CSC320 A3 PatchMatch

Slides by Yawen Ma

Based on Connelly Barnes, Eli Shechtman, Adam Finkelstein, Dan B Goldman(2009), PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing

Goal

- Find patch correspondences

Image A

Image B

Intuitions

- Neighboring pixels/patches are correlated
- One of many random guesses is likely to be a good guess

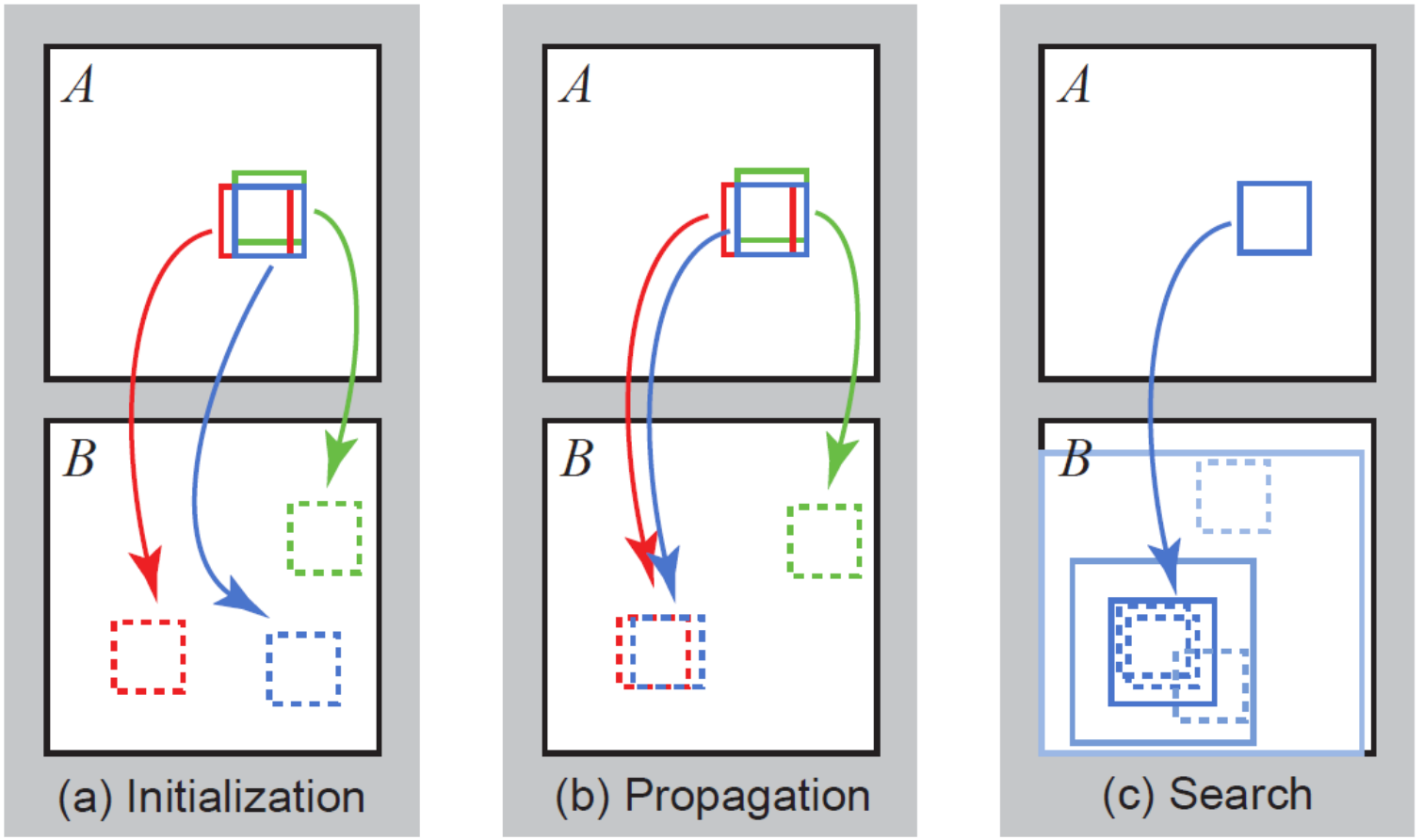
Nearest-neighbor field (NNF)

- A mapping: f: *A* -> $\mathbb{R}^2$
- "*A*": every patch center (pixel) on image A
- $\mathbb{R}^2$: the offset (differences between coordinates) of patches that has the smallest L_p distance

  - For a pixel at (x, y) in A, let v=($\Delta x, \Delta y$)=f(x, y) denote the current offset of the patch (x, y) in A.
  - Then the patch in B at (x, y)+v is the nearest patch of patch (x, y) in A

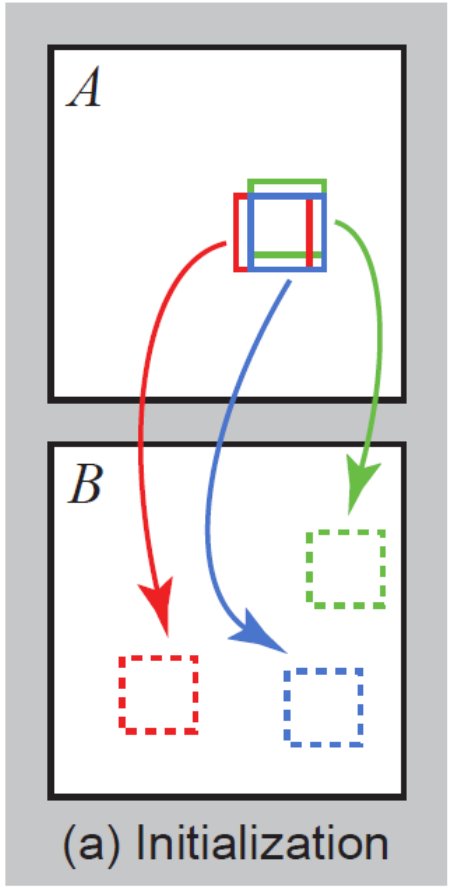
Approximate nearest-neighbor algorithm

1. Initialization
2. Iteration
   - Propagation
   - Random search



Initialization

- Fill NNF with
  - Random offsets
  - Or prior information



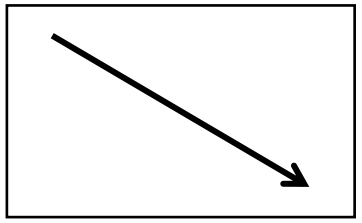
Iteration

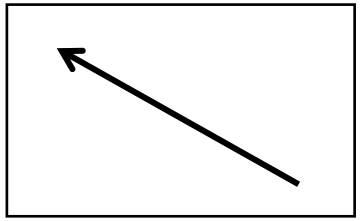
- Propagation
- Random Search

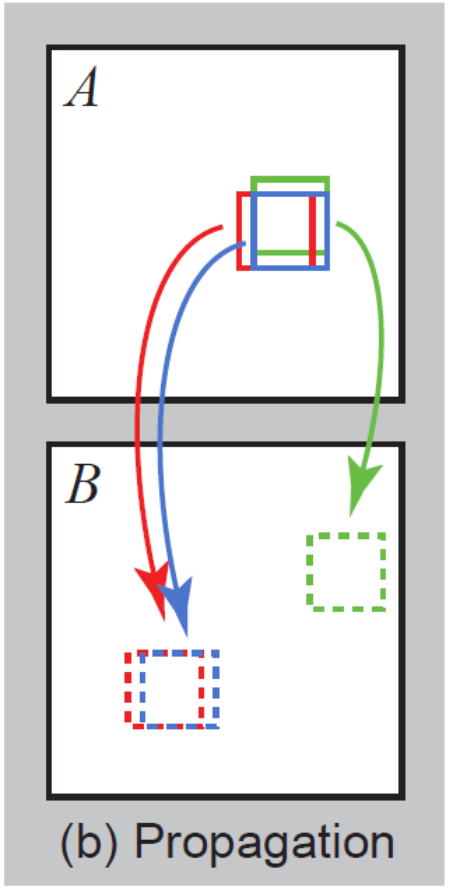
(Order: P1, S1, P2, S2, …, Pn, Sn)

Iteration - propagation

- For a pixel at (x, y) in A, let v=($\Delta x, \Delta y$)=f(x, y) denote the current offset of the patch (x, y) in A. Then the patch in B at (x, y)+v is the nearest patch of patch (x, y) in A
- Let D(v) denote the distance
- Odd # iterations:
  - argmin(D(f(x, y)), D(f(x-1, y)), D(f(x, y-1)))
- Even # iterations:
  - argmin(D(f(x, y)), D(f(x+1, y)), D(f(x, y+1)))







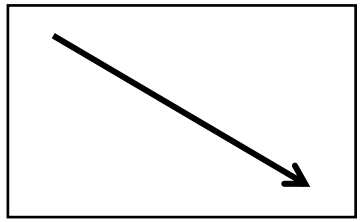
Exercise

Given 2 identical images A and B both of size (2n+1)*(2m+1).

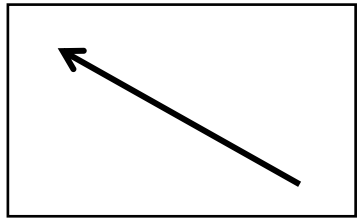
If it is known after some initialization the pixel at the center has

f(n+1, m+1)=(0, 0), and for all the other pixels f(x, y)≠(0, 0)

- What will happen after P1?
- What will happen after P1, P2 (skipping S1 for now)?



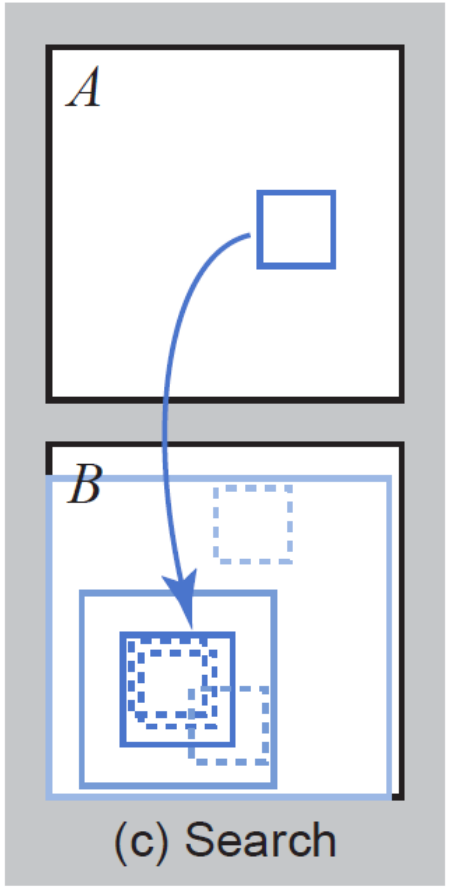
P1



P2

Iteration - random search

- Improve f(x, y) with several candidates:
- $u_i = v_0 + wa^i R_i$
  - $v_0 = \mathrm{f(x, y)}$ from last propagation
  - w: maximum image dimension
  - a: fixed ratio between search window sizes
  - R_i: uniform random in [-1, 1]*[-1, 1]
- Update f(x, y) with argmin{D(u_1),D(u_2),…,D(u_n)}



Questions?