Advanced Image and Video Processing Algorithms

Projects

List available now

Project proposal (2 pages): 1st of June

- LaTeX Template will be made available

Project idea presentation: 8th of June

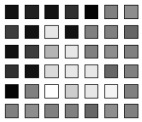
Final Project presentation: 20th of July

Project report

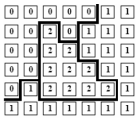
Advanced Processing Algorithms

- Graph Cuts
- Space-time manifolds
- Image-Based Priors

Image Segmentation



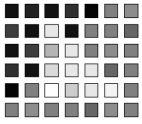
input image



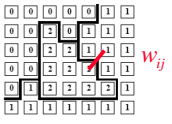
labeling

- How to find an optimal labeling?
- How to provide connected regions?

Image Segmentation

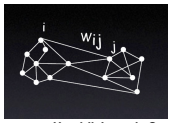


input image



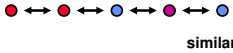
labeling

- based on similarity measure w_{ij}
- problem on graph
- look at two labels first



A Specialized Graph

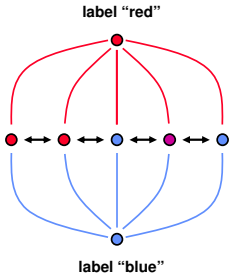
- n-links
- favor similar labels for neighboring pixels



similarity between pixels

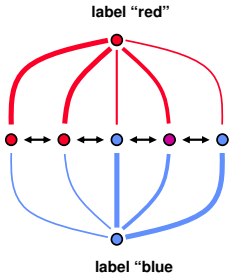
A Specialized Graph

- t-links: data term



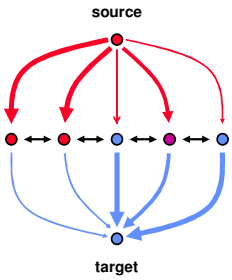
A Specialized Graph

- t-links: data term



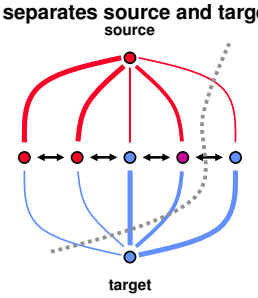
A Directed Graph

- minimize Potts energy $E(I) = \sum_{p \in P} |I_p - I_p^o| + \sum_{(p,q) \in N} K_{(p,q)} \cdot T(I_p \neq I_q)$



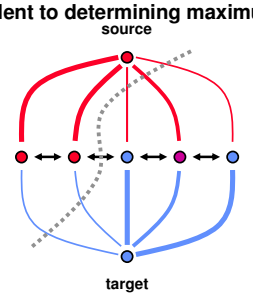
Graph Cut

- given graph G ={V,E} with capacities c(v1,v2)
- s-t cut separates source and target

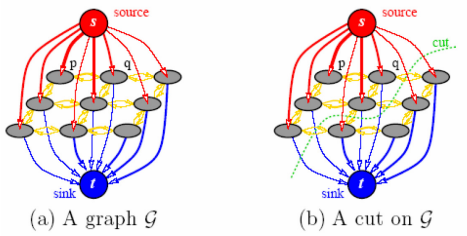


Minimum Cut

- determine cut with minimum energy
- equivalent to determining maximum flow

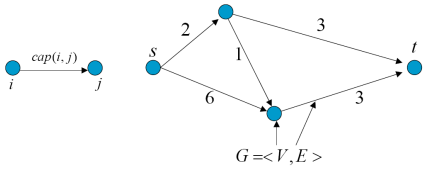


Graph Cut in 2D



s-t Graph

- A source node and a sink node
- Directed edge <i,j> from node i to node j
- Each arc <i,j> has nonnegative capacity c(i,j)
- cap(i,j) = 0 for non-existing arcs



Flow in s-t Graph

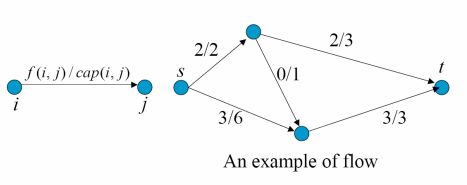
Flow is a real value function f that assign a real value f(i,j) to each arc such that

- capacity constraint: $f(i,j) \leq c(i,j)$
- mass balance constraint

$$\sum_{<i,j>\in E} f(i,j) - \sum_{<k,i>\in E} f(k,i) = \begin{cases} 0 & i \in V - \{s,t\} \\ |f| & i = s \\ -|f| & i = t \end{cases}$$

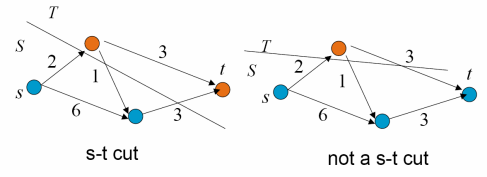
Flow in s-t Graph

- The maximum flow is the flow that has maximum value among all possible flow functions



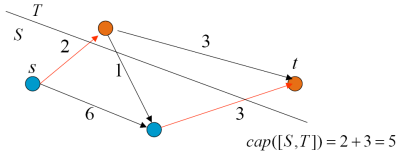
s-t Cut

- A cut is a partion of node set V which has two subsets S and T
- A cut is a s-t cut iff $s \in S, t \in T$



Capacity of s-t Cut (cost)

$$cap([S,T]) = \sum_{<i,j>\in E, i\in S, j\in T} cap(i,j)$$

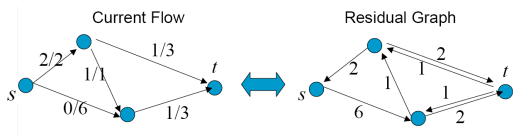


Minimum cut is the s-t cut whose capacity is minimum among all possible s-t cuts

Residual Graph

The residual Graph G_f :

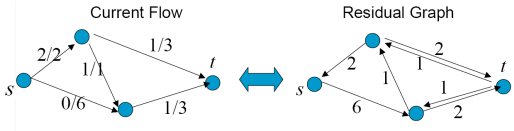
- ***cf(u,v) = c(u,v) – f(u,v)***
- **update flow along the inversed arc**



Residual Graph

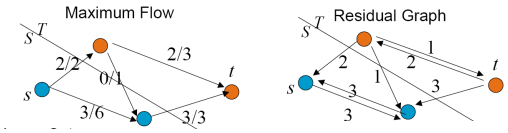
The residual Graph G_r :

- $cf(u,v) = c(u,v) - f(u,v)$
- update flow along the inversed arc



Max-Flow / Min Cut

- If there are no augmenting path -> max flow
- S is the set of all vertices that are reachable from s in residual graph



Ford-Fulkerson Algorithm

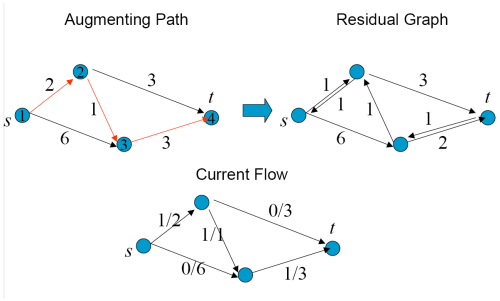
compute Residual Graph

repeat

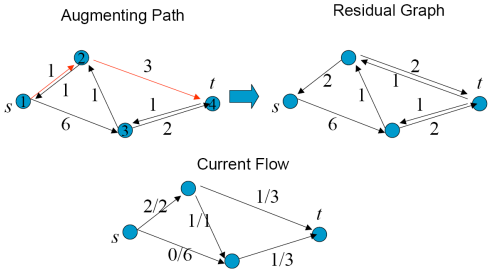
- find a direct path in Gf from source to sink in a depth-first search
- augment the flow of this path by an amount corresponding to the minimum residual capacity of the edges along this path
- update residual graph

until no augmenting paths exists

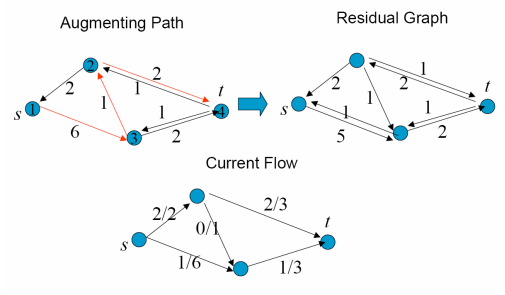
Ford-Fulkerson Algorithm I



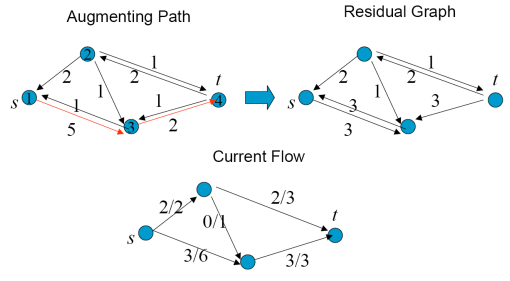
Ford-Fulkerson Algorithm II



Ford-Fulkerson Algorithm III



Ford-Fulkerson Algorithm IV



Ford-Fulkerson Algorithm IV

Further Demonstration

Extension to multiple labels

- start with random labeling
- iterative execution of the min-cut for each label α
- try to increase the number of α labels such that
  $E(f') \leq E(f)$
- α- expansion
- stop when no further update

[Boykov 01/02]

Applications

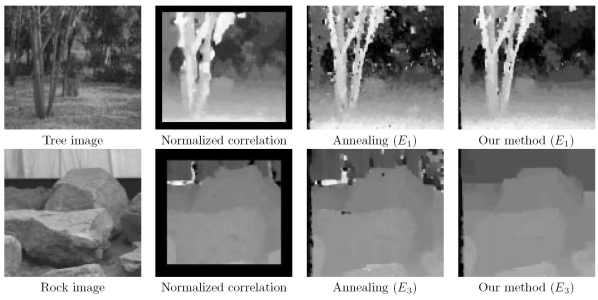
- denoising
- image-segmentation
- stereo
- texture synthesis
- ...

Denoising

(a) *Diamond* restoration (b) Original *Bell Quad* (c) "Restored" *Bell Quad*

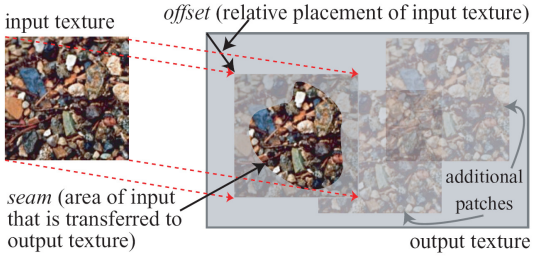
[Boykov 1999]

Stereo Reconstruction



[Boykov 1999]

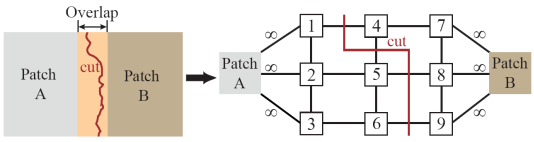
Patch-based Texture Synthesis



[Kwatra 03]

Patch-based Texture Synthesis

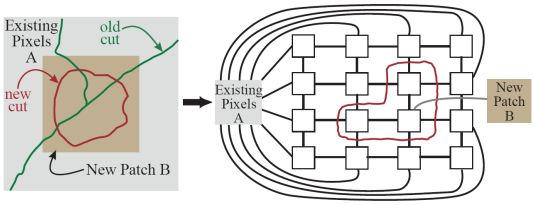
- minimize the energy across seems



[Kwatra 03]

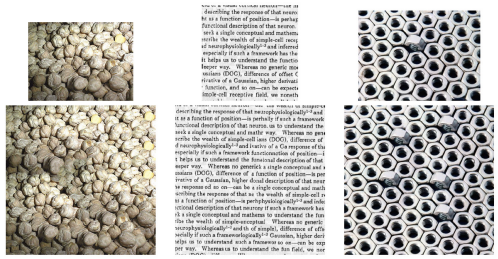
Patch-based Texture Synthesis

- insertion into existing region



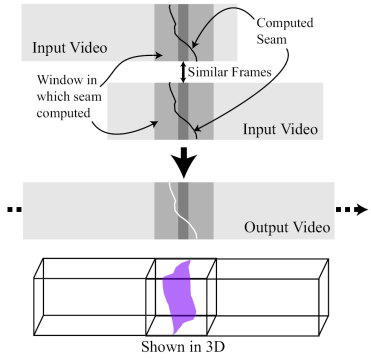
[Kwatra 03]

Texture Synthesis Results



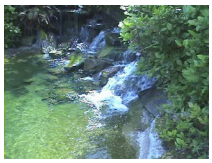
Texture Synthesis Results

Video Textures

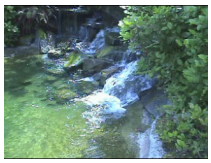


Video Texture Results

input



reconstruction



Space-time Manifolds

[Wexler 05]

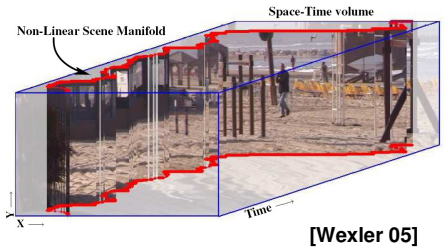
Space-time Manifolds

- how to extract this panorama?

[Wexler 05]

Space-time Manifolds

- determine optimal path through the space-time volume



[Wexler 05]

Algorithm

- transition from one column to the next within a frame is always possible
- transition from one frame to the next only allowed if some similar transition is found within one of the input frames:

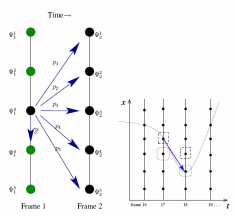
$$D(\psi_f^i, \psi_g^j) = \min_n \min_{UV \subset \Psi_n} ||UV - \psi_f^i \psi_g^j||$$

- simplified cost:

$$D(\psi_f^i, \psi_g^j) \leq C(\psi_f^i, \psi_g^j) = \min \left\{ \begin{array}{c} ||\psi_f^i \psi_g^j - \psi_f^i \psi_f^{i+1}||, \\ ||\psi_f^i \psi_g^j - \psi_g^{j-1} \psi_g^j|| \end{array} \right\}$$
$$\min \left\{ ||\psi_g^j - \psi_f^{i+1}||, ||\psi_f^i - \psi_g^{j-1}|| \right\}$$

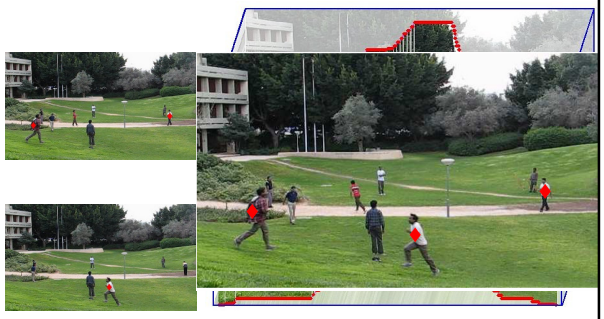
Algorithm

- find shortest path between selected start and end column



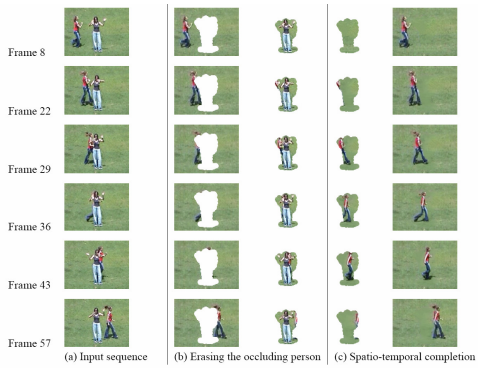
Space-time Manifolds Results

Space-time Manifolds Results



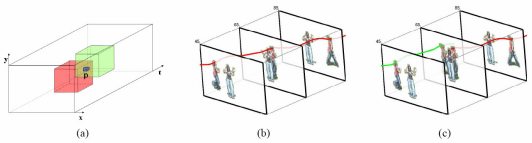
Hole Filling in Videos

Space-time Video Completion



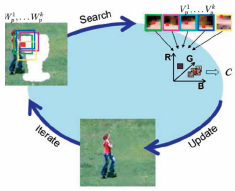
Local Space-time Consistency

- **fill up each voxel such that its space-time neighborhood matches a neighborhood in the input video**
- **make sure that this is true for all neighborhoods containing the voxel**



Algorithm

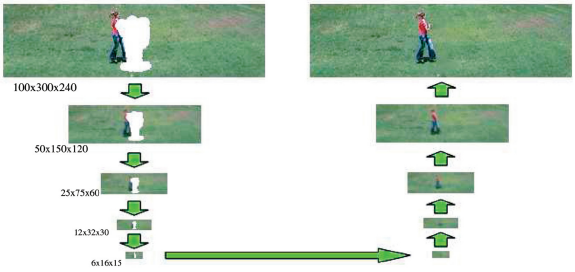
Input: video $\mathcal{S}$, hole $\mathcal{H} \subset \mathcal{S}$, database $\mathcal{D}$.
Compute space-time pyramids $\mathcal{S}_l$, $\mathcal{H}_l$, $\mathcal{D}_l$ $l = 1..L$.
$t \leftarrow 0$, $\mathcal{S}^t \leftarrow \mathcal{S}$
for pyramid level l, from top to bottom **do**
  repeat
    for all $p \in \mathcal{H}_l$ **do**
      Let $\{W_p^i\}_{i=1}^k$ be all windows s.t. $p \in W_p^i$
      Find $\{V^i\} \subseteq \mathcal{D}_l$ maximizing Eq. (2)
      Let $c^i \in V^i$ be the appropriate colors.
      Set $\omega_p^i = \alpha_p^i \cdot sim(W_p^i, V^i)$.
      $\mathcal{S}^{t+1}(p) \leftarrow ML(c^i, \omega_p^i)$ using Eq. (4)
    end for
    $t \leftarrow t + 1$
  until convergence
  Propagate solution to the next level
end for
Output: $\mathcal{S}^t$



$$sim(W_p, V_q) = e^{-\frac{ssd(W_p, V_q)}{2\sigma^2}}.$$

$$c = \frac{\sum_{i \in M} \omega_p^i c^i}{\sum_{i \in M} \omega_p^i}.$$

Hierarchical Approach



Results

Results