Computer Vision
for Visual Effects

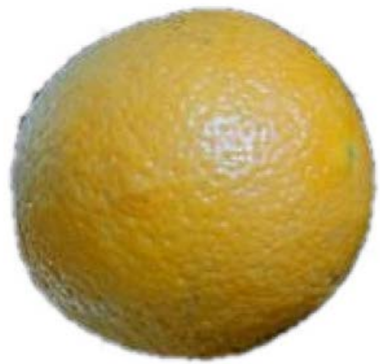
CVFX 2015

Related Work

- *Image Quilting for Texture Synthesis and Transfer*
  - Efros and Freeman
  - SIGGRAPH 2001
- *Graphcut Textures: Image and Video Synthesis Using Graph Cuts*
  - Kwatra, Sch¨odl, Essa, Turk, Bobick
  - SIGGRAPH 2003
- *Textureshop: Texture Synthesis as a Photograph Editing Tool*
  - Fang and Hart
  - SIGGRAPH 2004

Texture Transfer

+



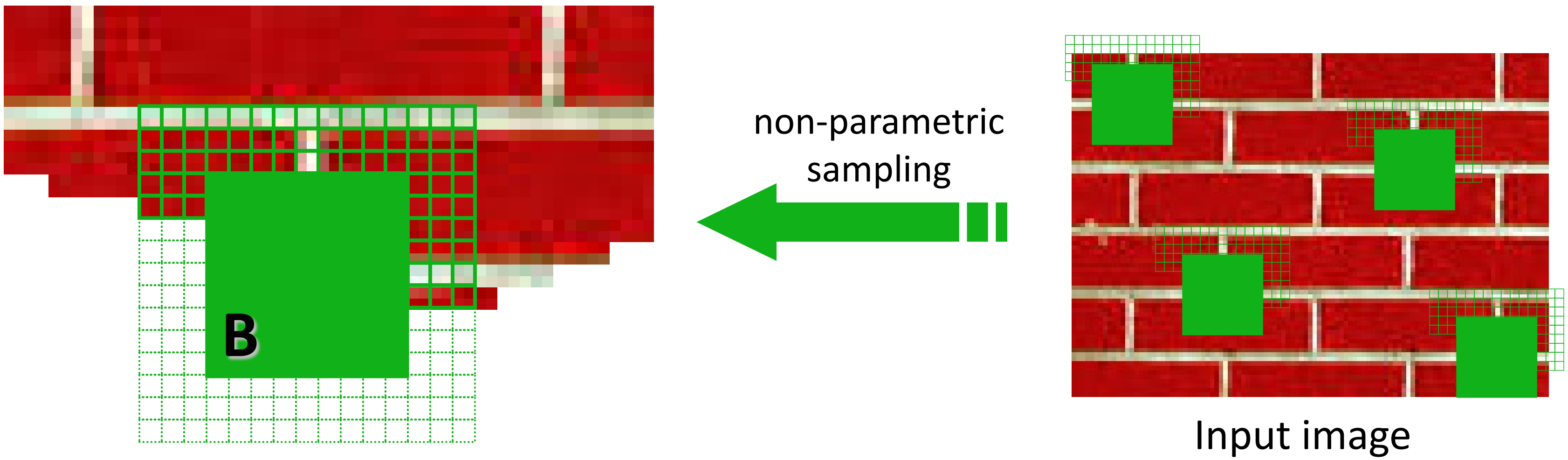
= ?

Idea

- Efros & Leung
  - Markov Random Fields
  - Nonparametric sampling

- Patch-based sampling, patch-by-patch synthesis
  - Texture blocks are by definition correct samples of texture
  - Avoiding a lot of searching work wasted on pixels that already "know their fate"
  - Stitching together small patches in a consistent way

Image Quilting



Synthesizing a block

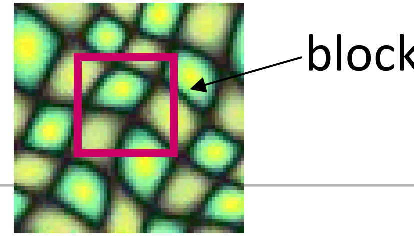
Input image

› Observation: neighbor pixels are highly correlated

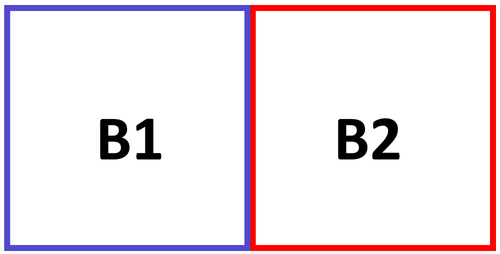
Idea: unit of synthesis = block

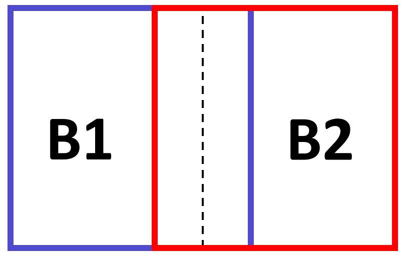
- Exactly the same but now we want P(**B**|N(**B**))
- Much faster: synthesize all pixels in a block at once
- Not the same as multi-scale!

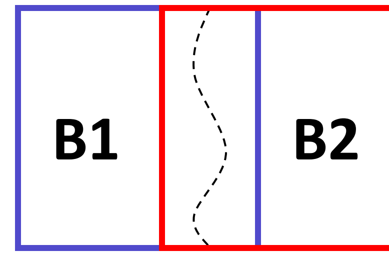
[Efros's slides]



Input texture



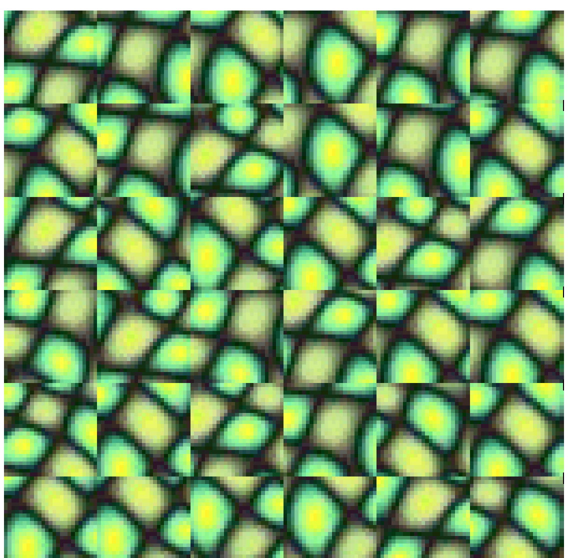


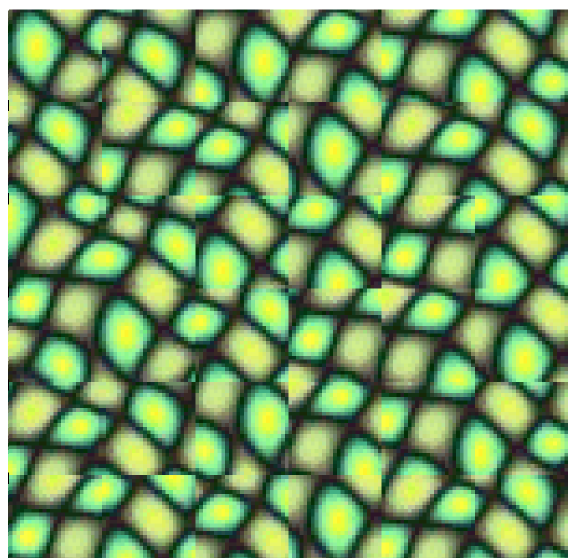


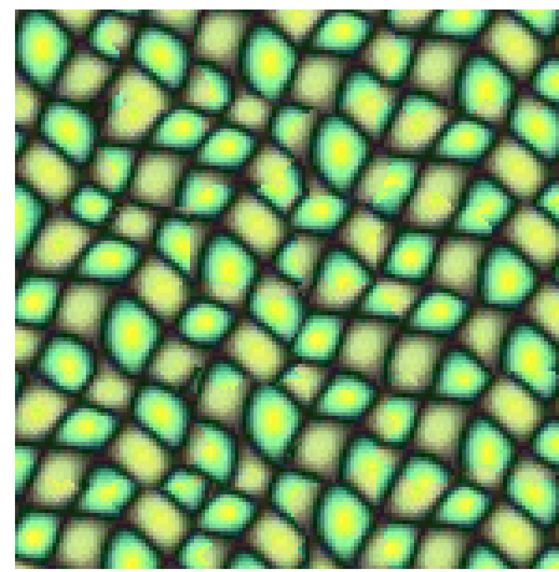
Random placement of blocks

Neighboring blocks constrained by overlap

Minimal error boundary cut



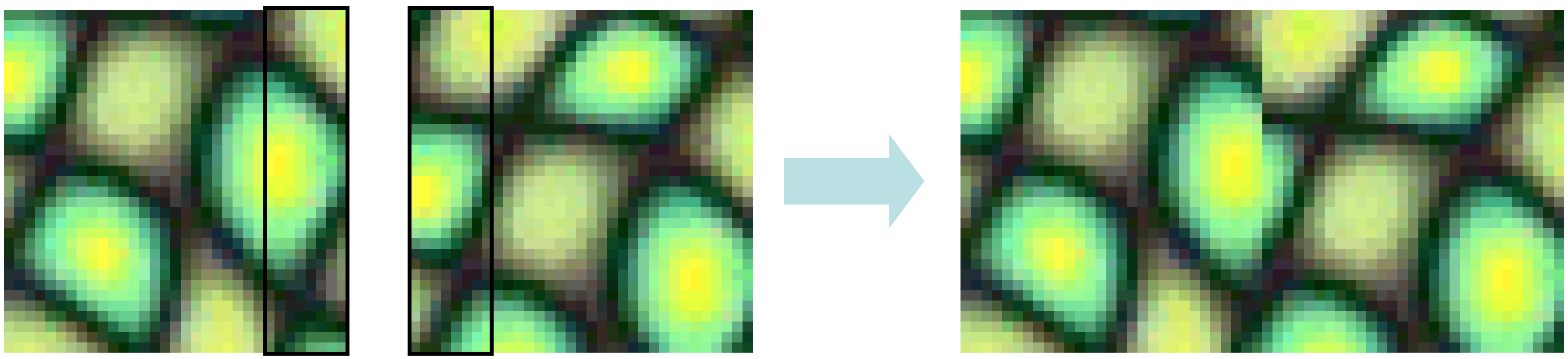


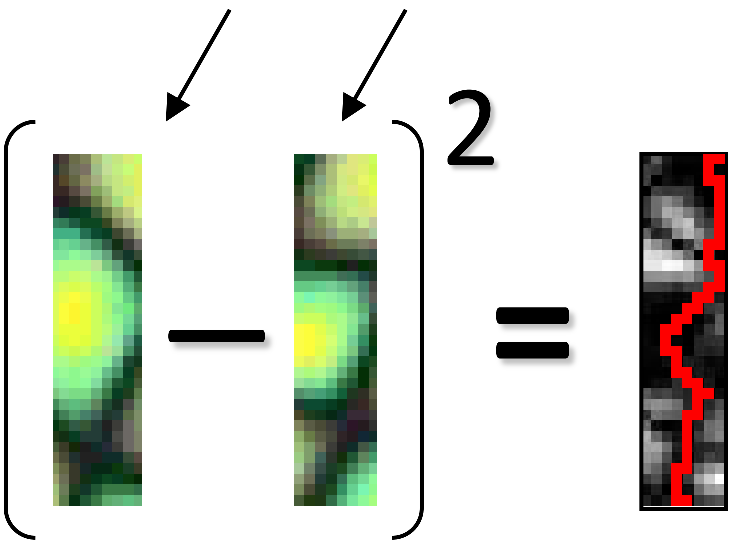


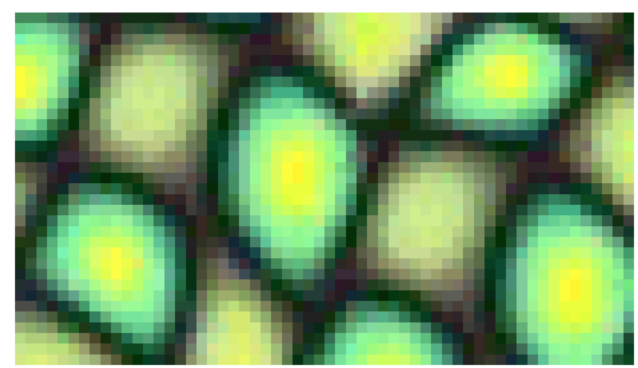
Minimal Error Boundary

overlapping blocks

vertical boundary







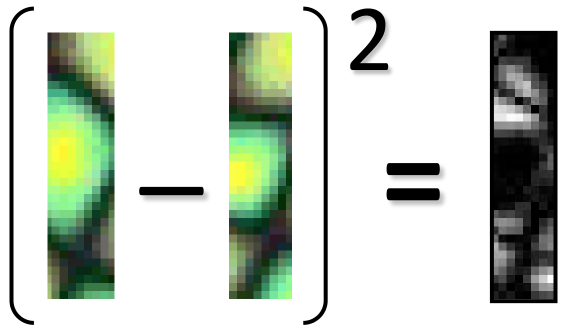
overlap error

min. error boundary

Minimum Error Boundary Cut

- Find the minimal cost path through the error surface

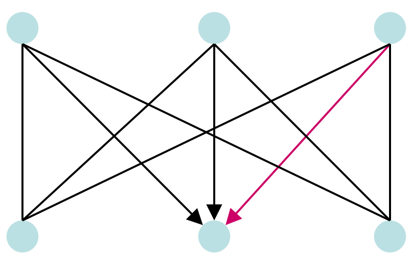
$$e = (B_1^{ov} - B_2^{ov})^2$$



overlap error

compute the cumulative minimum error for all paths

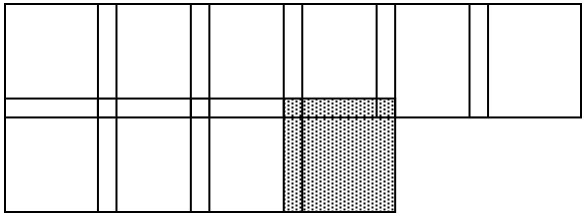
$$E_{i,j} = e_{i,j} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1}).$$



can be done with dynamic programming or Dijkstra's algorithm

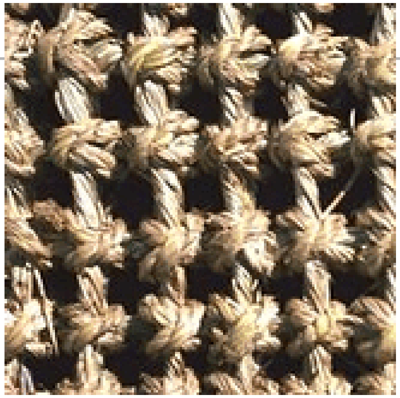
Algorithm

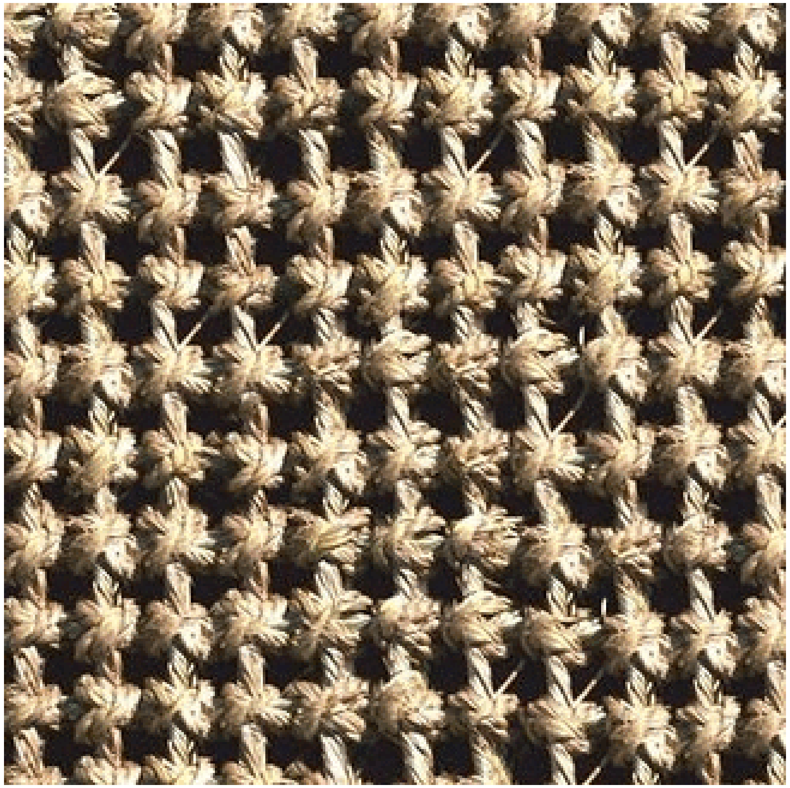
- Pick size of block and size of overlap
- Synthesize blocks in raster order



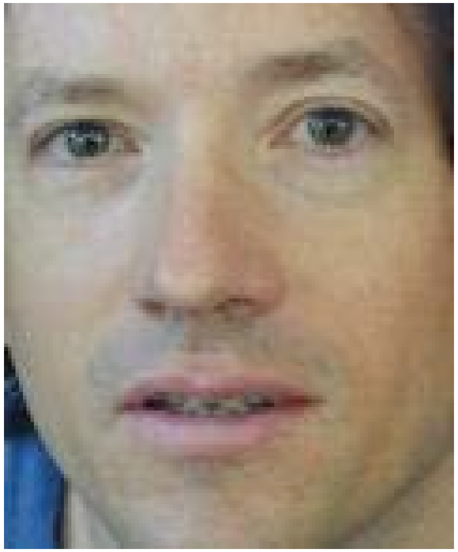
- Search input texture for block that satisfies overlap constraints (above and left)
  - Easy to optimize using nearest neighbor search
- Paste new block into resulting texture
  - Use dynamic programming to compute minimal error boundary cut

Texture Synthesis

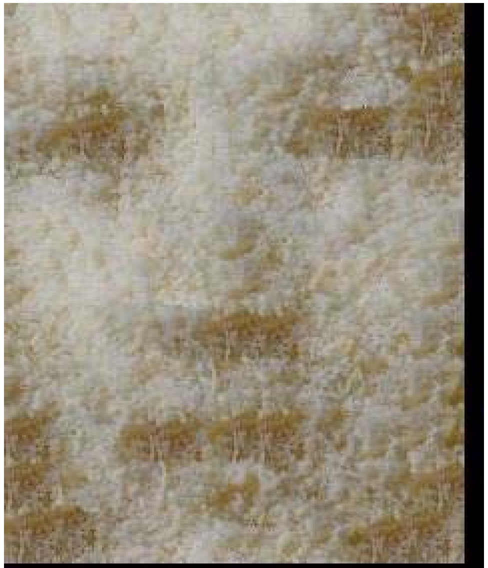


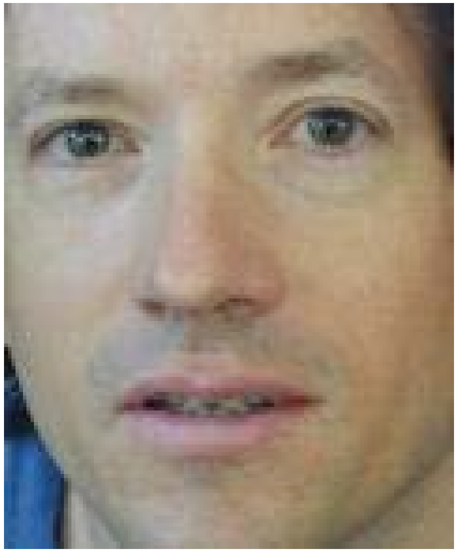


Texture Transfer



parmesan

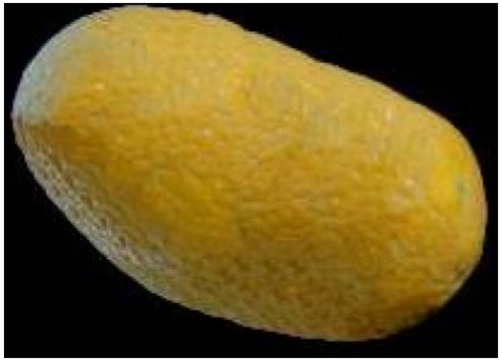


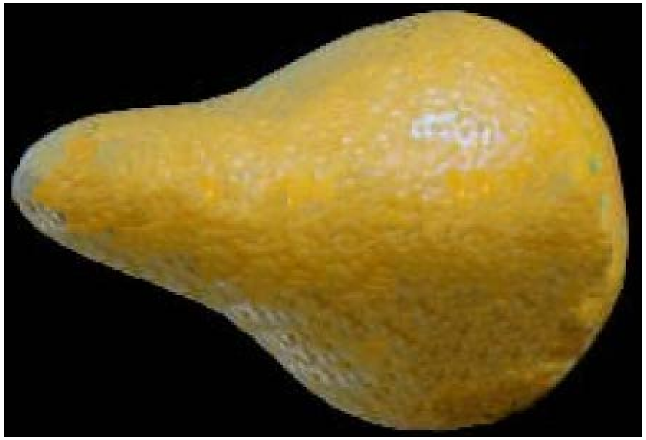


rice

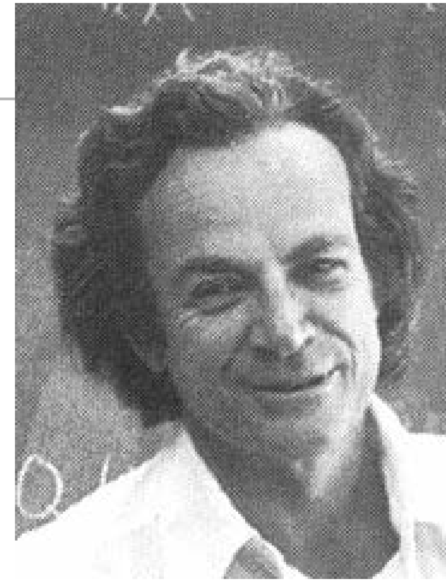
+

=



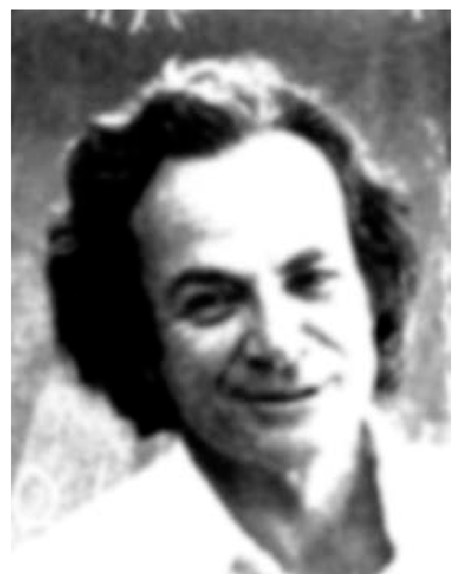


Source texture

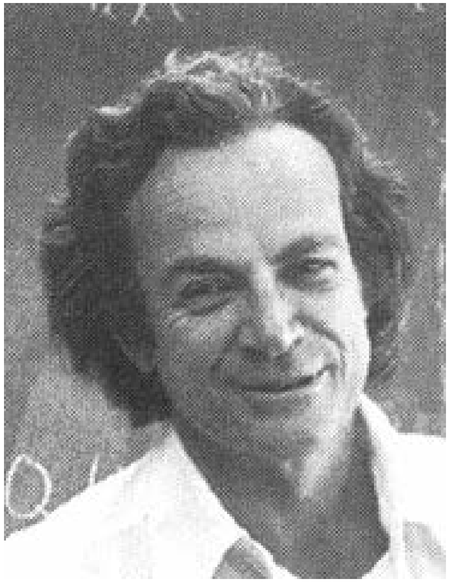


Target image

Source correspondence image

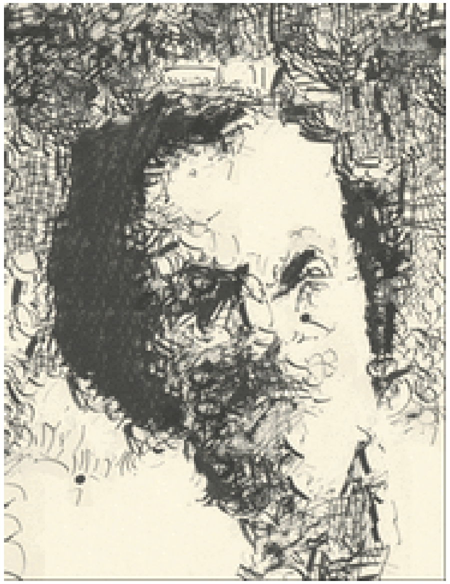


Target correspondence image



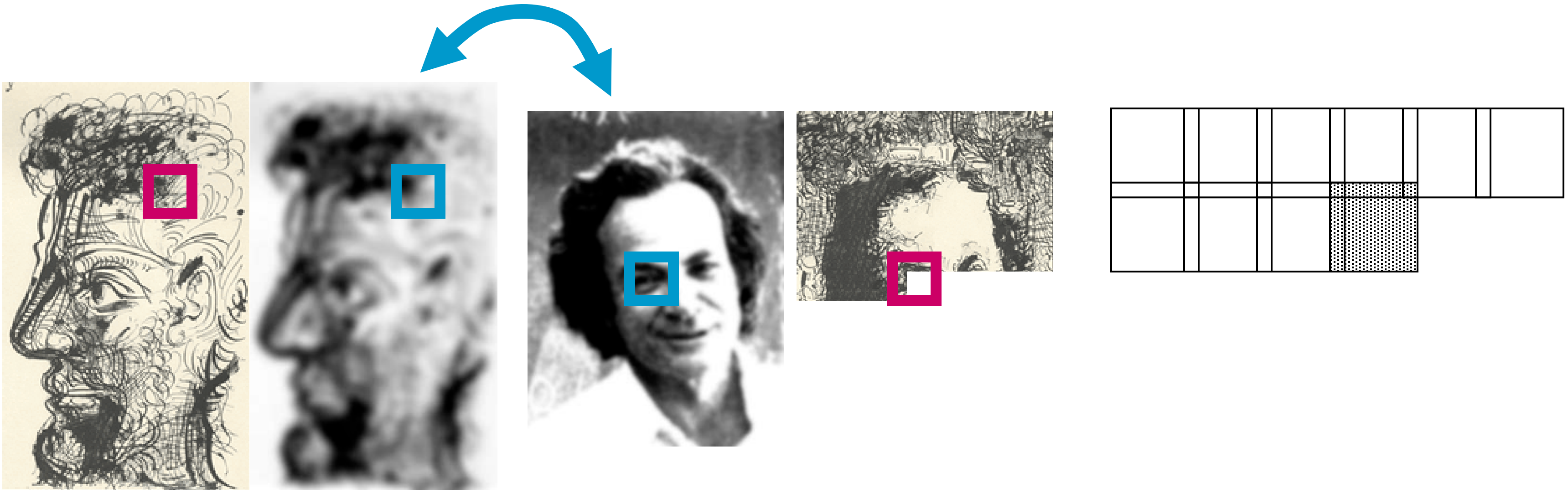
+

=



Texture Transfer

- Two independent constraints:
  - (a) the output are legitimate, synthesized examples of the source texture
  - (b) the correspondence image mapping is respected
- Iterative scheme
  - coarse-to-fine



Related Work

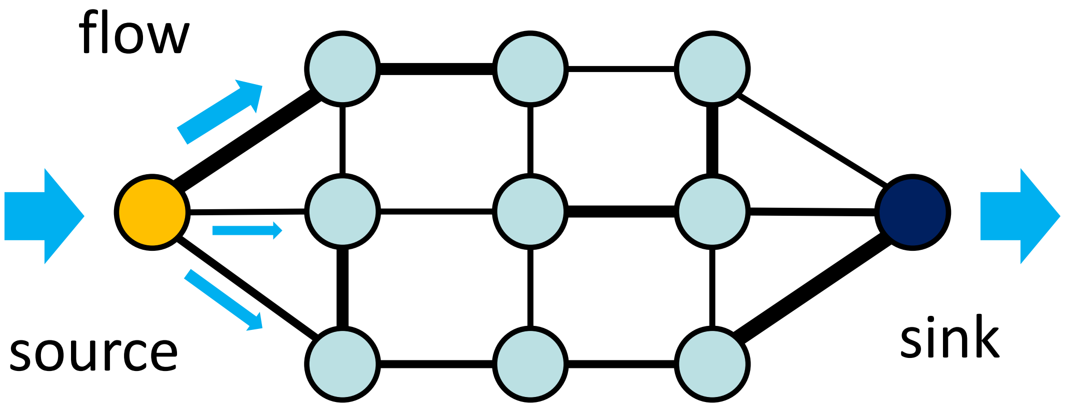
- *Image Quilting for Texture Synthesis and Transfer*
  - Efros and Freeman
  - SIGGRAPH 2001
- *Graphcut Textures: Image and Video Synthesis Using Graph Cuts*
  - Kwatra, Sch¨odl, Essa, Turk, Bobick
  - SIGGRAPH 2003
- *Textureshop: Texture Synthesis as a Photograph Editing Tool*
  - Fang and Hart
  - SIGGRAPH 2004

Graphcut Textures: Image and Video Synthesis Using Graph Cuts

Input Image Quilting Graph cut Input Image Quilting Graph cut

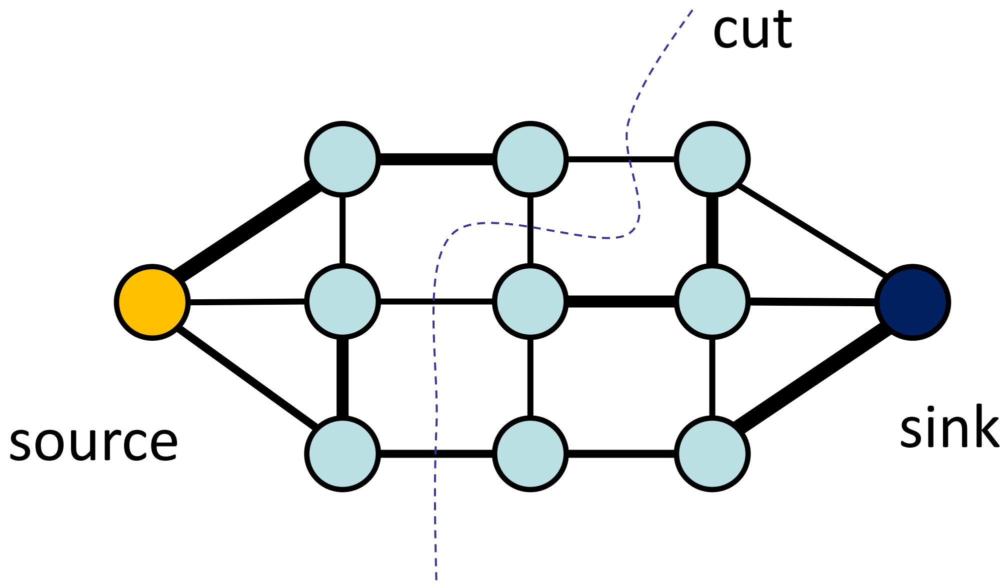
Max-Flow Problem

- Each edge is a *pipe*
- Find the largest *flow* F of *water* that can be sent from the *source* to the *sink* along the pipes
- The weight of an edge gives the pipe's capacity



Min-Cut Problem

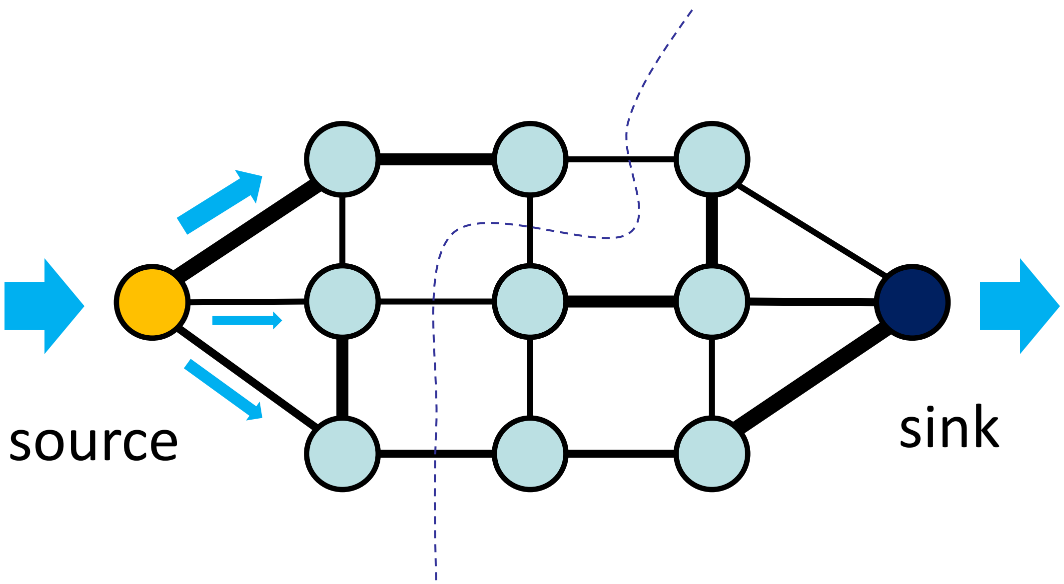
- Find the cheapest way to cut the edges so that the *source* is completely separated from the *sink*
- Edge weights now represent cutting *costs*



Based on Prof. Zabih's slides

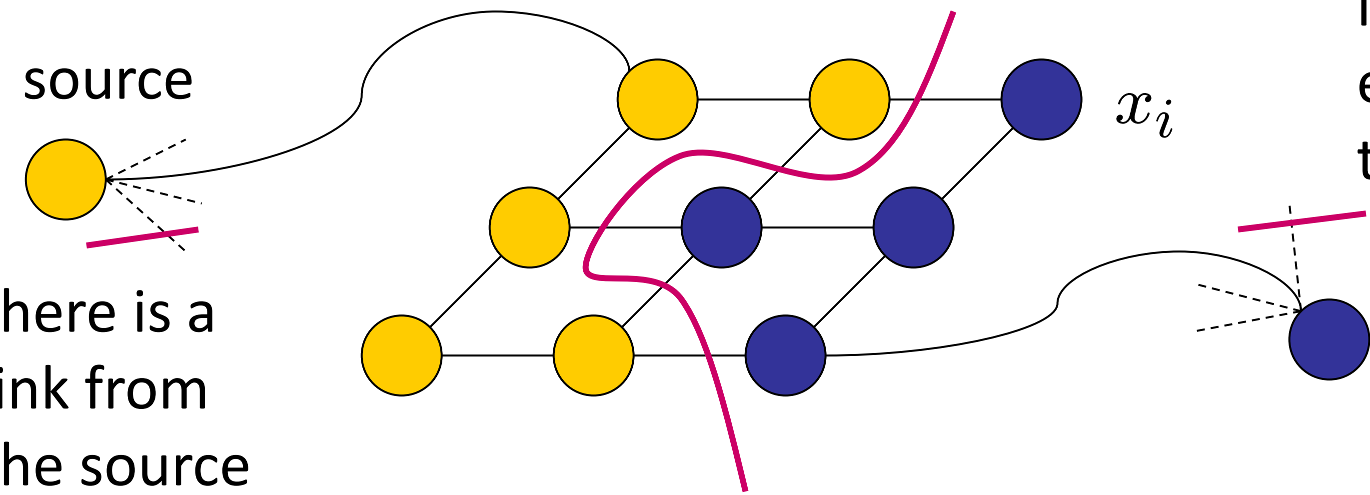
Max-Flow/Min-Cut Theorem

- Max Flow = Min Cut
  - Max-flow saturates the edges along the min-cut
- Ford and Fulkerson gave first polynomial time algorithm for globally optimal solution



Based on Prof. Zabih's slides

Pixel Labeling as a Min-Cut Problem



there is a link from the source to each node

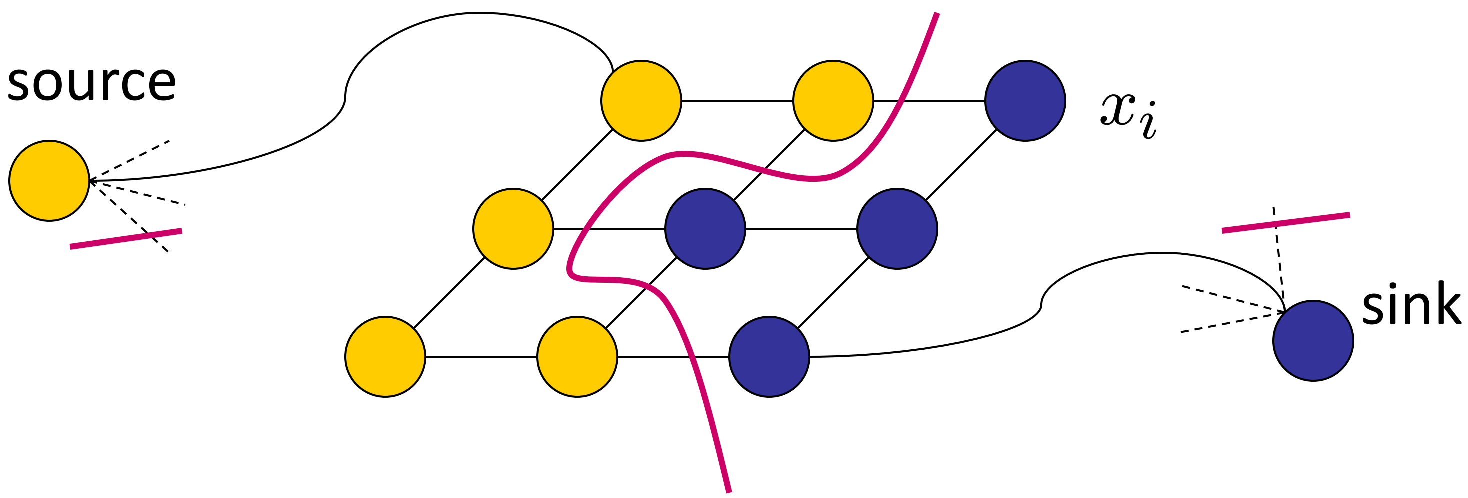
there is a link from each node to the sink

The nodes that still connect to the source (target) after the cutting will have the same label as the source (target).

Energy Minimization as a Min-Cut Problem

$$E(\mathbf{x}) = \sum_{\{i,j\}} V_{ij}(x_i, x_j) + \sum_i D_i(x_i)$$

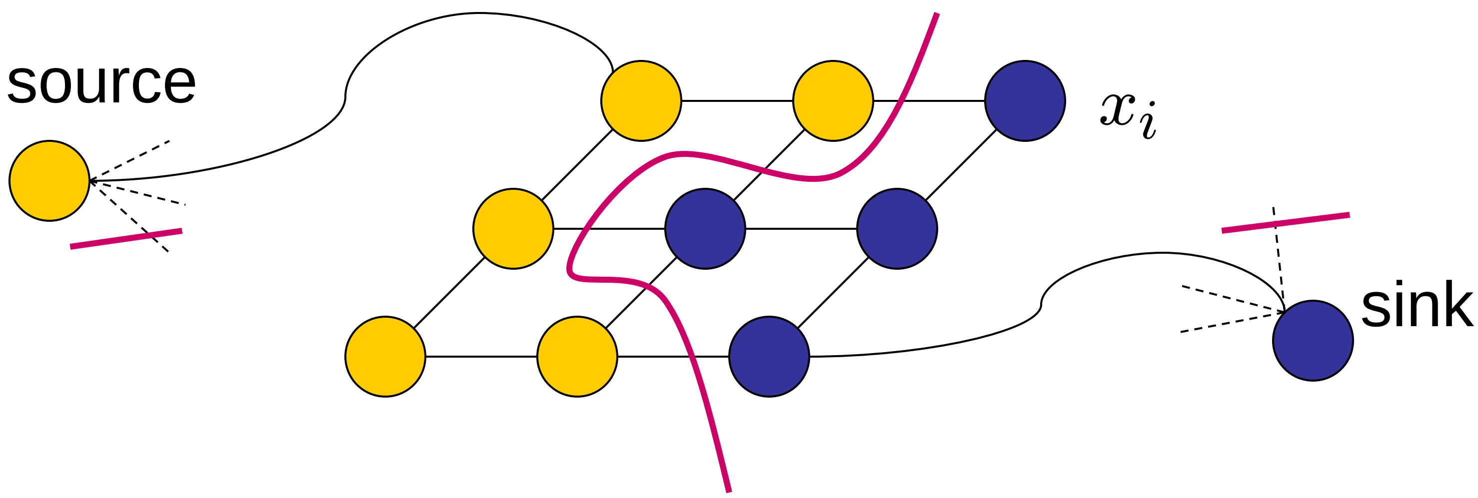
How to encode the energy terms into the graph such that finding the min-cut is equivalent to minimizing the energy for binary labeling?



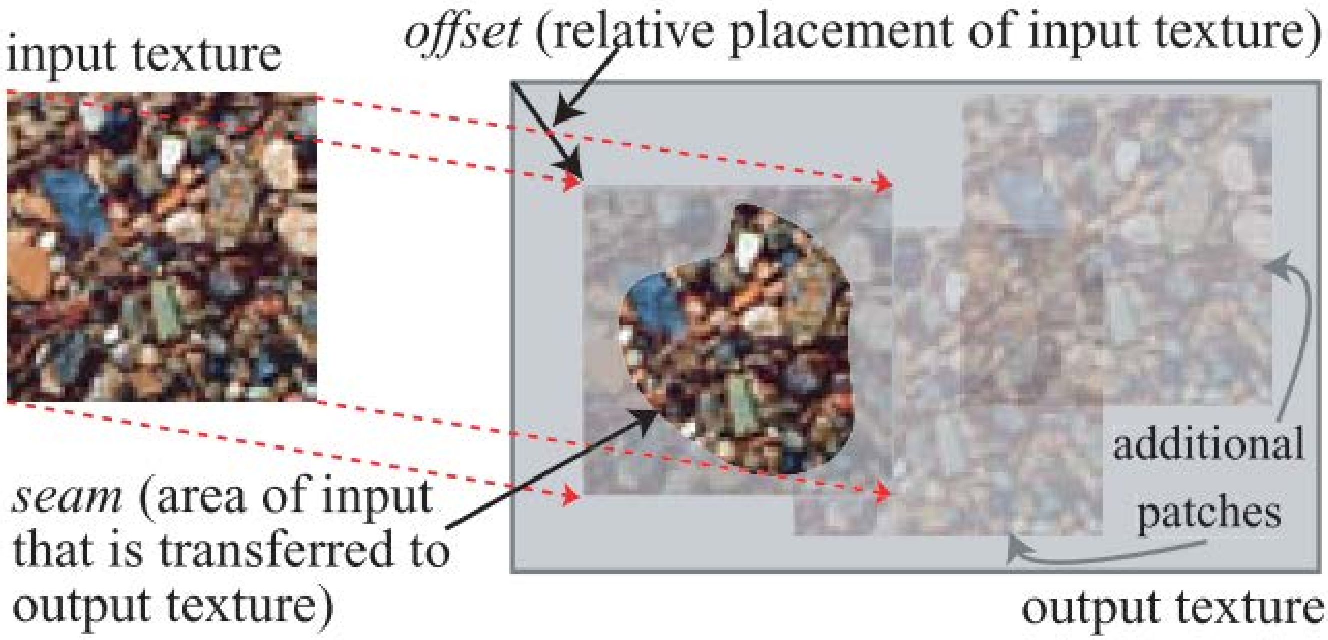
Energy Minimization as a Min-Cut Problem

$$E(\mathbf{x}) = \sum_{\{i,j\}} V_{ij}(x_i, x_j) + \sum_i D_i(x_i)$$

energy terms ← → costs on the edges



Copy, Paste, and Cut



Patch Fitting Using Graph Cuts

arc cost

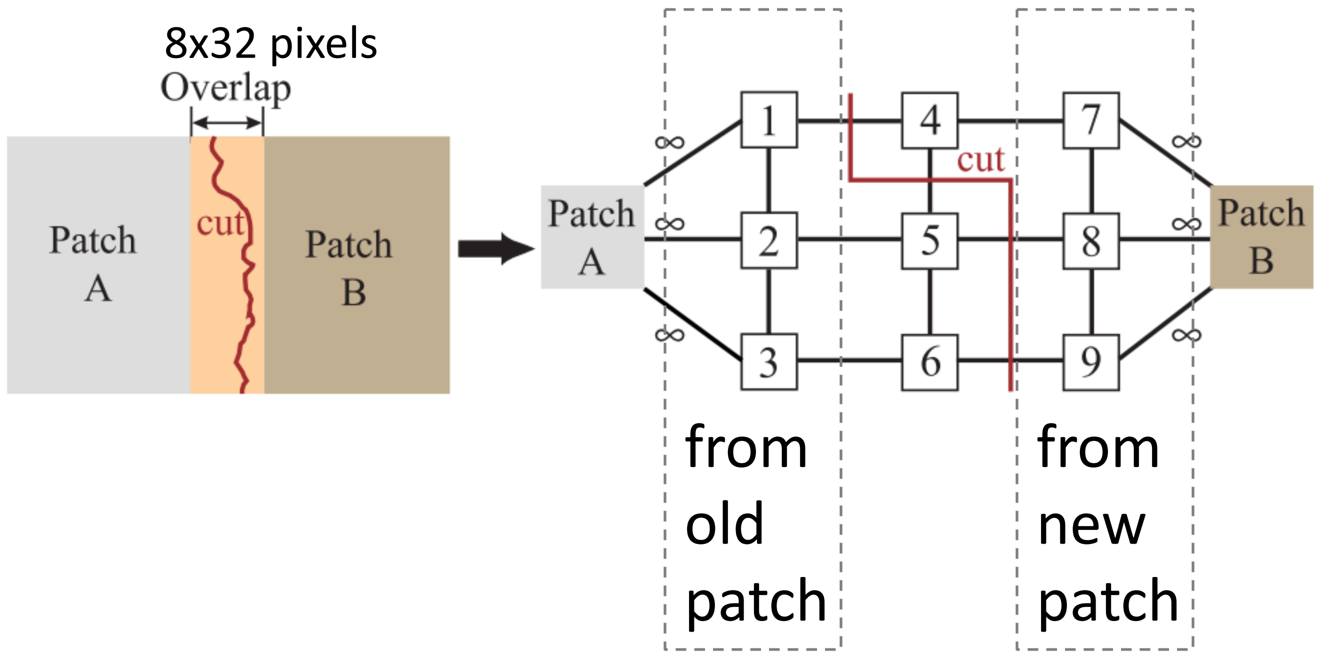
$$M(s,t,\mathbf{A},\mathbf{B}) = \|\mathbf{A}(s) - \mathbf{B}(s)\| + \|\mathbf{A}(t) - \mathbf{B}(t)\|$$

compare by color

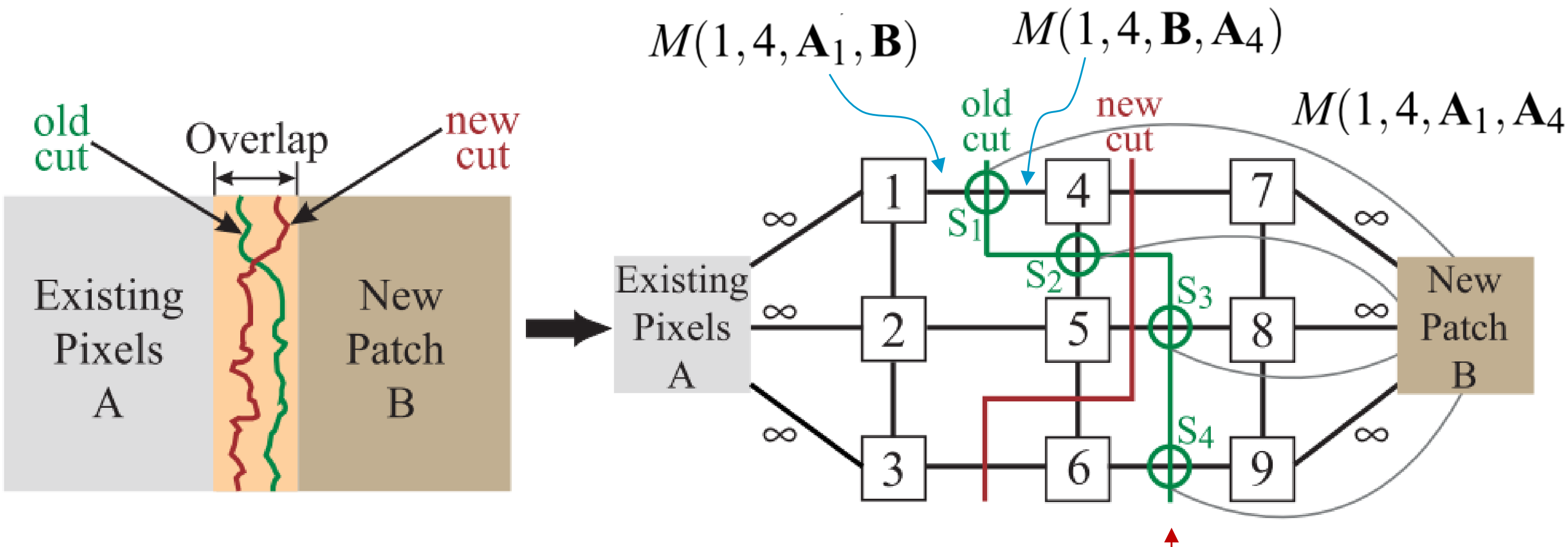
old

new

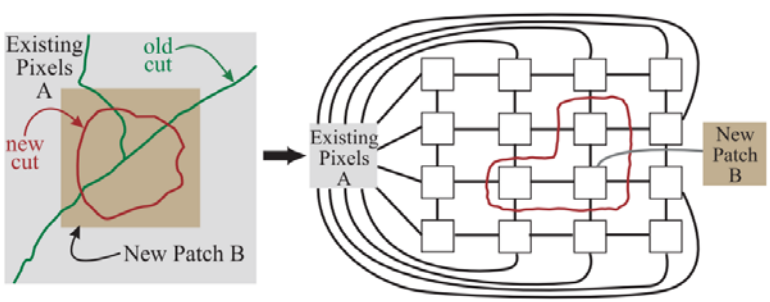
adjacent pixels



Old Seams



seam node



surrounded regions

Patch Placement and Matching

- Random placement
  - Random offset
- Entire patch matching
  - Pick the region with the largest error (cost of existing seams)
  - Decide the translation
    - Normalized SSD $$C(t) = \frac{1}{|\mathbf{A}_t|} \sum_{p \in \mathbf{A}_t} |\mathbf{I}(p) - \mathbf{O}(p+t)|^2$$
    - Pick the new patch location stochastically according to the probability $$P(t) \propto e^{-\frac{C(t)}{k\sigma^2}}$$
- Sub-patch matching

$$C(t) = \sum_{p \in \mathbf{S_O}} |\mathbf{I}(p-t) - \mathbf{O}(p)|^2$$

Extensions & Refinements

- Adapting the cost function by gradient magnitudes
  - Here, d indicates the direction of the gradient (direction of the edge between s and t)

$$M'(s,t,\mathbf{A},\mathbf{B}) = \frac{M(s,t,\mathbf{A},\mathbf{B})}{\|\mathbf{G}_{\mathbf{A}}^{d}(s)\| + \|\mathbf{G}_{\mathbf{A}}^{d}(t)\| + \|\mathbf{G}_{\mathbf{B}}^{d}(s)\| + \|\mathbf{G}_{\mathbf{B}}^{d}(t)\|}$$

  - To penalize seams going through low frequency region
- Feathering and multi-resolution spline
- FFT-based acceleration
  - Direct computation of SSD is slow
  - Use FFT

$$C(t) = \sum_{p} \mathbf{I}^2(p-t) + \sum_{p} \mathbf{O}^2(p) - 2\boxed{\sum_{p} \mathbf{I}(p-t)\mathbf{O}(p)}$$

Conclusion

- Stitch together patches of input image
  - At random or partly constrained (texture synthesis)
  - Constrained by another image (texture transfer)
- Image Quilting
  - No filters, no multi-scale, no one-pixel-at-a-time!
  - Fast and very simple
  - Results are not bad
- Graph Cuts
  - Fast
  - More flexible
  - Results are good

Another Interesting Approach

- *Textureshop: Texture Synthesis as a Photograph Editing Tool*
  - Fang & Hart
  - SIGGRAPH 2004