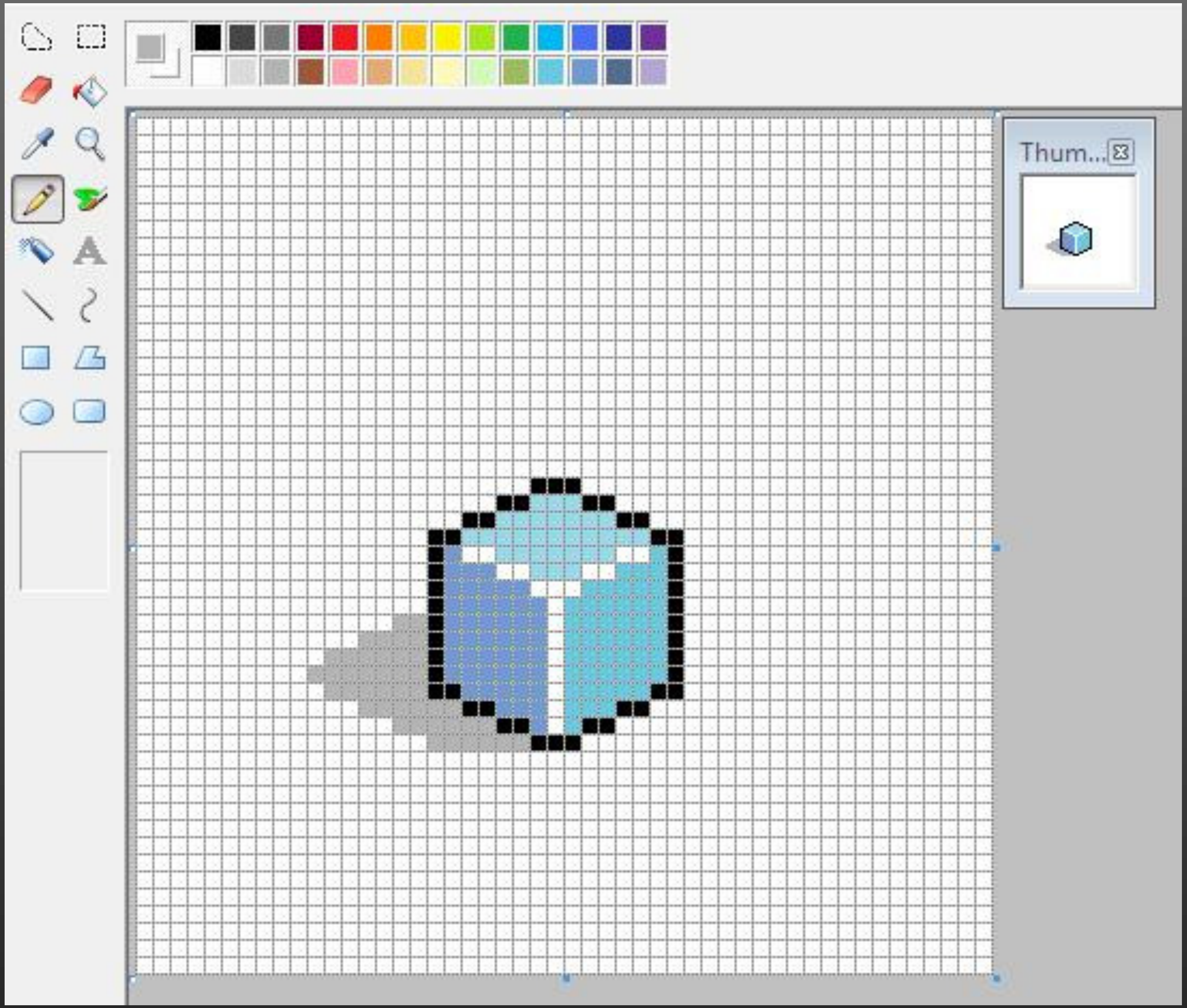
Normal Opacity: 100%

Lock: Fill: 100%

81.13% Doc: 5.05M/5.82M

THE WILD
LUCAS RODRIGUEZ

Why is Content Creation Hard?

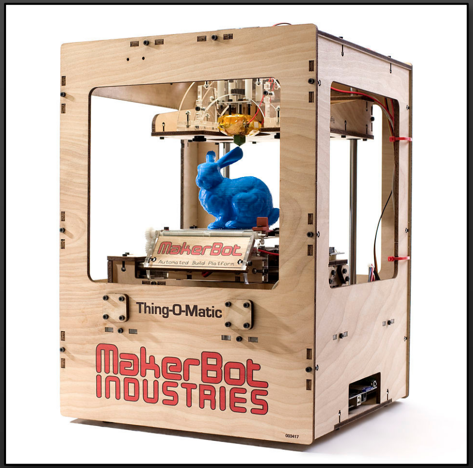


Why Does it Matter?

Movies

Games

Graphic Design



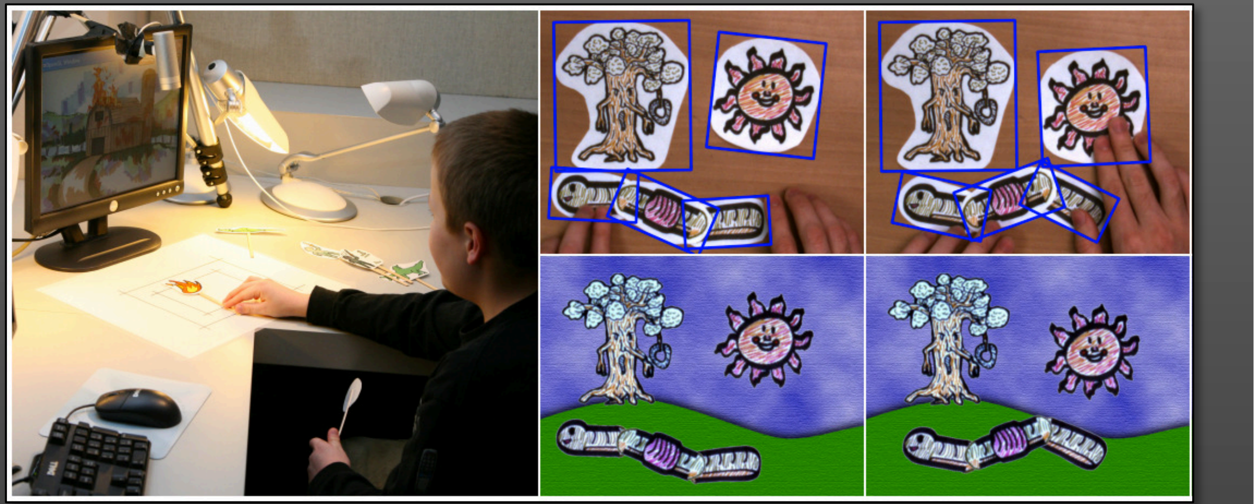
Fabrication

Research Goals

- Easier for everyone to create content
- Smarter tools with an understanding of images
- Strategies:
  - Exploit statistics of the natural world
  - Mid level (patch) image manipulation

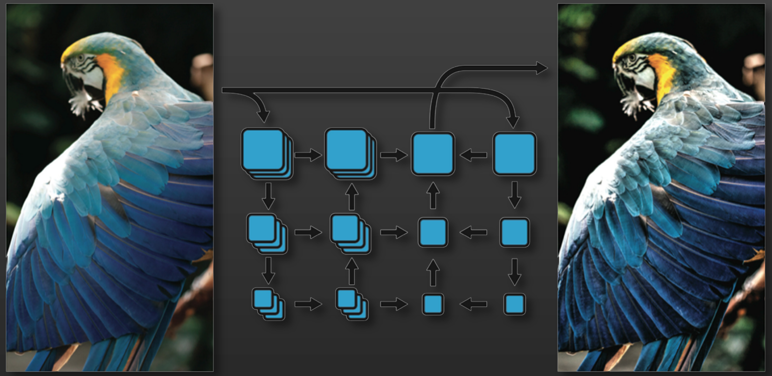
Other Research Projects

[Digital Bas-Relief from 3D Scenes, SIGGRAPH '07]



[Video Puppetry, A Performative… SIGGRAPH Asia '08]

[RealBrush, under review, Jingwan Lu at Princeton]



Tuned C++: 262 lines
Halide: 52 lines

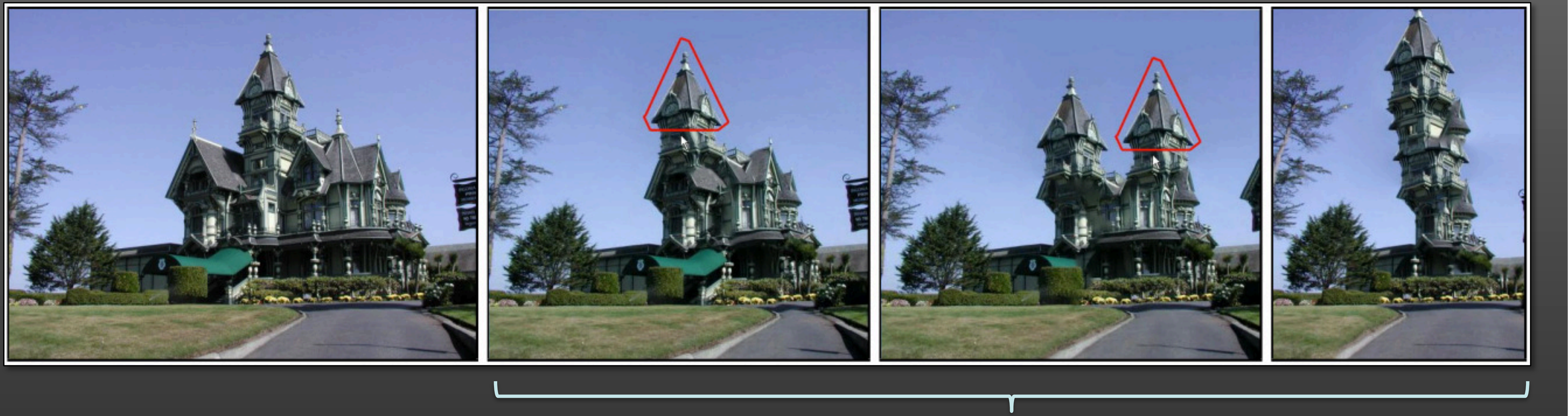
Expert, quad-core x86: 157 ms
Halide, quad-core x86: 86 ms

Halide, GPU: 21 ms

5x shorter, **2-7x** faster

[Optimizing Parallelism, Locality, …, ACM PLDI '13]

This Talk: PatchMatch

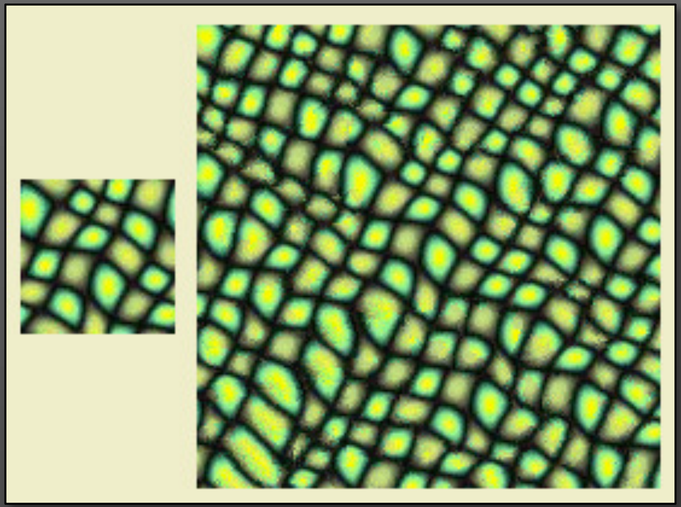


Input Photograph

Synthesized Output with User Constraints

[Barnes et al, PatchMatch: A Randomized Correspondence Algorithm, SIGGRAPH '09]

Previous Work: Texture and Patches



[Efros and Leung ‘99]

[Efros and Freeman ‘01]

[Hertzmann et al ‘01]

Three Example Applications

[Barnes et al, PatchMatch: A Randomized Correspondence Algorithm, SIGGRAPH ‘09]

Three Example Applications

[Photoshop demos]

Overview

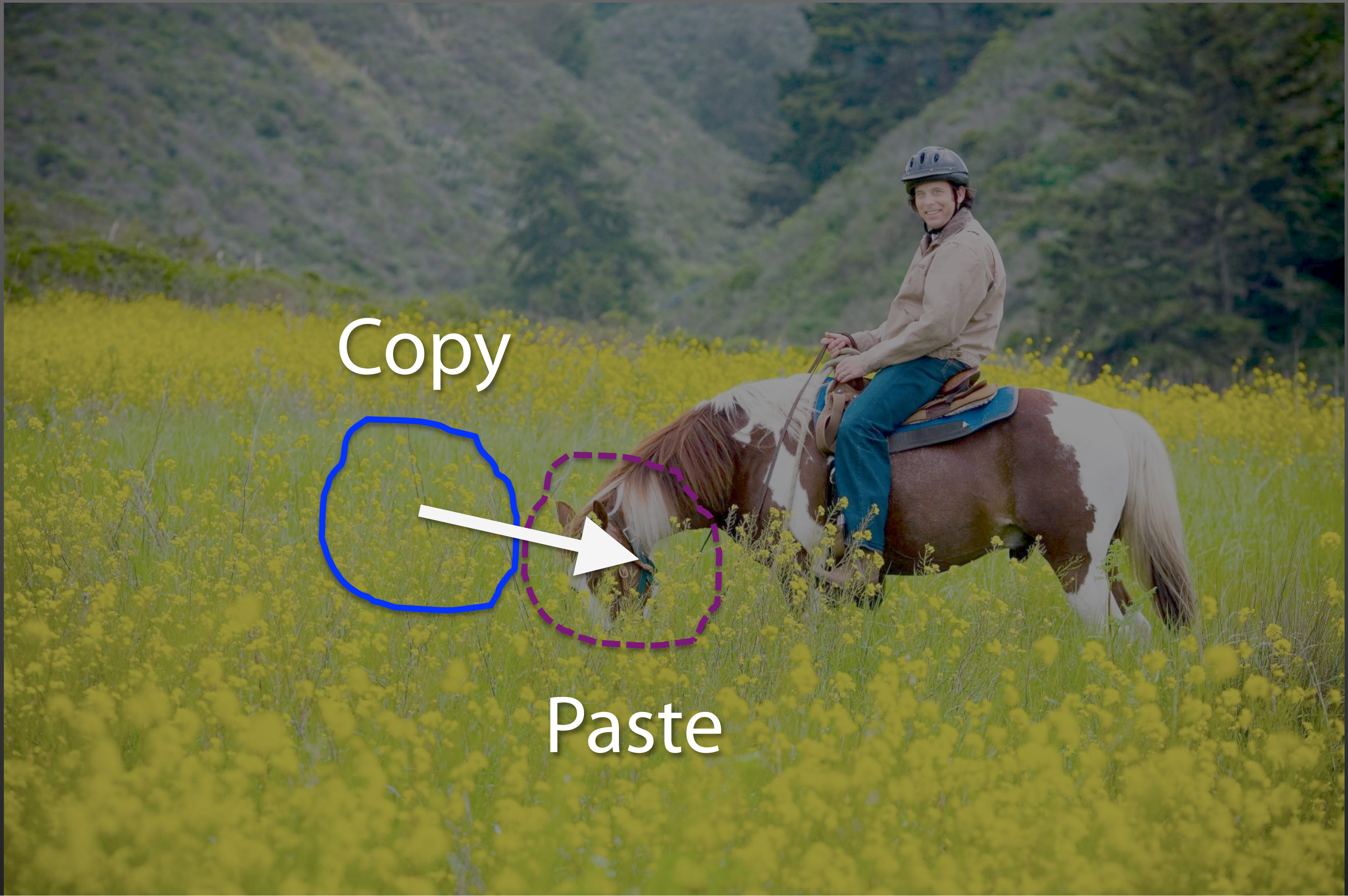
- Fast matching in images: the PatchMatch algorithm
    - Randomized algorithm
    - Applications: Image editing and video tapestries
- Matching under lighting and geometric variations
    - Dealing with higher dimensional spaces
    - Applications: computer vision, stitching and morphing, …
- Impact
- Future perspectives

Overview

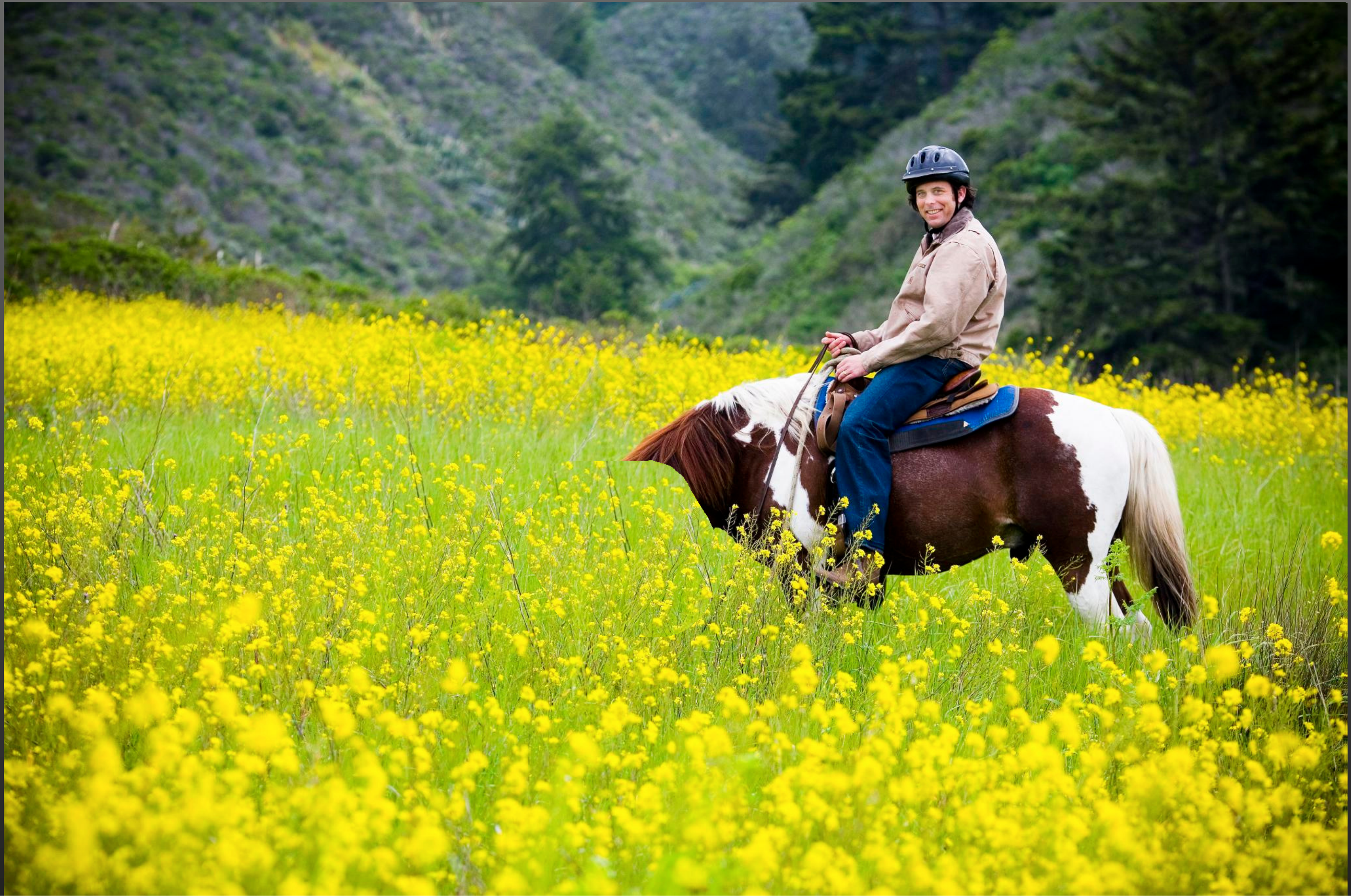
- Fast matching in images: the PatchMatch algorithm
  - Randomized algorithm
  - Applications: Image editing and video tapestries
- Matching under lighting and geometric variations
  - Dealing with higher dimensional spaces
  - Applications: computer vision, stitching and morphing, …
- Impact
- Future perspectives

Manual Hole Filling

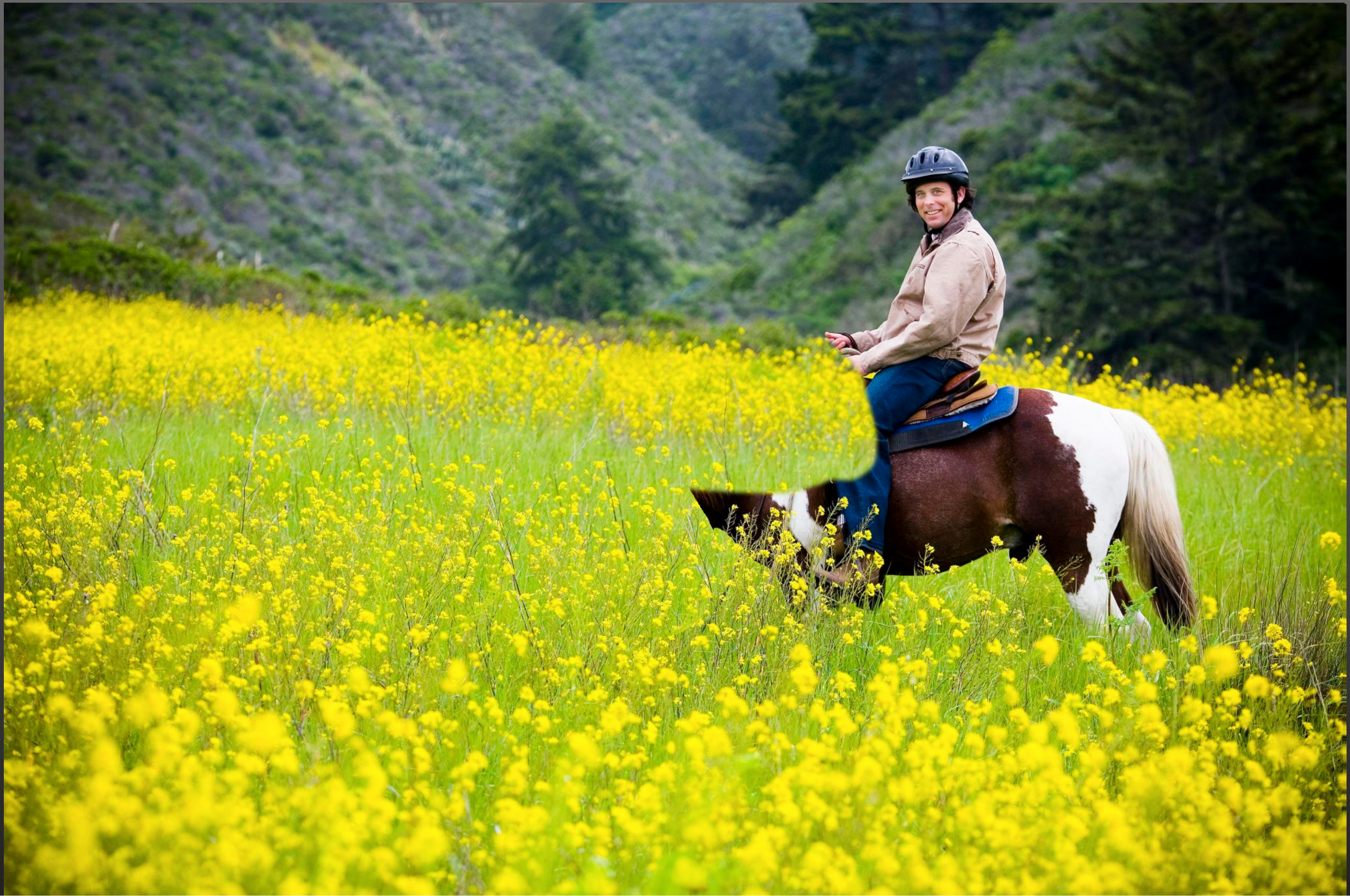
Manual Hole Filling



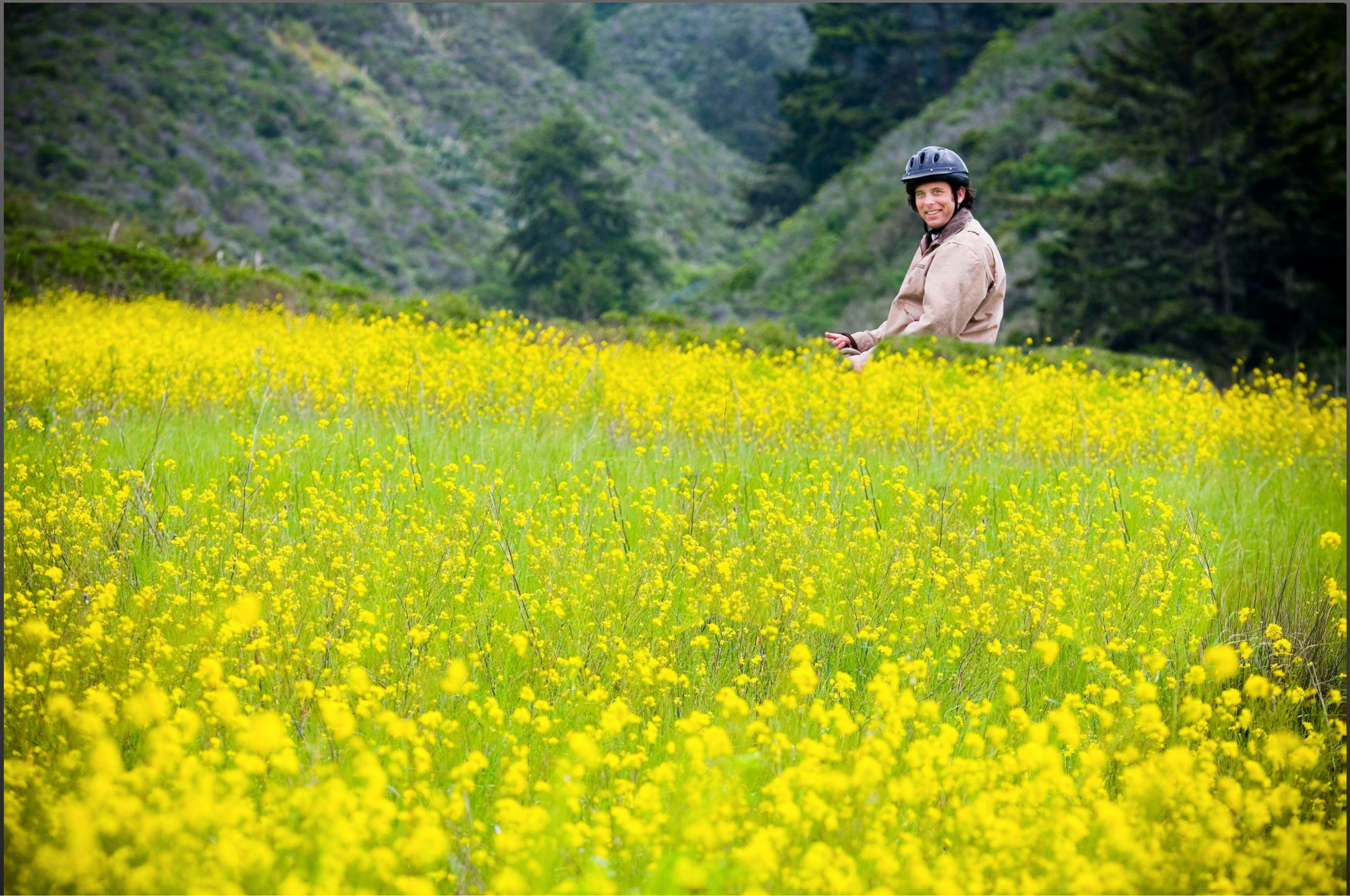
Manual Hole Filling



Manual Hole Filling



Manual Hole Filling



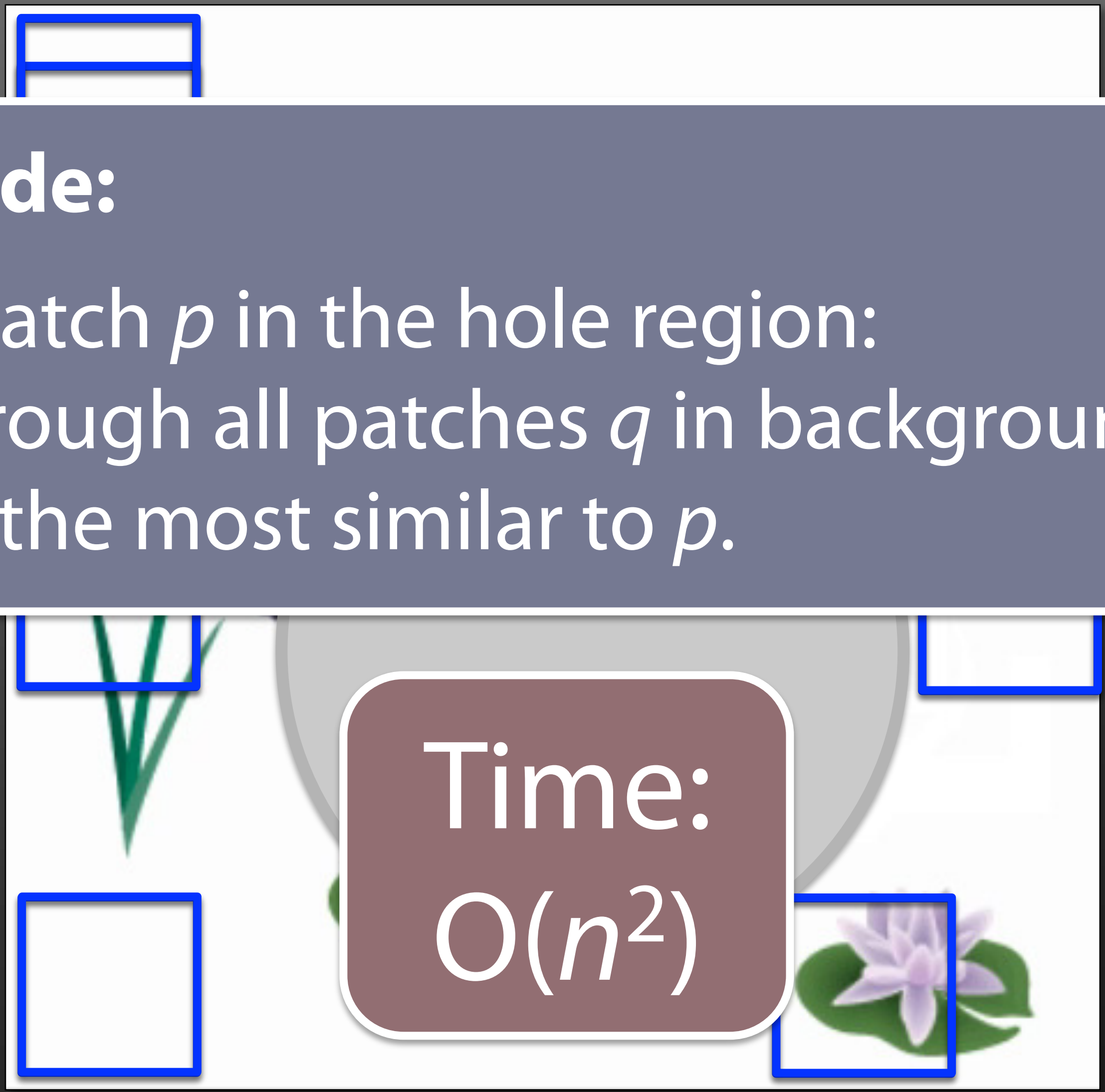
Manual Hole Filling

Manual Hole Filling

Intuition for Automatic Hole Filling

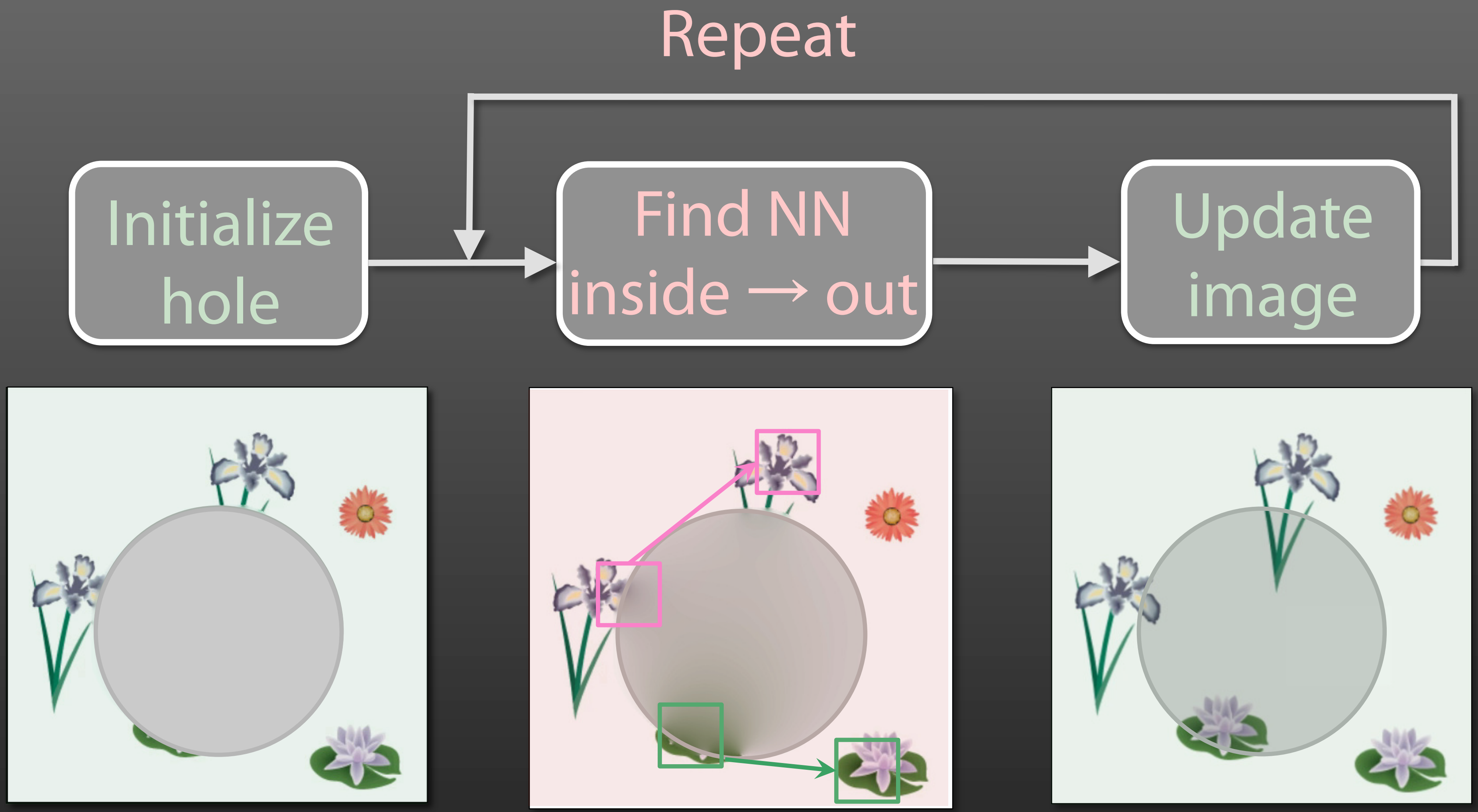
Pseudocode:

```
For each patch $p$ in the hole region:
  Loop through all patches $q$ in background region;
      Find the most similar to $p$.
```



Time:
O(n^2)

Hole Filling



Repeat

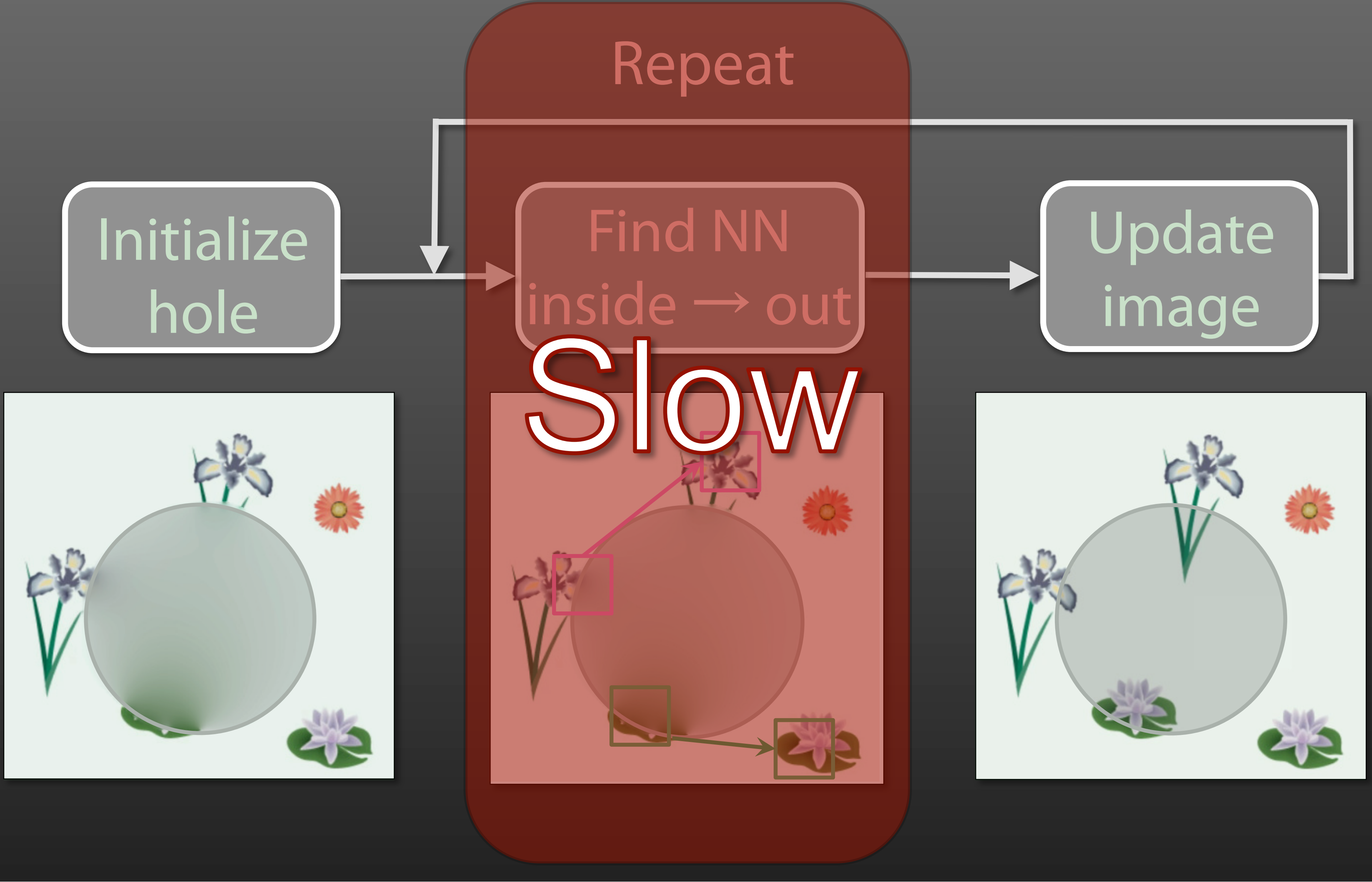
Initialize hole

Find NN inside → out

Update image

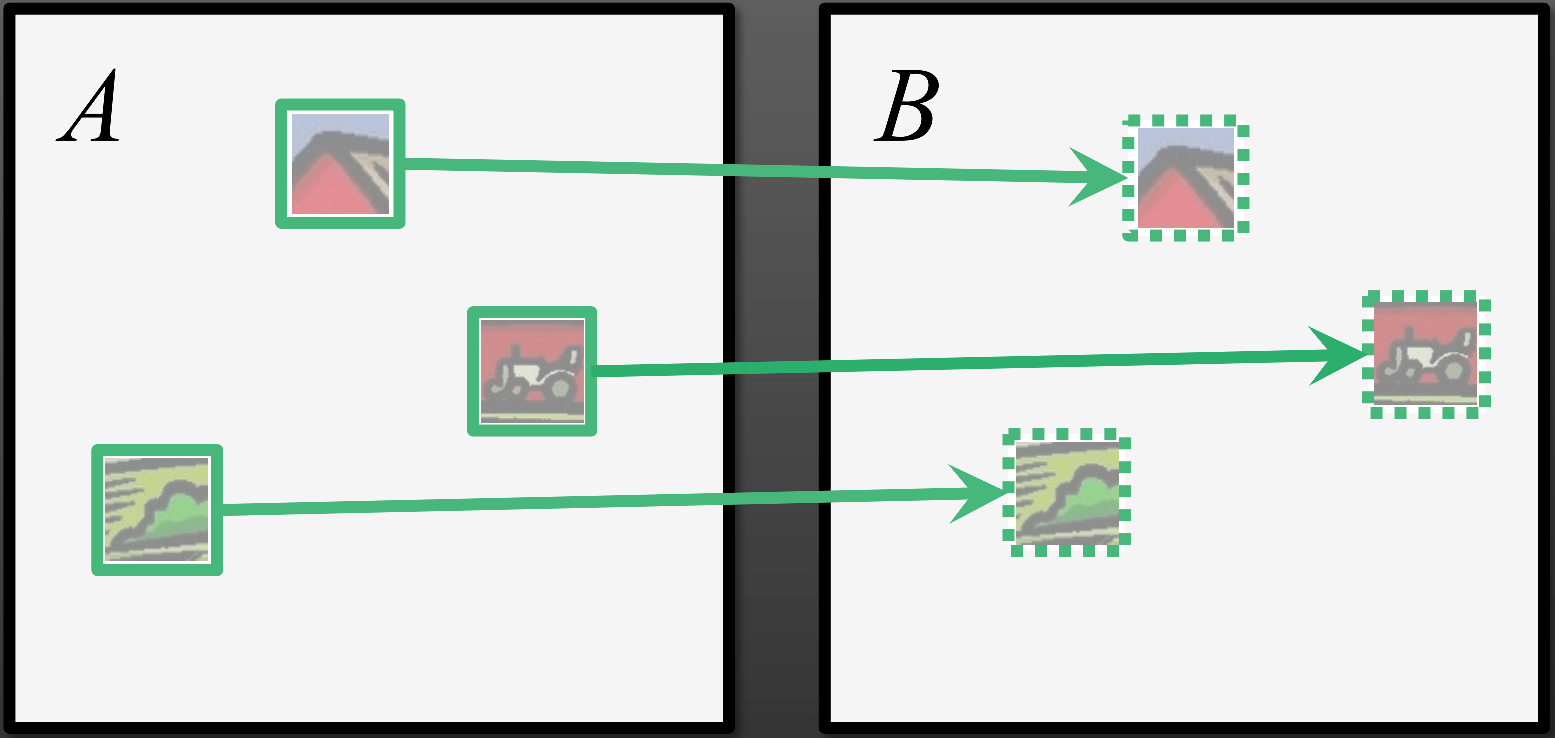
[Wexler '04]

Hole Filling



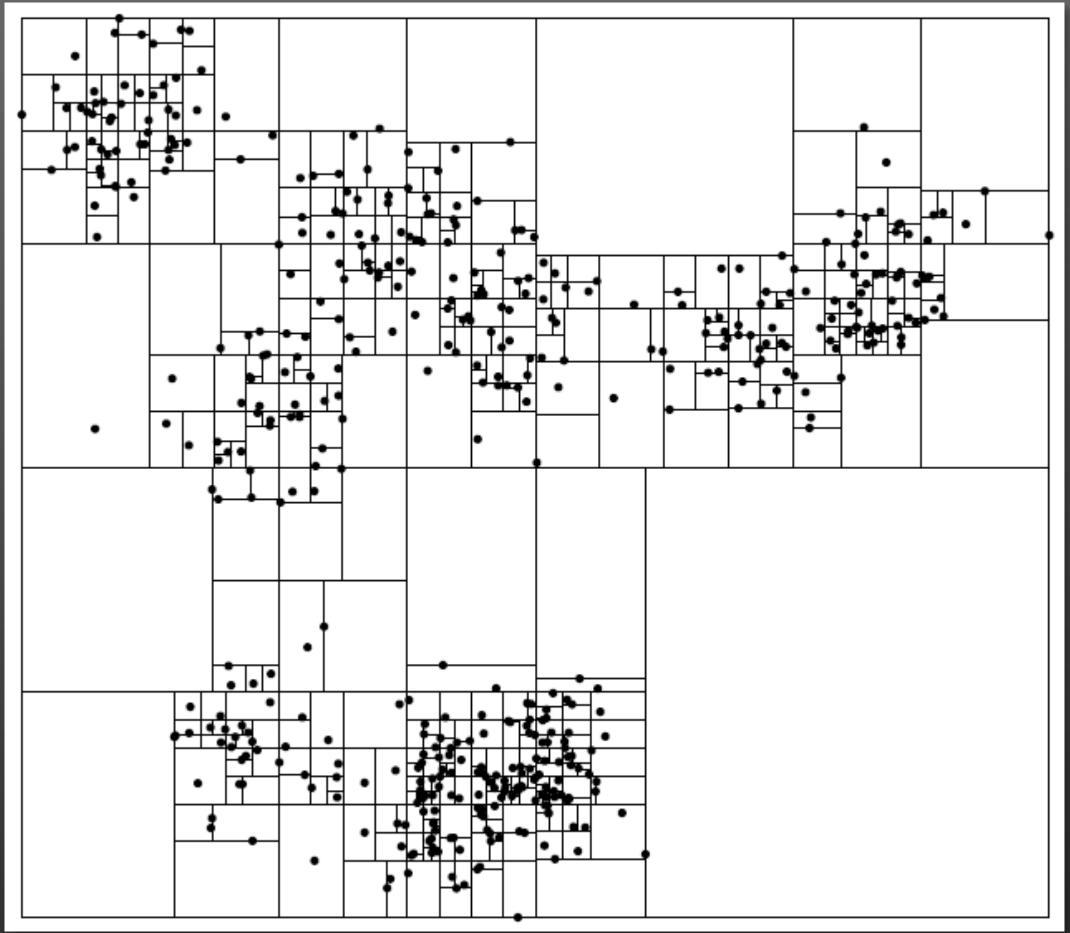
[Wexler '04]

Problem



“Hole” “Background”

Previous Work



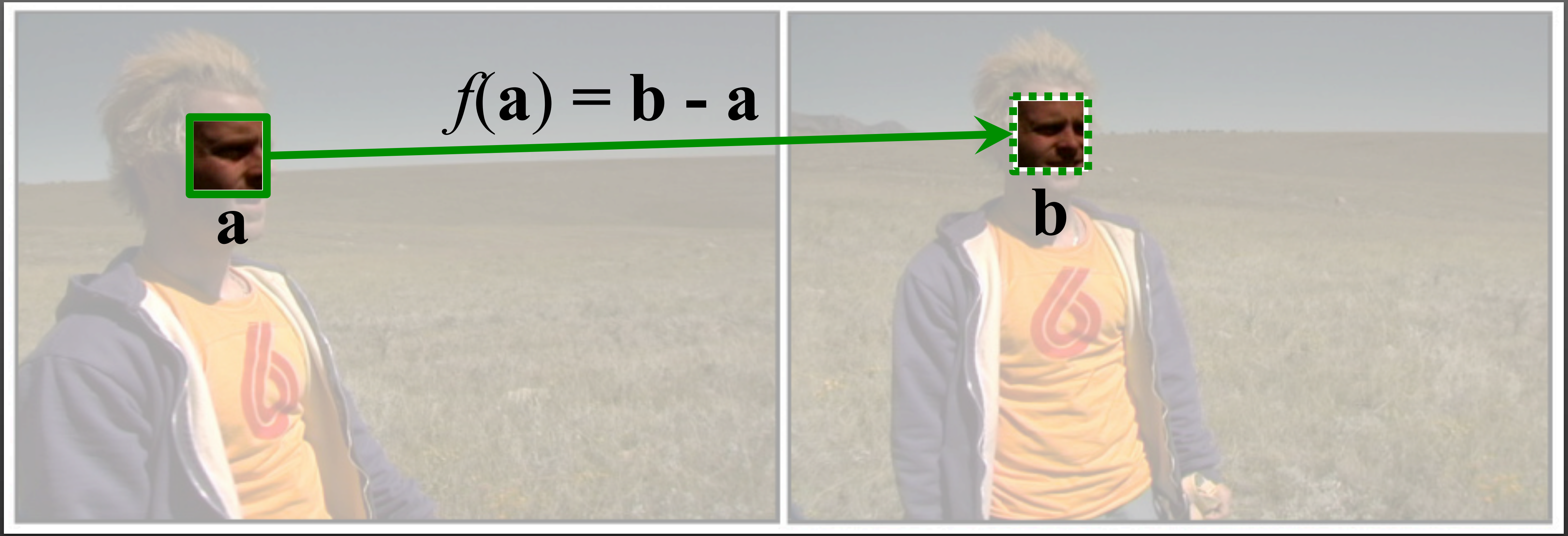
*k*d-tree with PCA

[Hertzmann '01]

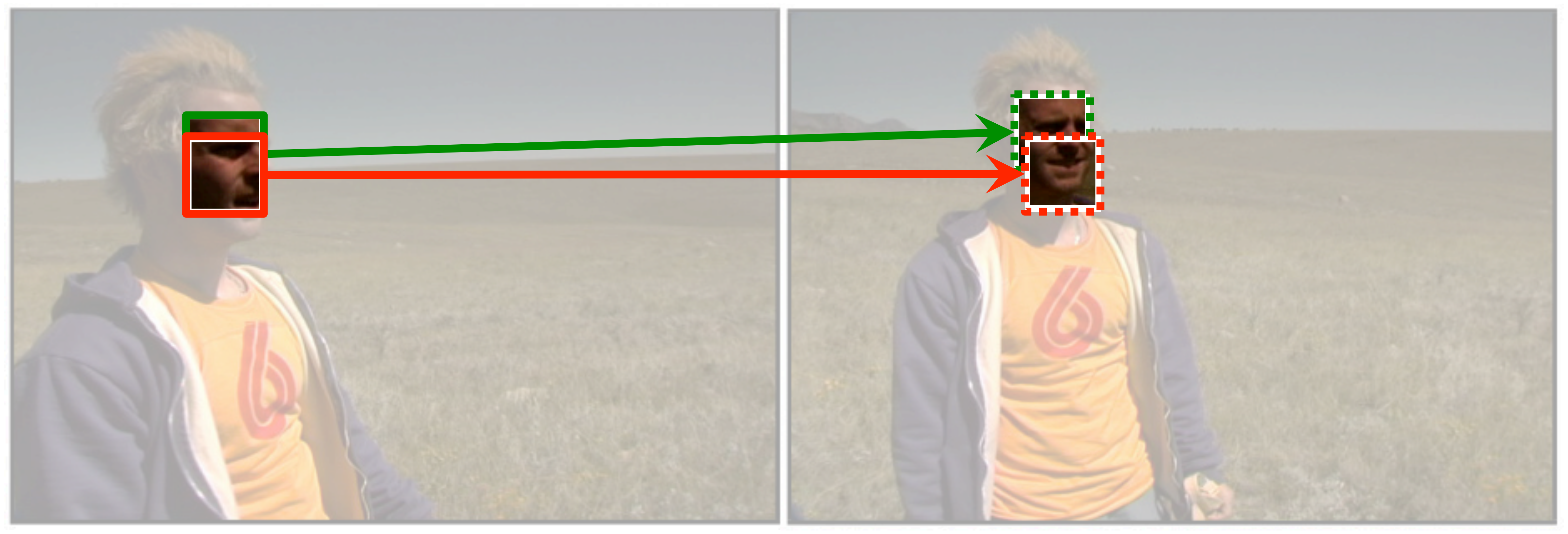
Time:
$O(n \log n)$

Observation about Image Statistics

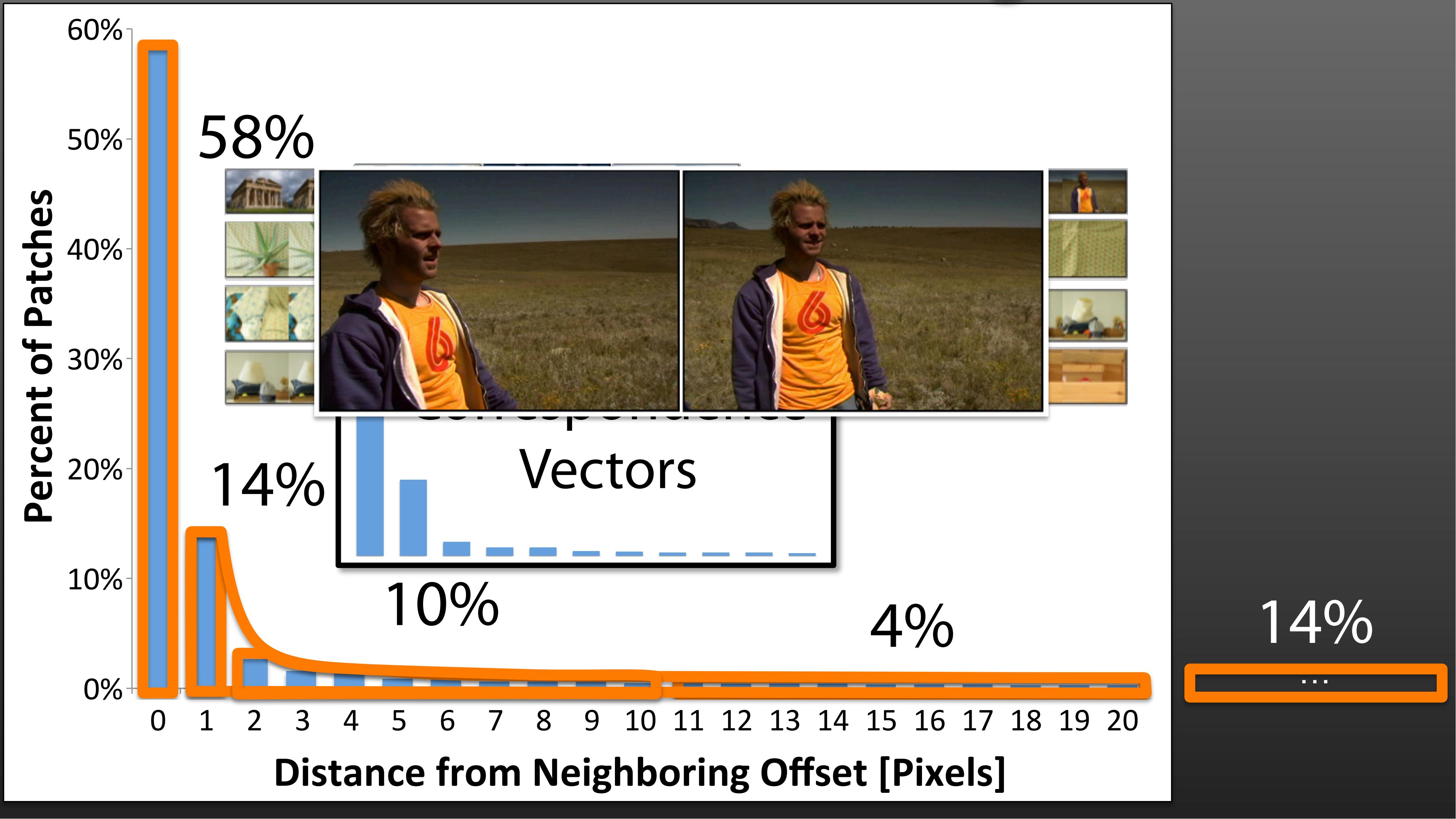
Observation about Image Statistics



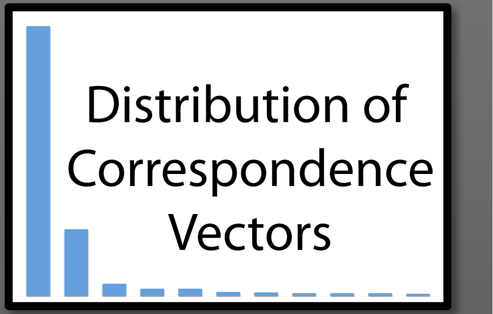
Observation about Image Statistics

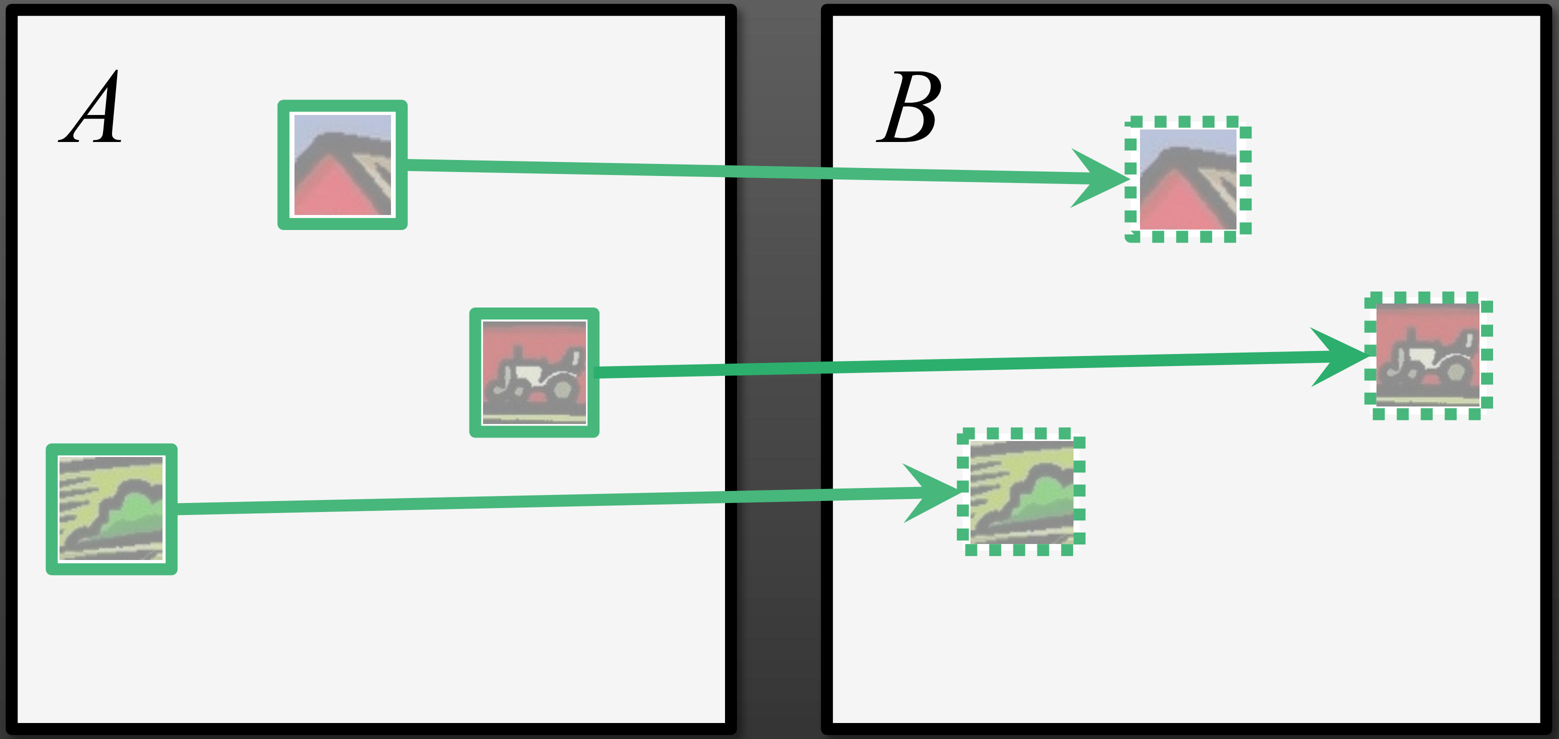


Use Statistics to Make Fast Algorithm

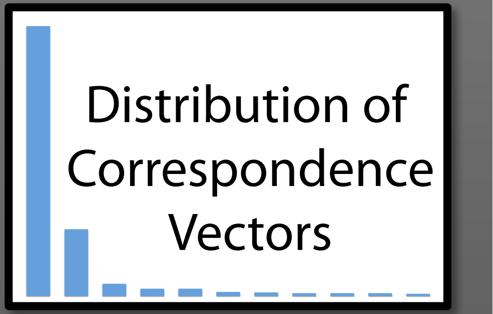


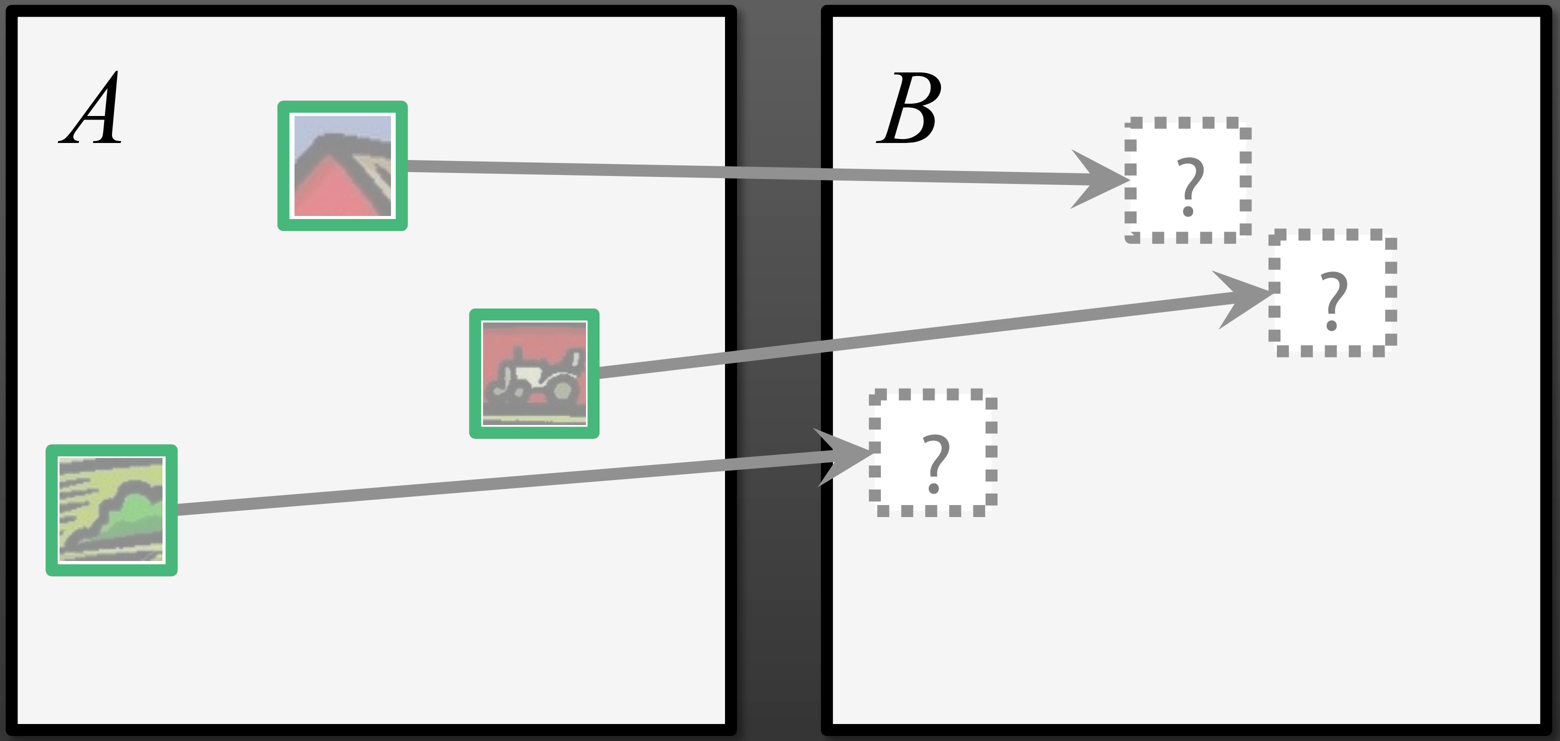
Matching: Translation Only



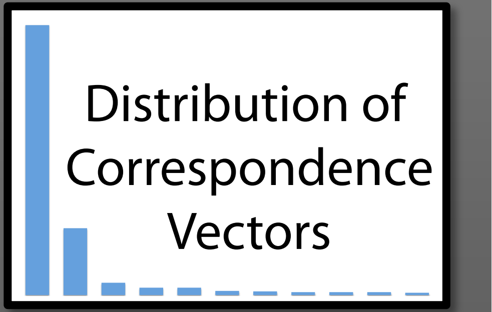


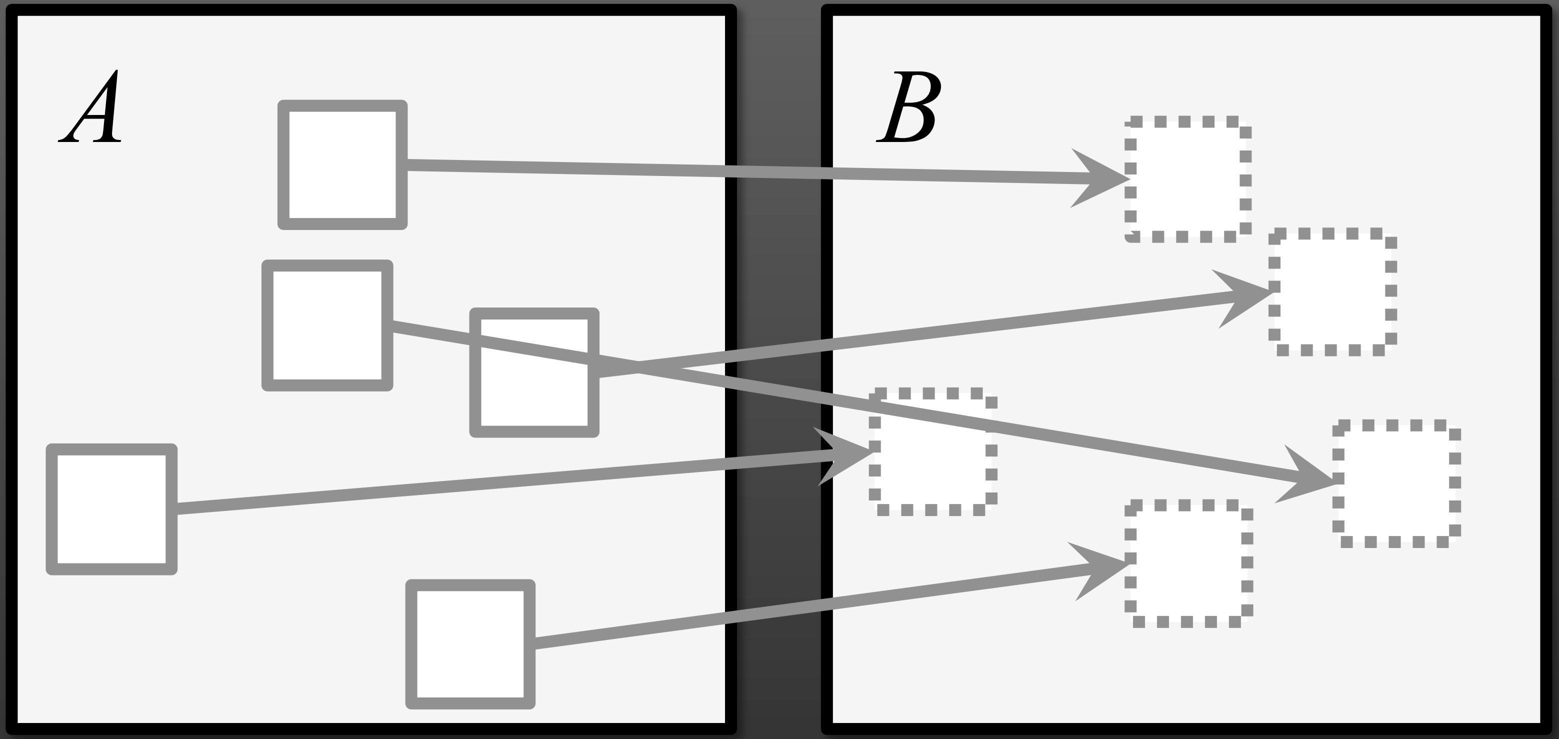
Intuition for PatchMatch



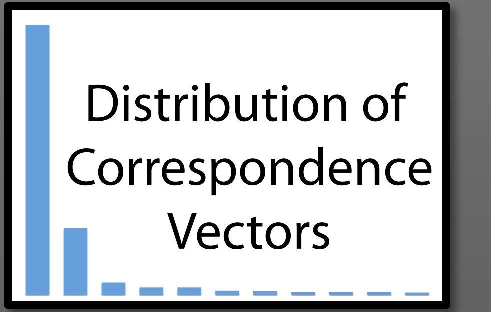


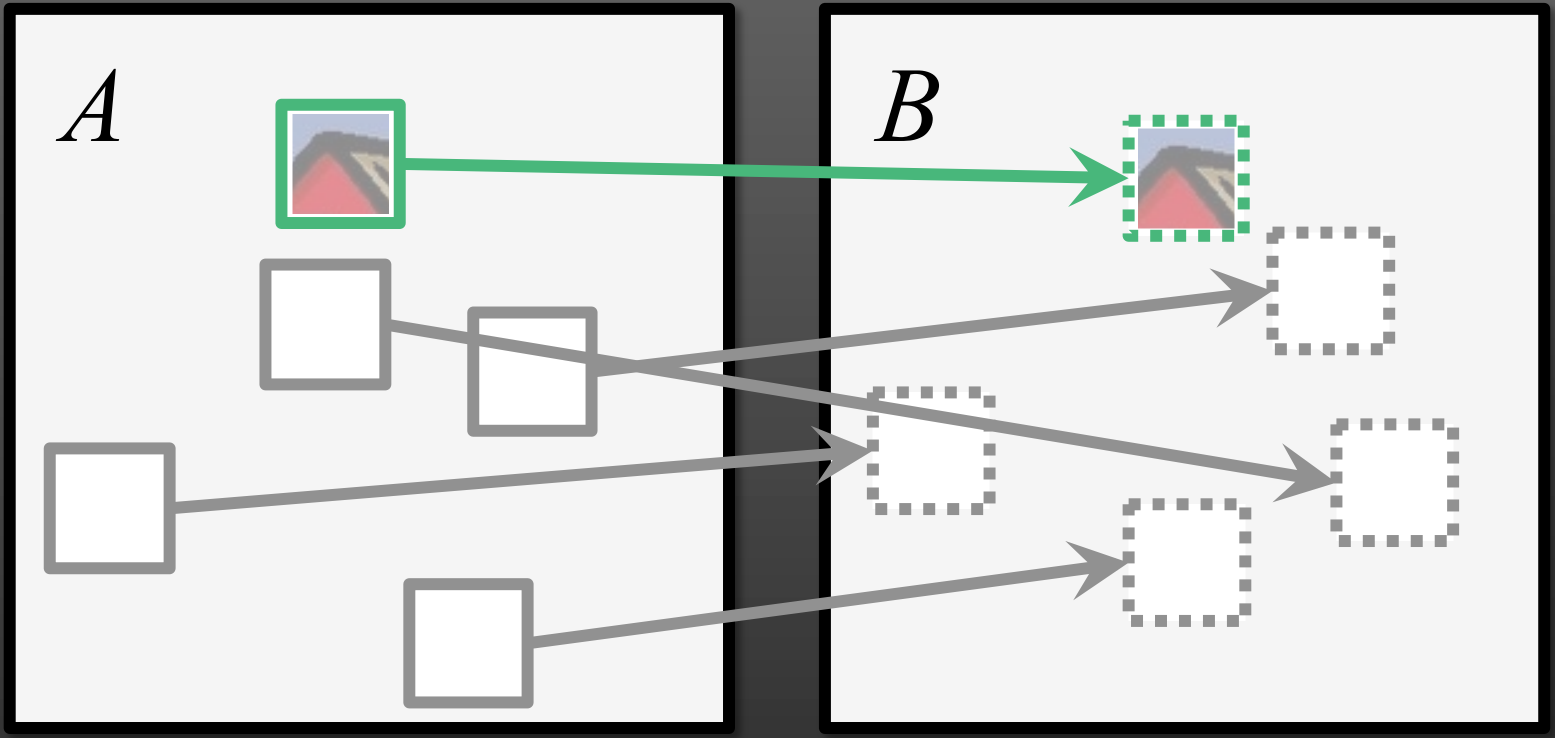
Intuition for PatchMatch



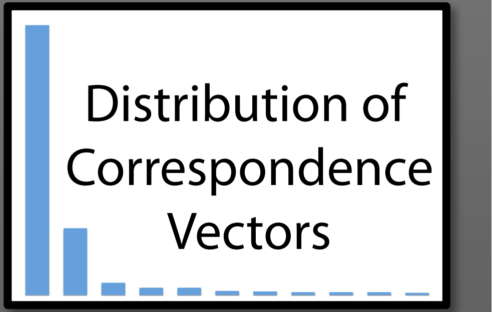


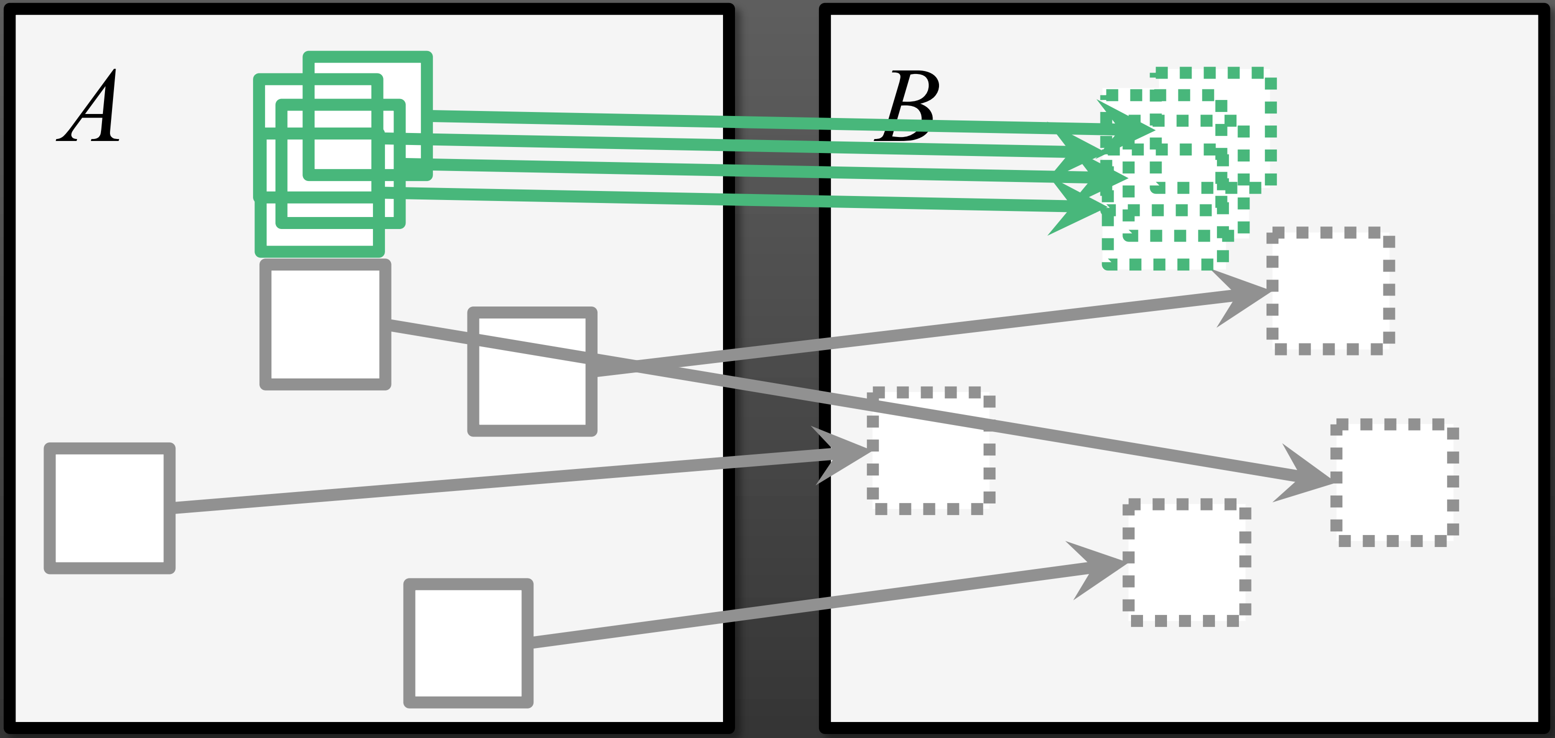
Intuition for PatchMatch



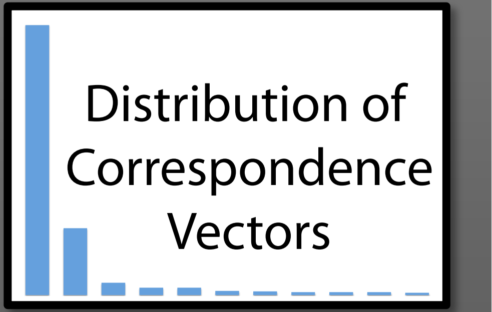


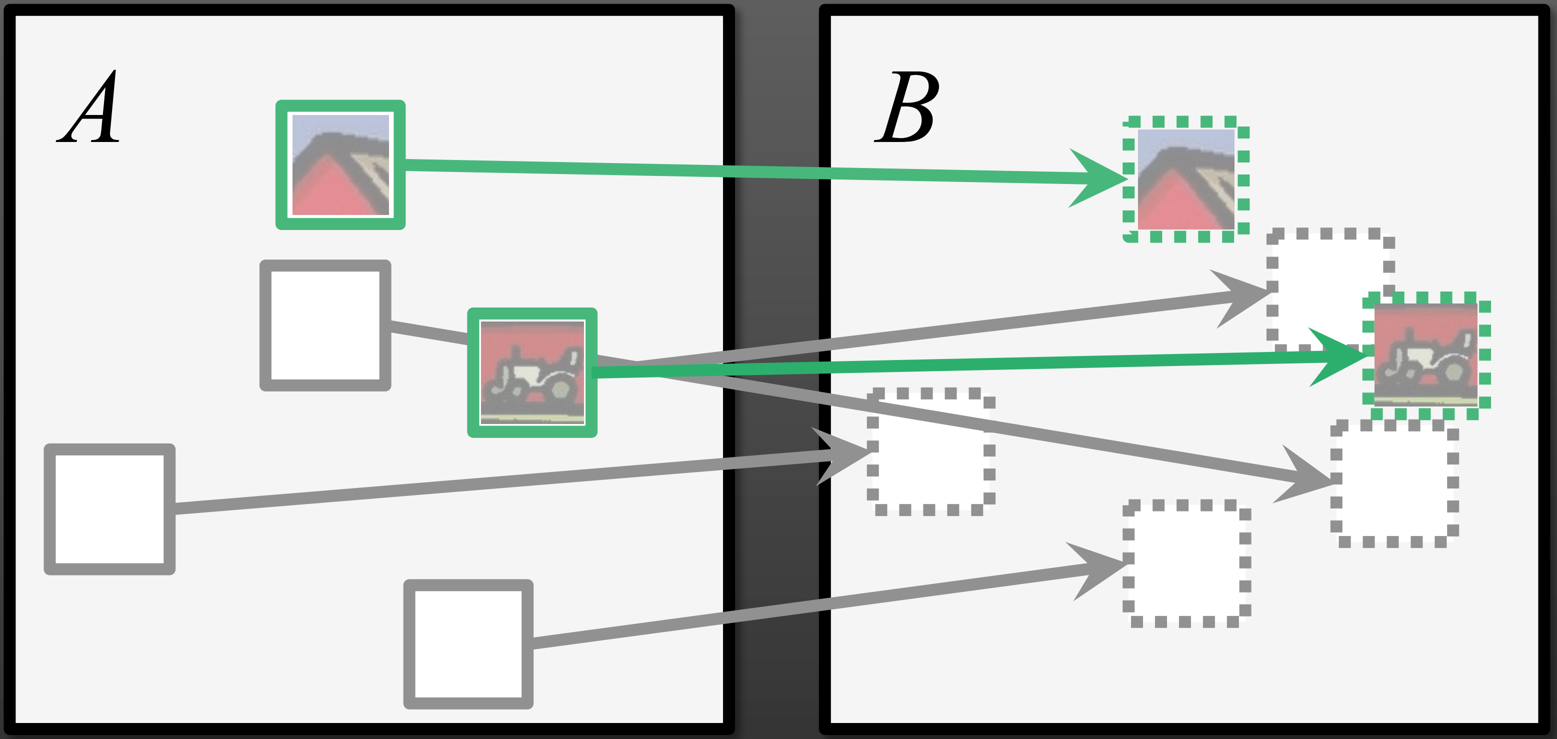
Intuition for PatchMatch



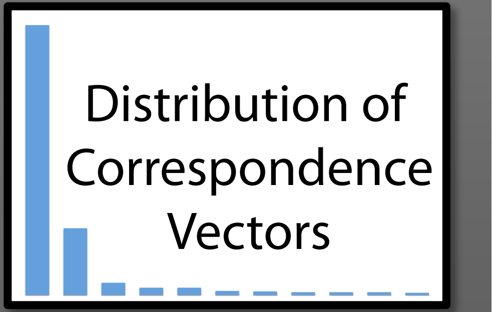


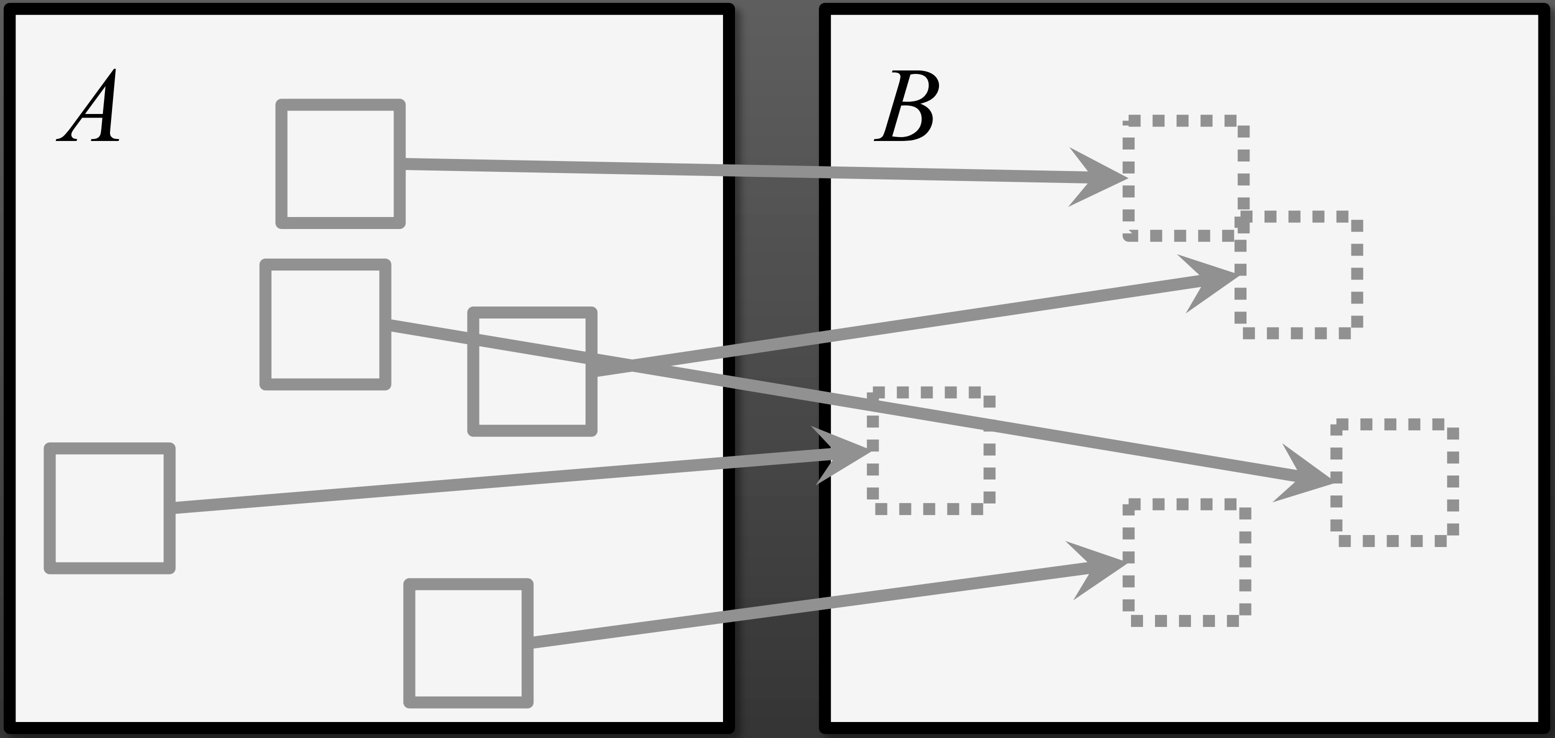
Intuition for PatchMatch



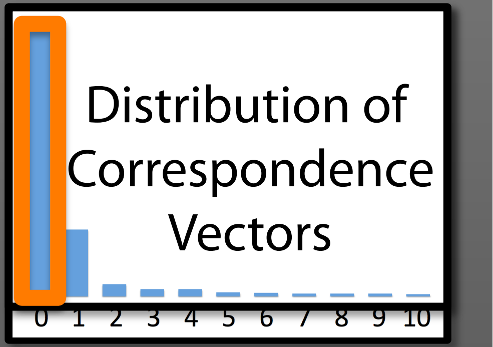


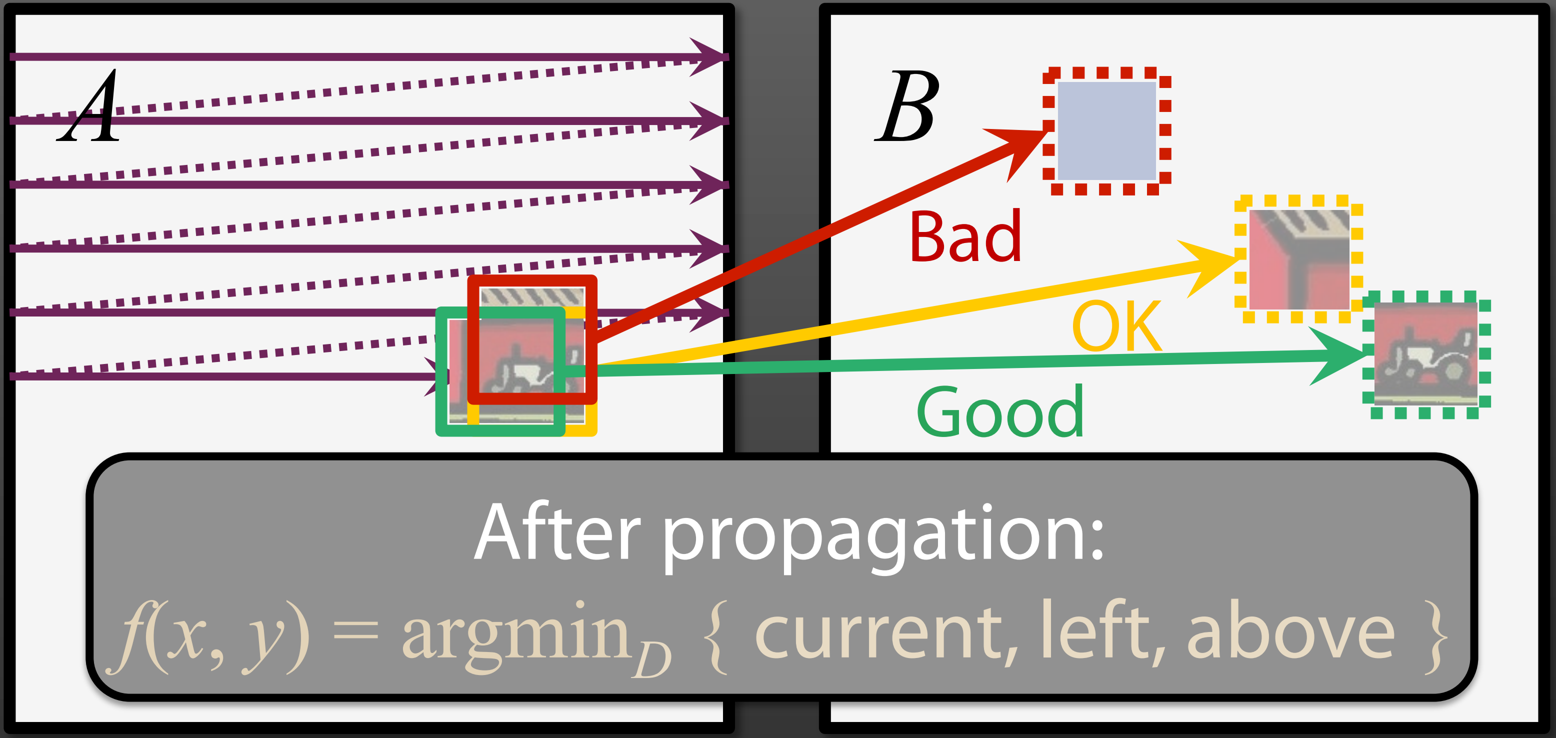
Step 1: Random Initialization





Step 2: Propagation

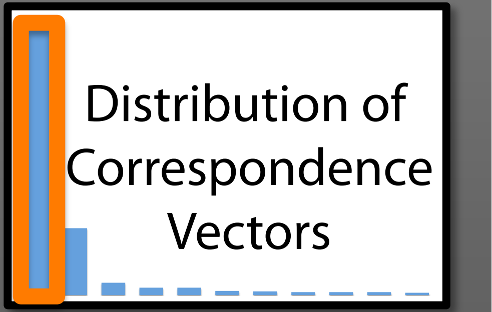


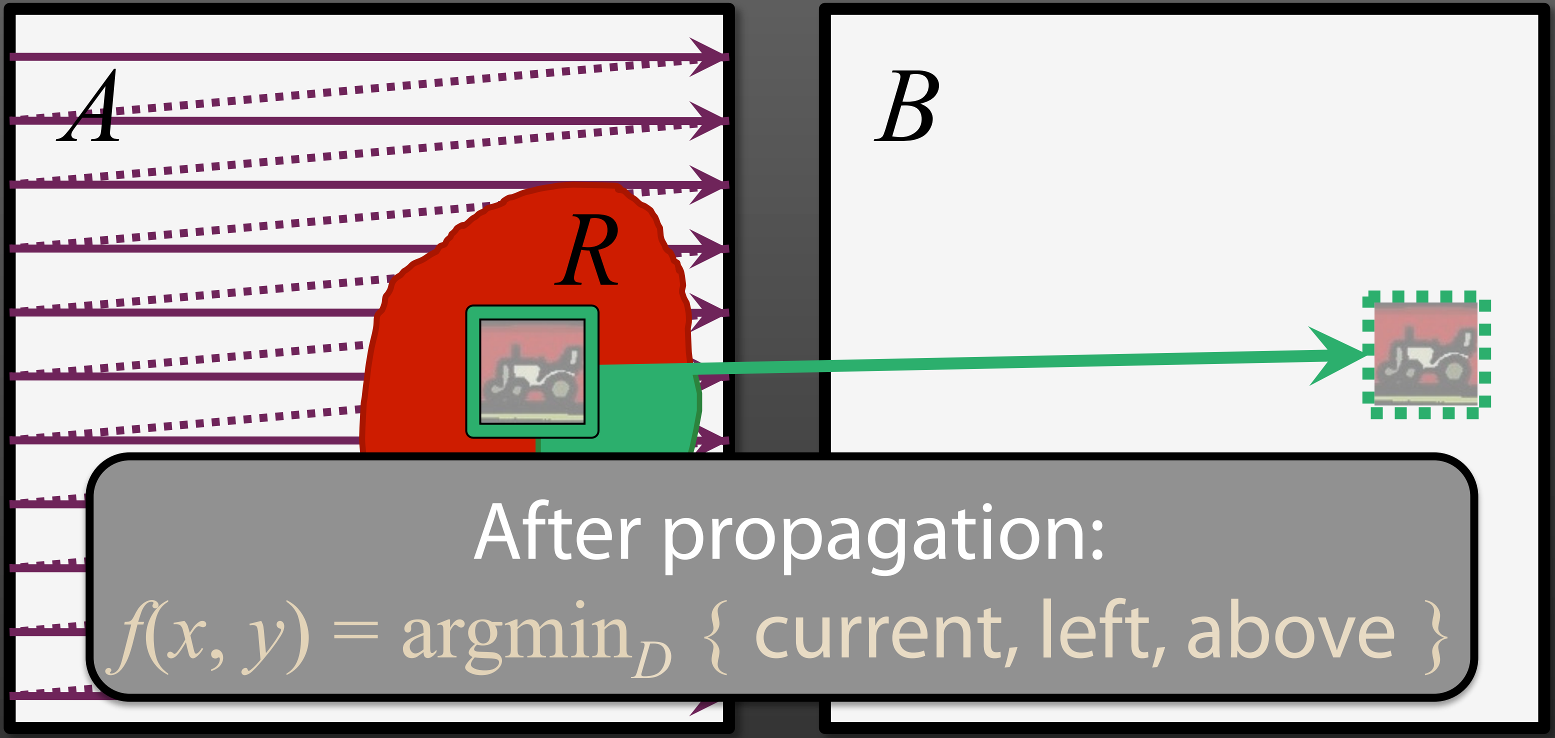


After propagation:

$f(x, y) = \mathrm{argmin}_D$ { current, left, above }

Step 2: Propagation

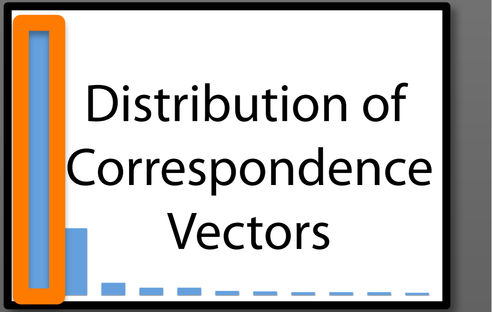


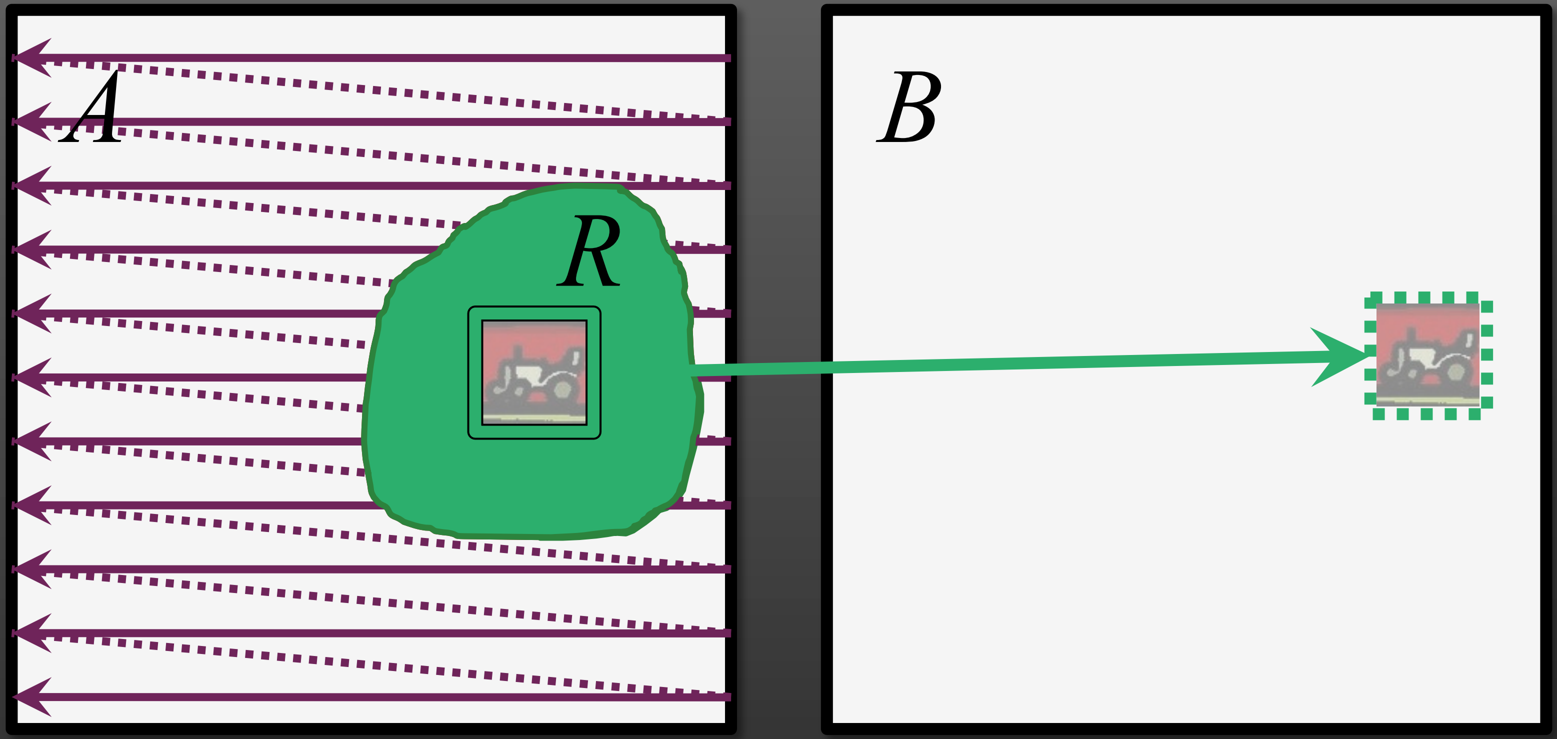


After propagation:

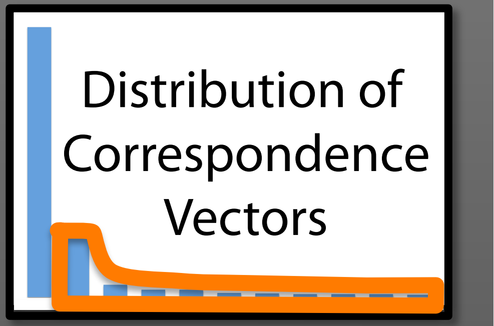
$f(x, y) = \text{argmin}_D$ { current, left, above }

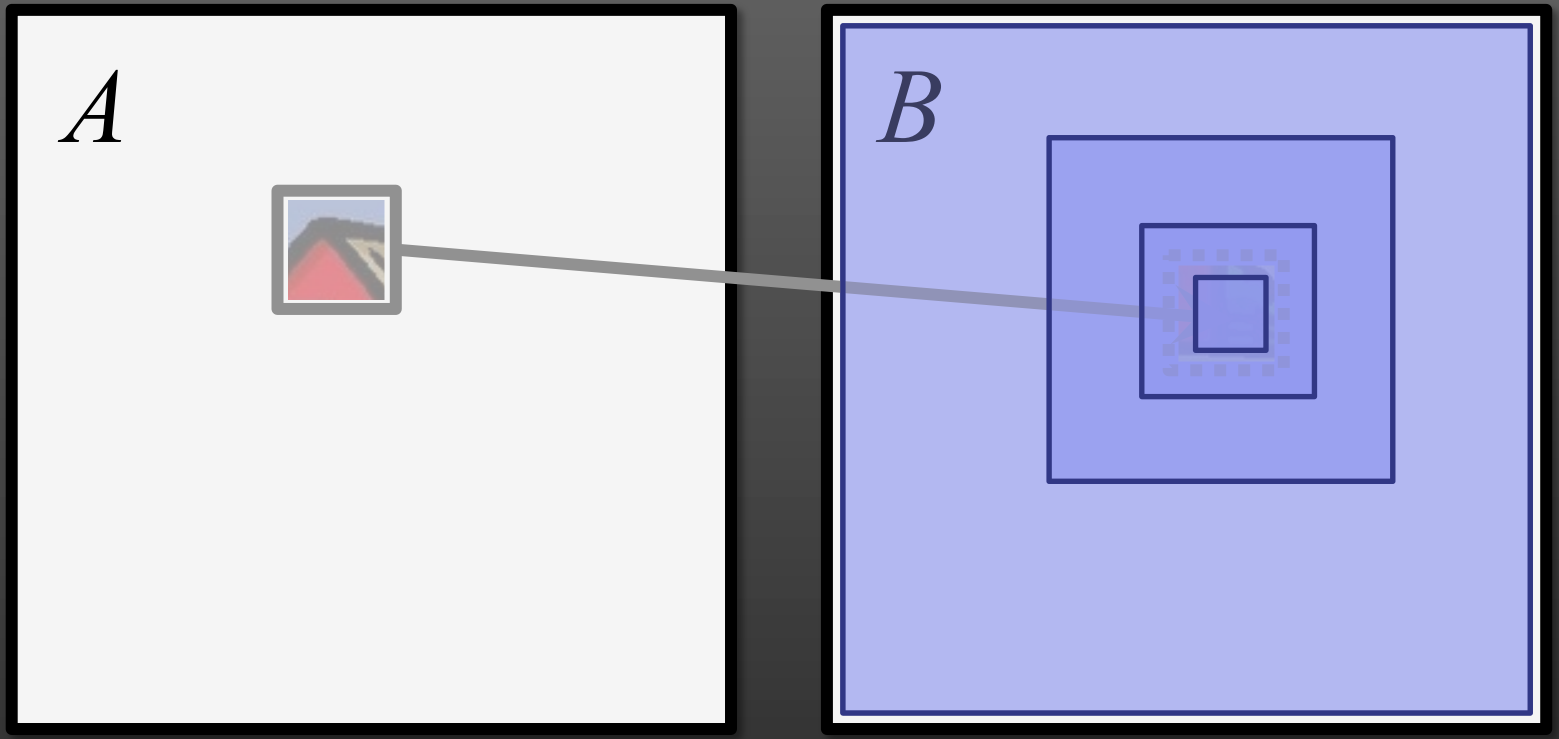
Step 2: Propagation



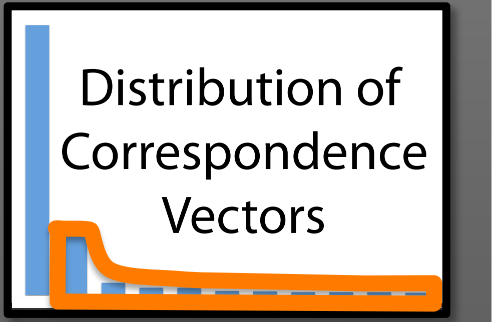


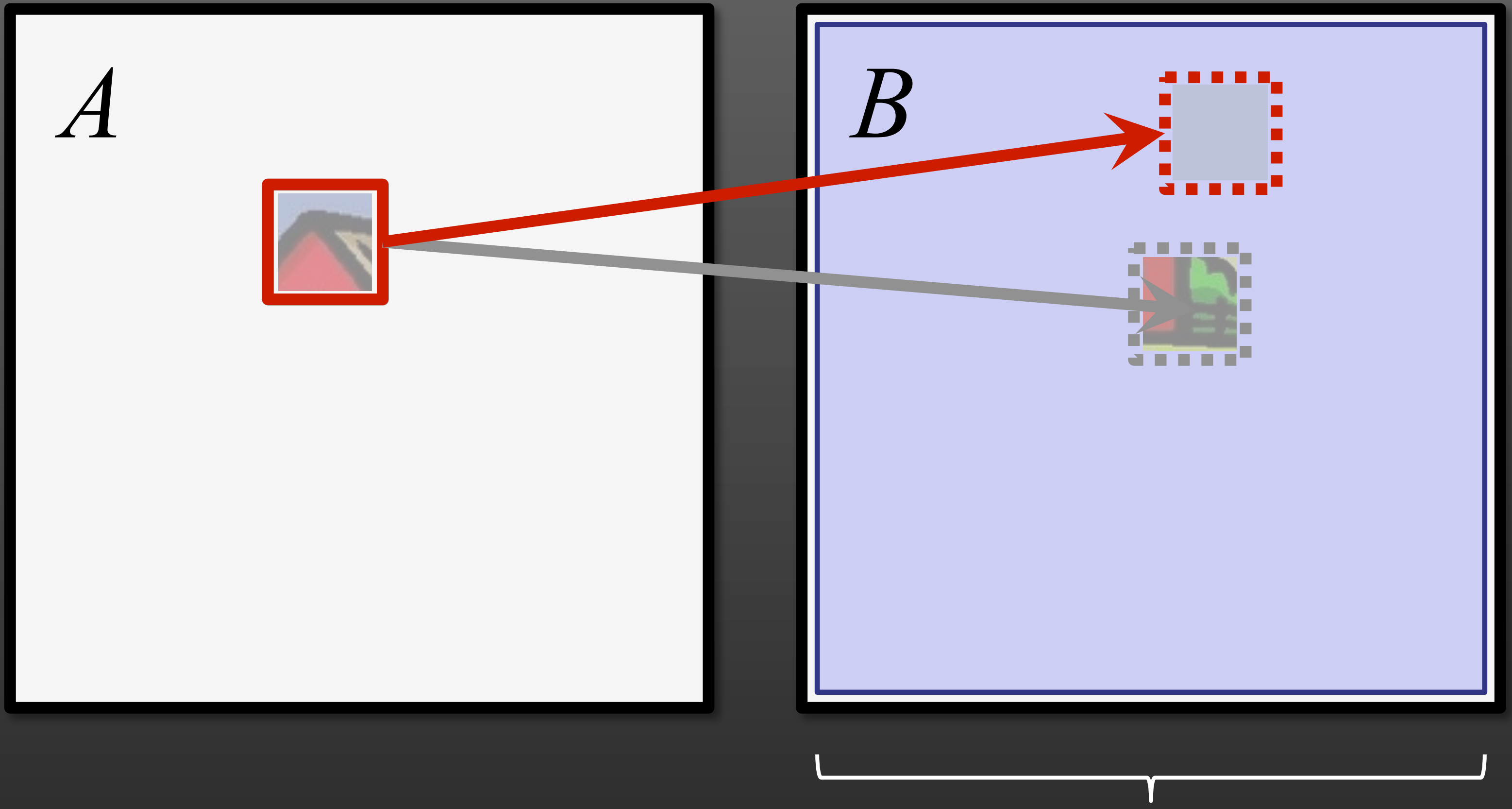
Step 3: Random Search





Step 3: Random Search

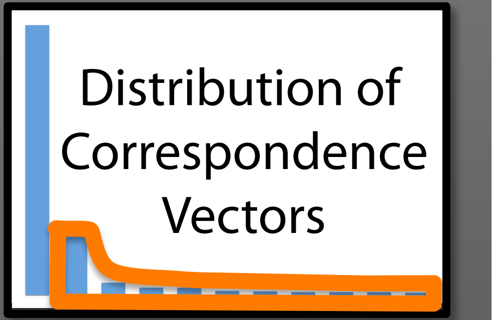


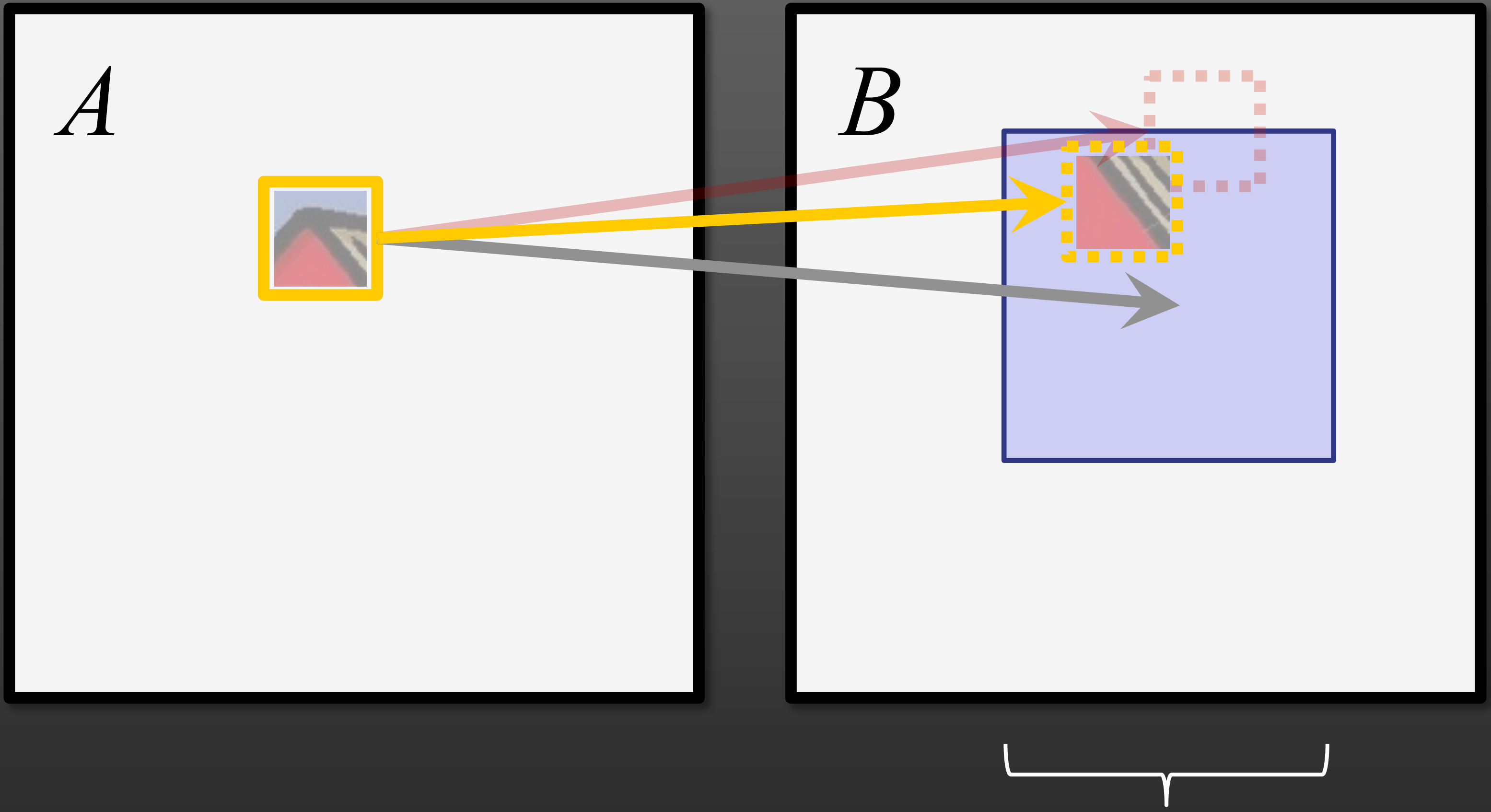


Box width: w

(image width)

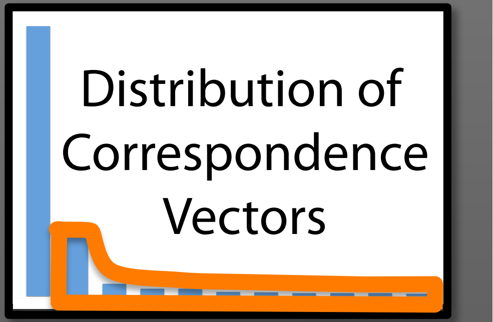
Step 3: Random Search



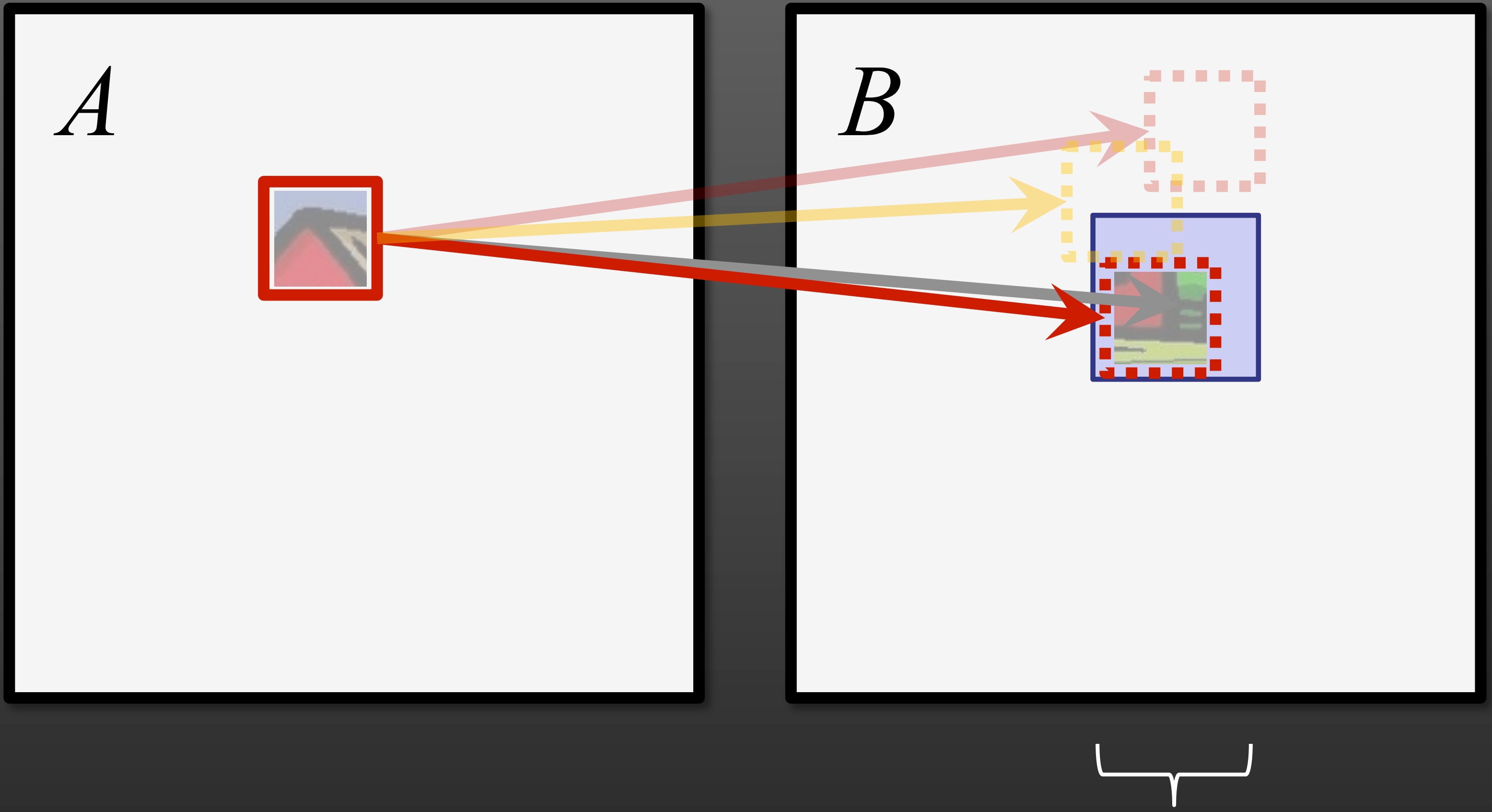


Box width: αw

Step 3: Random Search

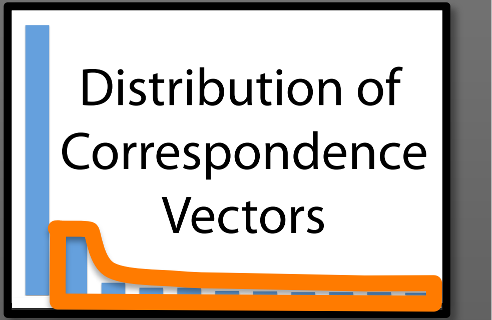


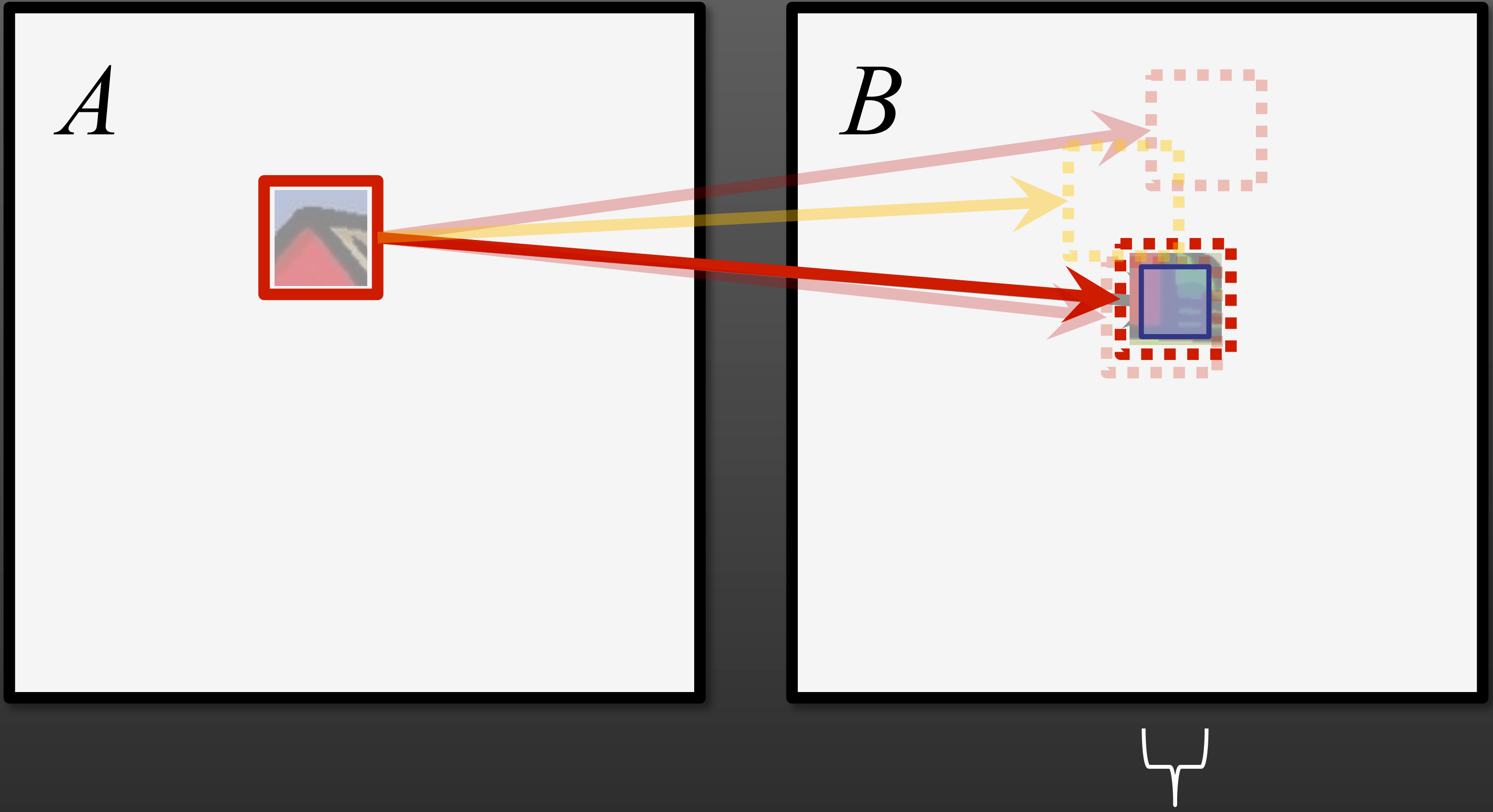
Distribution of Correspondence Vectors



Box width: $\alpha^2 w$

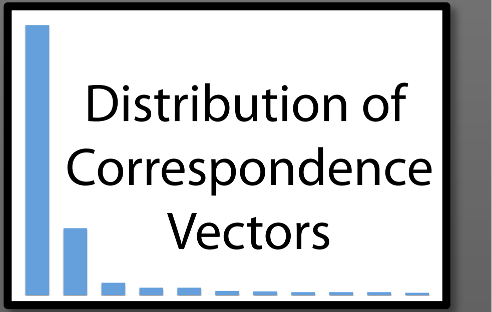
Step 3: Random Search

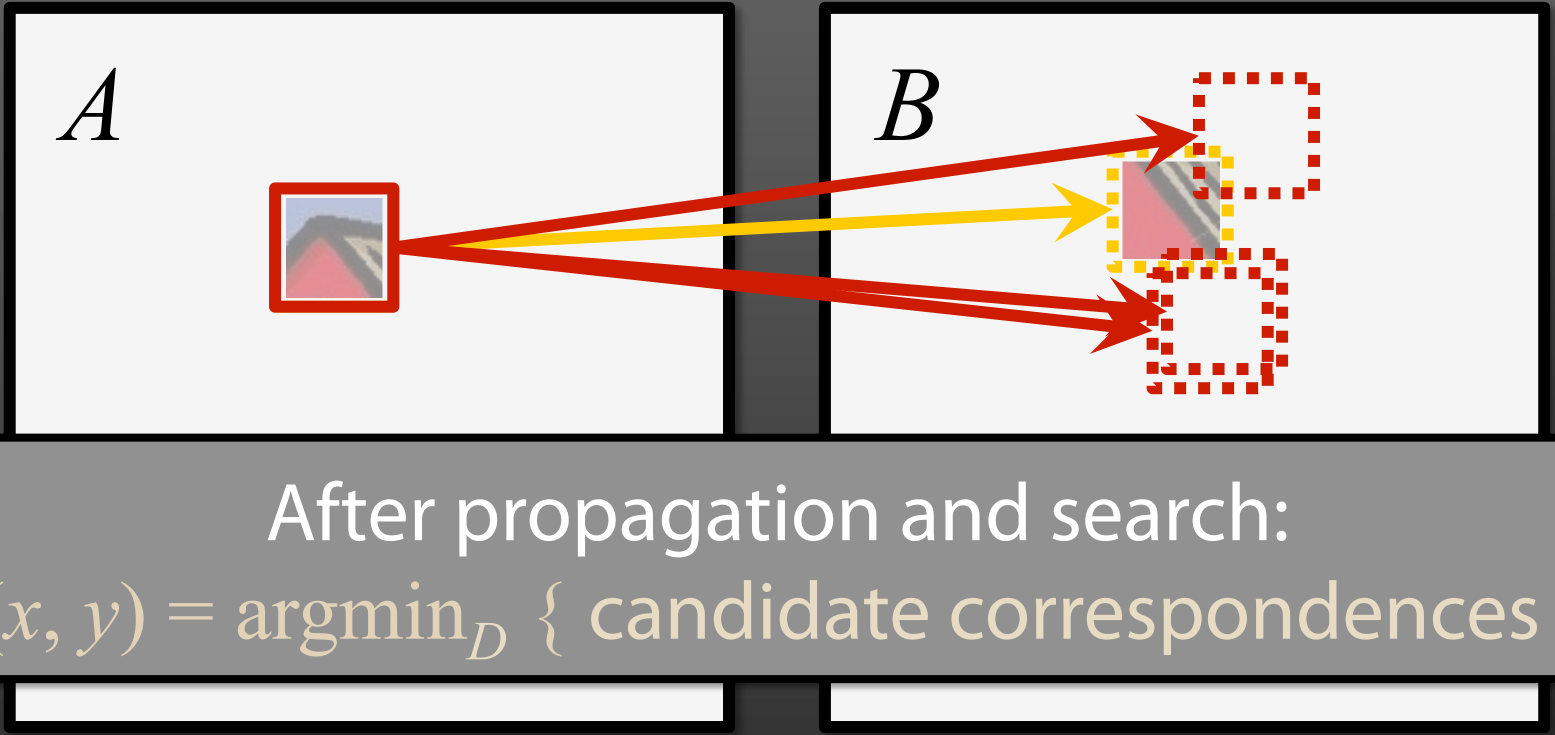




Box width: 1 pixel

Propagation and Search

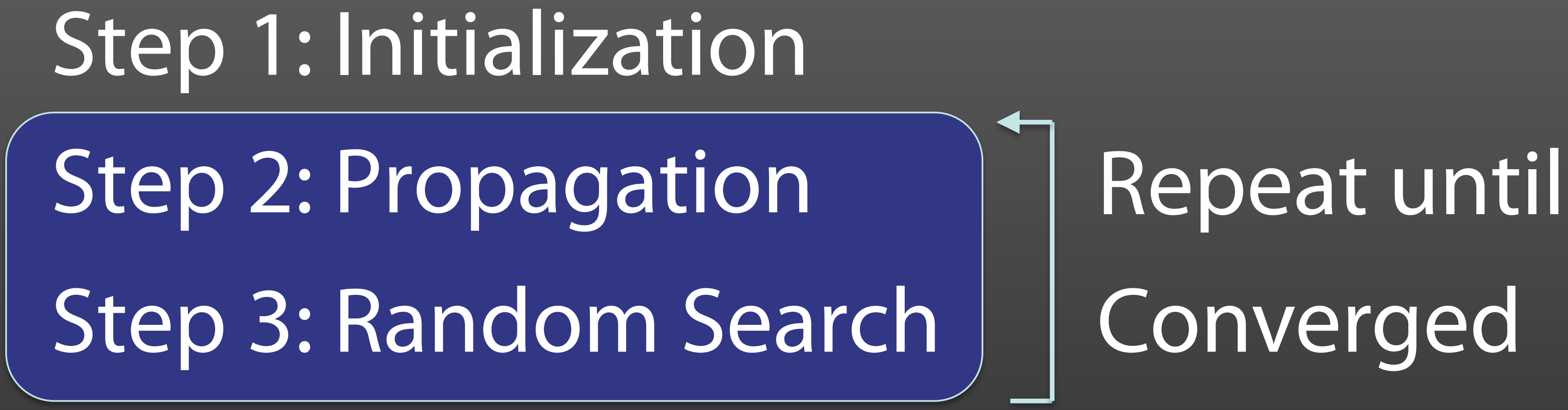




After propagation and search:

$f(x, y) = \text{argmin}_D$ { candidate correspondences }

PatchMatch: Summary



Step 1: Initialization

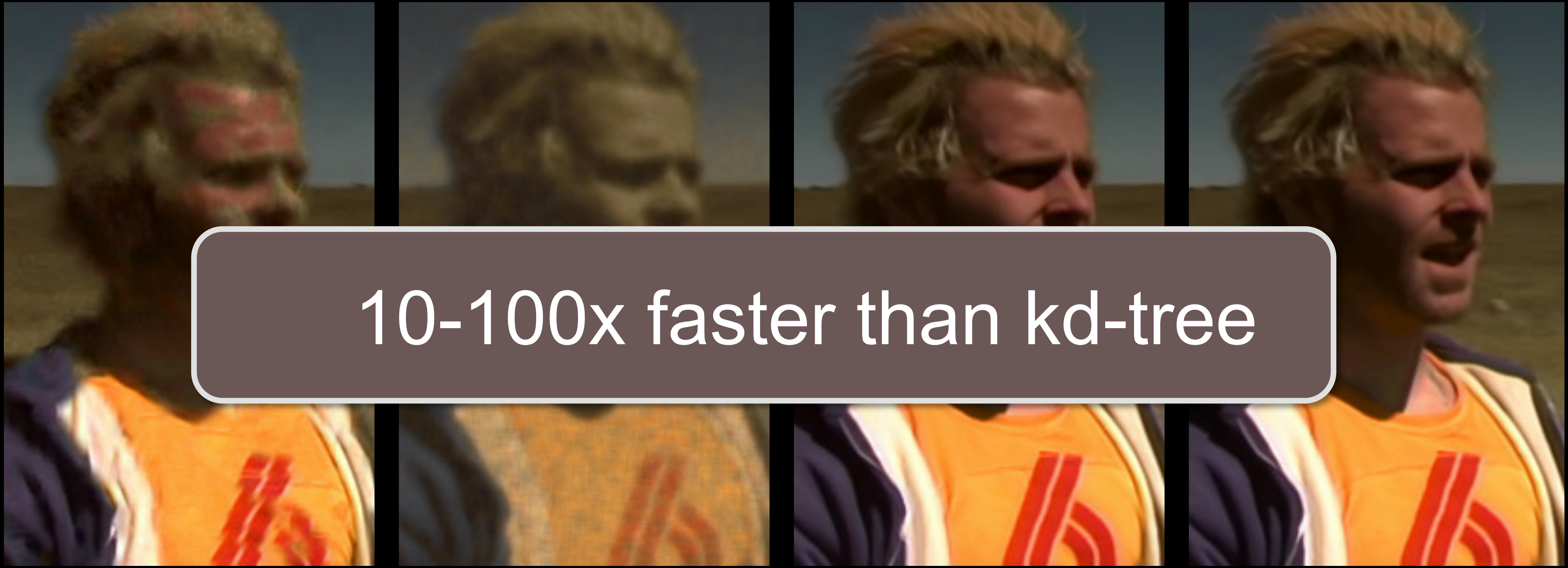
Step 2: Propagation

Step 3: Random Search

Repeat until Converged

Convergence

Convergence



10-100x faster than kd-tree

Propagation

Random Search

PatchMatch

Ground Truth

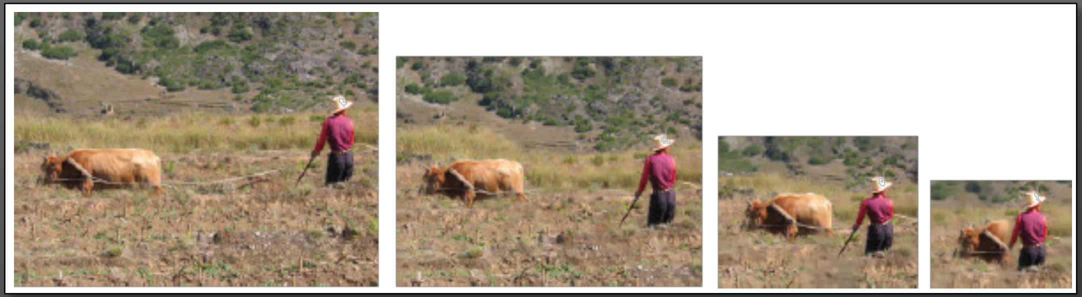
[After first iteration]

[After second iteration]

Overview

- Fast matching in images: the PatchMatch algorithm
  - Randomized algorithm
  - Applications: Image editing and video tapestries
- Matching under lighting and geometric variations
  - Dealing with higher dimensional spaces
  - Applications: computer vision, stitching and morphing, …
- Impact
- Future perspectives

Iterative Optimization Methods



Retargeting and Reshuffling

[Simakov '08]

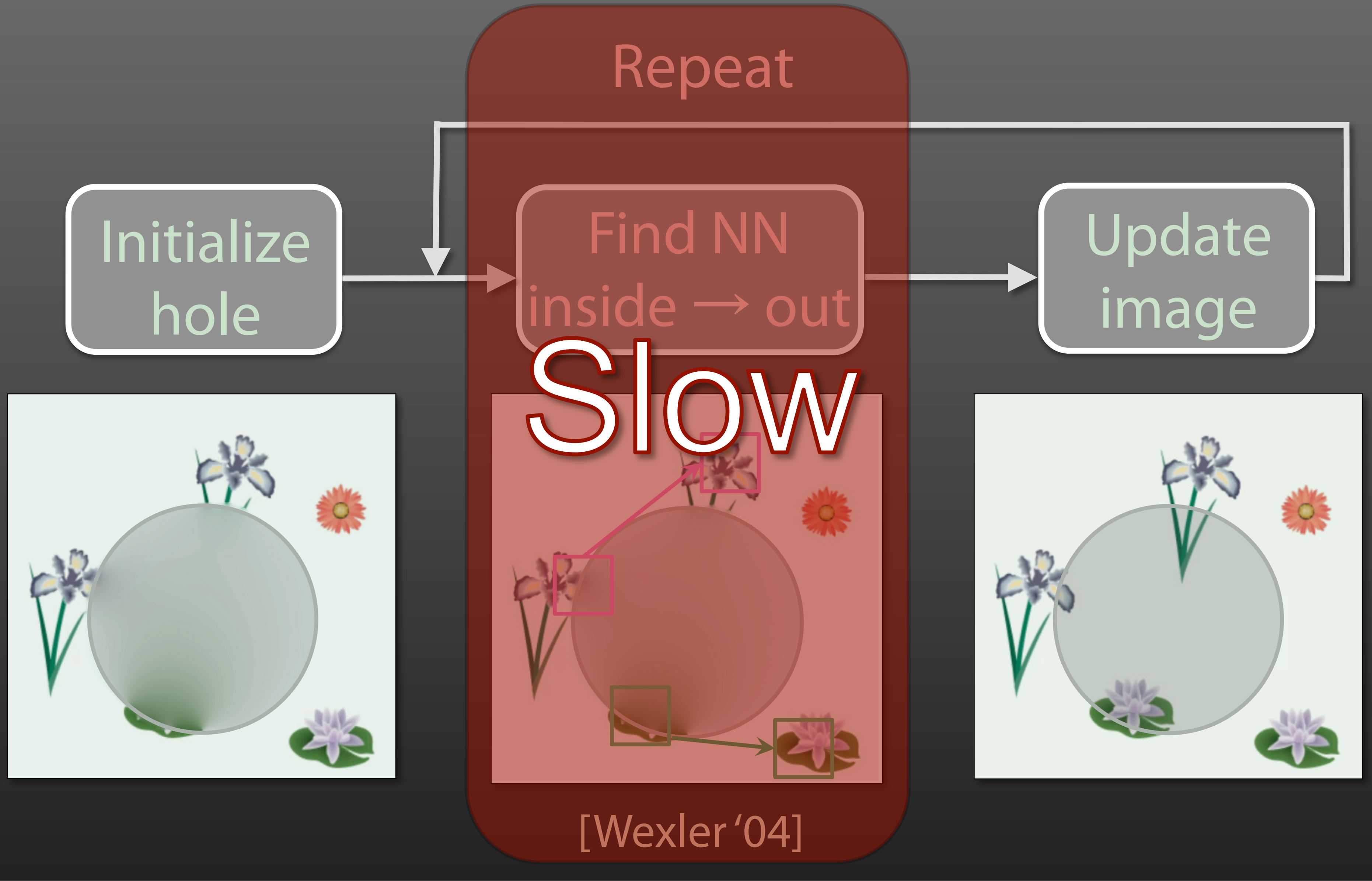
Texture Synthesis

[Kwatra '05]

Hole Filling

[Wexler '04]

Hole Filling



Repeat

Initialize hole

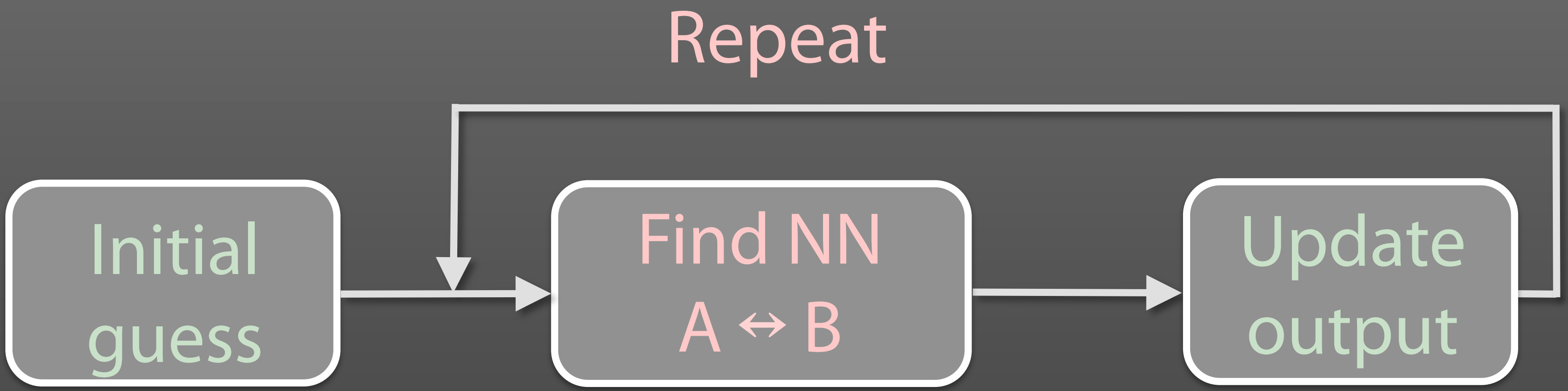
Find NN inside → out

Update image

Slow

[Wexler ‘04]

Retargeting / Reshuffling

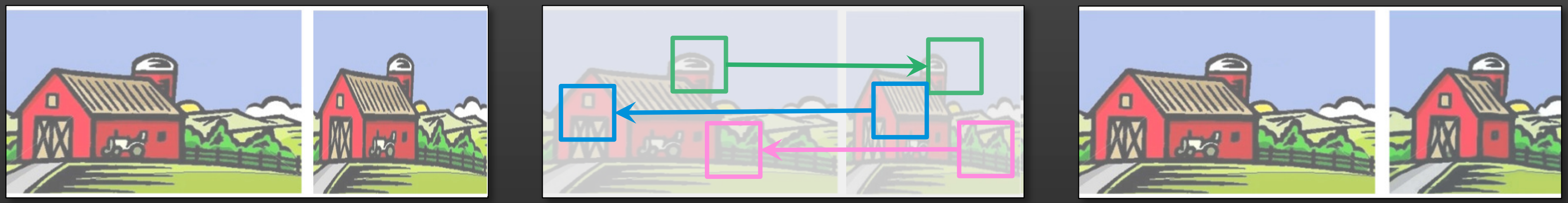


Repeat

Initial guess

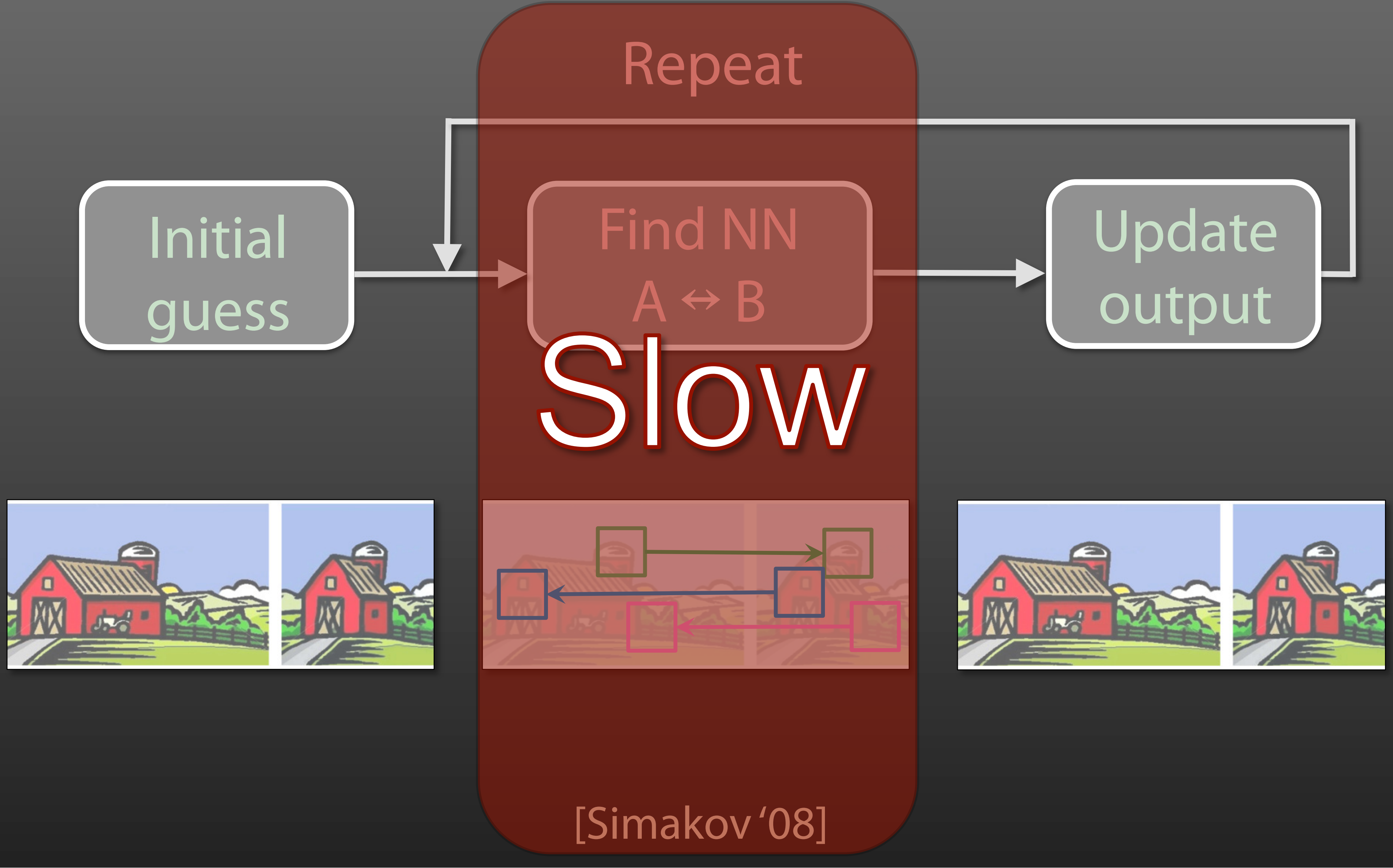
Find NN A ↔ B

Update output



[Simakov '08]

Retargeting / Reshuffling



Repeat

Initial guess

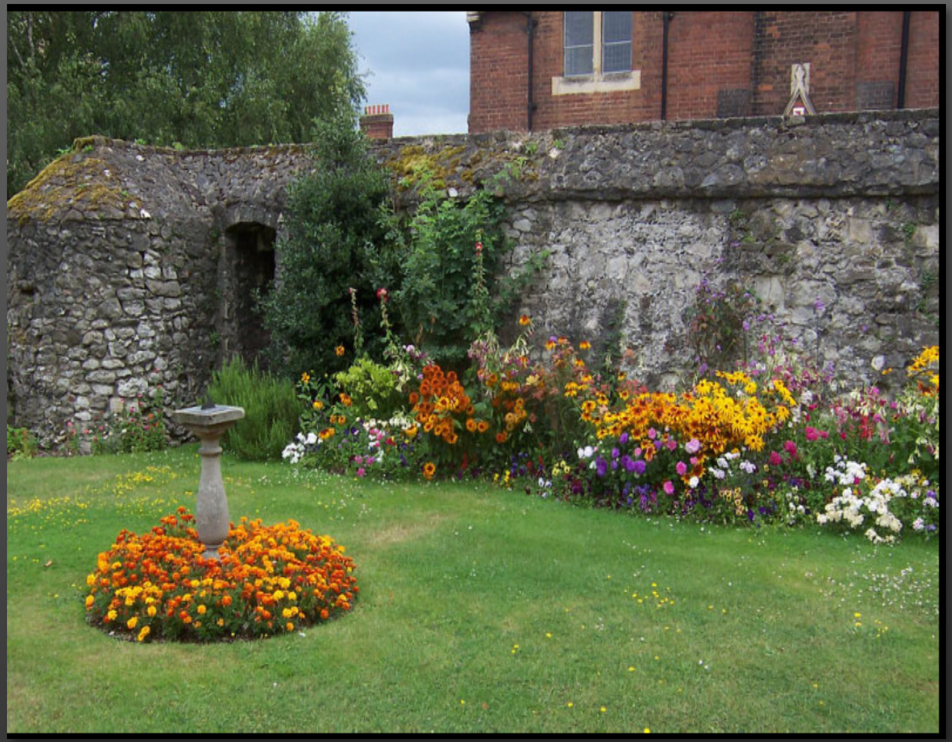
Find NN A ↔ B

Update output

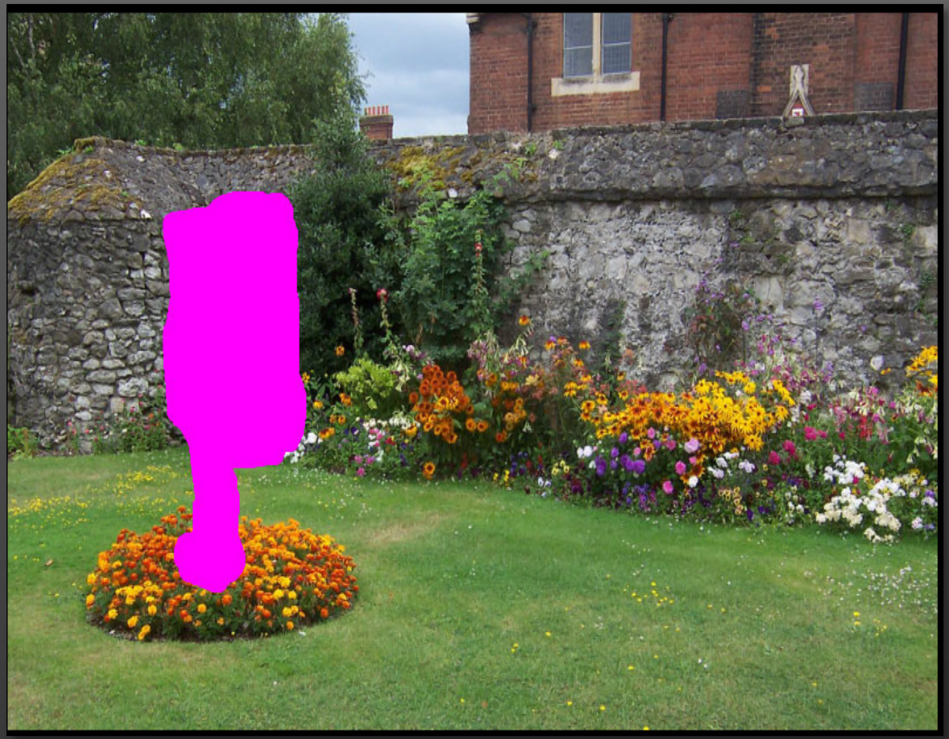
Slow

[Simakov '08]

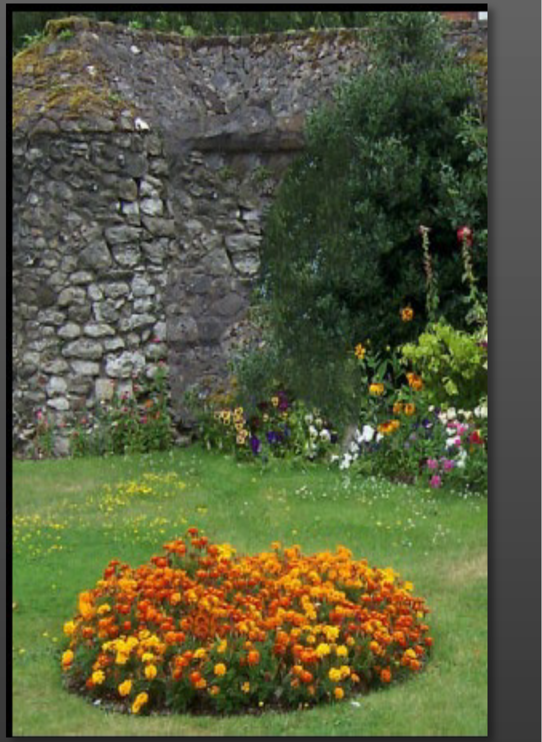
Hole Filling



Input



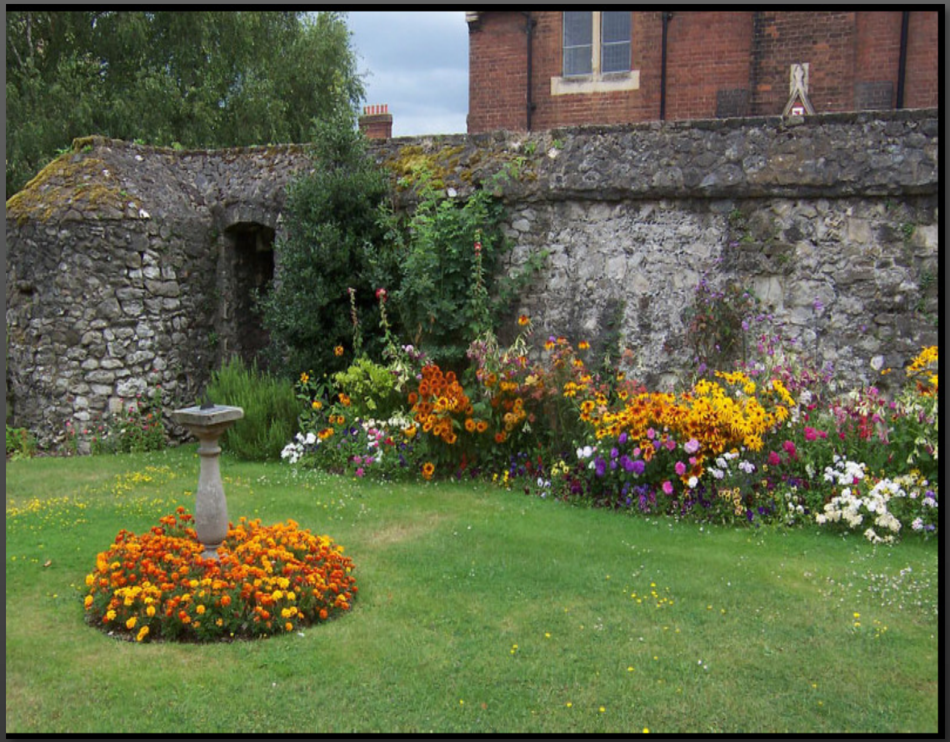
Hole



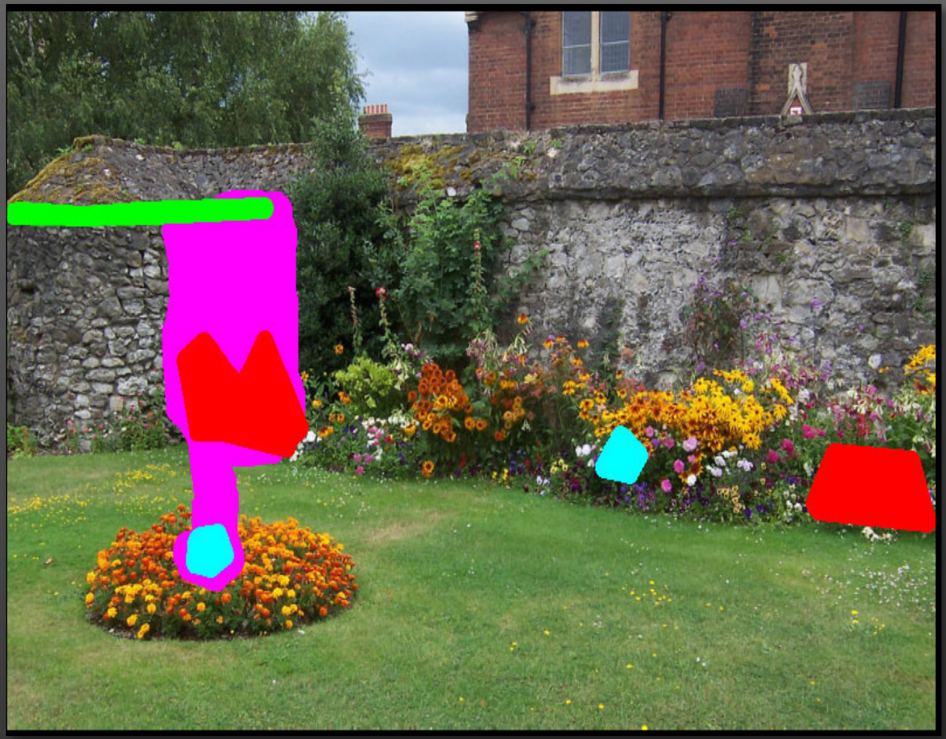
Output

(enlarged)

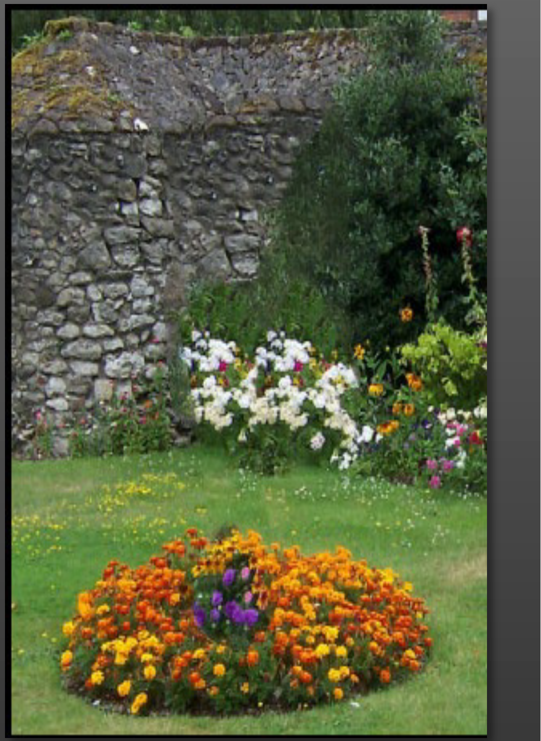
Hole Filling



Input

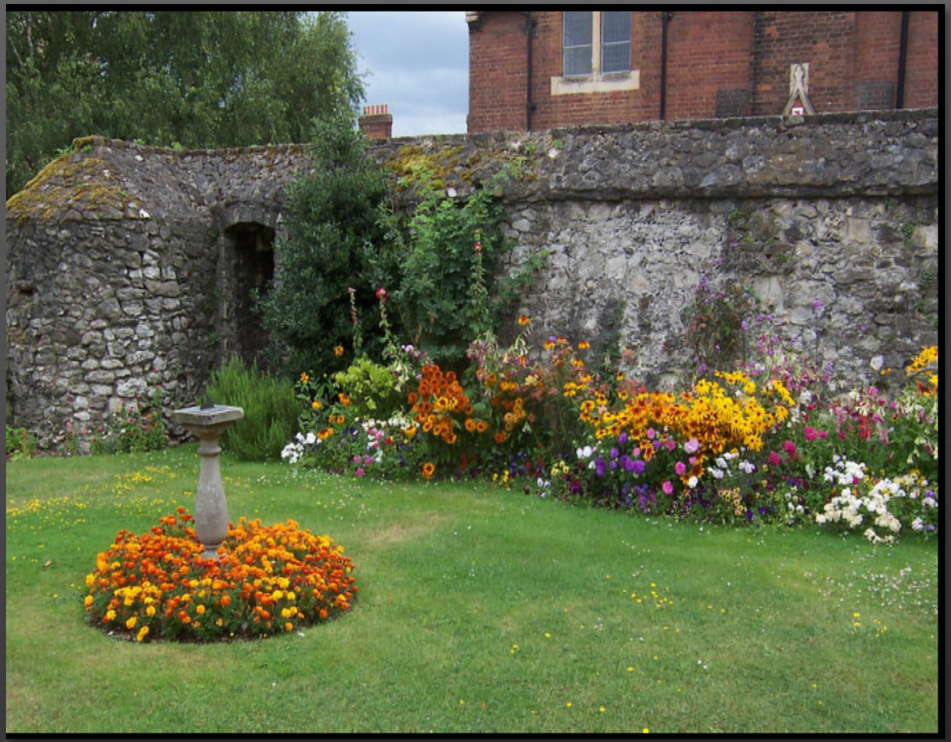


Hole

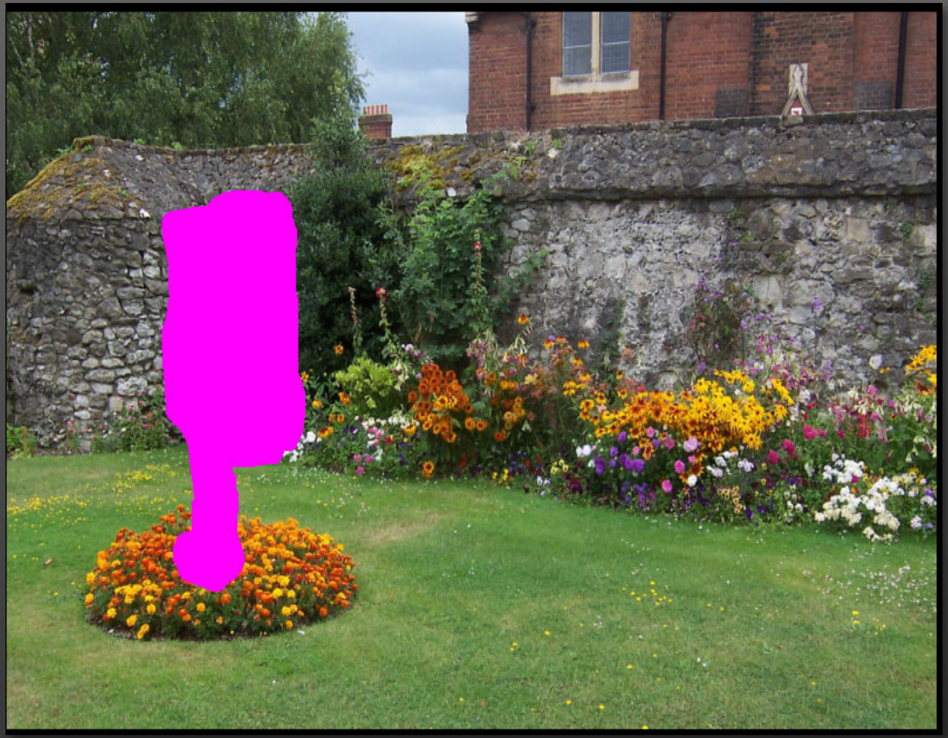


Output
(enlarged)

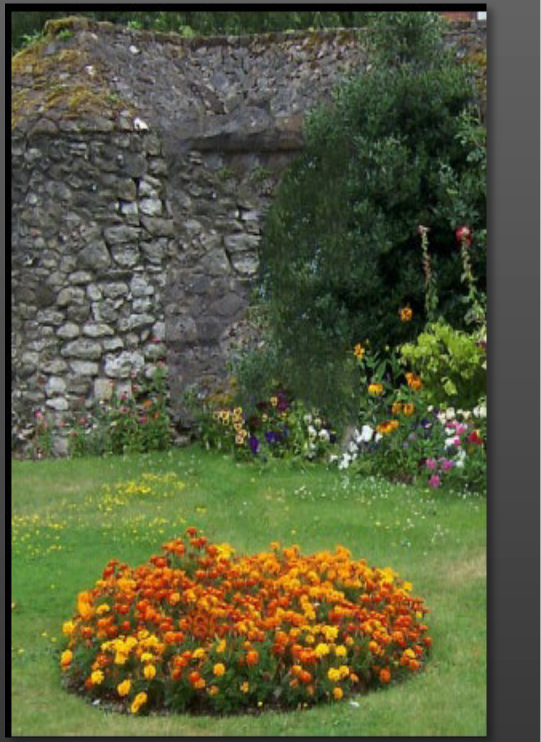
Hole Filling



Input

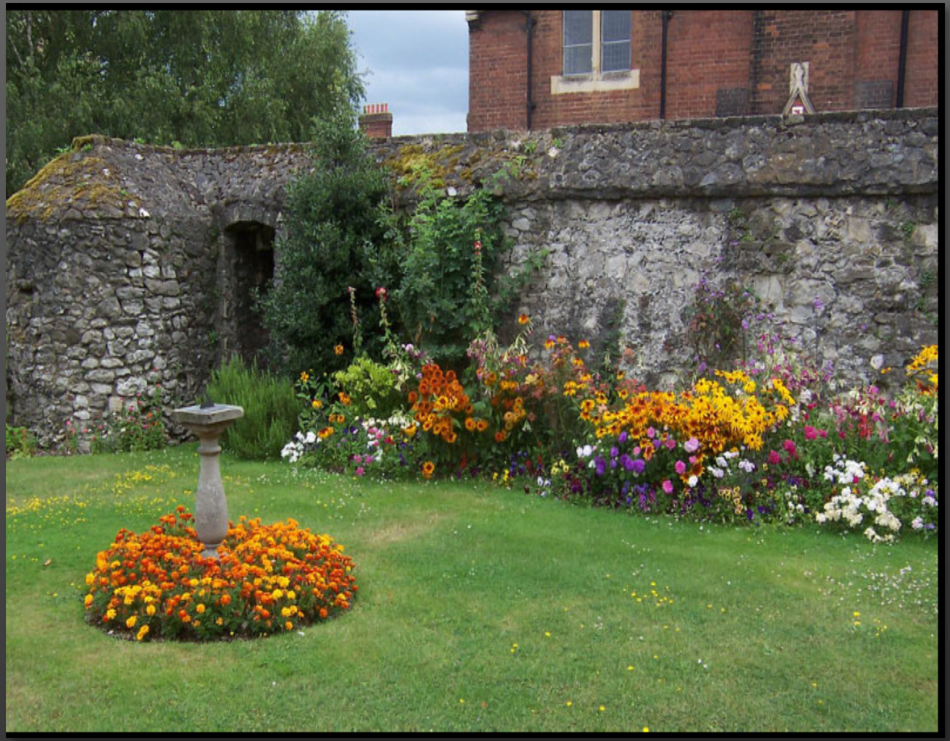


Hole

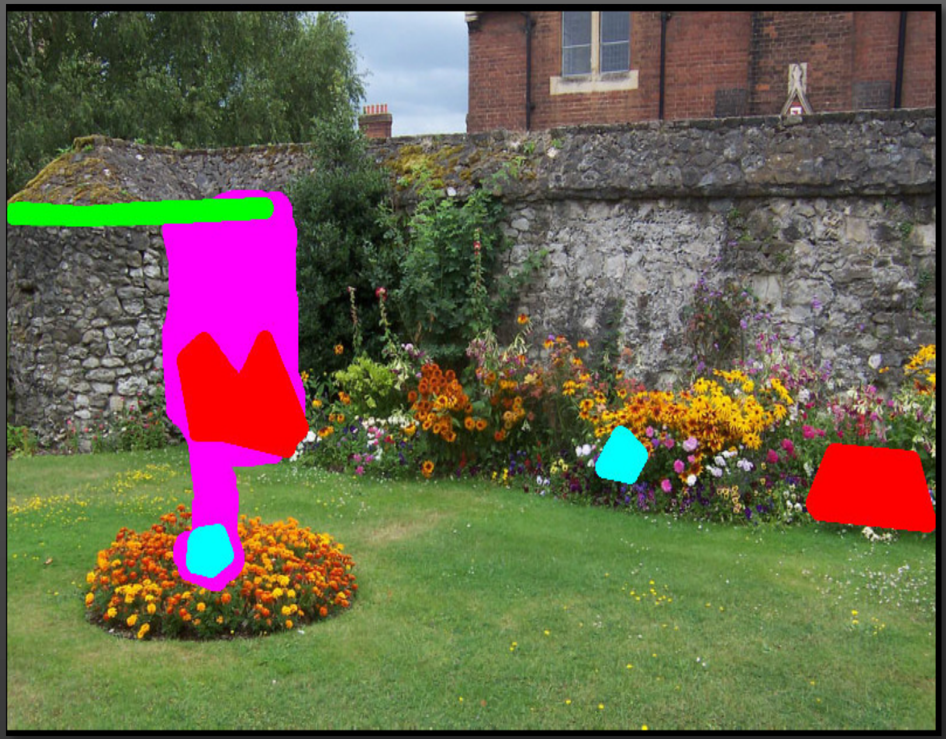


Output
(enlarged)

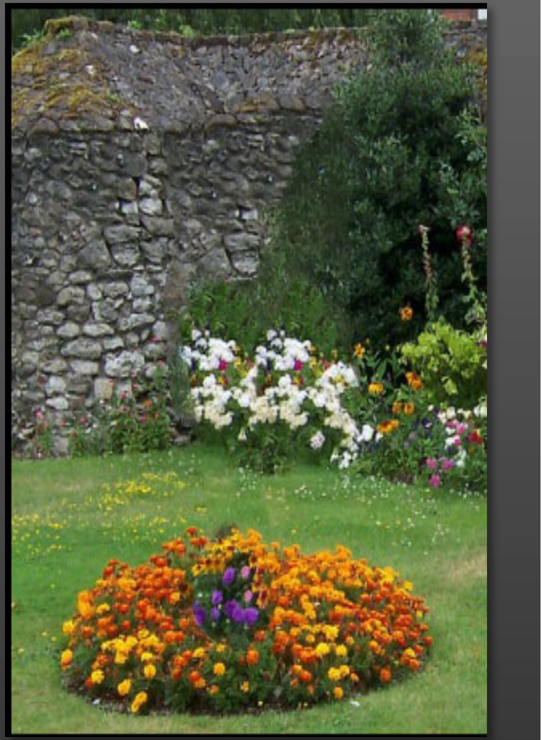
Hole Filling



Input

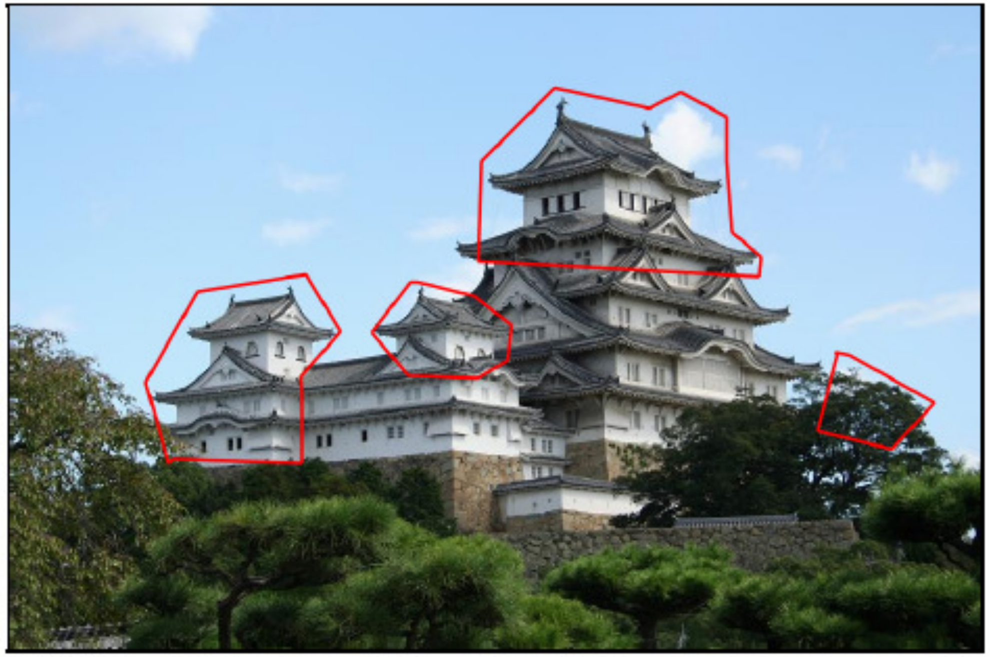


Hole

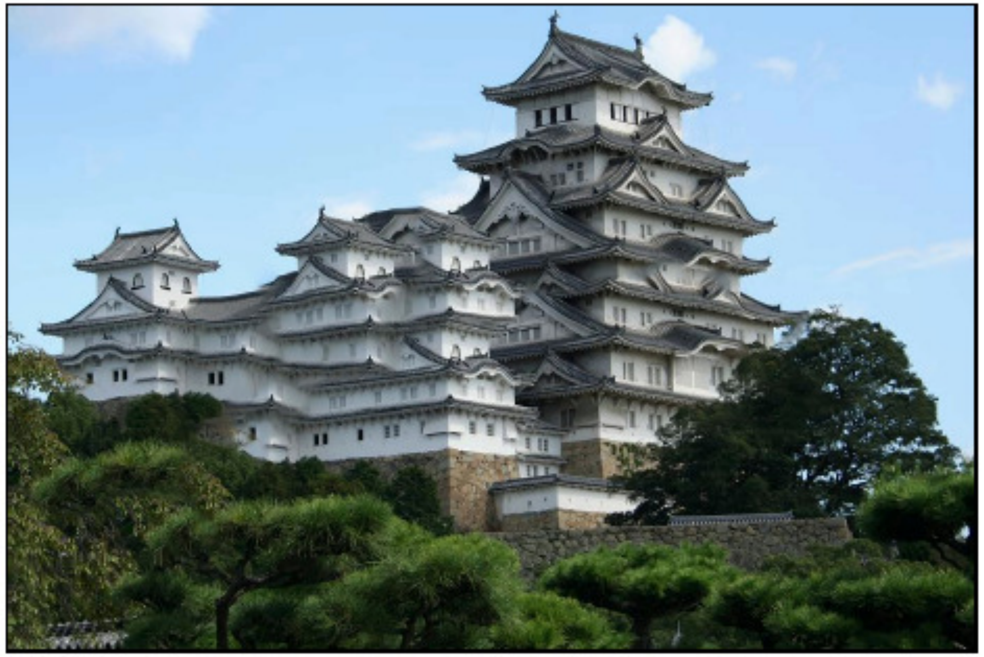


Output
(enlarged)

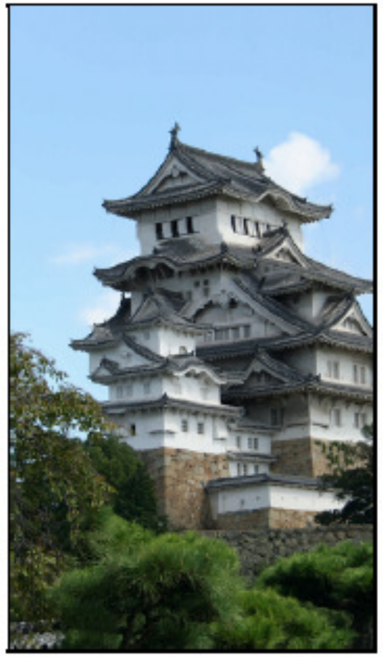
Reshuffling



Input

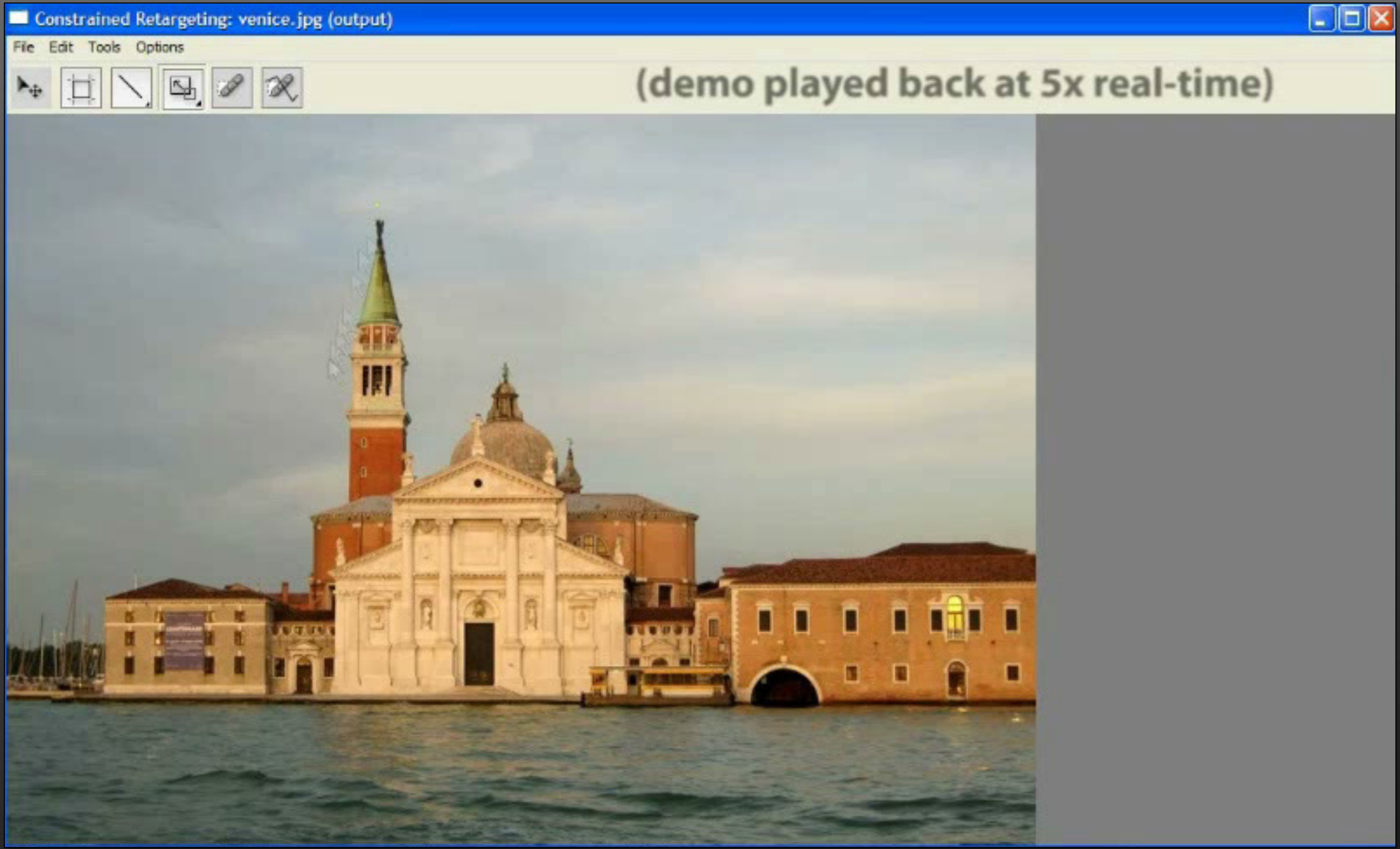


Enlarged



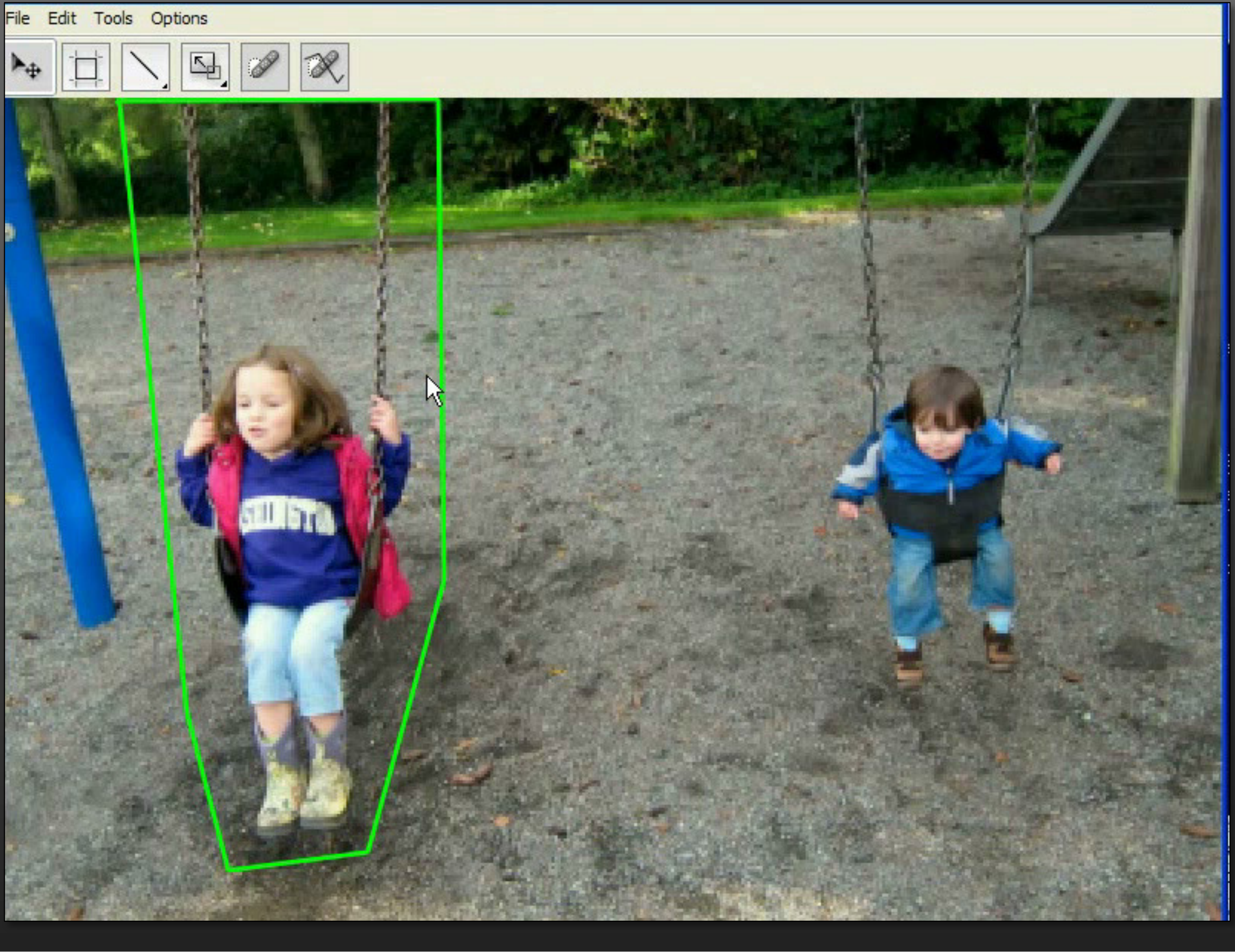
Reduced

Reshuffling



(demo played back at 5x real-time)

Example User Interaction



Overview

- Fast matching in images: the PatchMatch algorithm
  - Randomized algorithm
  - Applications: Image editing and video tapestries
- Matching under lighting and geometric variations
  - Dealing with higher dimensional spaces
  - Applications: computer vision, stitching and morphing, …
- Impact
- Future perspectives

Video Tapestries: Inspiration

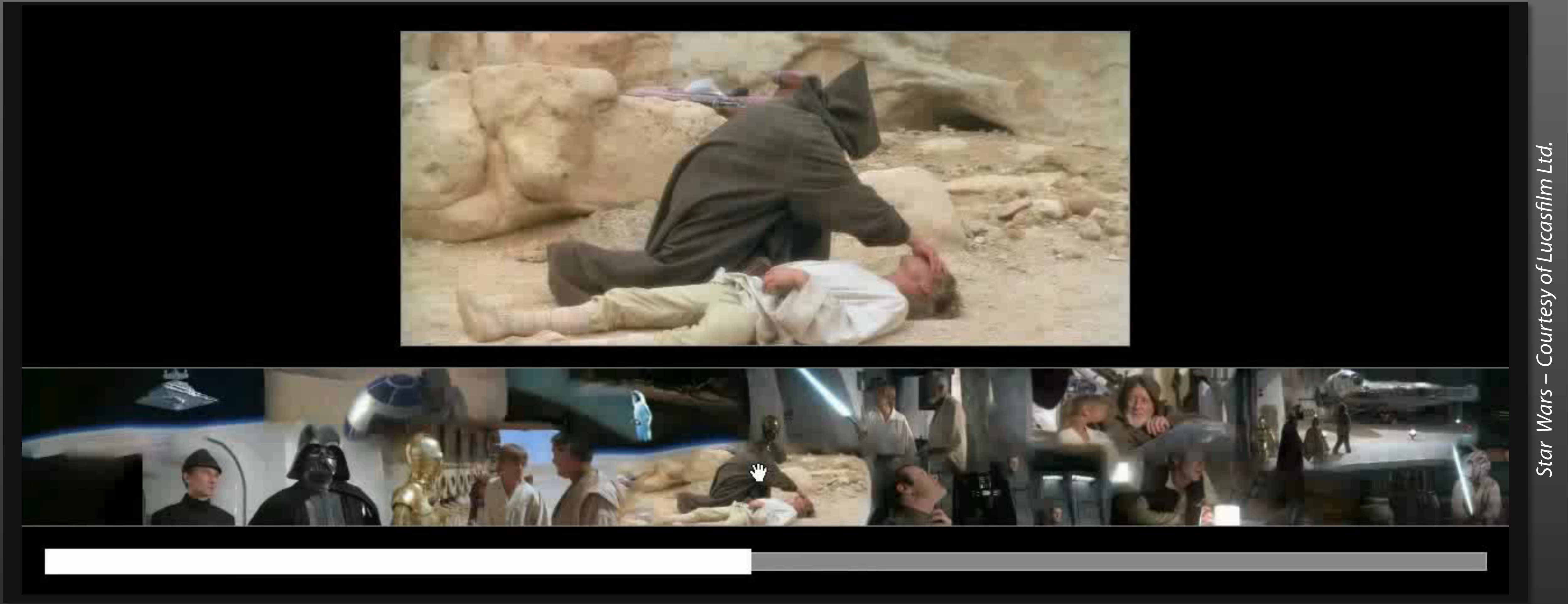
EDWARD REX: VBI: HAROLD DVX: ANGLORVM: ET SVI MILITES: EQVITANT: AD BOS HAM:

The Bayeux Tapestry

Trajan’s Column

“The Achievement of the Grail” – Burne-Jones

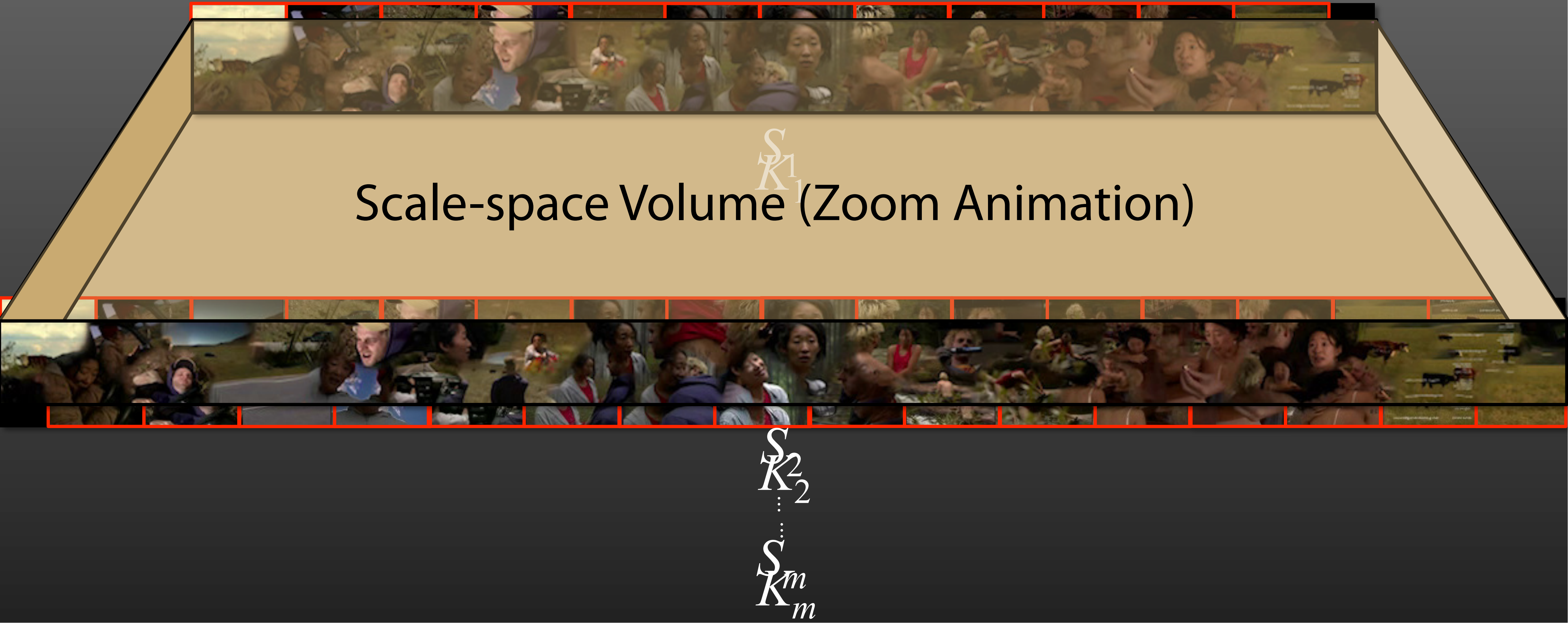
Video Tapestries



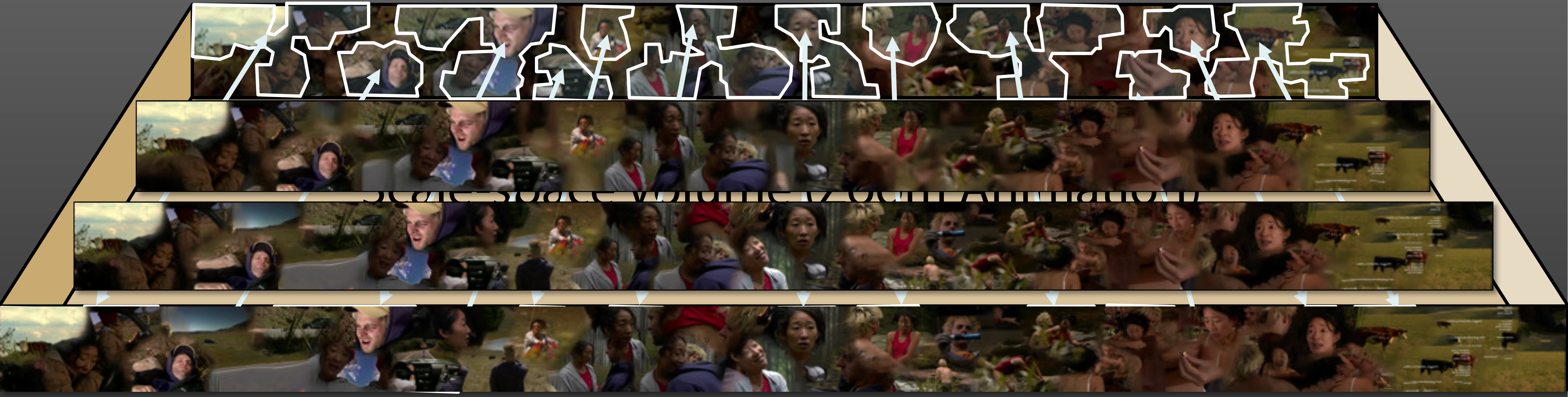
Star Wars – Courtesy of Lucasfilm Ltd.

[Barnes et al, Video Tapestries with Continuous Temporal Zoom, SIGGRAPH ‘10]

Multi-scale Tapestries

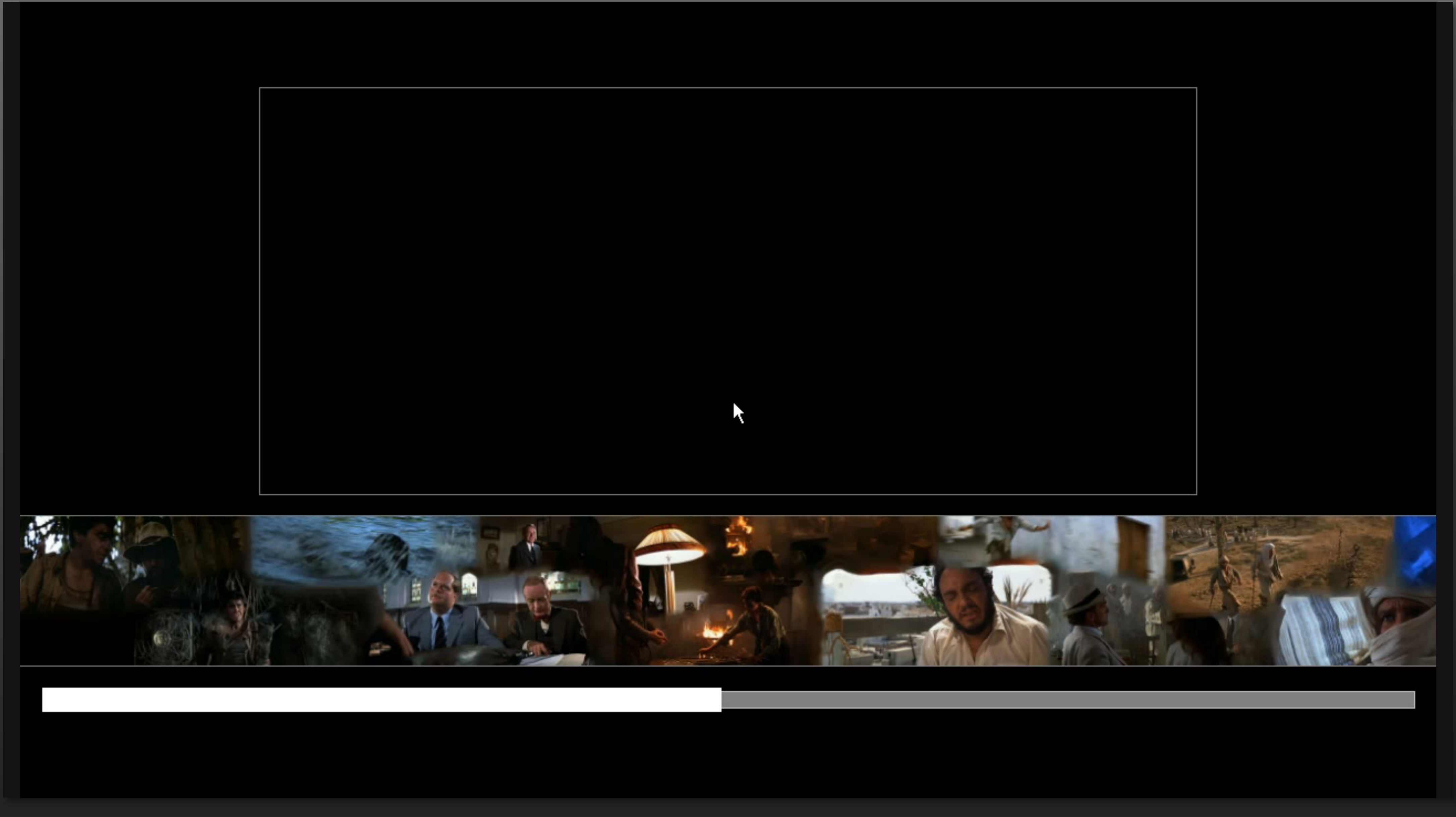


Continuous Zoom



Zooming reduces cognitive load

Further Examples

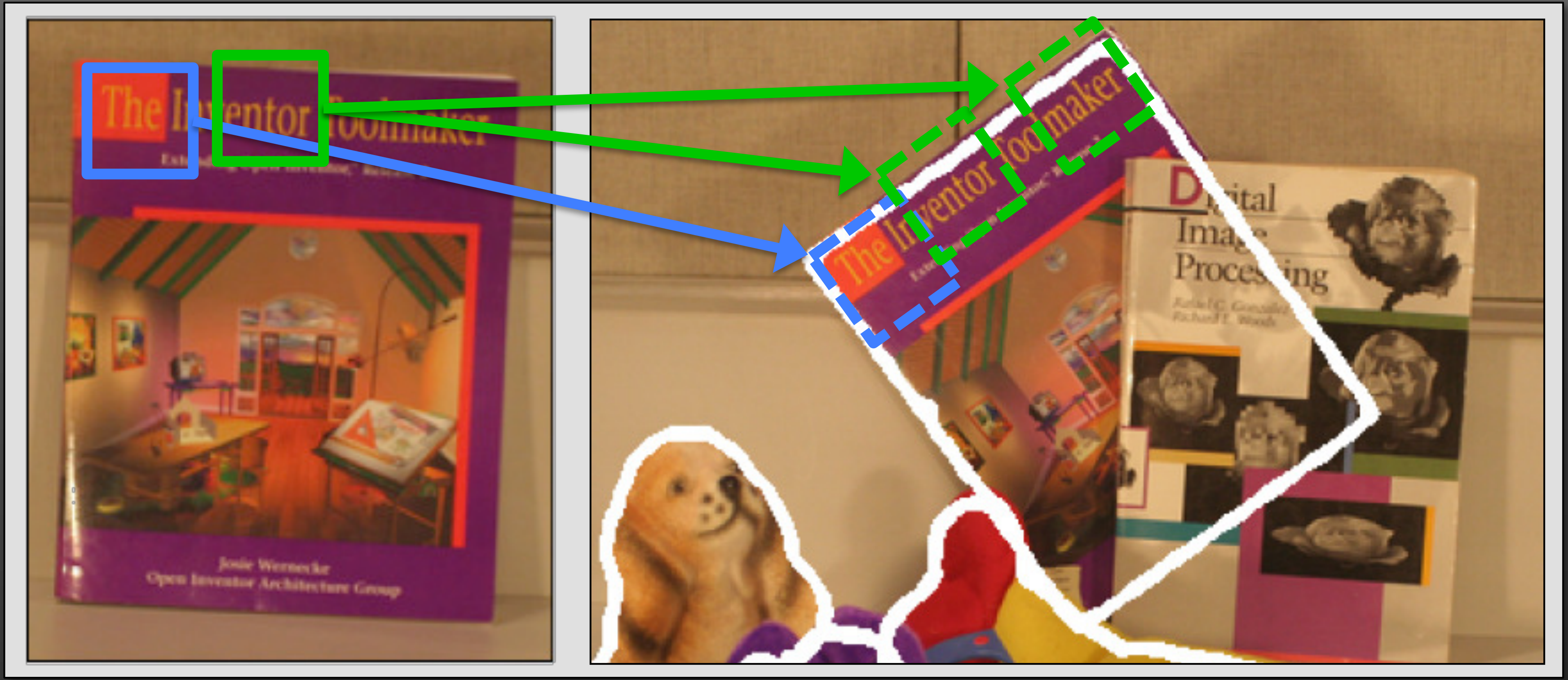


Raiders of the Lost Ark – Courtesy of Lucasfilm Ltd

Overview

- Fast matching in images: the PatchMatch algorithm
  - Randomized algorithm
  - Applications: Image editing and video tapestries
- Matching under lighting and geometric variations
  - Dealing with higher dimensional spaces
  - Applications: computer vision, stitching and morphing, …
- Impact
- Future perspectives

Generalized Matching: Motivation



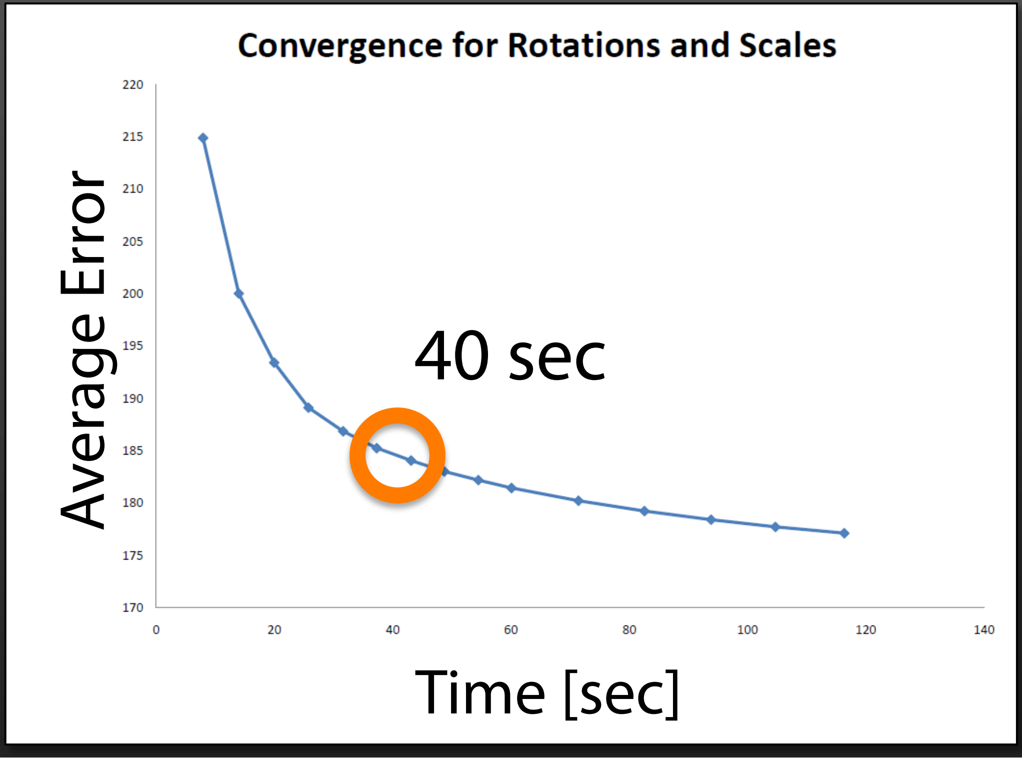
Rotations and Scales

k-Nearest Neighbors

[Barnes et al, The Generalized PatchMatch Correspondence Algorithm, ECCV '10]

Scales and Rotations

- Converges efficiently: extend random search window from (x, y) to (x, y, θ, s)



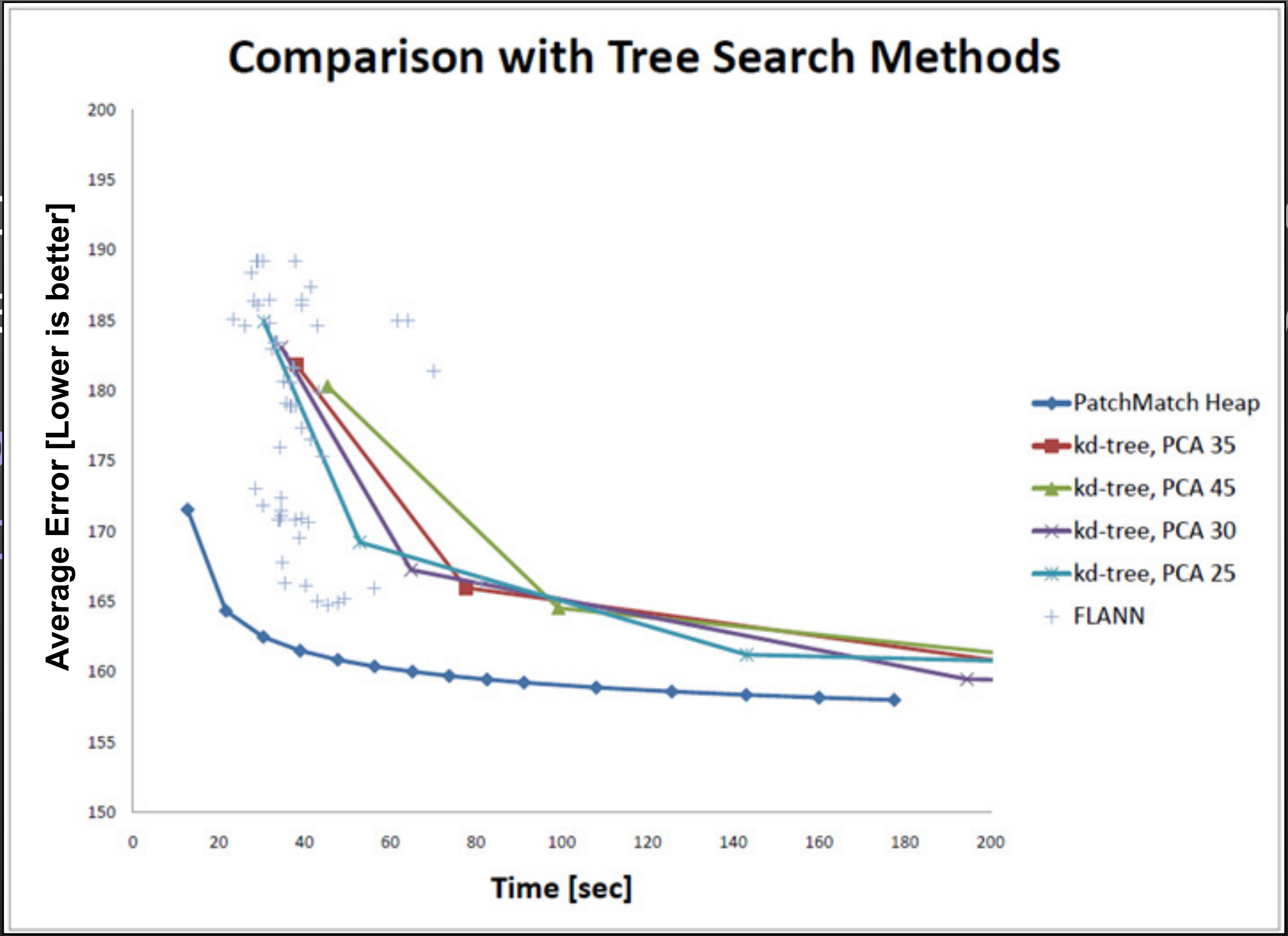
Arbitrary Descriptors

- Any distance function (pairwise comparisons), e.g:
  - SIFT, a popular computer vision descriptor
  - Patches invariant to lighting changes
- Convergence is output sensitive

k-Nearest Neighbors

- Benchm
- Store *k* c ce match on top. If ne w candidate.

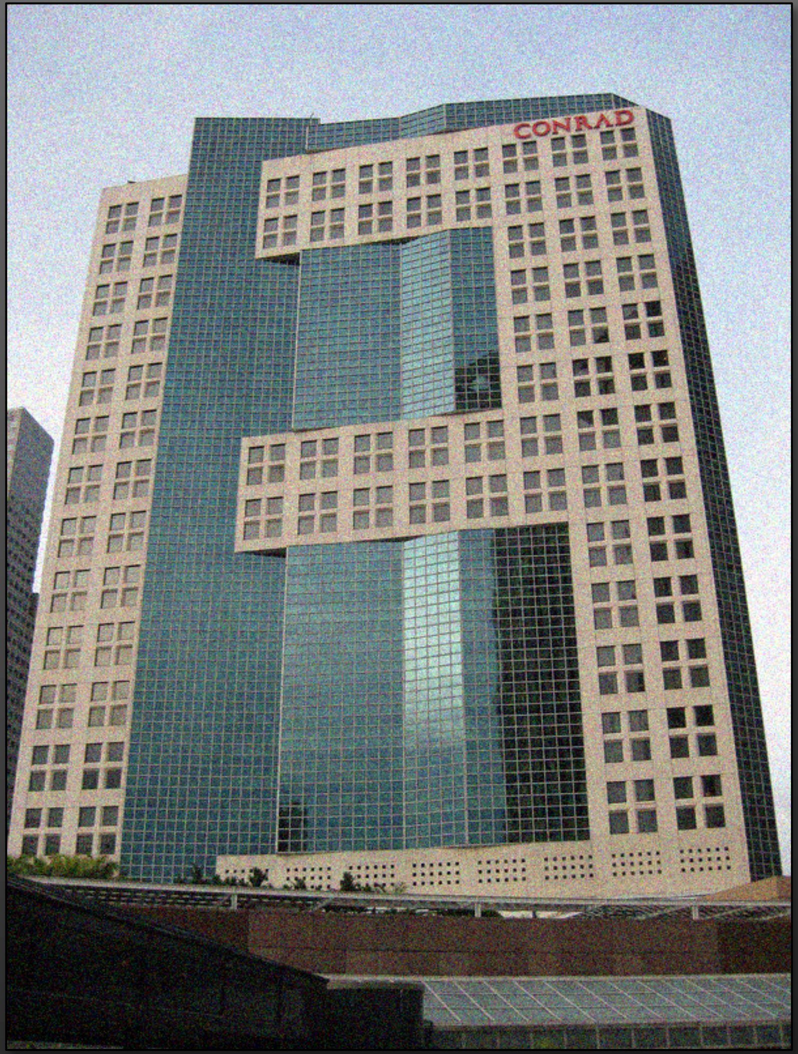
Comp candic



Overview

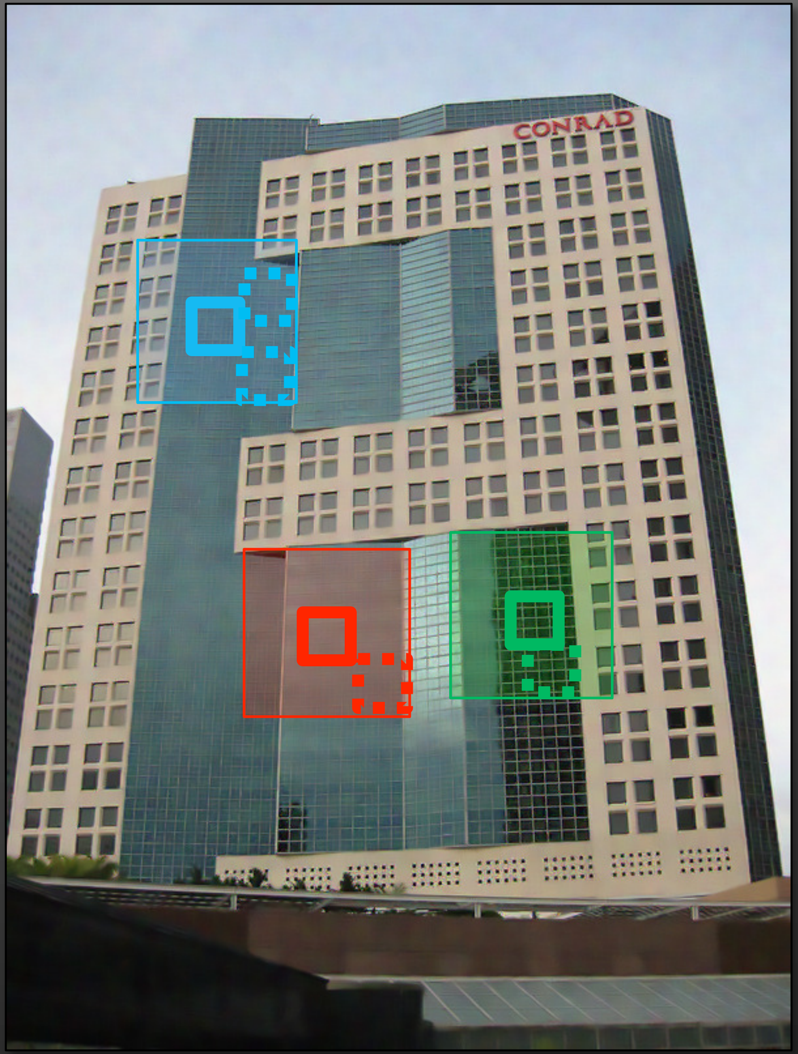
- Fast matching in images: the PatchMatch algorithm
  - Randomized algorithm
  - Applications: Image editing and video tapestries
- Matching under lighting and geometric variations
  - Dealing with higher dimensional spaces
  - Applications: computer vision, stitching and morphing, …
- Impact
- Future perspectives

Nonlocal Means Denoising



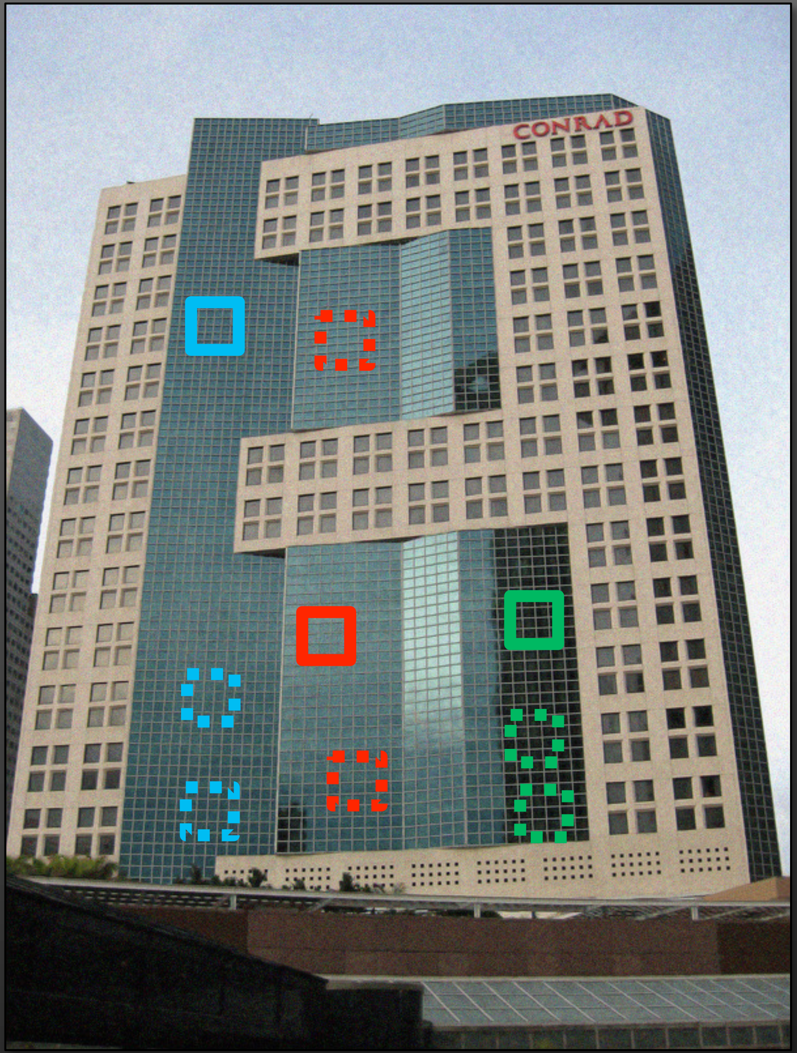
Noisy input

Nonlocal Means Denoising



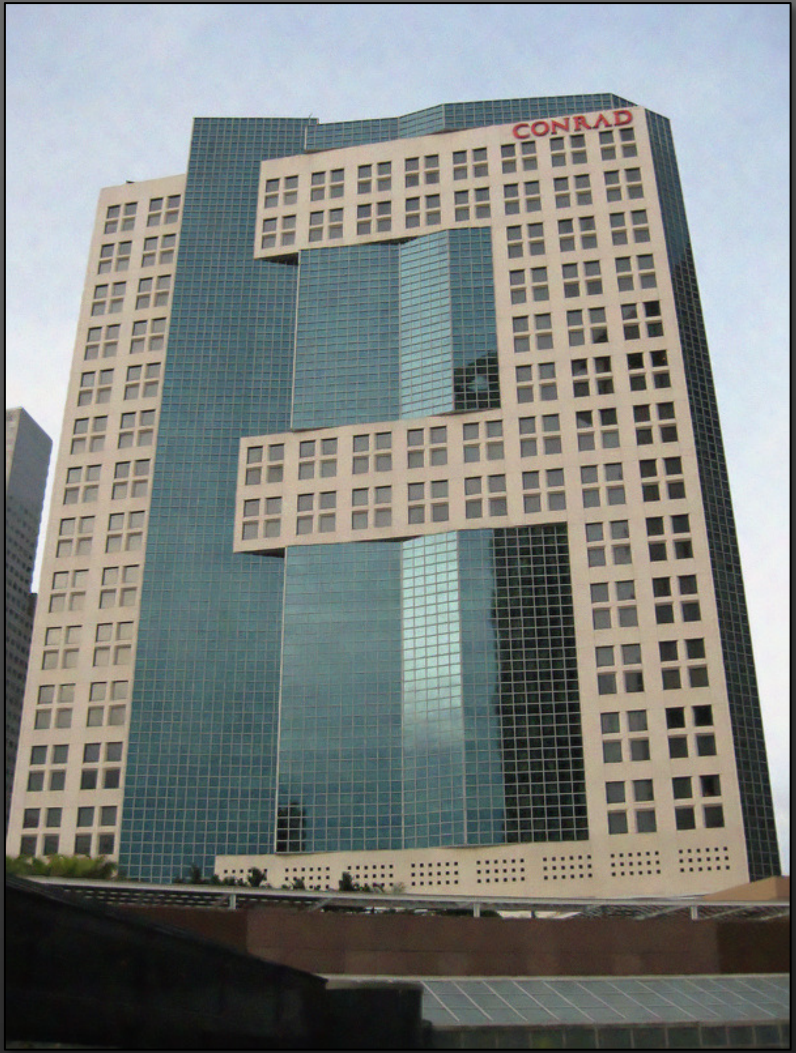
Buades et al (11x11 search window), PSNR 28.9 dB

Nonlocal Means Denoising



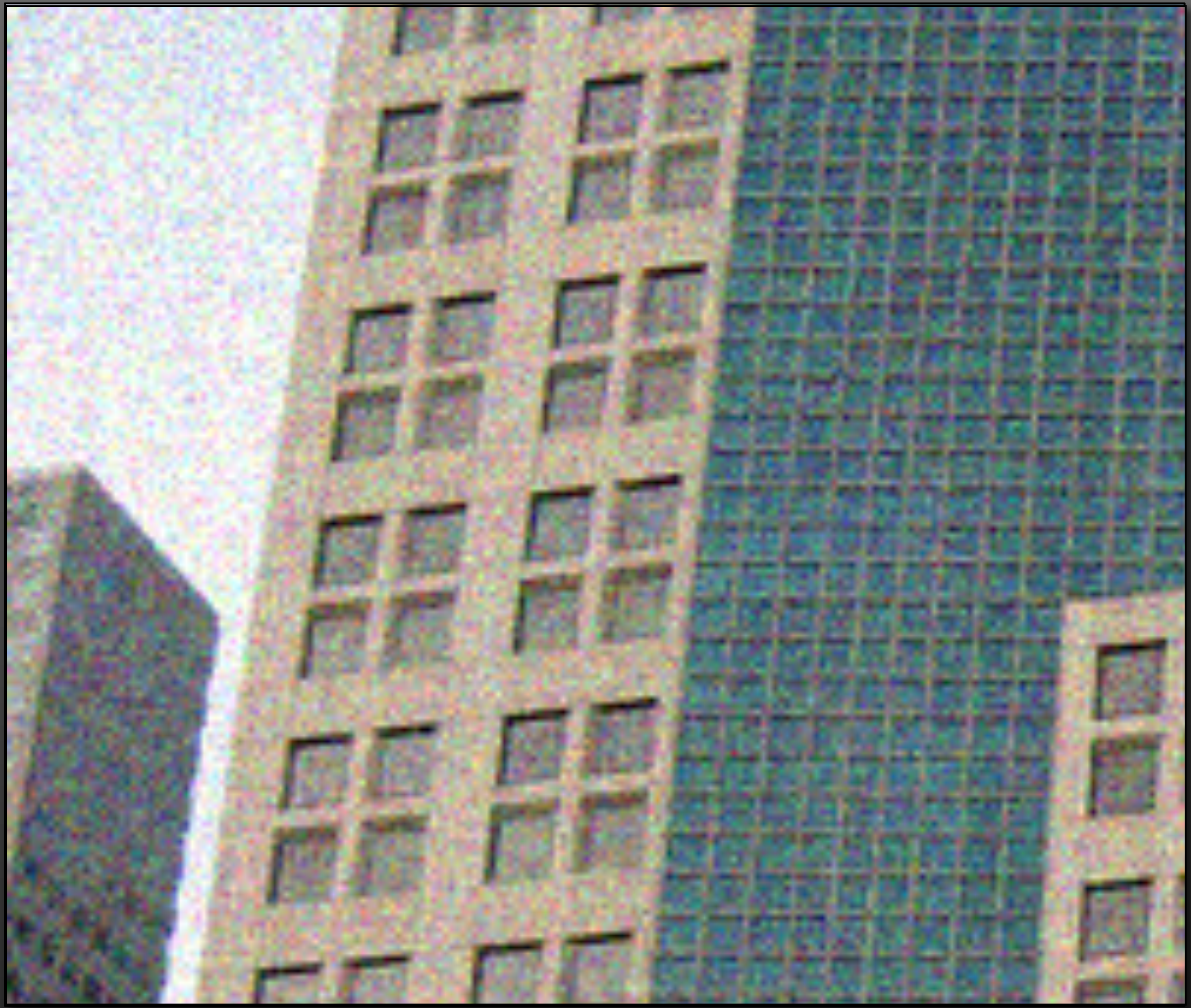
Our *k*NN search (*k*=32), PSNR 29.1 dB

Nonlocal Means Denoising



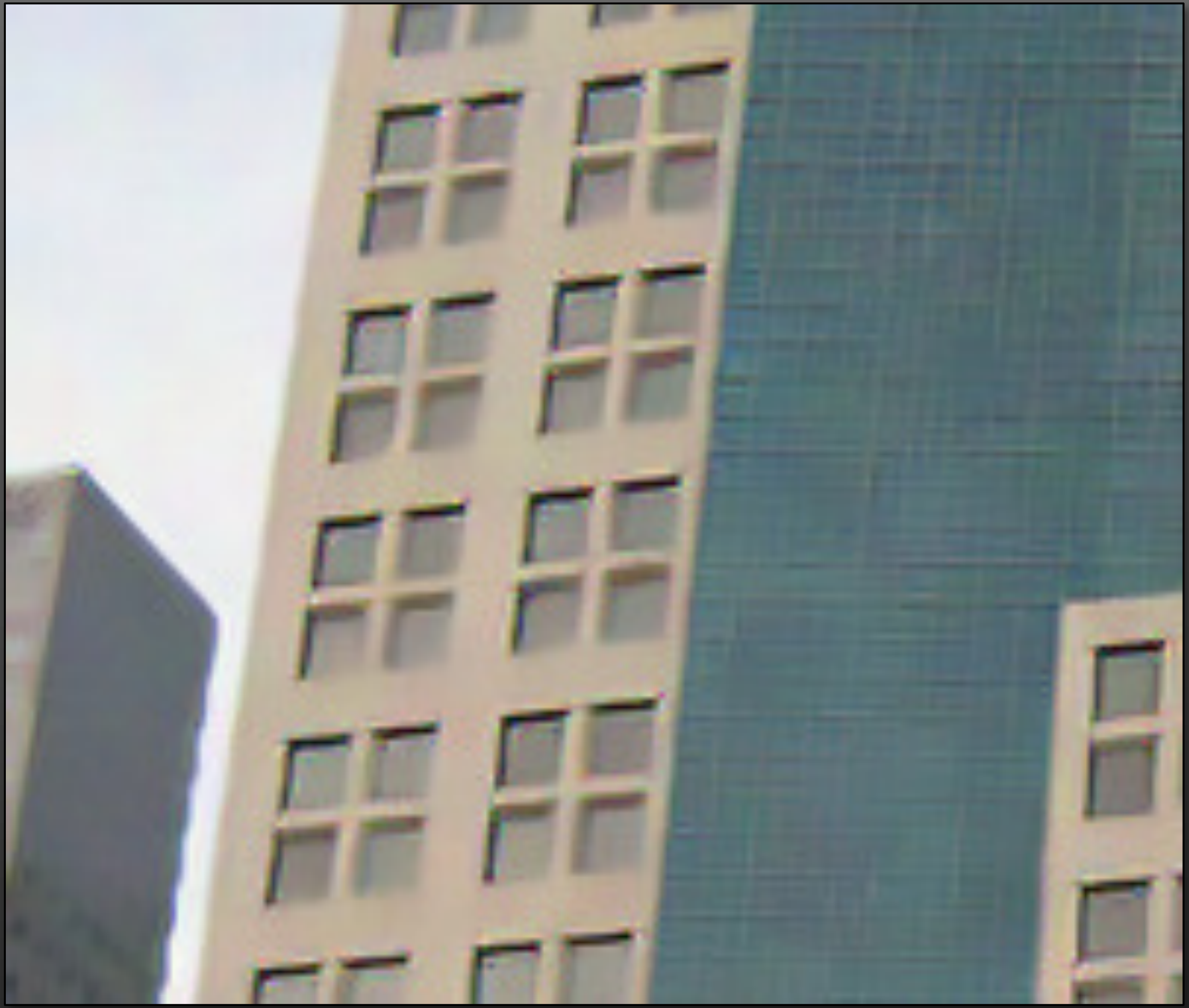
Combine both, PSNR 30.9 dB

Nonlocal Means Denoising



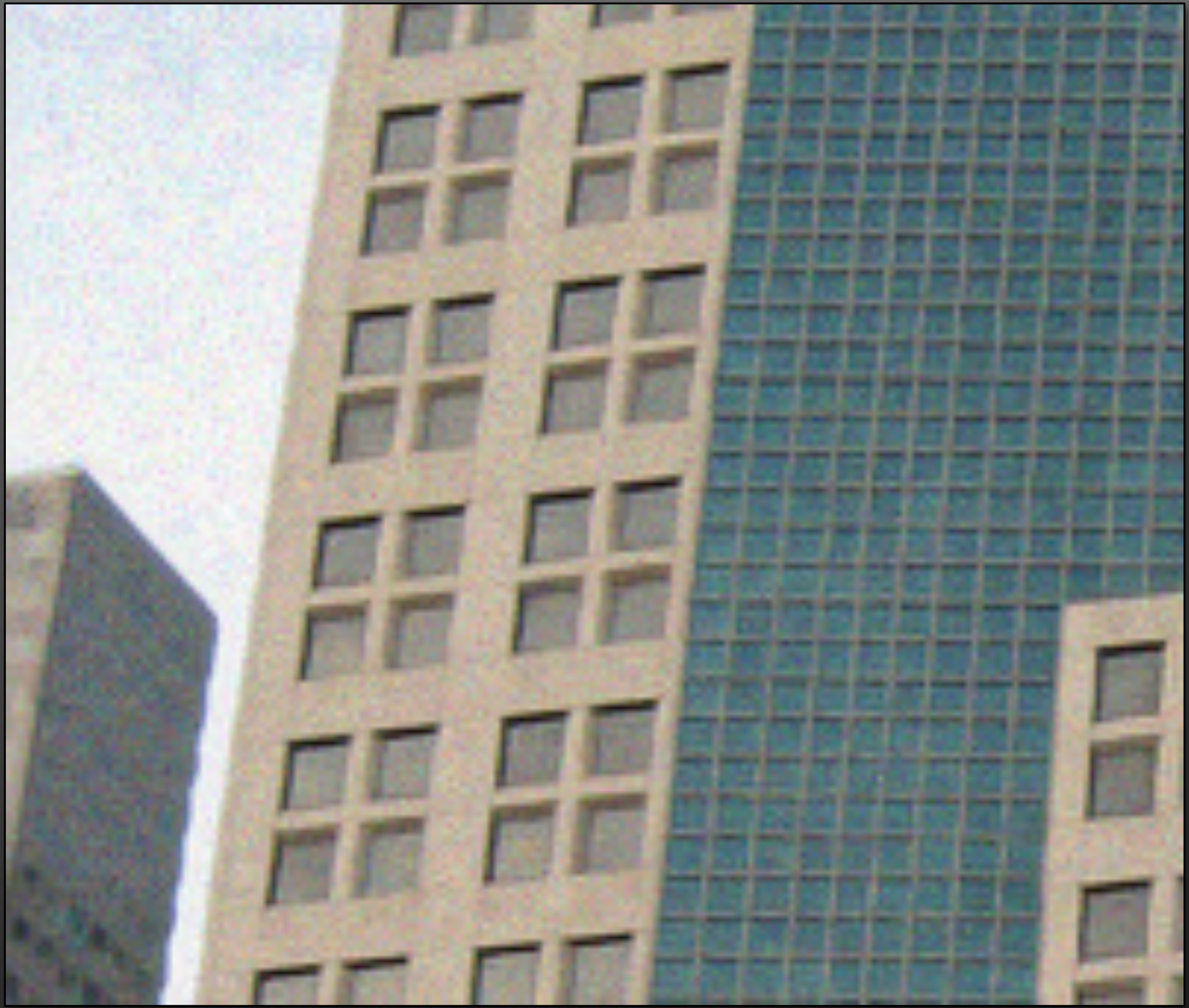
Noisy Input

Nonlocal Means Denoising



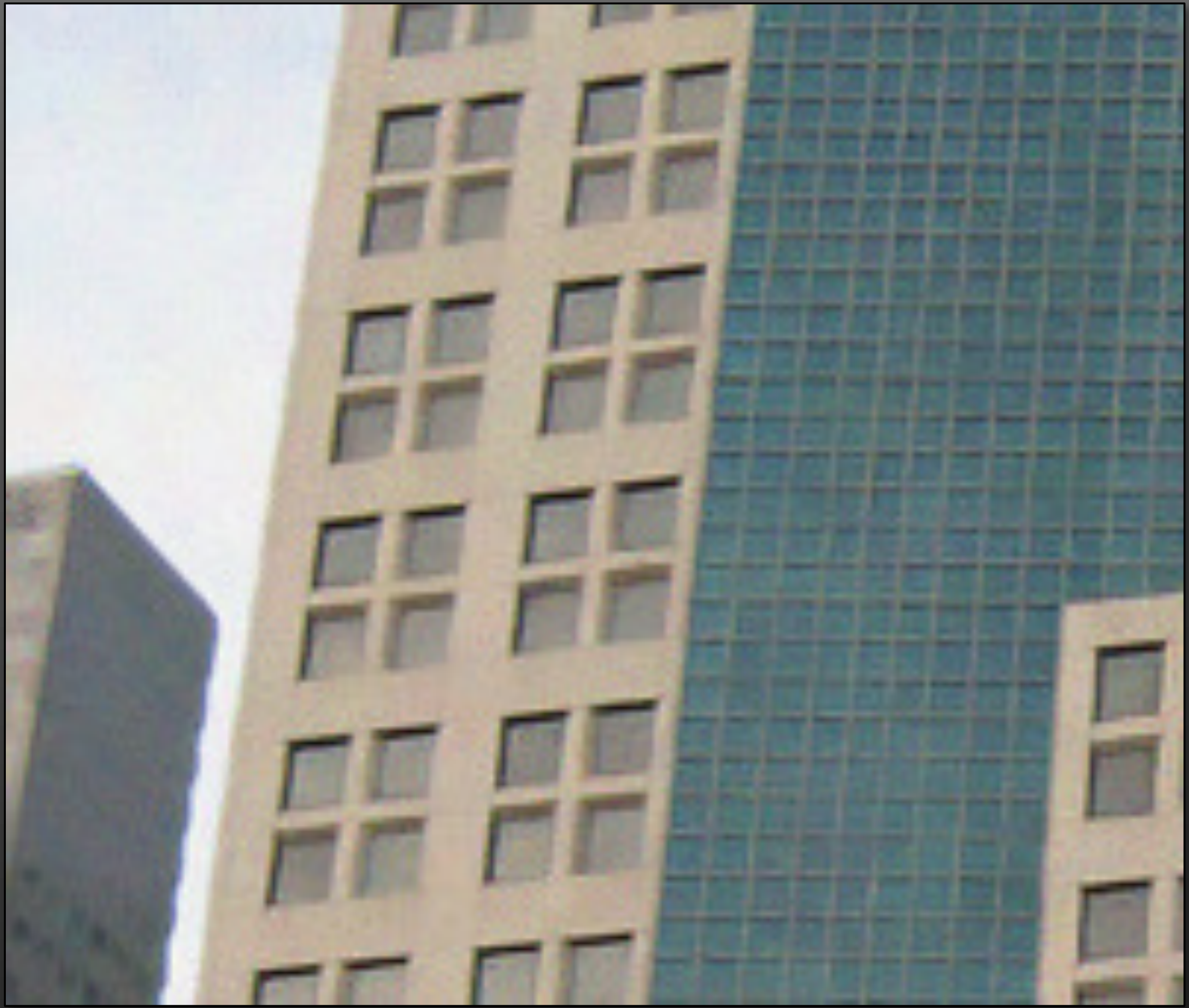
Local Search

Nonlocal Means Denoising



Our *k*NN (Global)

Nonlocal Means Denoising

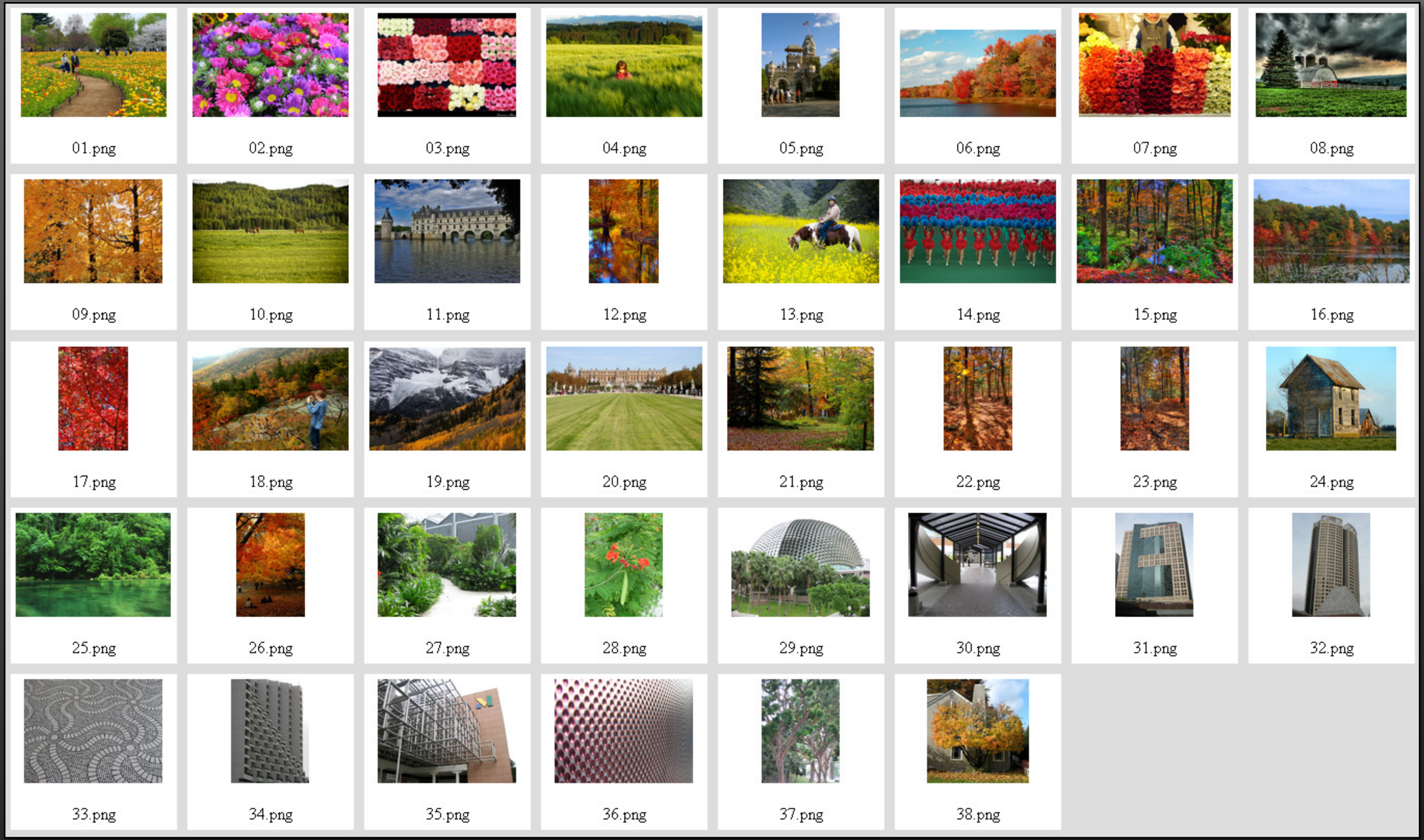


Combine both

Nonlocal Means Denoising

Ground truth

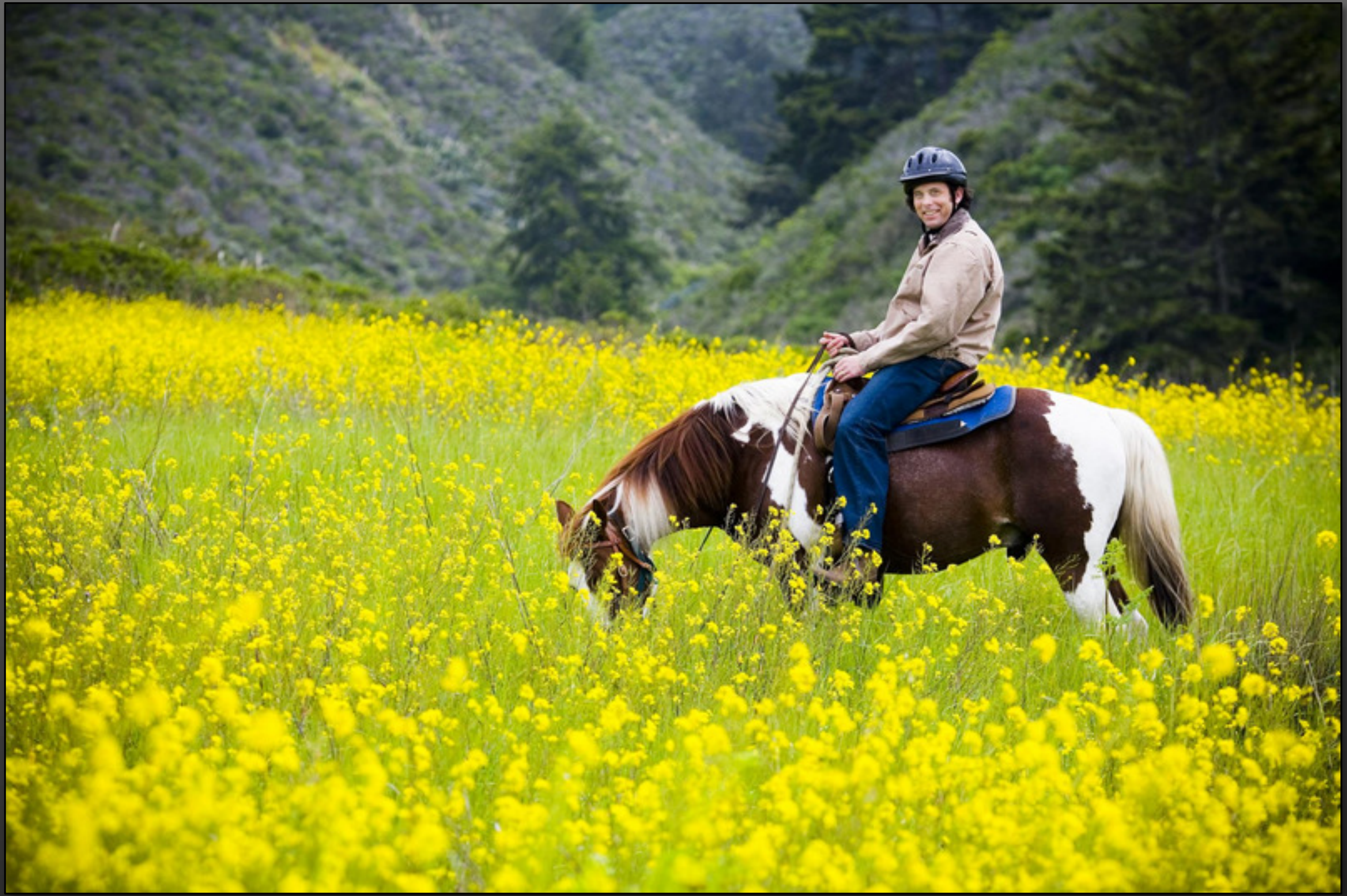
Nonlocal Means Denoising



Nonlocal Means Denoising

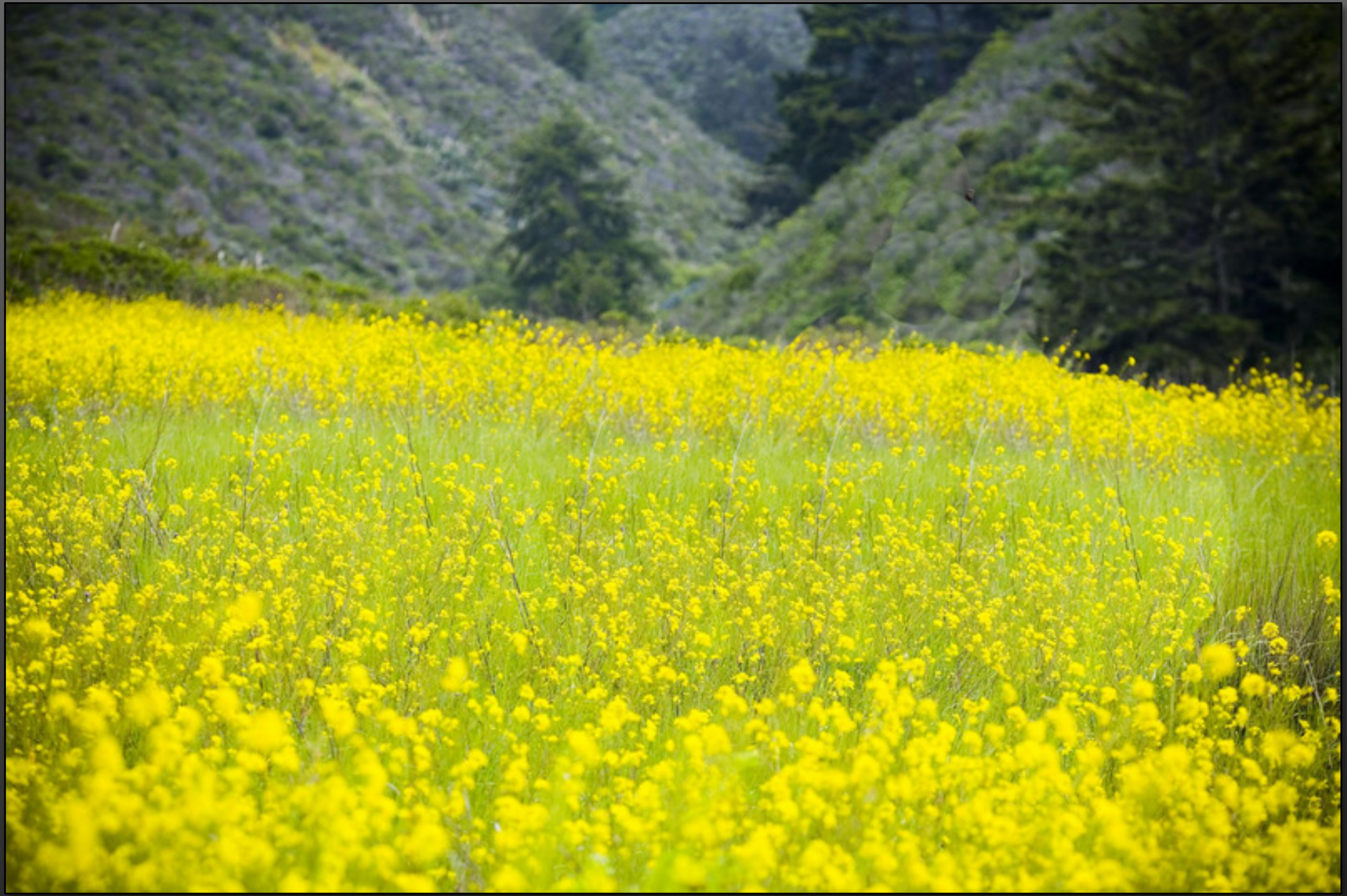
Local window search [Buades et al],	PSNR: 27.8 dB
Global search [our kNN],	PSNR: 27.4 dB
Combine local+global [our kNN],	PSNR: 28.4 dB
State of the art [BM3D],	PSNR: 29.9 dB

Detecting Forgeries



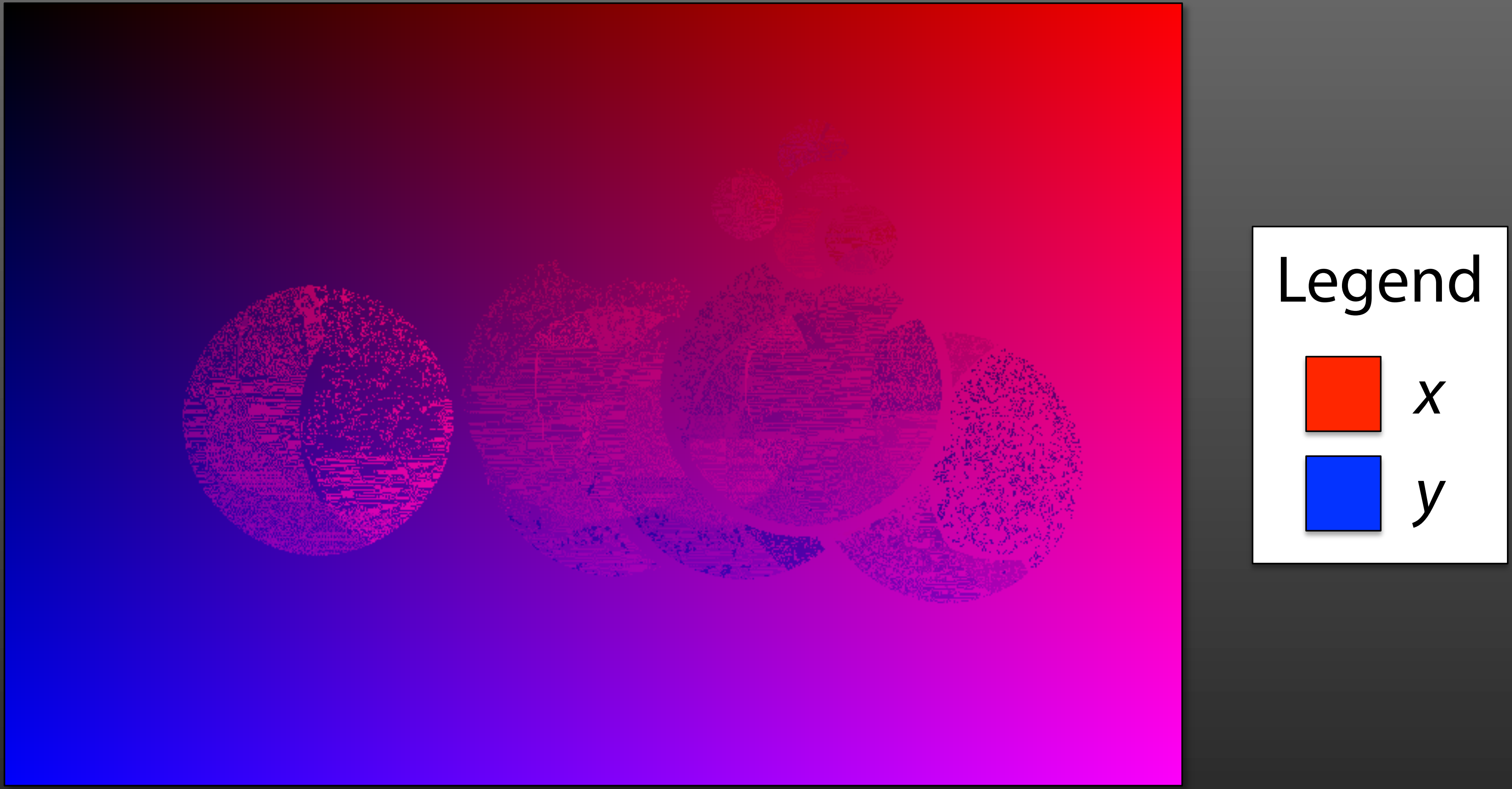
Original image (undoctored)

Detecting Forgeries



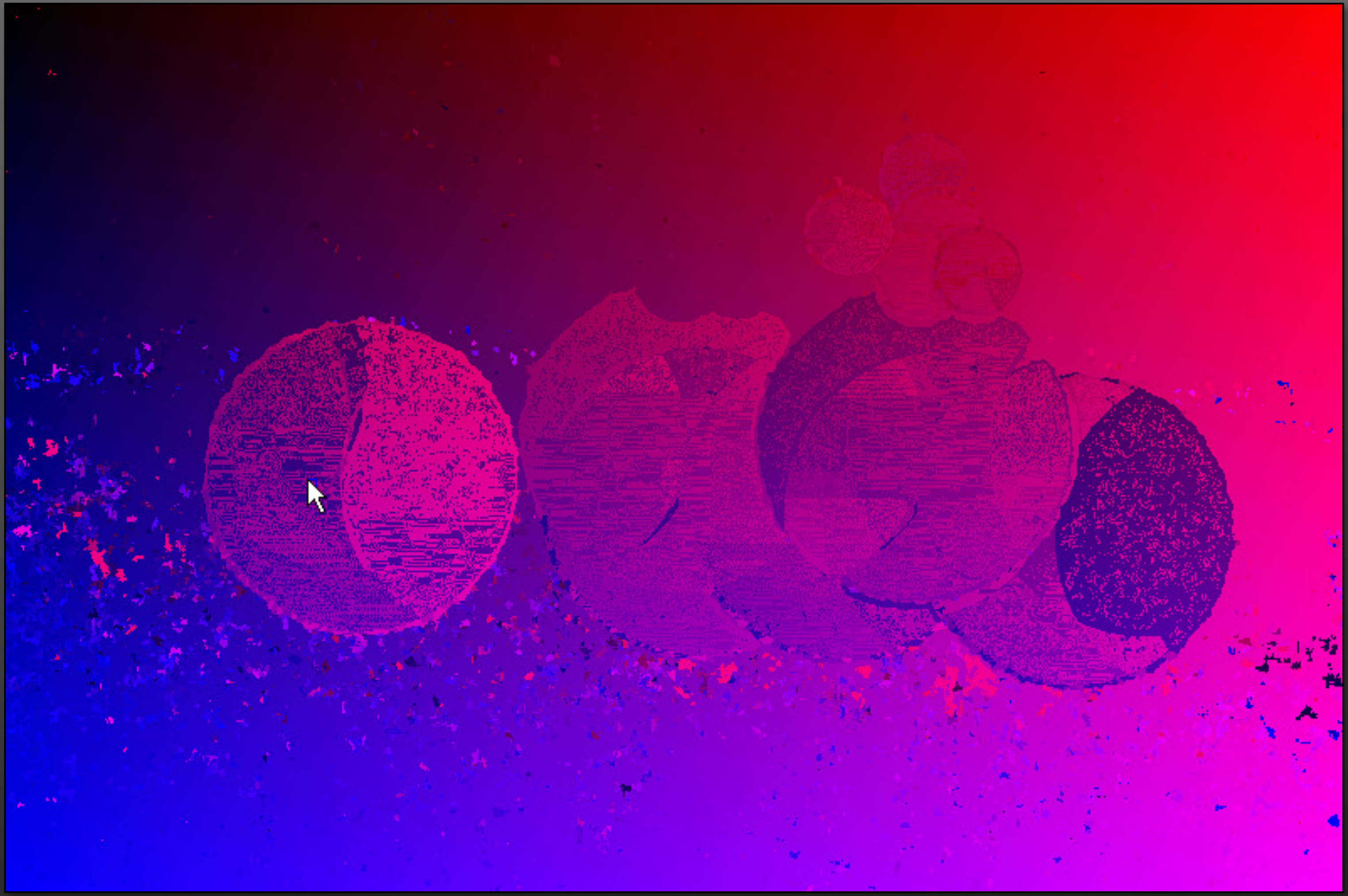
Doctored image (input)

Detecting Forgeries



*k*NN, neighbor 1

Detecting Forgeries



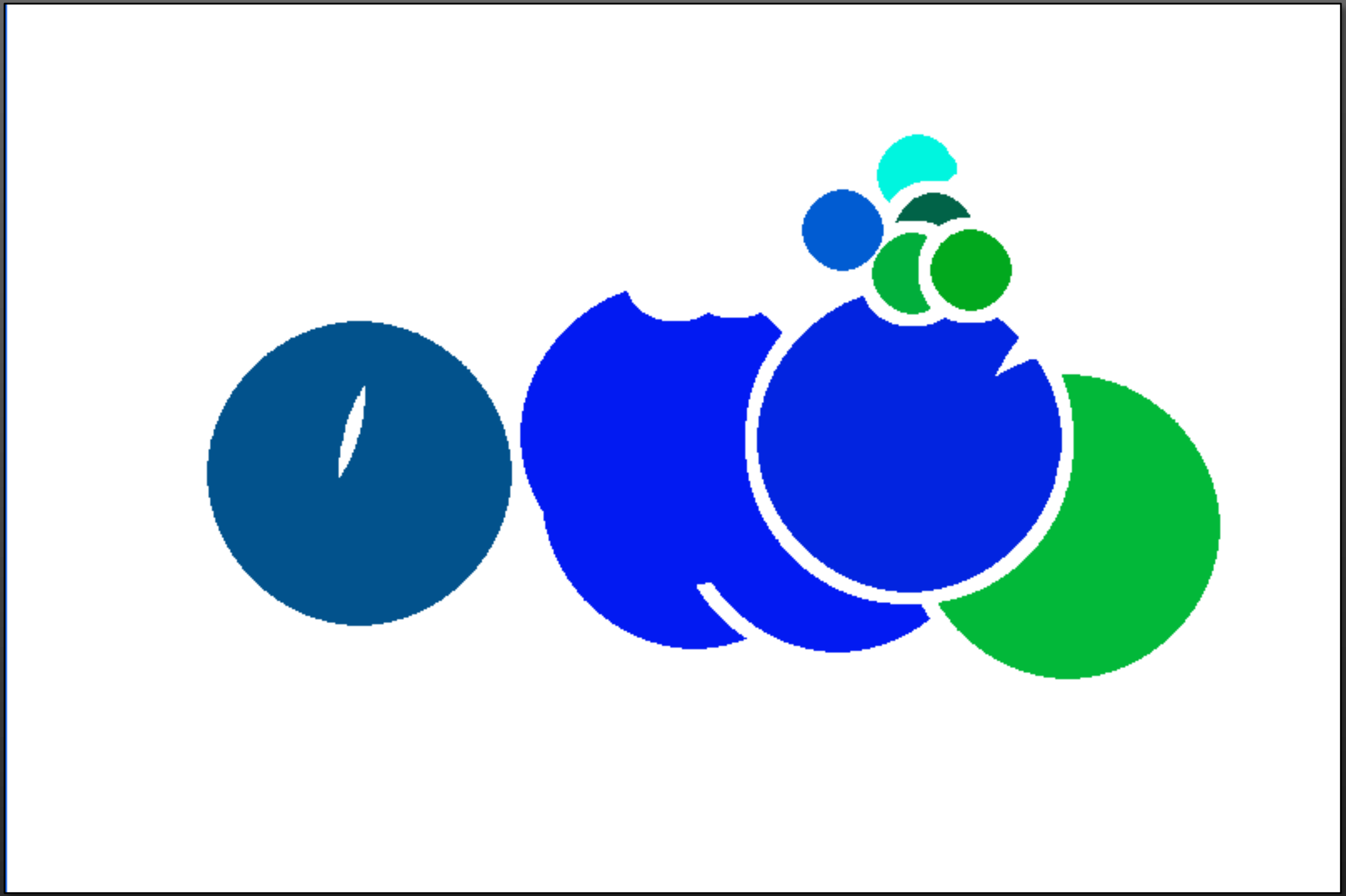
Legend

🟥 x

🟦 y

*k*NN, neighbor 2

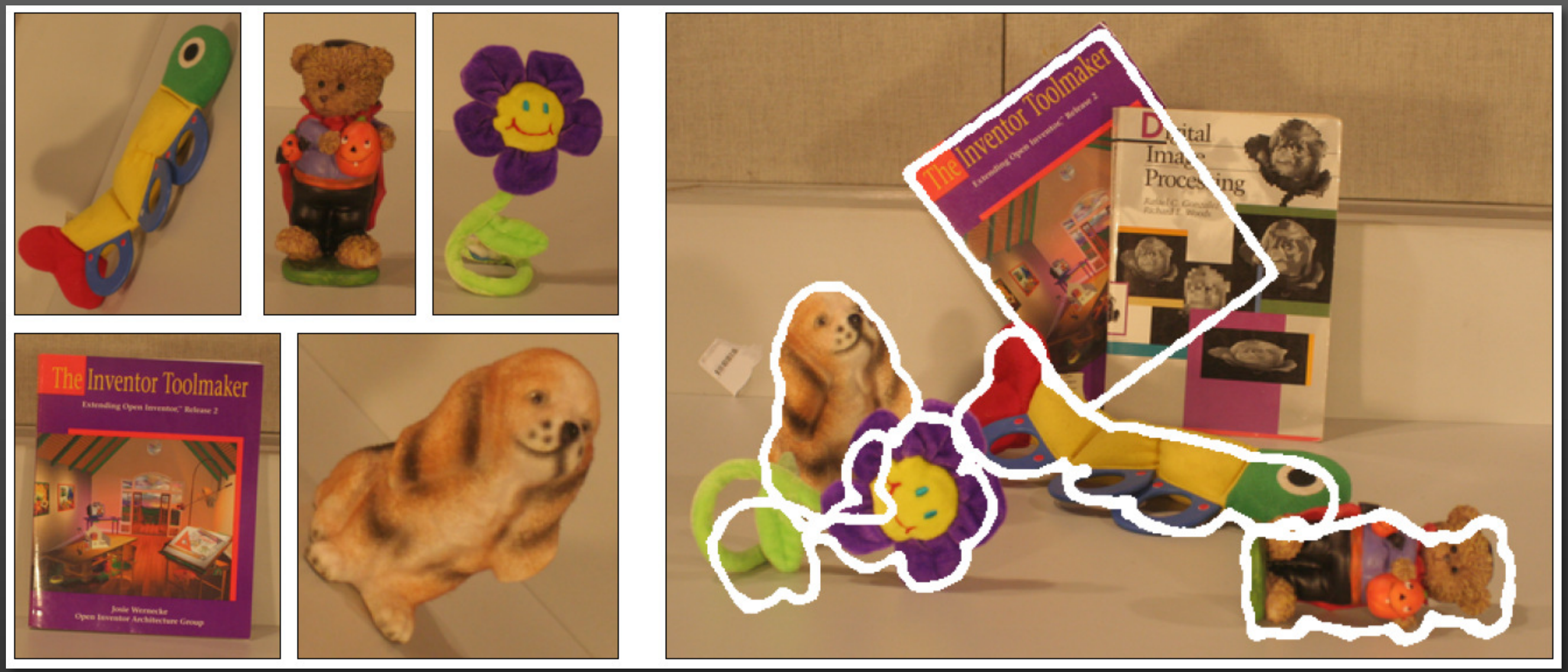
Detecting Forgeries



Mask of Likely Forgery

Object Detection

- Detect a template in a scene:

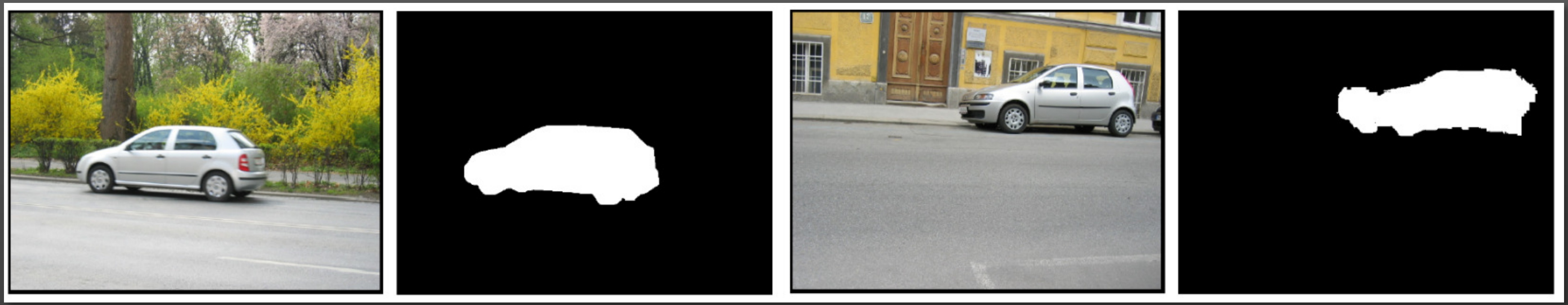


Template

Scene

Object Detection

- Label transfer:



Labeled Photo

Known Label

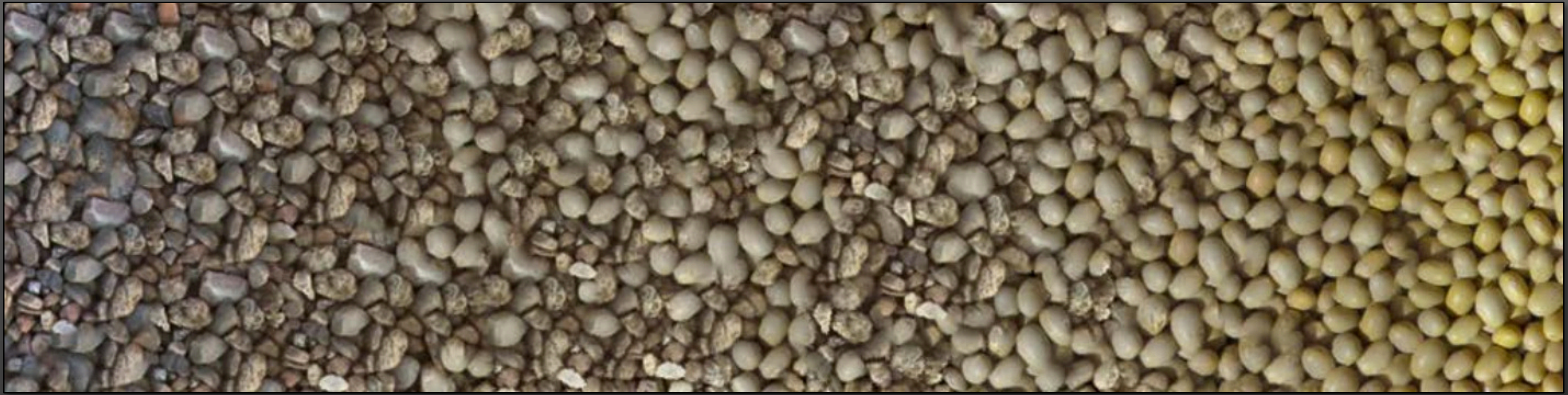
Unlabeled Photo

Transferred Label

Overview

- Fast matching in images: the PatchMatch algorithm
  - Randomized algorithm
  - Applications: Image editing and video tapestries
- Matching under lighting and geometric variations
  - Dealing with higher dimensional spaces
  - Applications: computer vision, stitching and morphing, …
- Impact
- Future perspectives

Stitching and Morphing

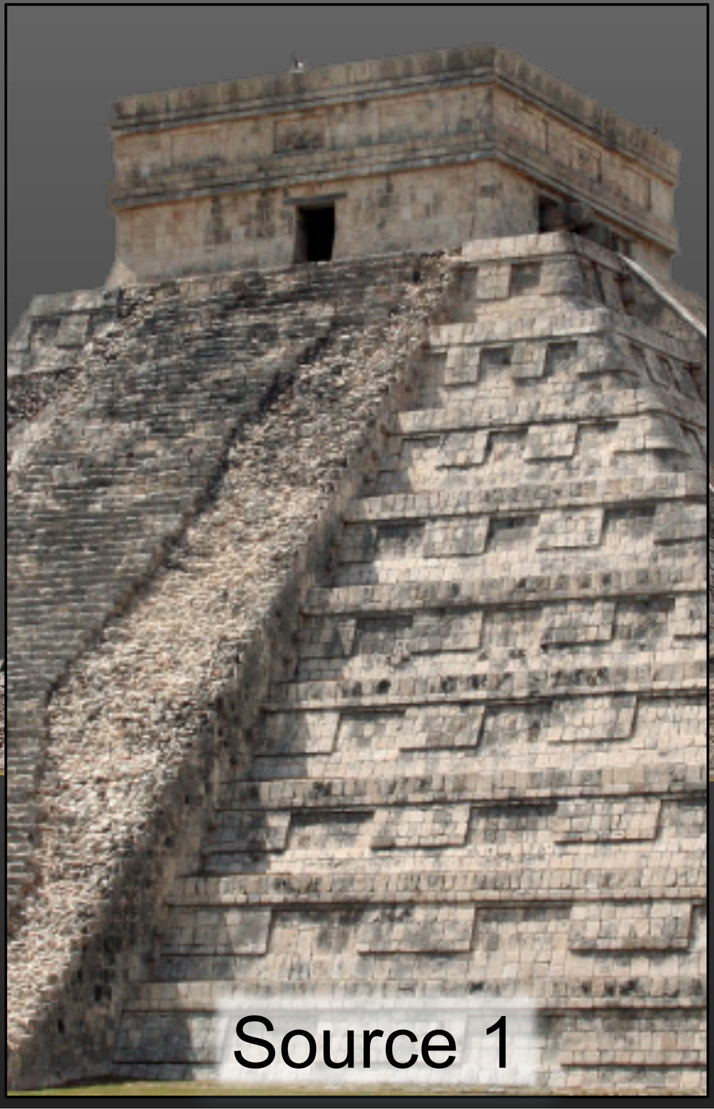


Improvements:

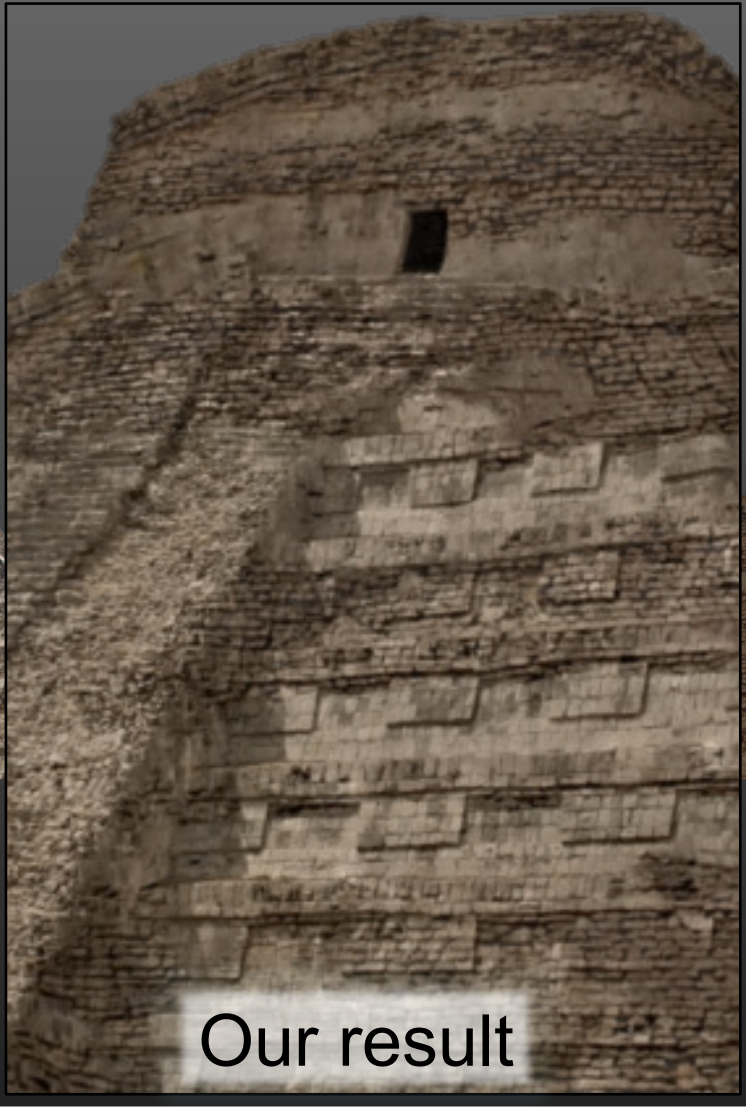
(1) Match across geometrical changes, (2) Use image gradients

[Darabi et al, Image Melding: Combining Inconsistent Images…, SIGGRAPH '12]

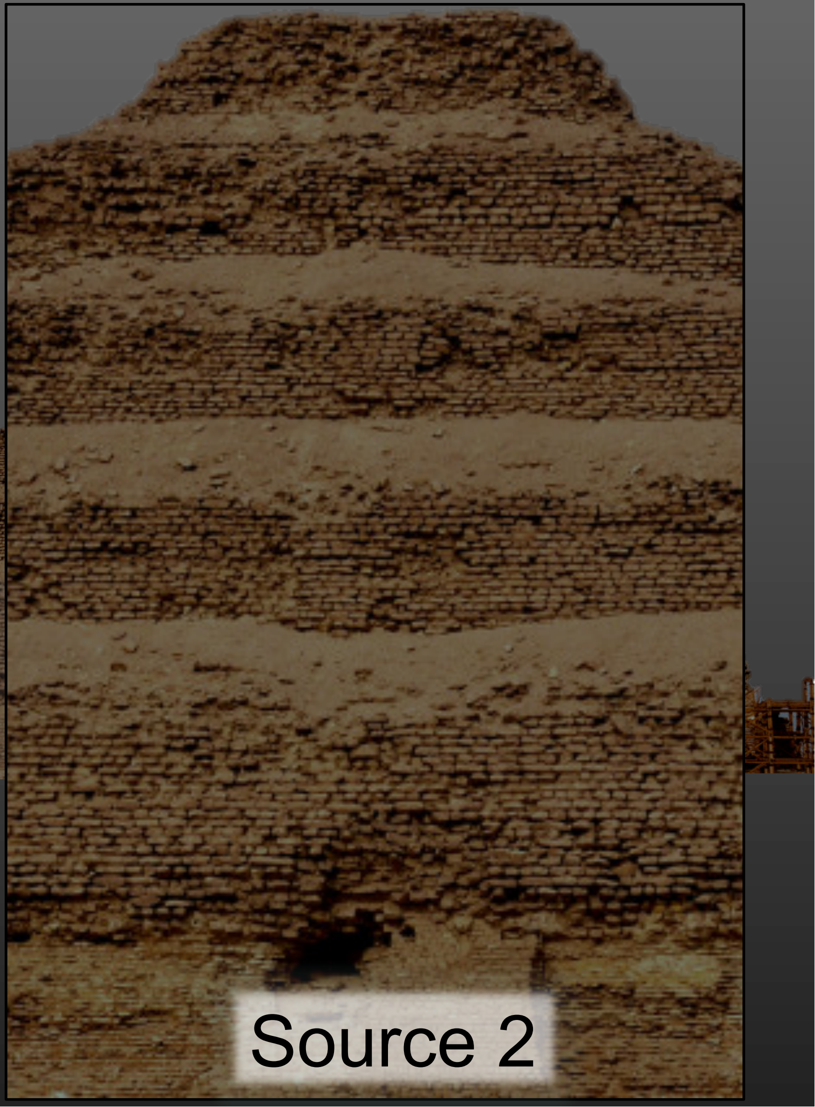
Texture Transition



Source 1

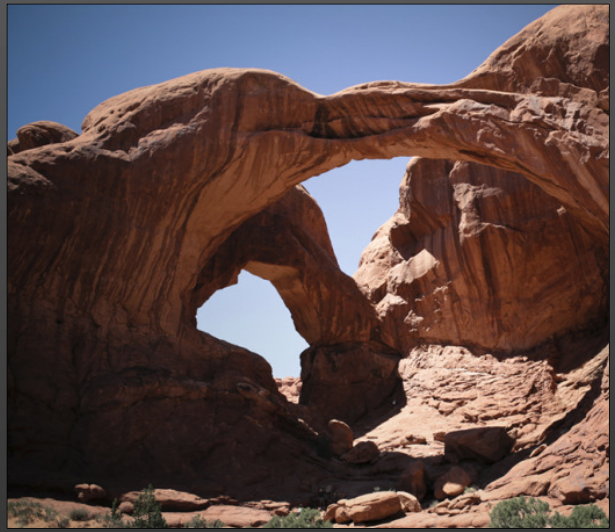


Our result

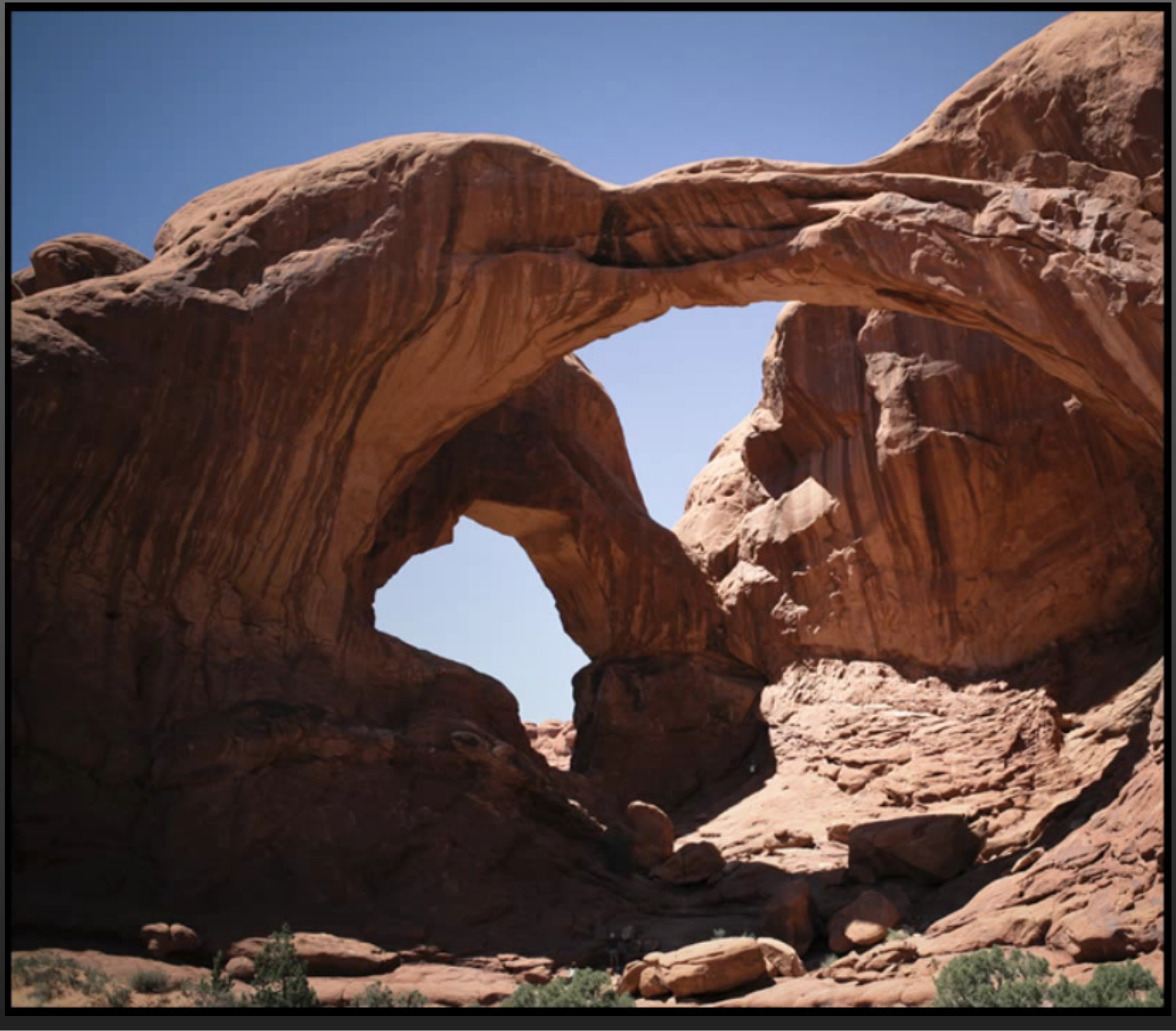


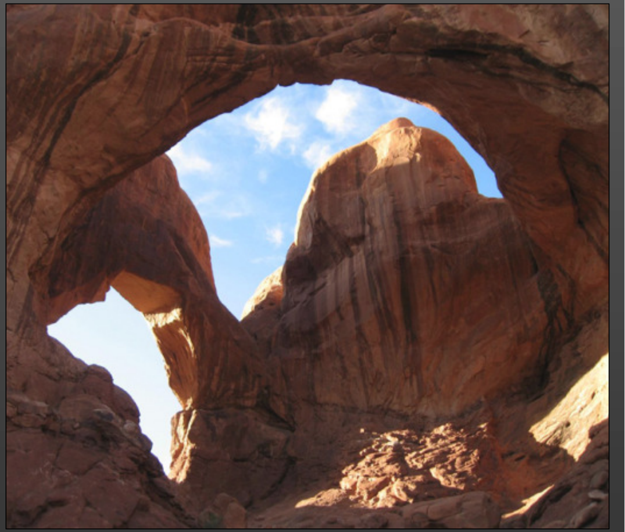
Source 2

Morphing Result



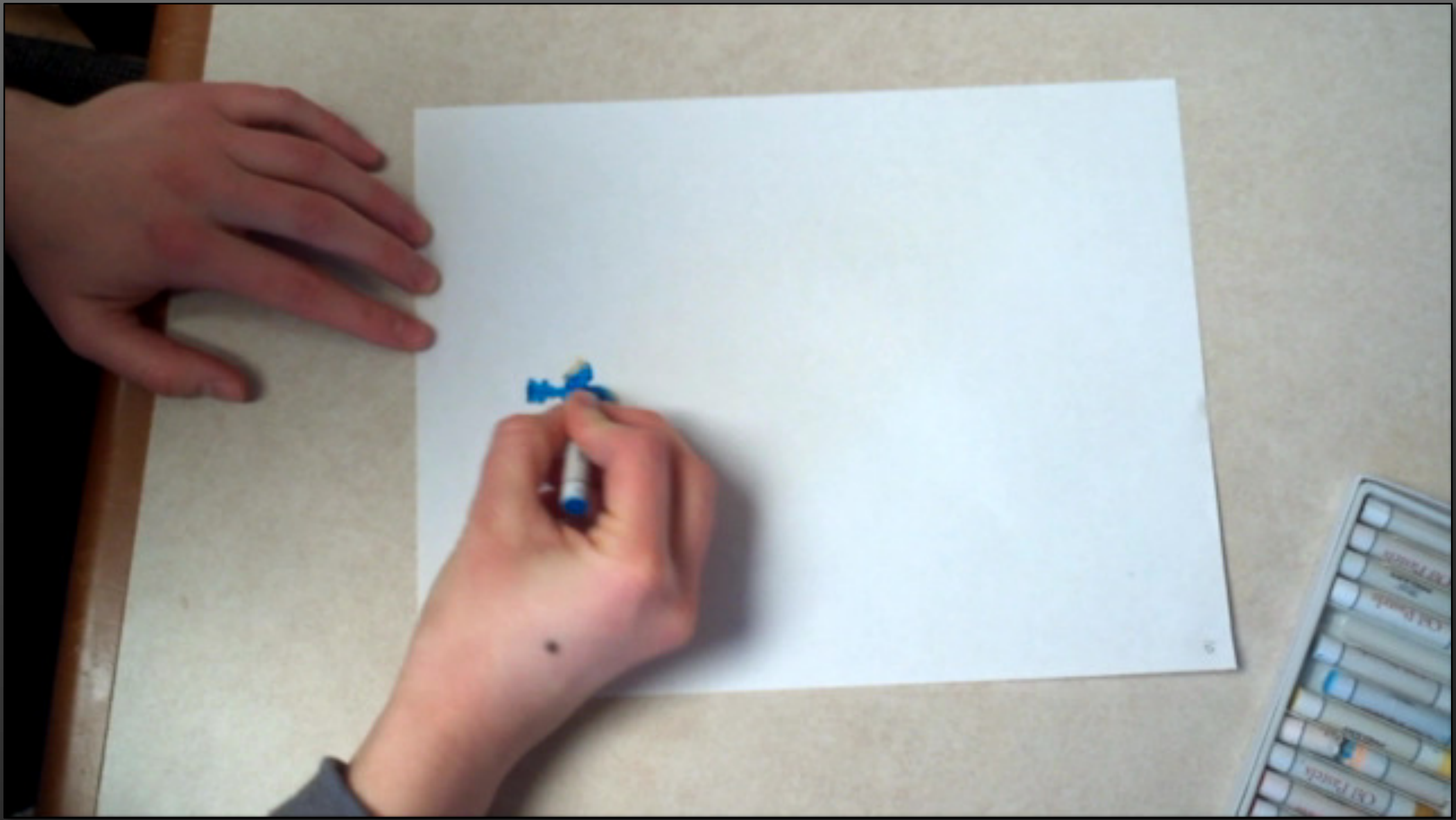
Source 1





Source 2

Morphing for Fluid Animation



[NPR Fluids, under review, with Mark Browning at Princeton]

Overview

- Fast matching in images: the PatchMatch algorithm
  - Randomized algorithm
  - Applications: Image editing and video tapestries
- Matching under lighting and geometric variations
  - Dealing with higher dimensional spaces
  - Applications: computer vision, stitching and morphing, …
- Impact
- Future perspectives

Interactivity Matters!

- Minutes to second: now a creation tool
  - Demos of our research: 8 million views on YouTube
  - #1 feature of Photoshop CS5
  - Award PC Magazine

Research Impact

Direct Follow-Ons

1. PatchMatch Stereo - Stereo Matching with Slanted Support Windows, Bleyer et al, BMVC 2011
2. PMBP: PatchMatch Belief Propagation for Correspondence Field Estimation, Besse et al, BMVC 2012
3. PatchMatch-Based Content Completion of Stereo Image Pairs, Morse et al, IEEE 3DIMPVT 2012
4. PatchMatchGraph: Building a Graph of Dense Patch Correspondences for Label Transfer, Gould, ECCV 2011
5. Parallel-Friendly Patch Match Based on Jump Flooding, Yu et al 2012
6. Feature match: an efficient low dimensional PatchMatch technique, Ramakanth, Indian Conf. on Vision 2012
7. Mixed-Resolution Patch-Matching, Sureka et al, ECCV 2012
8. Non-parametric Texture Transfer Using MeshMatch, Chen et al, Adobe TR 2012
9. TreeCann - k-d tree Coherence Approximate Nearest Neighbor algorithm, Olonetsky et al, ECCV 2012
10. Coherency sensitive hashing, Korman et al, ICCV 2011
11. Computing nearest-neighbor fields via Propagation-Assisted KD-Trees, He et al, CVPR 2012

Direct Follow-Ons

1. PatchMatch Stereo - Stereo Matching with Slanted Support Windows, Bleyer et al, BMVC 2011
2. PMBP: PatchMatch Belief Propagation for Correspondence Field Estimation, Besse et al, BMVC 2012
3. PatchMatch-Based Content Completion of Stereo Image Pairs, Morse et al, IEEE 3DIMPVT 2012
4. PatchMatchGraph: Building a Graph of Dense Patch Correspondences for Label Transfer, Gould, ECCV 2011
5. Parallel-Friendly Patch Match Based on Jump Flooding, Yu et al 2012
6. Feature match: an efficient low dimensional PatchMatch technique, Ramakanth, Indian Conf. on Vision 2012
7. Mixed-Resolution Patch-Matching, Sureka et al, ECCV 2012
8. Non-parametric Texture Transfer Using MeshMatch, Chen et al, Adobe TR 2012
9. TreeCann - k-d tree Coherence Approximate Nearest Neighbor algorithm, Olonetsky et al, ECCV 2012
10. Coherency sensitive hashing, Korman et al, ICCV 2011
11. Computing nearest-neighbor fields via Propagation-Assisted KD-Trees, He et al, CVPR 2012

Papers Using Components of PatchMatch Algorithm

12. RepFinder: finding approximately repeated scene elements for image editing, Cheng et al, SIGGRAPH 2010
13. A high-quality video denoising algorithm based on reliable motion estimation, Liu and Freeman, ECCV 2010
14. Regenerative morphing, Shechtman et al, CVPR 2010
15. Synthesizing structured image hybrids, Risser et al, SIGGRAPH 2010
16. NRDC: Non-Rigid Dense Correspondence with Applications for Image Enhancement, HaCohen et al, SIGGRAPH 2011
17. Transforming image completion, Mansfield et al, BMVC 2011
18. Randomized motion estimation, Boltz and Nielsen, ICIP 2010
19. Decoupled coarse-to-fine matching and nonlinear regularization for efficient motion estimation, Mariano and Sapiro, ICIP 2012
20. Object Removal by Depth-guided Inpainting, He, Bleyer, Gelautz, Australian Association for Pattern Recognition 2011
21. Interactive images: cuboid proxies for smart image manipulation, Zheng, Chen, Cheng, Zhou, Hu, Mitra, SIGGRAPH 2012
22. A global sampling method for alpha matting, He, Rhemann, Rother, Tang, Sun, CVPR 2011
23. ImageAdmixture: Putting Together Dissimilar Objects from Groups, Cheng et al, TVCG 2012
24. Statistics of Patch Offsets for Image Completion, He et al, TVCG 2012
25. Interactive image completion with perspective constraint, Hao et al VRCAI 2012
26. Super-Resolution-based Inpainting, Le Meur et al, ECCV 2012
27. Robust patch-based hdr reconstruction of dynamic scenes, Sen et al, SIGGRAPH Asia 2012

Conclusion

- Patch nearest neighbor problem central to vision/graphics
- New approach (PatchMatch) based on statistical sampling
- Changing the way people look at image processing
- Applications:

Image Editing

Video Tapestries

Computer Vision

Stitching and Morphing

Collaborators

Frédo Durand *(MIT)*

Saman Amarasinghe *(MIT)*

Adam Finkelstein *(Princeton)*

Szymon Rusinkiewicz *(Princeton)*

Maneesh Agrawala *(Berkeley)*

Pradeep Sen *(U. New Mexico)*

Dan B Goldman *(Adobe)*

Eli Shechtman *(Adobe)*

Sylvain Paris *(Adobe)*

Jonathan Ragan-Kelley *(MIT)*

Andrew Adams *(MIT)*

Jingwan Lu *(Princeton)*

Mark Browning *(Princeton)*

Sam Ritter *(Princeton)*

David E Jacobs *(Berkeley)*

Jason Sanders *(Berkeley)*

Soheil Darabi *(U. New Mexico)*

Stephen DiVerdi *(Google)*