Modern Approaches for Media Retargeting

Ariel Shamir
The Interdisciplinary Center

Olga Sorkine
ETH Zurich

Alexander Hornung
Disney Research Zurich

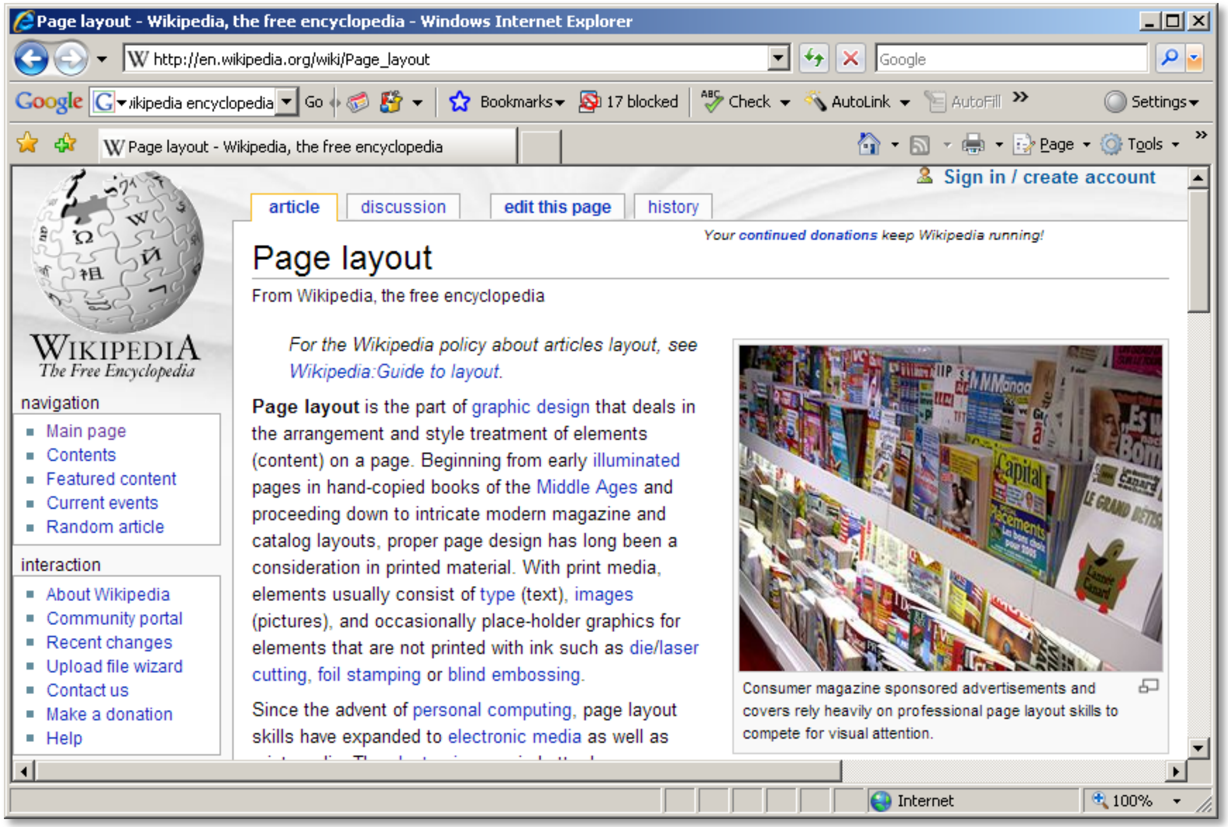
Overview

- Session 1
  - Introduction
  - Seam Carving
  - Saliency Measures

-- Break --

- Session 2
  - Discrete approaches
  - Continuous approaches

Motivation

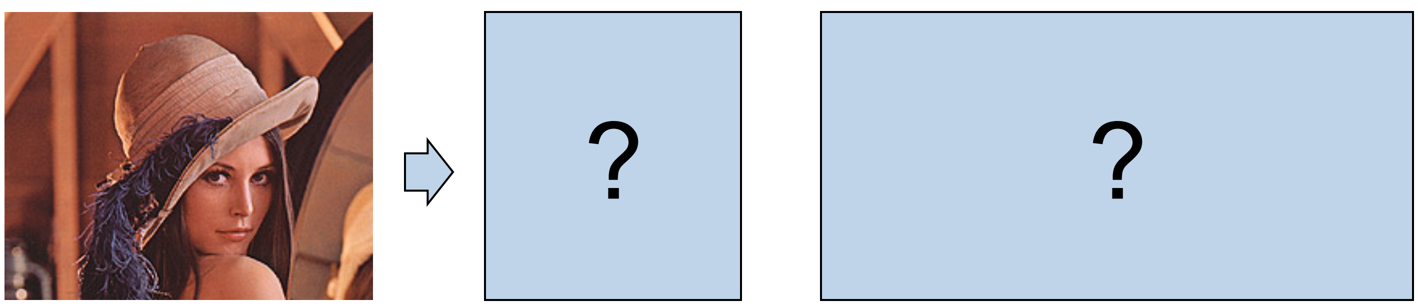


Page layout - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/Page_layout
Sign in / create account
article
discussion
edit this page
history
Your continued donations keep Wikipedia running!
Page layout
From Wikipedia, the free encyclopedia
For the Wikipedia policy about articles layout, see Wikipedia:Guide to layout.
Page layout is the part of graphic design that deals in the arrangement and style treatment of elements (content) on a page. Beginning from early illuminated pages in hand-copied books of the Middle Ages and proceeding down to intricate modern magazine and catalog layouts, proper page design has long been a consideration in printed material. With print media, elements usually consist of type (text), images (pictures), and occasionally place-holder graphics for elements that are not printed with ink such as die/laser cutting, foil stamping or blind embossing.
Since the advent of personal computing, page layout skills have expanded to electronic media as well as
Consumer magazine sponsored advertisements and covers rely heavily on professional page layout skills to compete for visual attention.
WIKIPEDIA
The Free Encyclopedia
navigation
Main page
Contents
Featured content
Current events
Random article
interaction
About Wikipedia
Community portal
Recent changes
Upload file wizard
Contact us
Make a donation
Help
Internet
100%

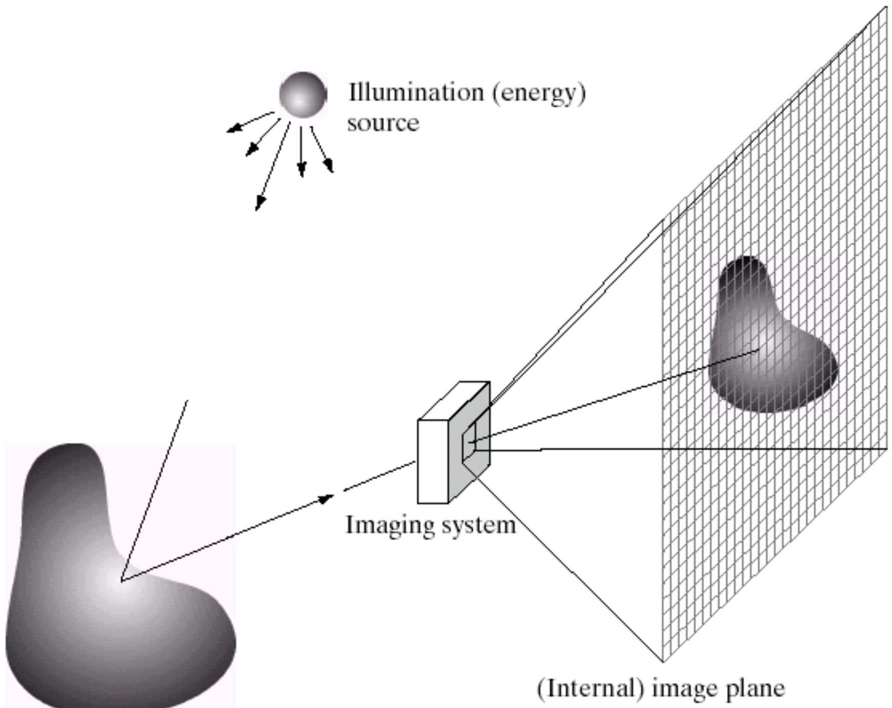
Motivation

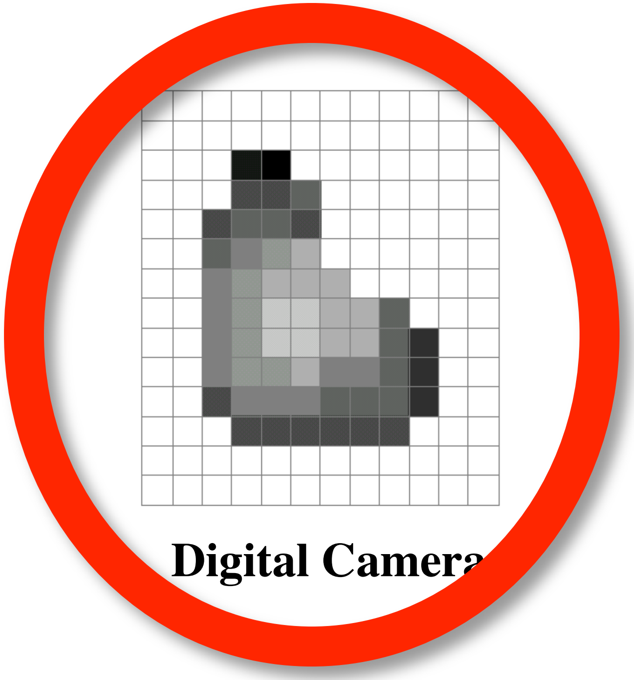
Retargeting

- Given the original media in size m$\times$n resize it to size m'$\times$n' where m' ≠ m or n' ≠ n or both.



What is an image?

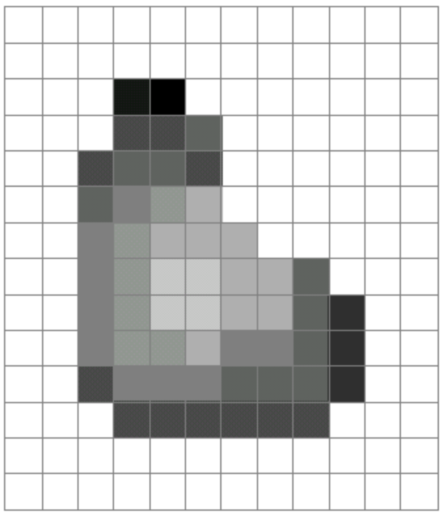




Source: A. Efros

A Grid of Intensity Values

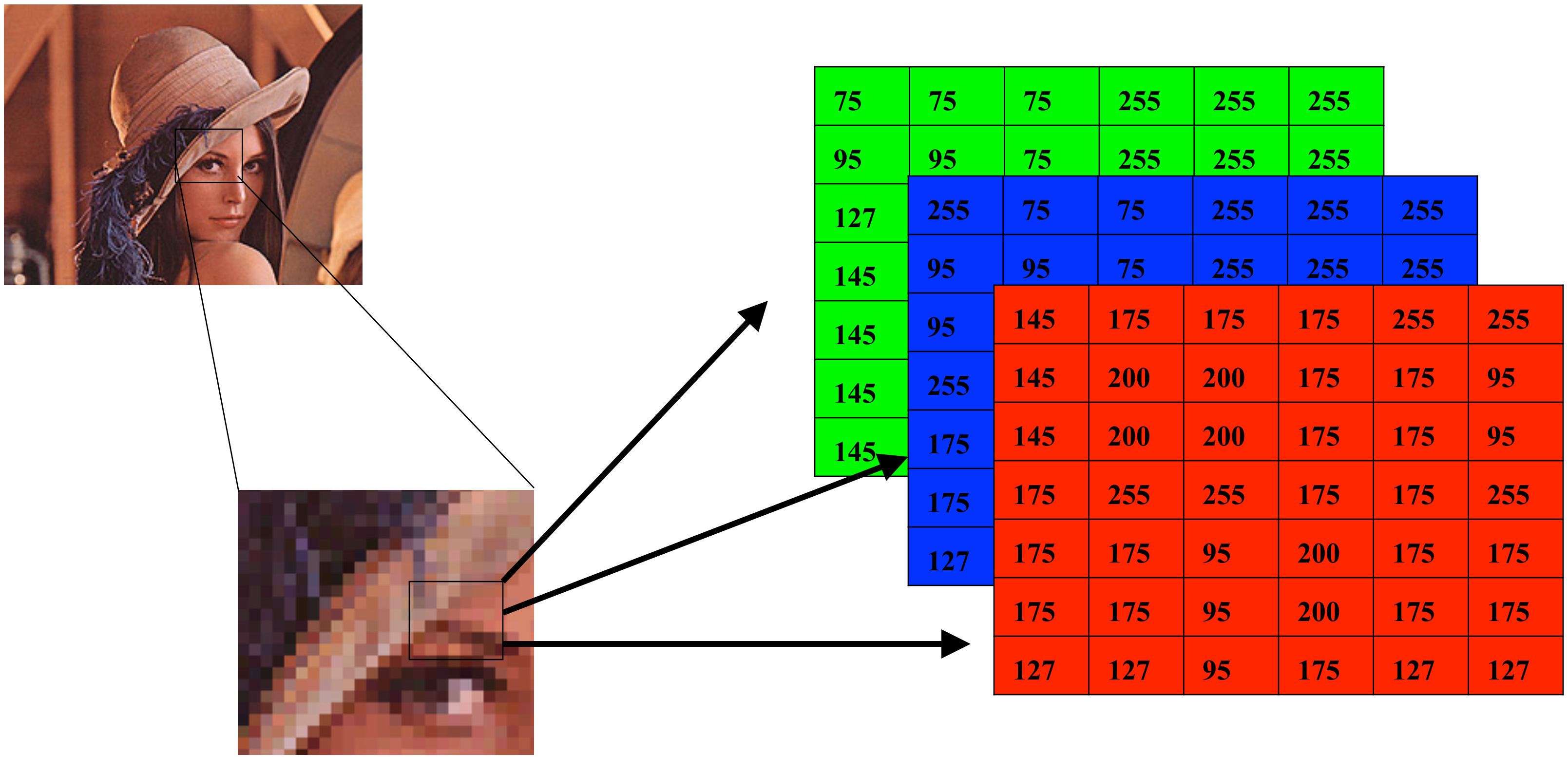
(common to use one byte per value: 0 = black, 255 = white)



=

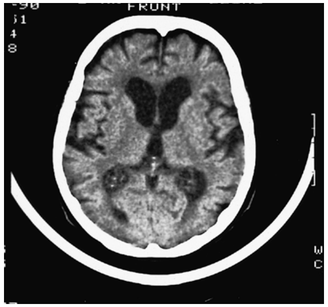
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	20	0	255	255	255	255	255	255	255
255	255	255	75	75	75	255	255	255	255	255	255
255	255	75	95	95	75	255	255	255	255	255	255
255	255	96	127	145	175	255	255	255	255	255	255
255	255	127	145	175	175	175	255	255	255	255	255
255	255	127	145	200	200	175	175	95	255	255	255
255	255	127	145	200	200	175	175	95	47	255	255
255	255	127	145	145	175	127	127	95	47	255	255
255	255	74	127	127	127	95	95	95	47	255	255
255	255	255	74	74	74	74	74	74	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255

Image as Grid of Values

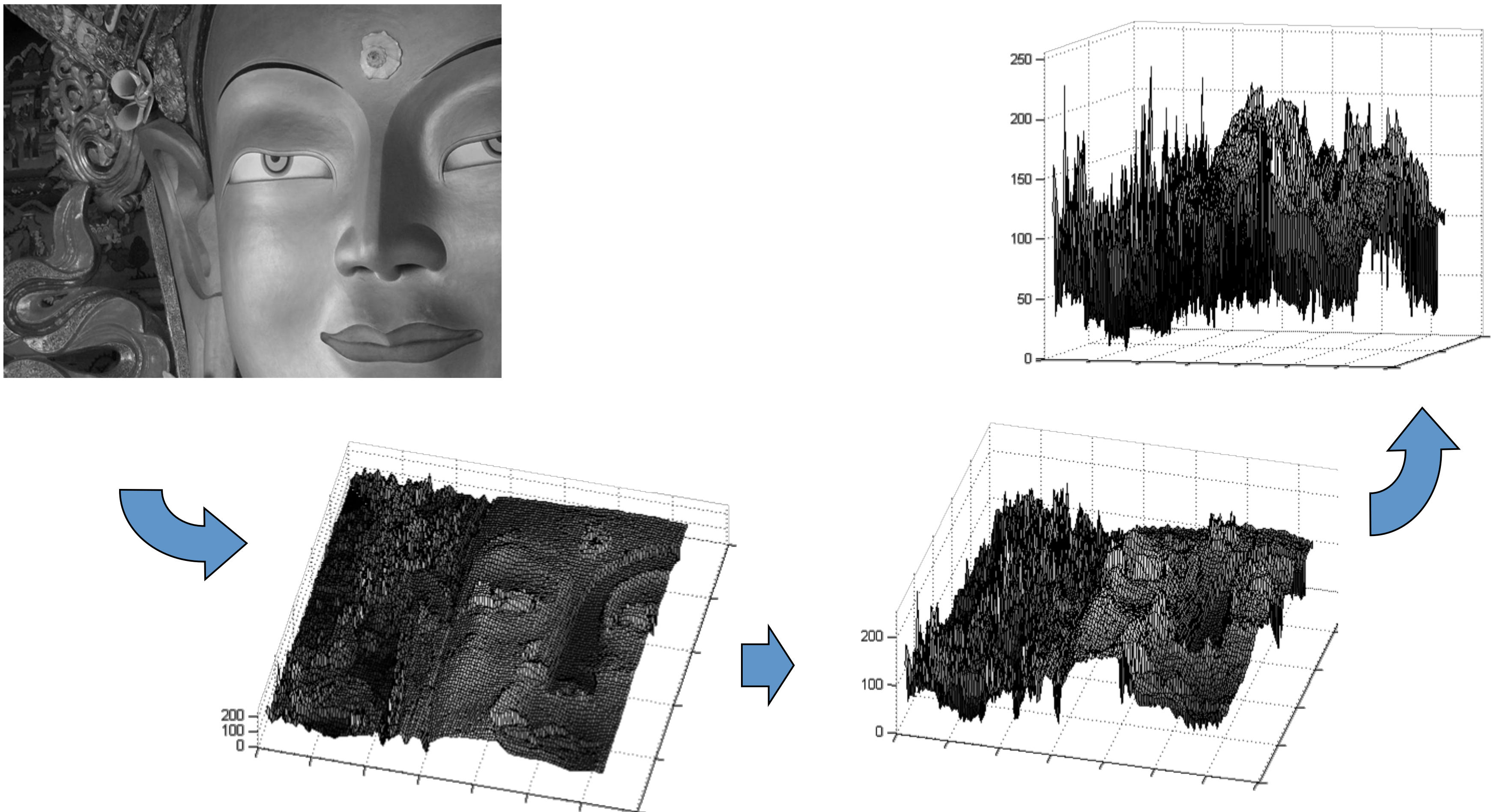


Images As Samples

- All images can in fact be seen as point sample representation of some function, but they are mostly defined on planar regular grids and we can assume some blending function which defines some function on the whole space.



An Image as a 2D Function



Basic Distinction: Discrete vs. Continuous

- Pixels are treated as discrete entities
- Pixels are treated as sample of a continuous function
- Following this we will see two major approaches for retargeting that we term "discrete" vs. "continuous"

Content Aware

Scale (continuous)

Crop (discrete)

Enlarging?

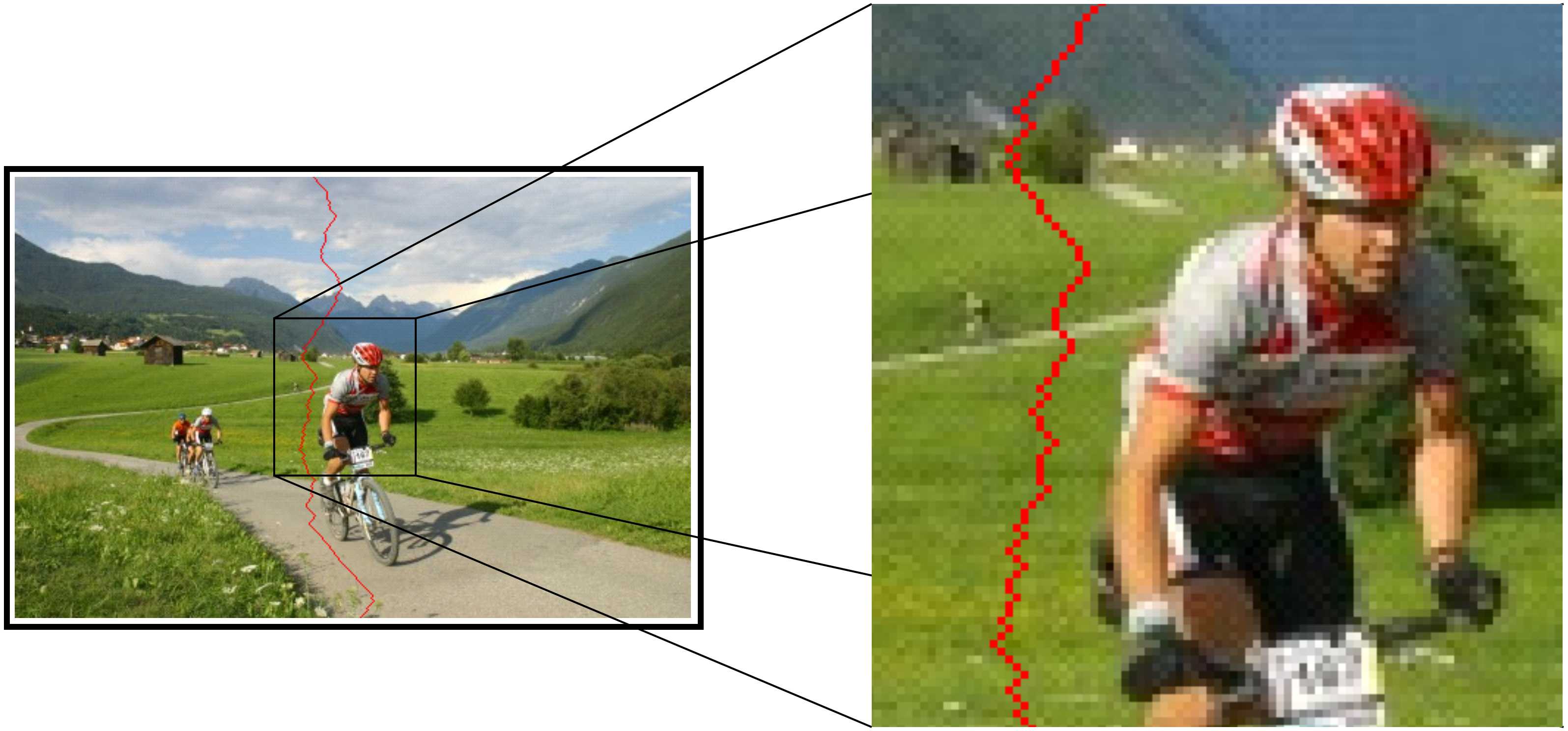
Key Idea: Content Aware

- Remove (or Insert) “less important” parts and preserve more important ones
- In effect this means we are creating ... ***<u>content aware</u>*** resizing

- **Key questions: what is important?**

Resizing?

A Seam

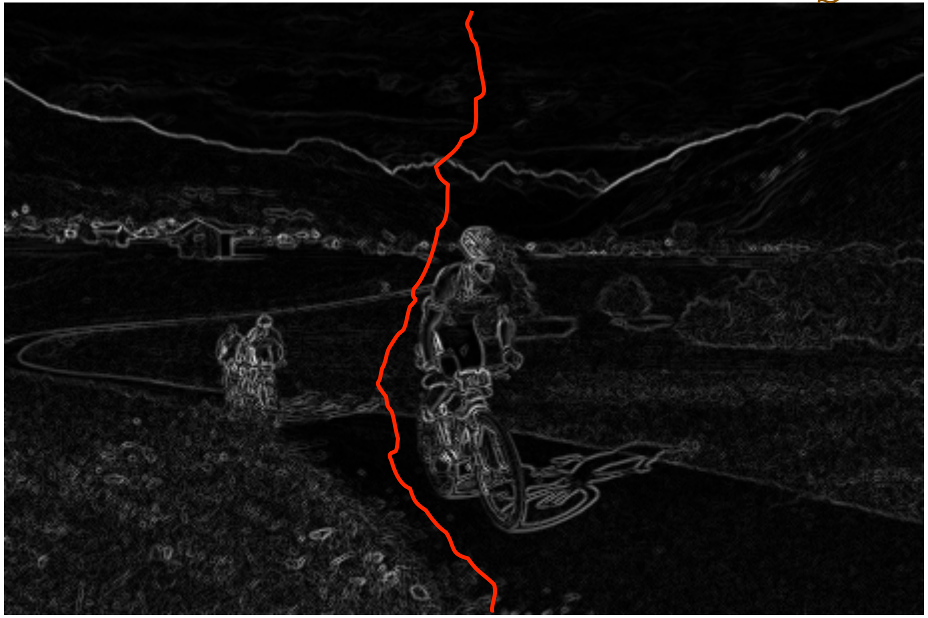


Seam Carving

Finding the Seam?

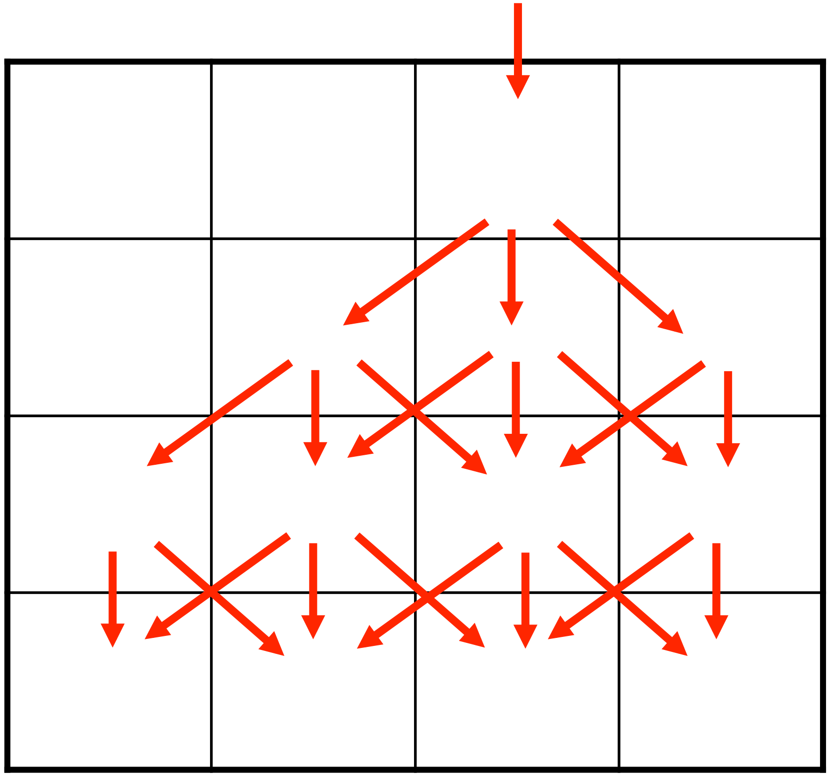
The Optimal Seam

$$E(\mathbf{I}) = \left|\frac{\partial}{\partial x}\mathbf{I}\right| + \left|\frac{\partial}{\partial y}\mathbf{I}\right| \Rightarrow s^* = \arg\min_s E(s)$$



Naïve Approach

- Loop over all seams and check their energy E(s). Choose the one with smallest energy.
- How many seams?
- Exponential (~3^h for wxh image)



However… Pixel Attributes → Dynamic Programming

$$M(i, j) = e(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1))$$

5	8	12	3
9	2	3	9
7	3	4	2
6	5	7	8

Dynamic Programming

$$M(i, j) = e(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1))$$

5	8	12	3
9	2+5	3	9
7	3	4	2
6	5	7	8

Dynamic Programming

$$M(i, j) = e(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1))$$

5	8	12	3
9	7	3+3	9
7	3	4	2
6	5	7	8

Dynamic Programming

$$M(i, j) = e(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1))$$

5	8	12	3
9	7	6	12
14	9	10	8
15	14	15	8+8

Searching for Minimum

$$M(i, j) = e(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1))$$

5	8	12	3
9	7	6	12
14	9	10	8
15	14	15	16

Backtracking the Seam

$$M(i, j) = e(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1))$$

5	8	12	3
9	7	6	12
14	9	10	8
15	14	15	16

Backtracking the Seam

$$M(i, j) = e(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1))$$

5	8	12	3
9	7	6	12
14	9	10	8
15	14	15	16

Backtracking the Seam

$$M(i, j) = e(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1))$$

5	8	12	3
9	7	6	12
14	9	10	8
15	14	15	16

H & V Cost Maps

Horizontal Cost

Vertical Cost

A Local Operator!

Aspect Ratio Change

Aspect Ratio Change

Original

Seam Carving

Scaling

Aspect Ratio Change

Cropping

Seams

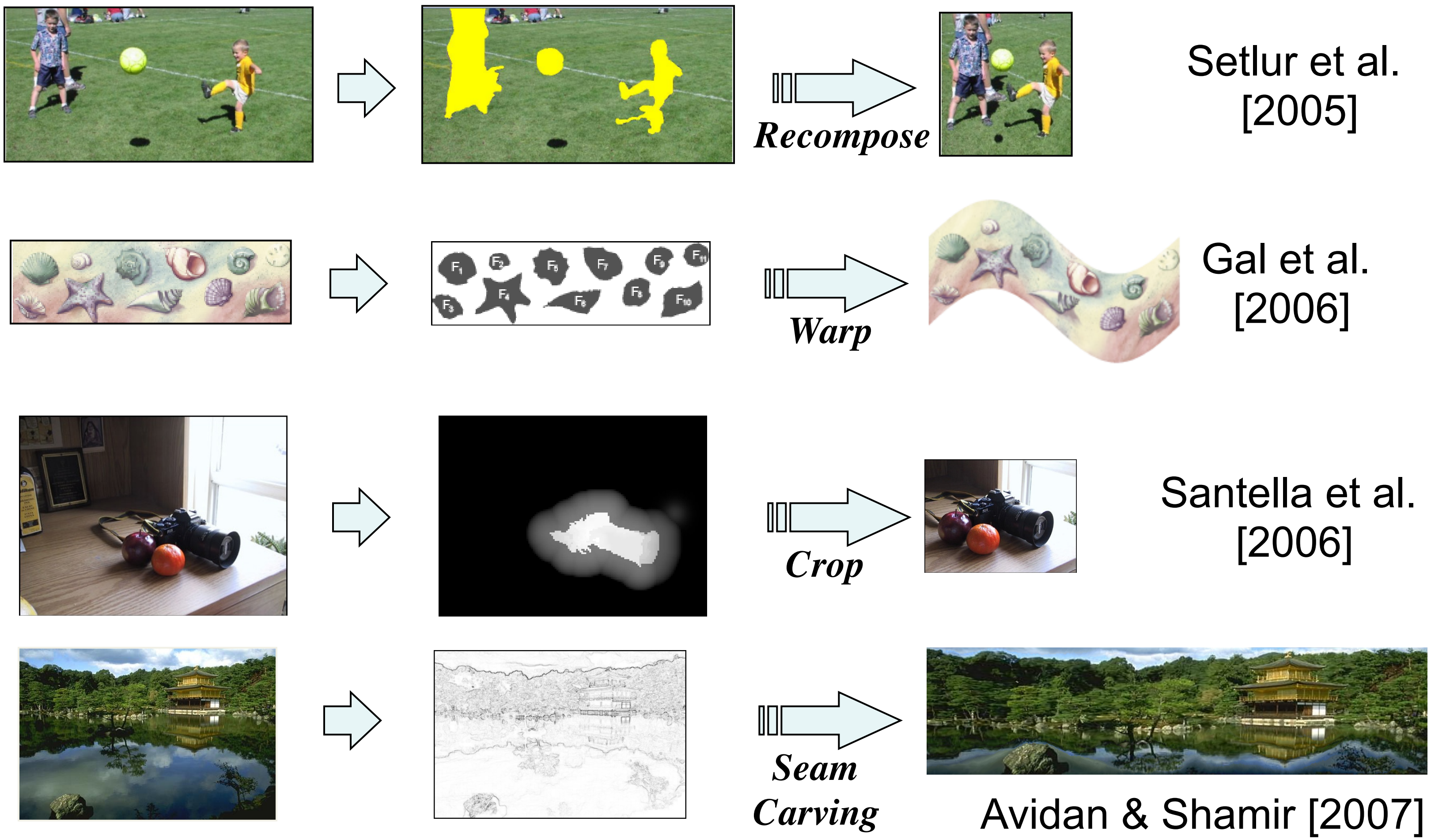
Scaling

Two Step Approach

1. Define what is important
2. Change the size by applying an operator

1. Define an energy function **E(I)** (interest, importance, saliency…)	⇨	2. Use some operator(s) to change the image **I**

General Scheme



Papers…

- Discrete methods
- Continuous methods
- Images
- Video
- Hybrid

Papers

- 2003*: Suh et al.* – Thumbnail creation
- 2003*: Chen et al.* – Cropping for Mobile
- *…*
- 2007: *Avidan & Shamir* – Seam carving
- 2007*: Wolf et al.* – Video
- 2008: *Wang et al*. – Scale & stretch
- …
- 2010: *Rubinstein et al.* – RetargetMe benchmark
- …
- 2012: *Panozzo et al.* – Axis-Aligned Deformation

SIGGRAPH ASIA2012

Embracing the digital convergence

CONFERENCE 28 Nov - 1 Dec
EXHIBITION 29 Nov - 1 Dec
Singapore EXPO

www.SIGGRAPH.org/ASIA2012

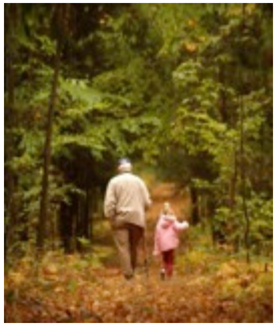
Visual Importance Measures

Alexander Sorkine-Hornung

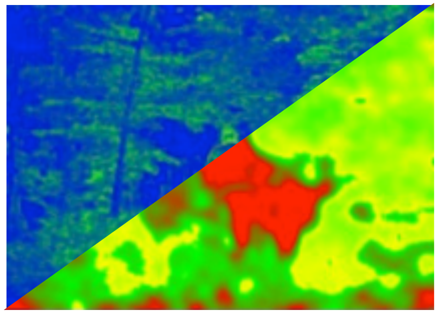
Disney Research

Visual saliency

- Content-aware rescaling
  - Preserve visually important parts of an image



- Importance map
  - Indicate how salient a pixel or area is



- Content-aware operators
  - Protect important areas
  - Allow deformations on less important parts

How to quantify visual importance?

- Dependent on many factors
- Subjective judgment
- Image semantics & context
- Application!
  - Image segmentation
  - Medical applications
  - Driving assistance systems
  - Advertising
  - Retargeting

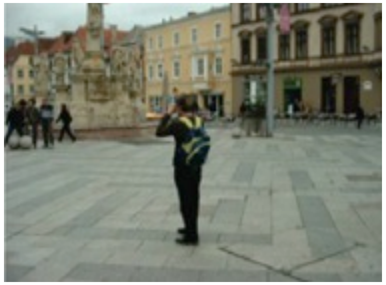
How to quantify visual importance?

- Eye tracking to measure attention
  - Few examples for retargeting
  - “Hot spots” only…
    important structures?
  - Does not tell us which regions to protect in order to avoid noticeable artifacts
- Need (preferably automatically and easily) computable measures

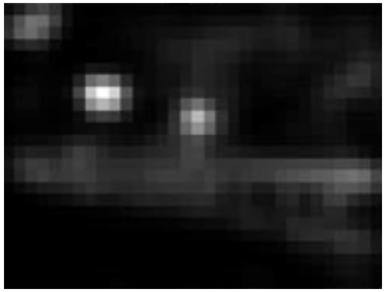
“Learning to predict where humans look”, Judd et al., ICCV 2009
“Using Eye-Tracking to Assess Different Image Retargeting Methods”, Castillo et al., APGV 2011

High- or low-level?

- Top-down, high level models
  - Need to be founded on neurosciences, biology, computer vision, …
  - Recent results combining learning and object detection for saliency



- Bottom-up, low-level stimuli driven
  - Successful / useful in many application scenarios (including retargeting)



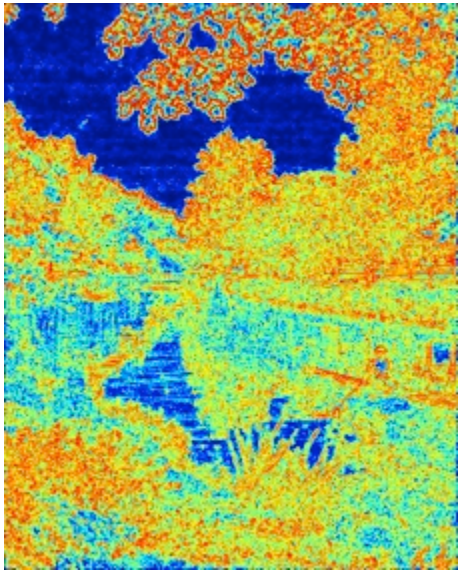
Low-level visual saliency

- Low level visual system processes basic features
  - Color, orientation of edges, direction of movement
- Perceptual research indicates that contrast is most influential factor
- Define various contrast measures
  - Intensity gradient, histograms, spectral properties, …
- Combine into saliency map
  - Winner-take-all, thresholds, nonlinear operations, …
- Simple definitions and efficient to compute

Intensity gradients

- Assumption: humans are sensitive to edges
- Saliency is simply the magnitude of gradients

$$L_1 : \left|\frac{\partial I}{\partial x}\right| + \left|\frac{\partial I}{\partial y}\right| \qquad L_2 : \left(\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2\right)^{1/2}$$



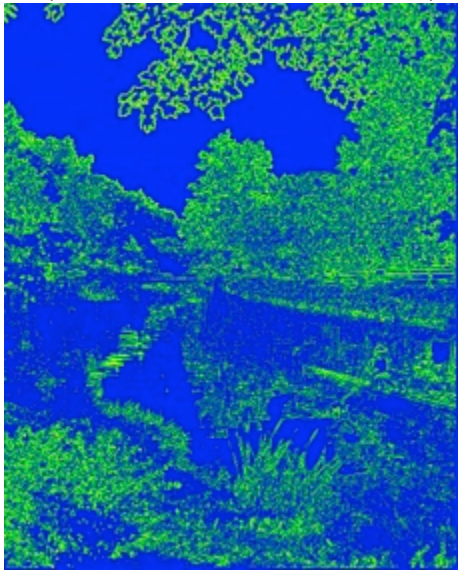
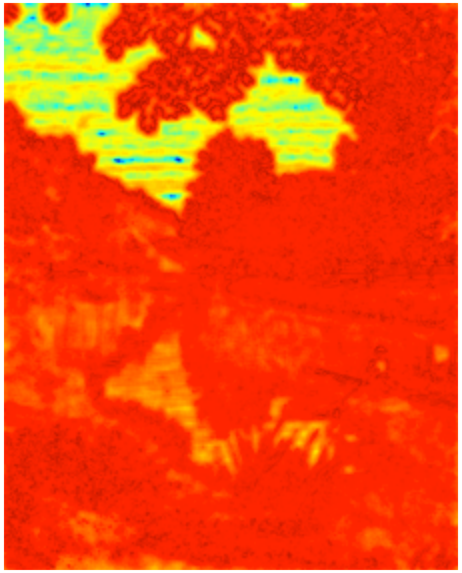


Image entropy

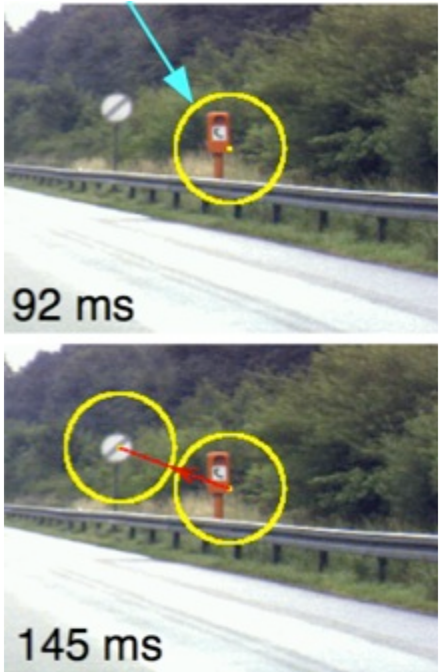
- Statistical measure of the intensity histogram
- For each pixel compute entropy $-\sum p \log_2(p)$ around it in a $k \times k$ window
- Measures how "busy" or textured the image is



- Gradients and entropy sensitive to noise and small scale detail

Low-level attention model

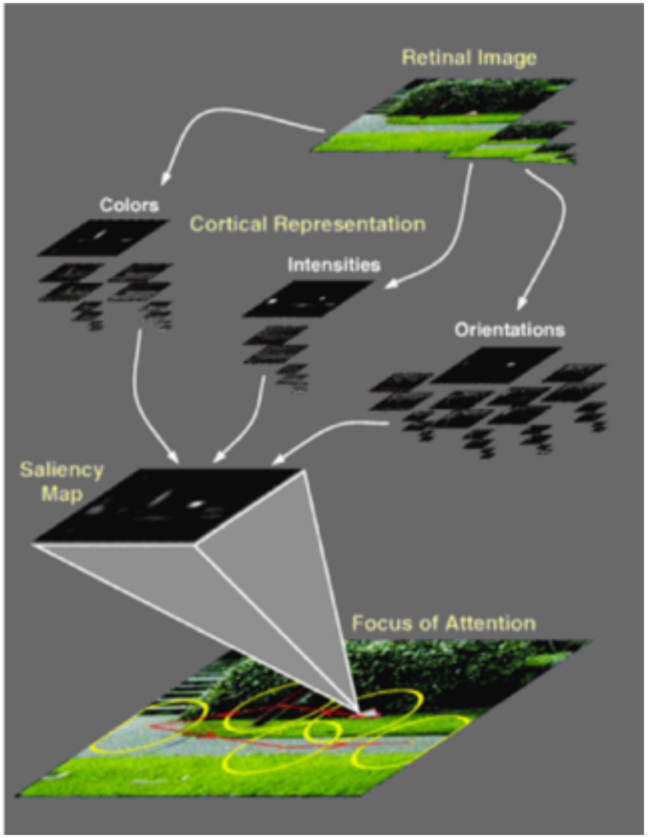
- Target application: rapid scene analysis
- React to basic stimuli
  - Inspired by neuronal architecture of early primate visual system
- Multi-scale image features
  - Color
  - Intensity
  - Orientations



"A model of saliency-based visual attention for rapid scene analysis", Itti et al., PAMI 1998

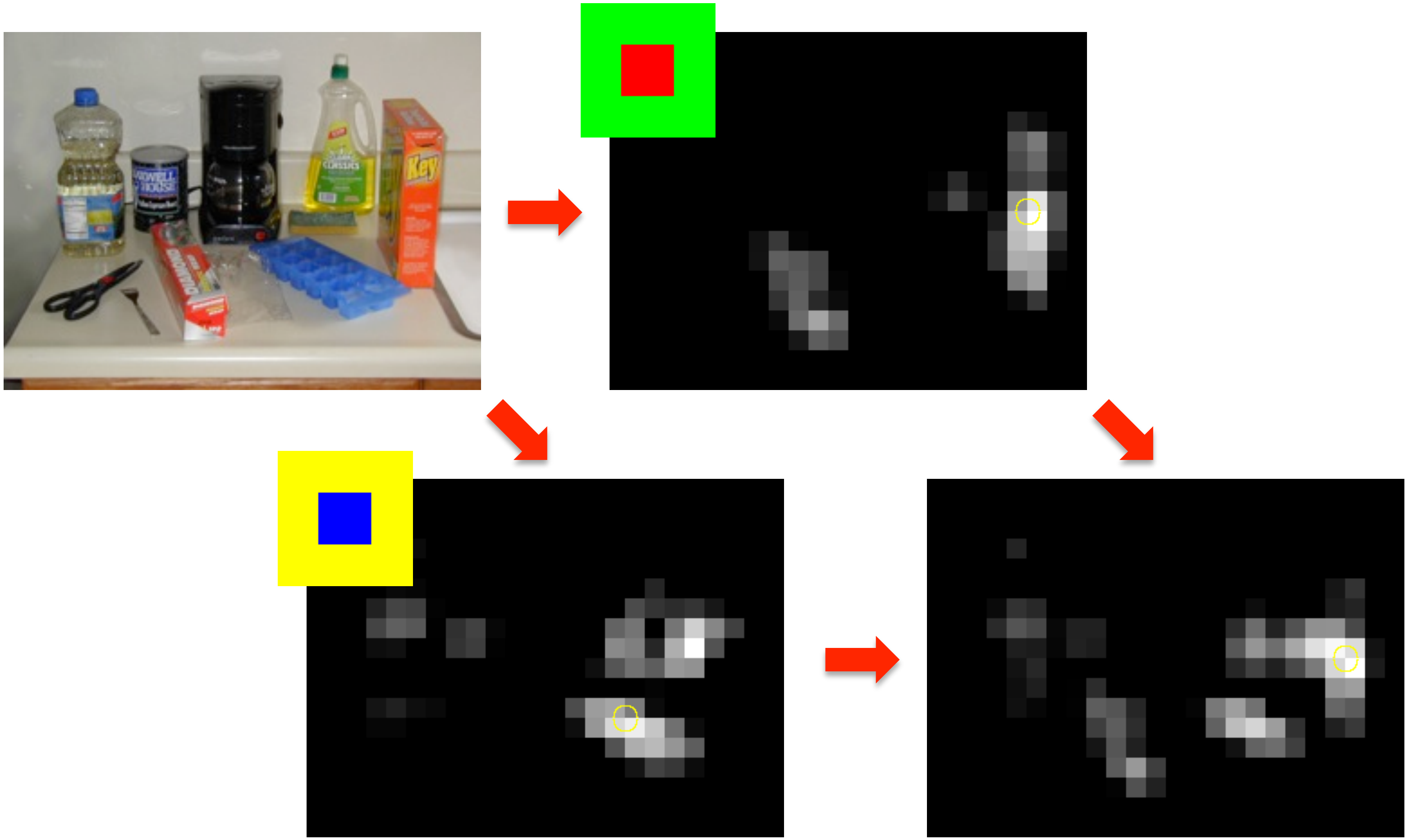
Low-level attention model

- Compute multi-res pyramid of the image
- On and between levels compute local filters like color differences, edges, etc.
- Combine response in saliency map

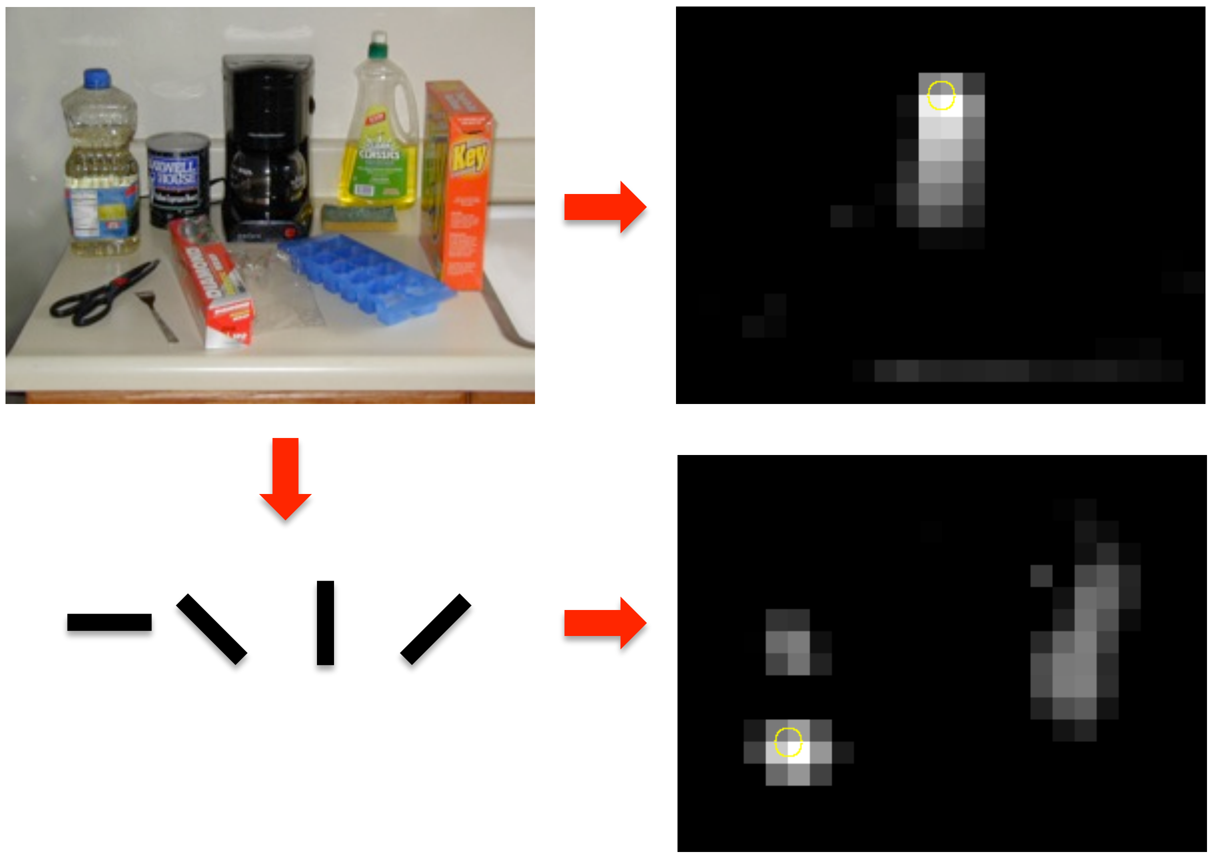


"A model of saliency-based visual attention for rapid scene analysis", Itti et al., PAMI 1998

Color contrast

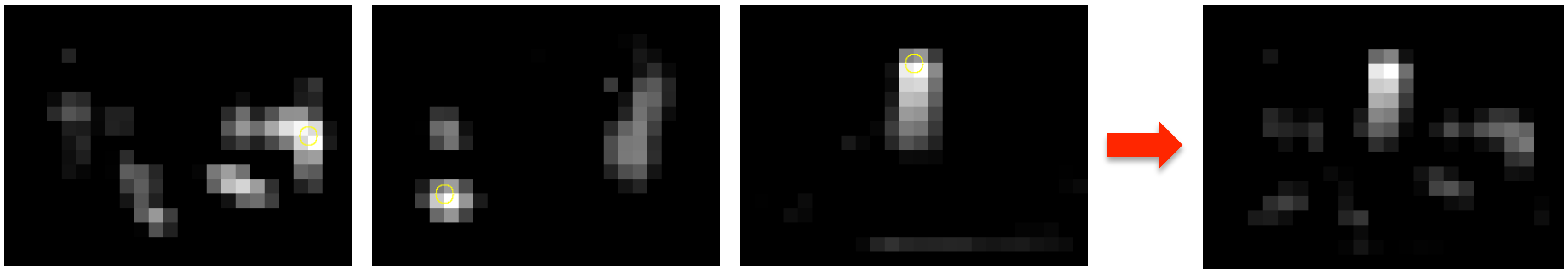


Intensity and orientations

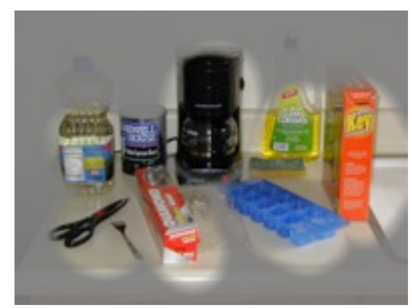


Low-level attention model

- Combine intermediate results for final saliency

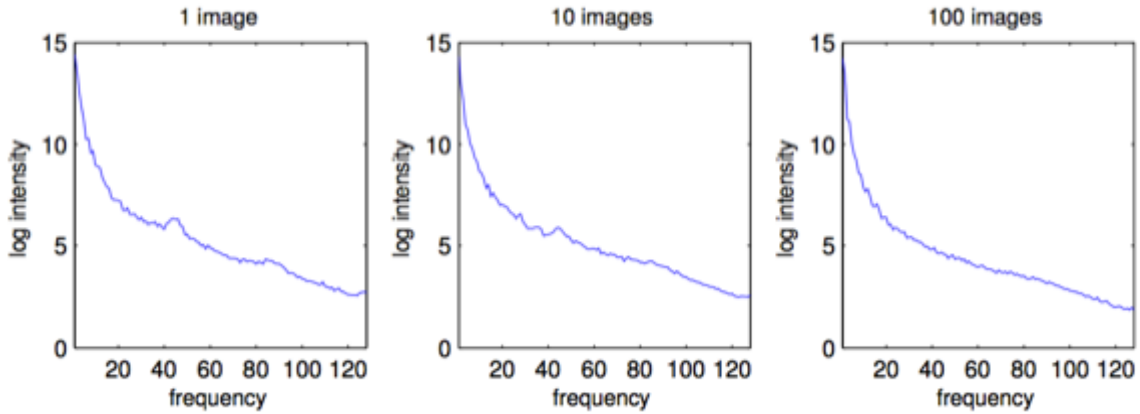


- Very efficient to compute
- Considers a more global scale (multi-res image pyramid)
- Quite coarse, blurry saliency maps
- No clear objects or structures



Spectral approaches

- Frequency spectra of natural images

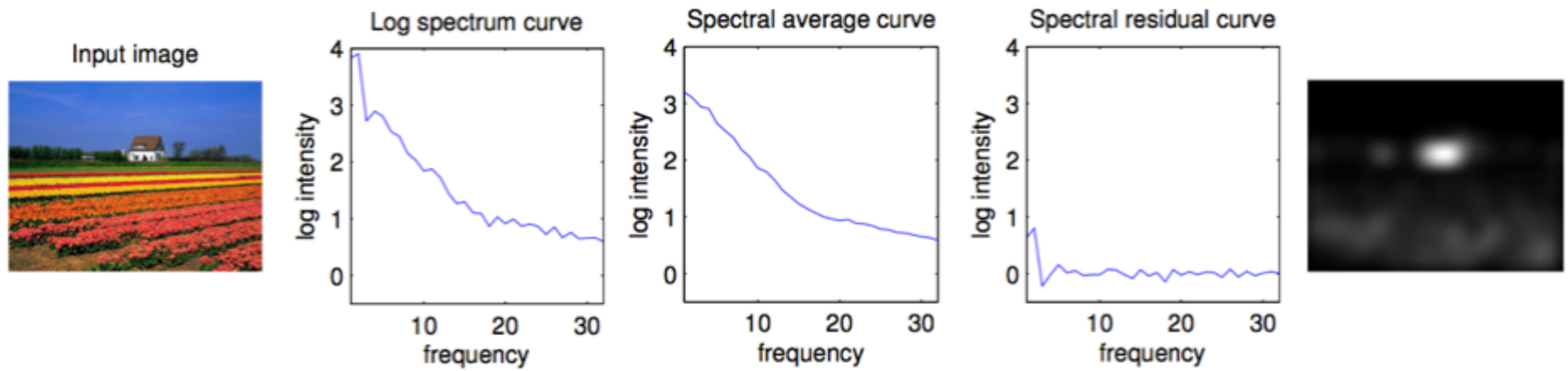


- Separate statistically redundant components from those carrying information
- Spectral singularities represent salient regions

“Saliency Detection: A spectral residual approach”, Hou and Zhang, CVPR 2007
“Spatio-temporal Saliency Detection using Phase Spectrum […]”, Guo et al., CVPR 2008

Spectral approaches

- Compute log spectrum and averaged log spectrum by convolution
- Saliency as spectral residual $R(f) = L(f) - A(f)$



Spectral approaches

- Often more "intuitive" response than Itti et al. (at least for retargeting)
- Efficient implementation
- Generally blurry and low resolution

"Saliency Detection: A spectral residual approach", Hou and Zhang, CVPR 2007
"Spatio-temporal Saliency Detection using Phase Spectrum […]", Guo et al., CVPR 2008

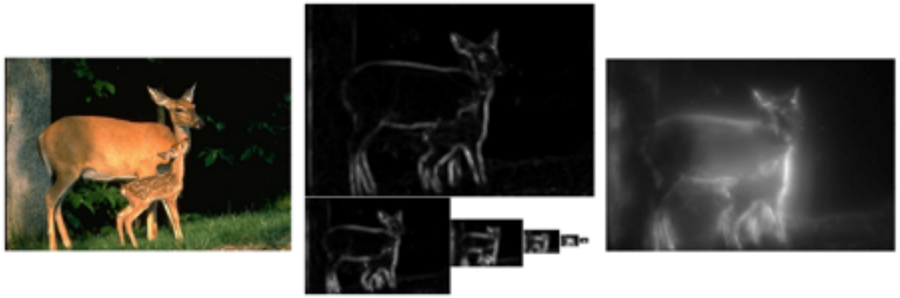
Learning combinations of features

- Large image database with ground truth labeling
- Compute set of features
  - Multi-scale contrast (edges at various scales)
  - Center surround histogram
  - Color spatial distribution
  - Captures local to global
- CRF to learn optimal linear combination of features

"Learning to detect a salient object", Liu et al., CVPR 2007

Learning combinations of features

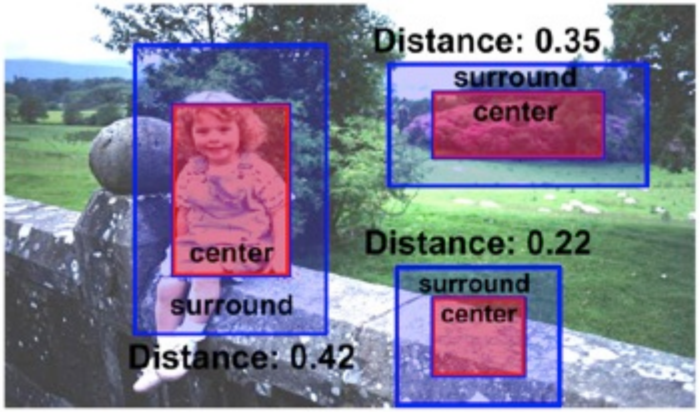
- Large image database with ground truth labeling
- Compute set of features
  - Multi-scale contrast (edges at various scales)
  - Center surround histogram
  - Color spatial distribution
  - Captures local to global
- CRF to learn optimal linear combination of features



"Learning to detect a salient object", Liu et al., CVPR 2007

Learning combinations of features

- Large image database with ground truth labeling
- Compute set of features
  - Multi-scale contrast (edges at various scales)
  - Center surround histogram
  - Color spatial distribution
  - Captures local to global



- CRF to learn optimal linear combination of features

"Learning to detect a salient object", Liu et al., CVPR 2007

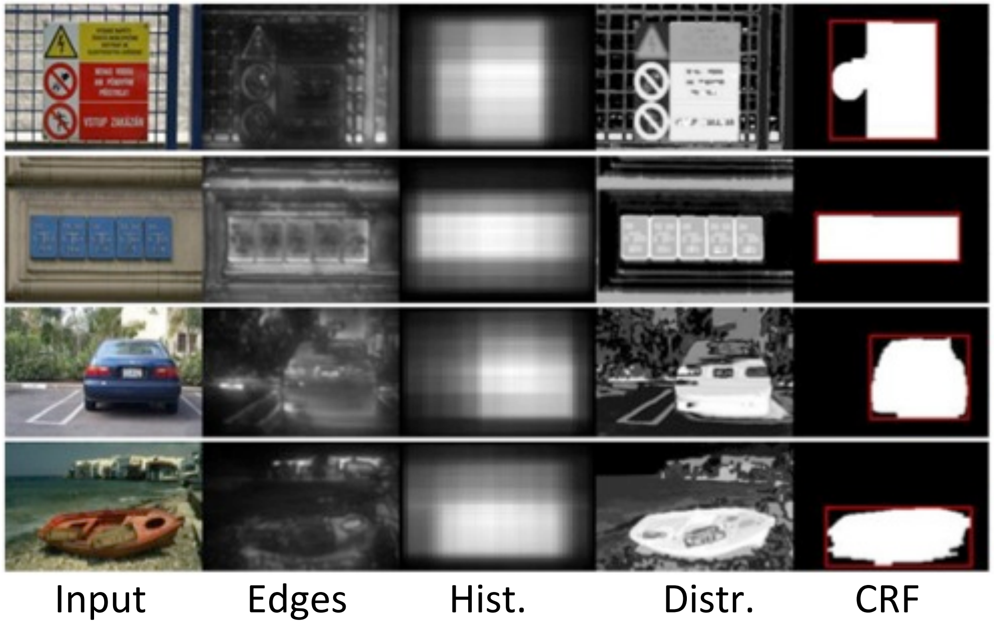
Learning combinations of features

- Large image database with ground truth labeling
- Compute set of features
  - Multi-scale contrast (edges at various scales)
  - Center surround histogram
  - Color spatial distribution
  - Captures local to global
- CRF to learn optimal linear combination of features

“Learning to detect a salient object”, Liu et al., CVPR 2007

Learning combinations of features

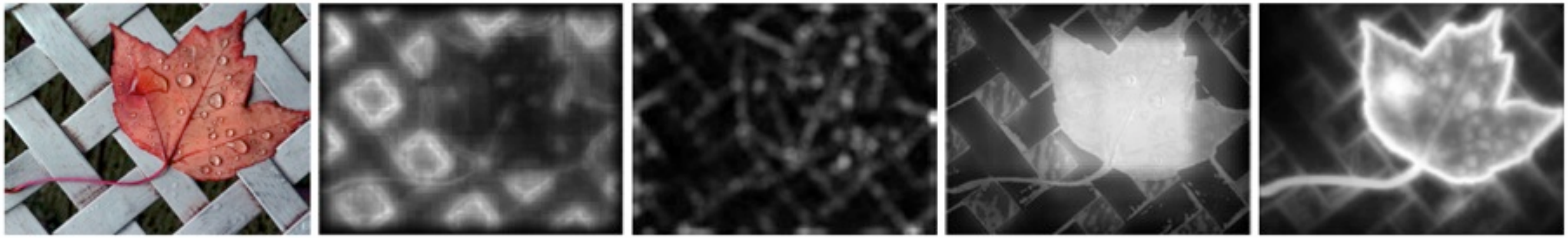
- Ground truth data
- Object segmentation rather than blurry attention maps
- Single salient object
- Sensitivity to high frequency content like edges or noise



"Learning to detect a salient object", Liu et al., CVPR 2007

Patch based approaches

- Consider also global image structures
  - Local: low-level contrast
  - Global: suppress frequently occuring content
  - Visual organziation: take context into account
- For each pixel, compare surrounding patch to K most similar patches at different scales

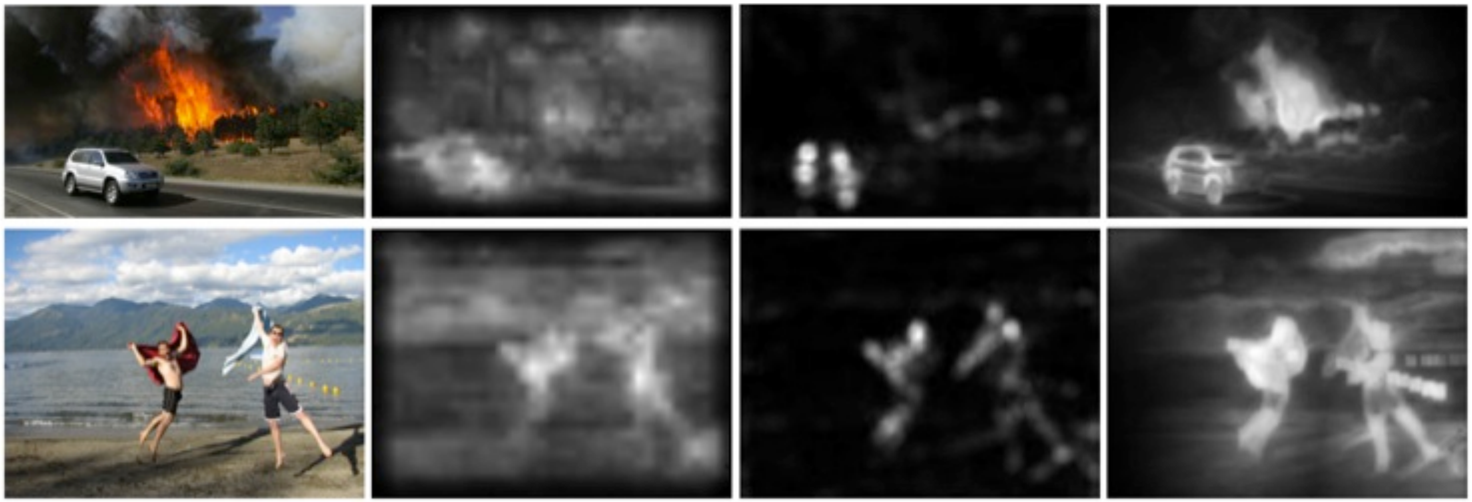


Input | Itti et al. | Spectral | Learning | Patch-based

“Context-Aware Saliency Detection”, Goferman et al., CVPR 2010

Patch based approaches

- No training required, context useful for retargeting
- Suffers from involved combinatorial complexity
  - Low resolution, may loose details

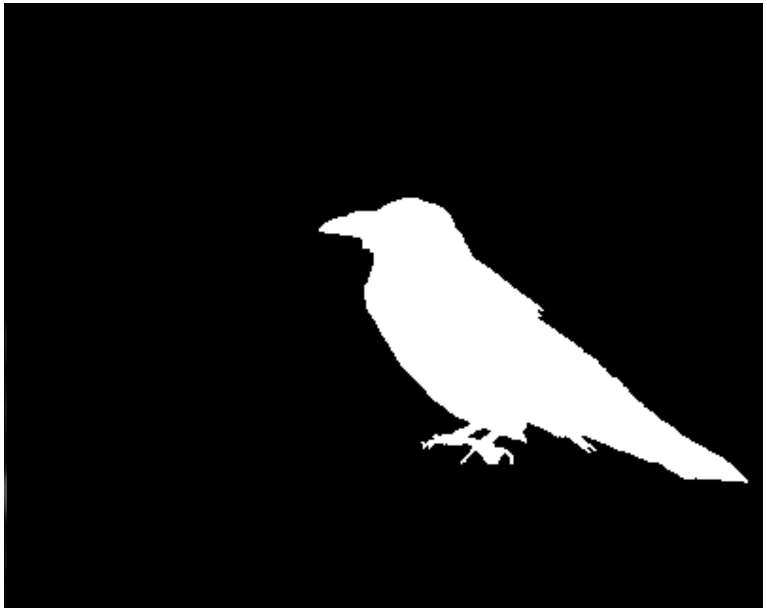


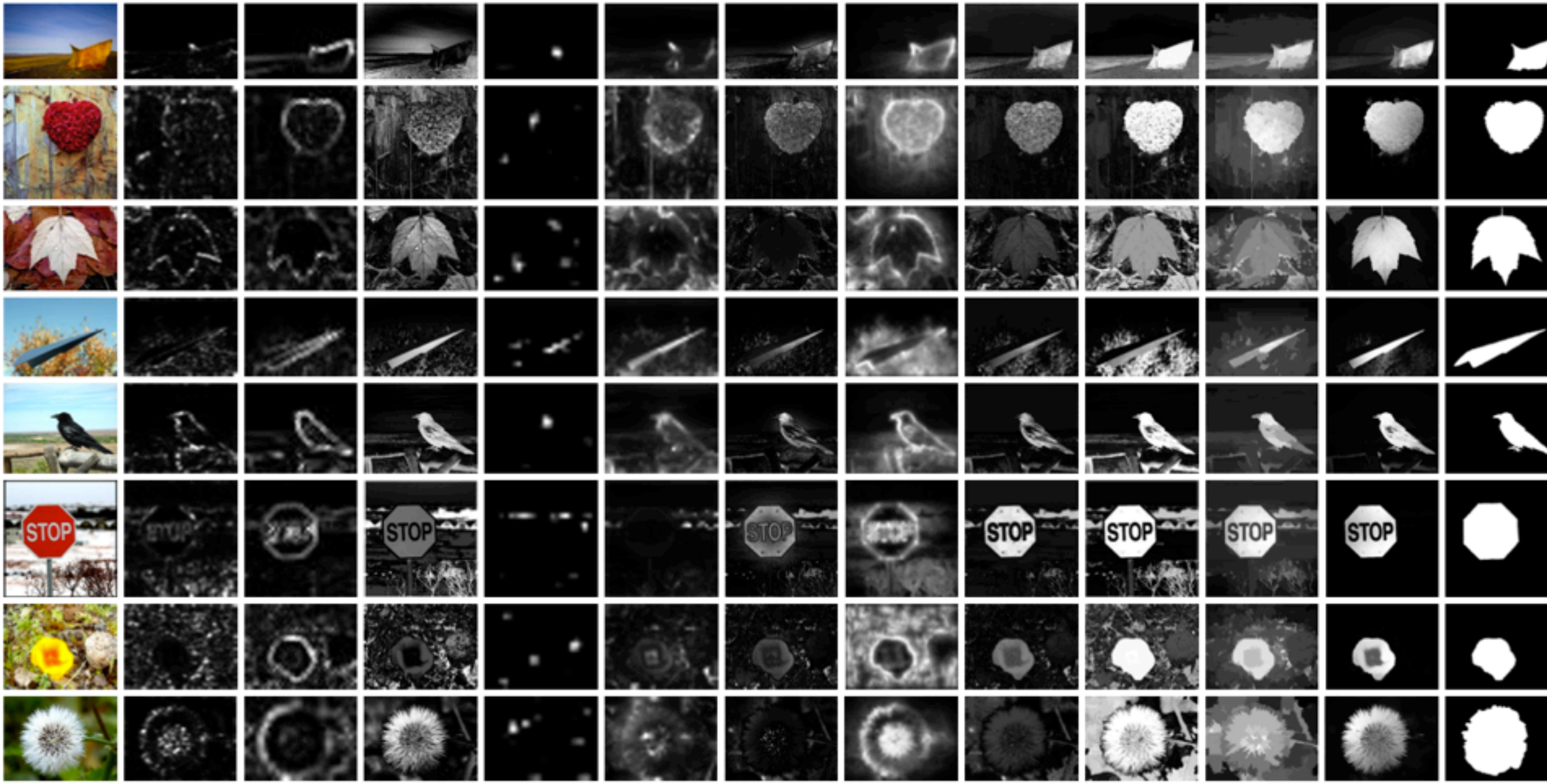
Input Itti et al. Spectral Patch-based

… and many more

- Definition and estimation of contrast based on various types of image features
  - Color variation of individual pixels
  - Histograms
  - Edges and gradients
  - Frequency spectra
  - Structure and distribution of image patches
  - Multi-scale descriptors
  - Combinations thereof
- Significance of those features unclear, similar approaches with considerably varying performance

Which one is best?

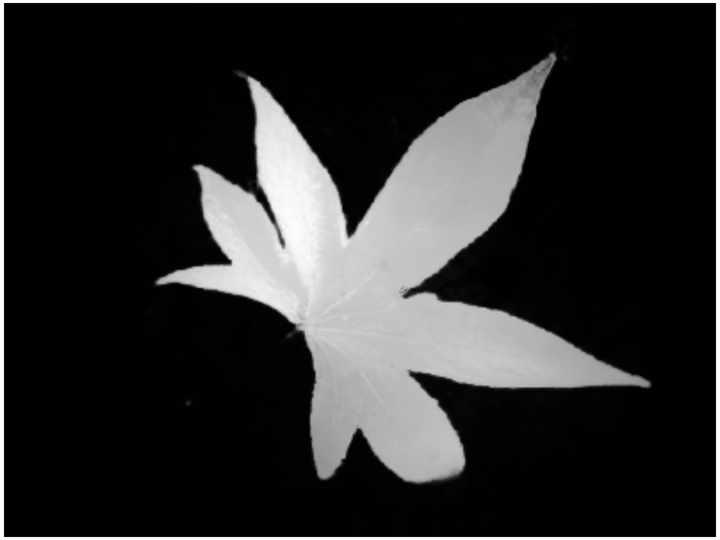




(a) SRC (b) SR [16] (c) MZ [22] (d) LC [29] (e) IT [18] (f) GB [15] (g) AC [1] (h) CA [12] (i) FT [2] (j) HC [7] (k) RC [7] (l) SF (m) GT

Saliency Filters
Contrast from basic image elements

- Reconsider relevance of individual contrast measures
  - Sensitivity to detail and noise
  - Larger-scale edges & global relations
- Abstraction
  - Decompose image into structurally representative elements
- Contrast
  - Uniqueness of elements
  - Spatial distribution of elements
- Up-sample to pixel-level



"Saliency Filters: Contrast Based Filtering for Salient Region Detection", Perazzi et al., CVPR 2012

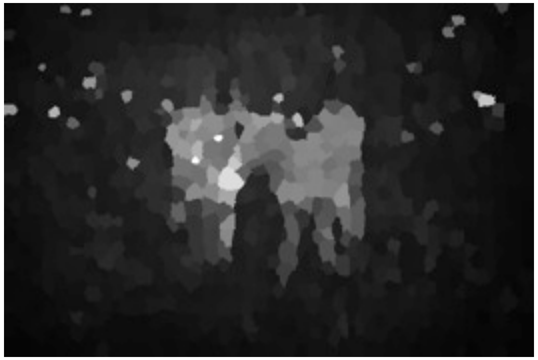
Abstraction

- Decompose image into elements that
  - Preserve relevant structure
  - Abstract undesirable detail
- Cluster pixels (e.g. based on color) into perceptually homogeneous regions
- Discontinuities between those regions should be preserved
- Constraints on shape and size
- Superpixel segmentation
  - k-means clustering in 5D space (CIE colors and position)
- Content-adaptive scale space

Element uniqueness

- Measure the "rarity" of an element
- Element color and position $\mathbf{c}_i$ $\mathbf{p}_i$
- Local/global control $w(\cdot)$
  - Radius of uniqueness operator
- Element uniqueness

$$U_i = \sum_{j=1}^{N} \|\mathbf{c}_i - \mathbf{c}_j\|^2 \cdot w(\mathbf{p}_i, \mathbf{p}_j)$$



- Problem: quadratic complexity
- Implemented as high dimensional Gaussian blurring

Element distribution

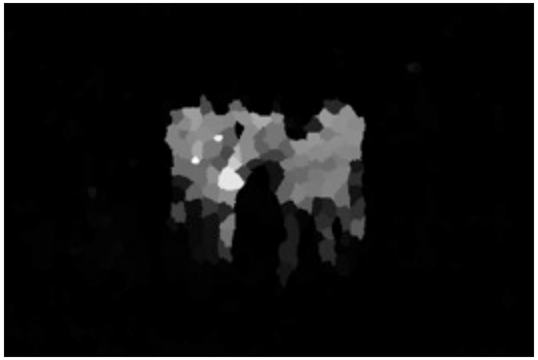
- Unique elements not always salient
  - Background colors distributed
  - Foreground colors more compact
- Spatial variance of the color of a segment

$$D_i = \sum_{j=1}^{N} \|\mathbf{p}_j - \mu_i\|^2 w(\mathbf{c}_i, \mathbf{c}_j)$$

- Weighted mean of similar elements

$$\mu_i = \sum_{j=1}^{N} w(\mathbf{c}_i, \mathbf{c}_j)\mathbf{p}_j$$

- Again evaluated by Gaussian blurring



Saliency assignment

- Combined element saliency

$$S_i = U_i \cdot \exp(-k \cdot D_i)$$

- Segmentation-based techniques often limited to elements
- Upsample to pixel resolution
  - Recover abstracted detail
  - Weighted combination of elements
  - Does not carry over segmentation errors
- Again, Gaussian filtering

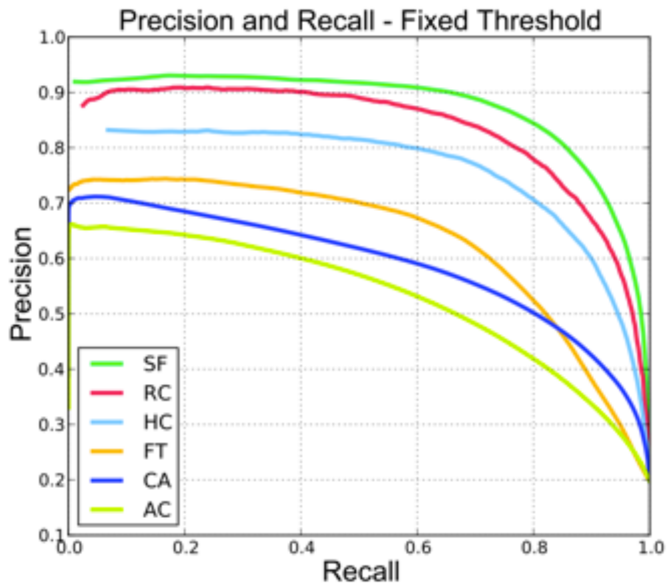
$$\tilde{S}_i = \sum_{j=1}^{N} w_{ij} S_j$$

Evaluation

- Closest to user segmentation
- Comparably simple algorithm
  - Abstraction
  - Two contrast measures
  - Up-sampling
- How to evaluate saliency maps?
  - Precision and Recall
  - Mean absolute error

“A database of human segmented images”, Martin et al., ICCV 2001
“Frequency-tuned salient region detection”, Achanta et al., CVPR 2009

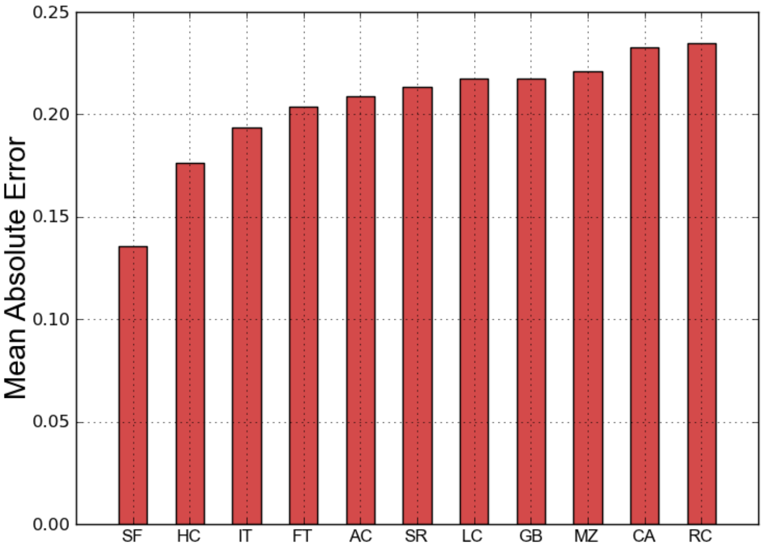
Precision and Recall



- Precision
  - How many of computed salient pixels are actually salient?
- Recall
  - How many of actually salient pixels were computed?
- For attention detection precision is most important
- For retargeting we need both
- Adaptive thresholds and F-measure

Mean Absolute Error

- Precision and Recall does not consider the true negative saliency assignments
- P&R favors methods that are
  - Successful with salient pixels
  - May fail to correctly identify non-salient regions
- Evaluate mean absolute error between continuous saliency map and binary ground truth
- Different results than P&R!



Limitations

- Low-level measures do not necessarily correspond to human attention

- Is segmentation of the most salient object best choice for retargeting?

"Learning to predict where humans look", Judd et al., ICCV 2009
"Using Eye-Tracking to Assess Different Image Retargeting Methods", Castillo et al., APGV 2011

High level saliency

- Global / semantic / structural information about image content
  - Line detection
  - Symmetries
  - Face / object detection
  - User input

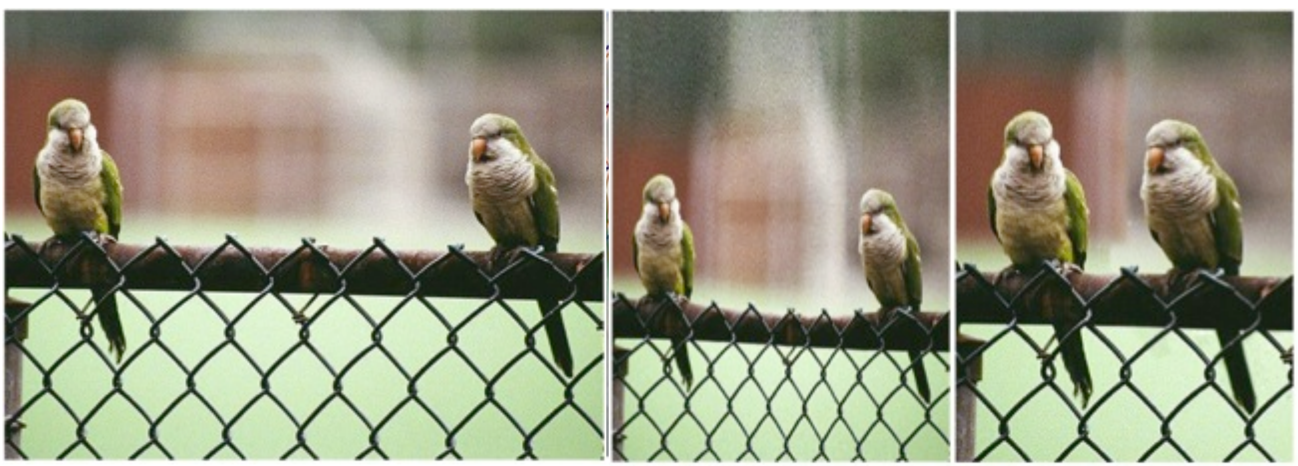
Line detection

- Lines are prominent features, especially when man-made objects are present in the image
- Local measure of line strength
- Similarity transforms for lines

“A system for retargeting of streaming video”, Krähenbühl et al., SIGGRAPH Asia 2009
“A line structure preserving approach to image retargeting”, Chang and Chuang, CVPR 2012

Symmetry detection

- Regular repeating structures become very "salient" when broken
- Analyze translational symmetries and detect underlying lattice structures



"Resizing by symmetry-summarization", Wu et al., SIGGRAPH Asia 2010

Face detection

- Predefined class of objects like faces
- Object detection using machine learning
  - Classifiers trained on a dataset
  - Frameworks such as Viola-Jones
  - OpenCV
- Simply "protect" detected faces in the output saliency map

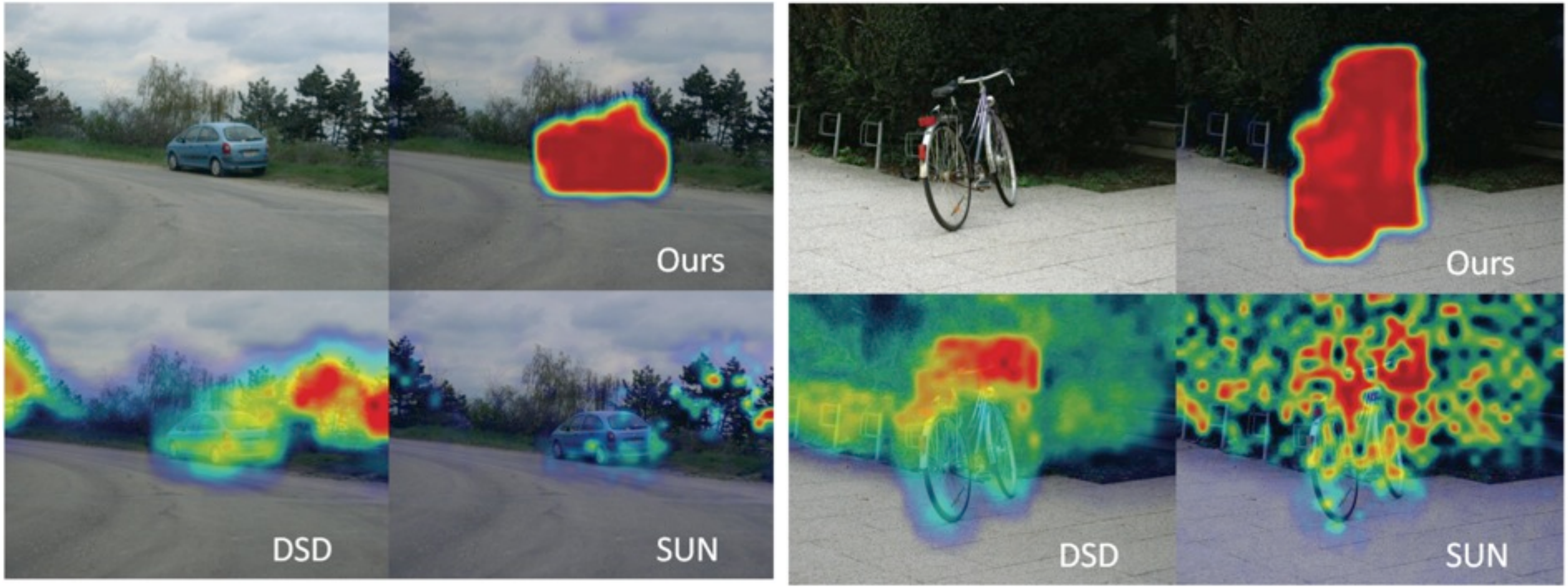
Saliency based on object detection

- Cluttered scenes with occlusions and various objects
- Combine feature learning and saliency computation
  - For each target object class learn a dictionary of patches
  - CRF for spatial consistency

Input Itti et al. Spectral "Bike" "Person"

"Top-Down Visual Saliency via Joint CRF and Dictionary Learning", Yang and Yang, CVPR 2012

Saliency based on object detection

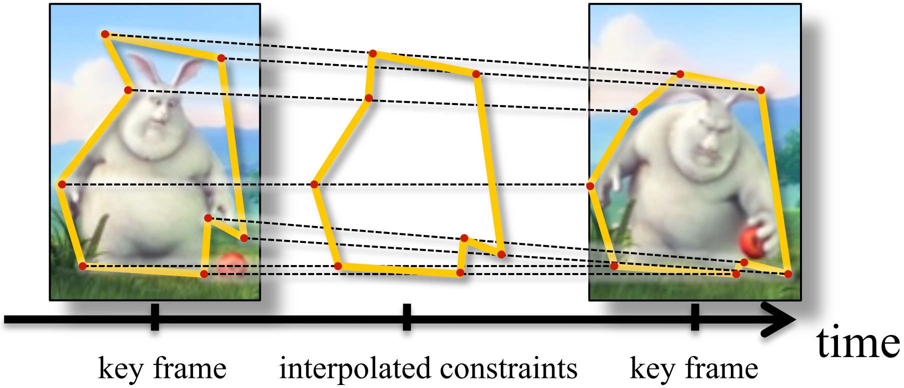


Saliency based on object detection

- Boundary / differences between object detection and saliency?
- Hard to learn everything
  - What objects are important to be preserved?
  - Still no global structures

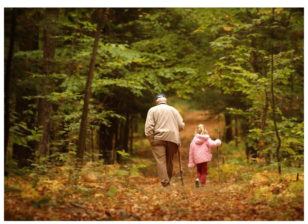
User Input

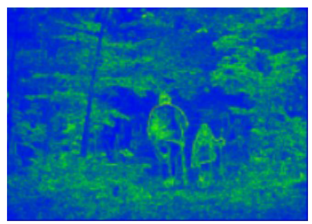
- If all else fails – brush interface to mark important areas, lines, structures, etc.
- In video: key framing to propagate



Combining saliency measures

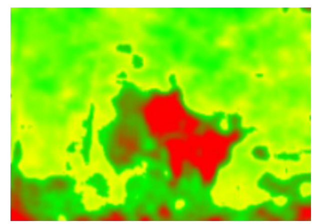
- Several importance maps can be combined
- Strongly dependent on input and desired result
- Optimal combinations are hard to learn automatically





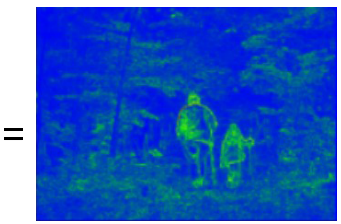
$((\frac{\partial}{\partial x}I)^2 + (\frac{\partial}{\partial y}I)^2)^{1/2}$

x



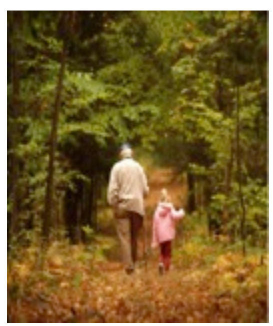
saliency map
[Itti et al. 98]

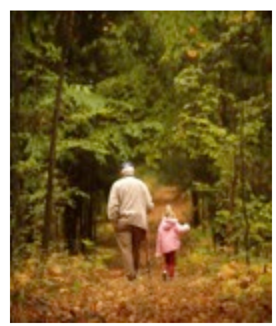
=



combined importance map

gradients only



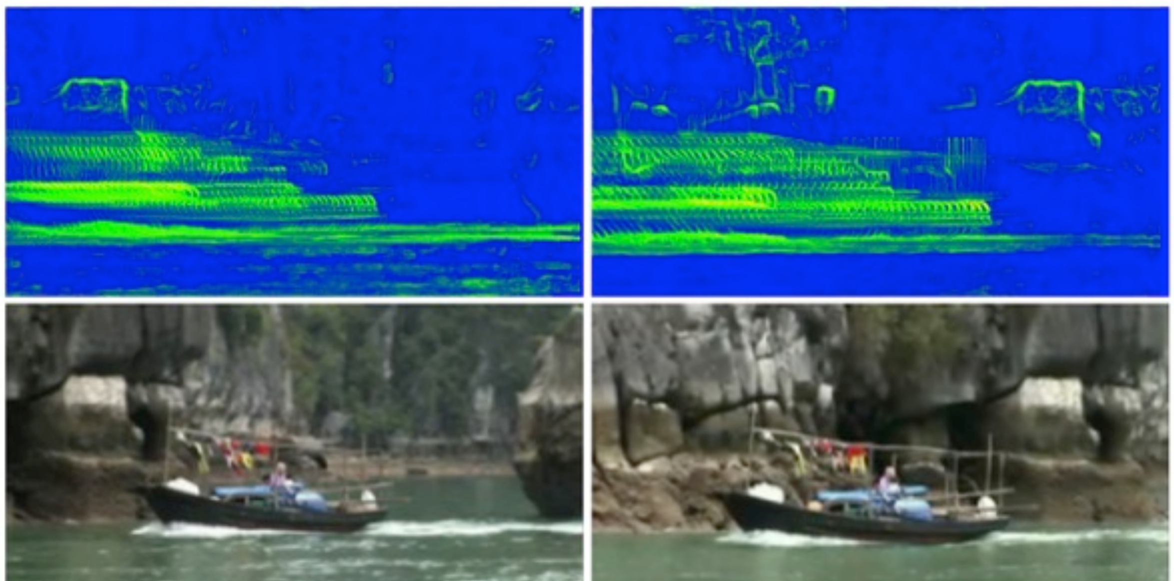


combined importance map

from [Wang et al. 2008]

Video: temporal coherence

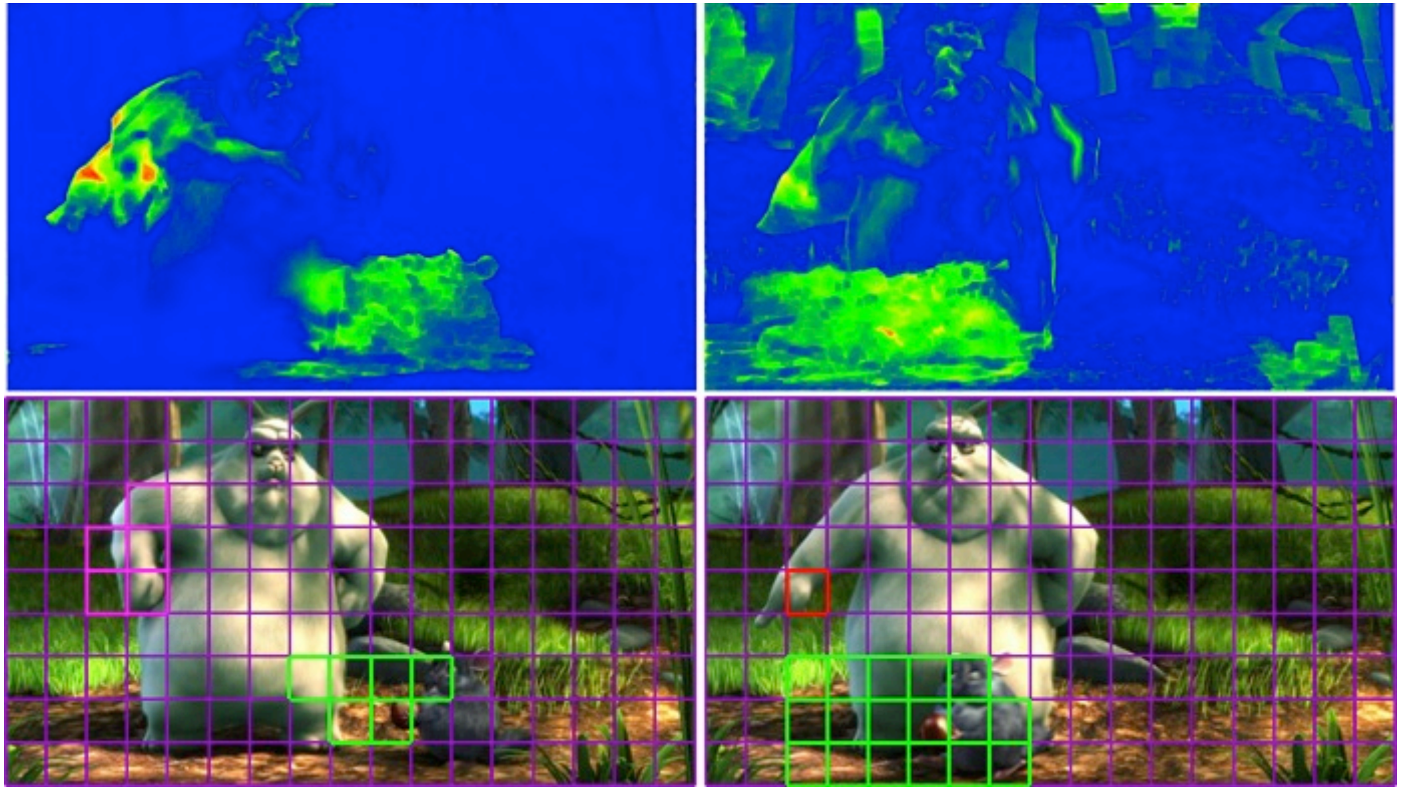
- Temporal coherence is key
- Blending and propagation (e.g., using optical flow) of saliency maps over several frames



from [Wang et al. 2009]

Video: motion saliency

- Moving objects are important



from [Wang et al. 2009]

Saliency summary

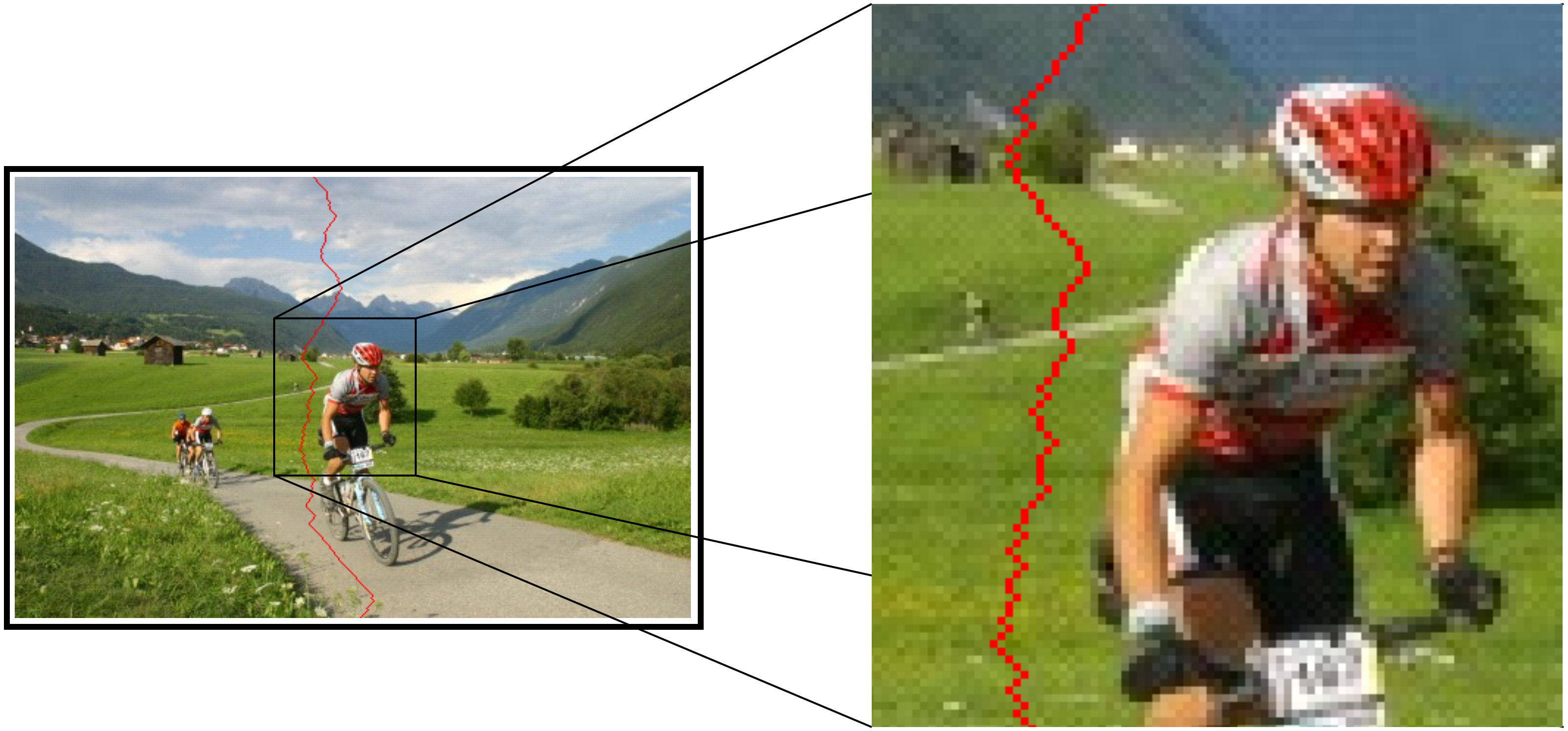
- Low-level vs high-level importance measures
  - Low-level measures are easy and fast to compute, but may miss actually important content
  - High-level capture semantics, but are more difficult to define and compute
- How to combine different measures?
  - Requires deeper understanding of our perception
- Interaction between importance map and the actual retargeting method
  - What type of saliency is optimal for retargeting?

Discrete Retargeting Operators

Overview

- More Seam carving
- Graph cut
- Video carving
- Shift Maps
- Multi-Operator

Seam Carving



$$M(i,j) = e(i,j) + \min(M(i-1,j-1), M(i-1,j), M(i-1,j+1))$$

Both Dimensions?

- Remove horizontal seam first?
- Remove vertical seams first?
- Alternate between the two?
- The optimal order can be found! → Dynamic Prog.

Optimal Order Map

*Removal of **vertical** seams*

*Removal of **horizontal** seams*

0	13	16	19		
16	17	22	28		
19	31	25	35		
24	28	29	???		
32	35	33			
41	38	35			

Optimal?

- Greedy in iterative sense we assume the cost function is monotonic!
- In fact there are many (exponential) ways to get to the desired size (m x n) – we must check all of them but we store only the best of two:
  - (m+1 x n) + (row seam cost)
  - (m x n+1) + (col seam cost)
- Key idea: ratio (of row & column) is more important than order

Example

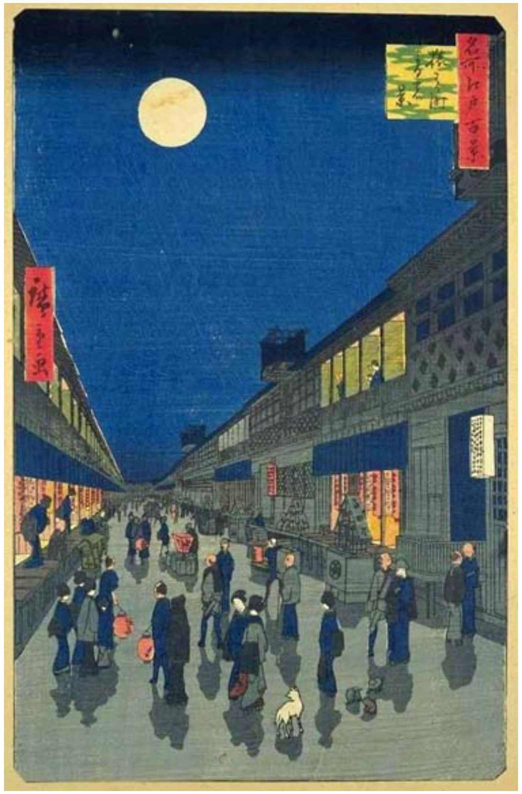
- We find best path: **RCR** by checking RCR against RRC.
- but maybe **CRR** is better than **RCR**? – we didn't check it because we chose **RC** over **CR** in the previous stage – and we are bound to this choice!

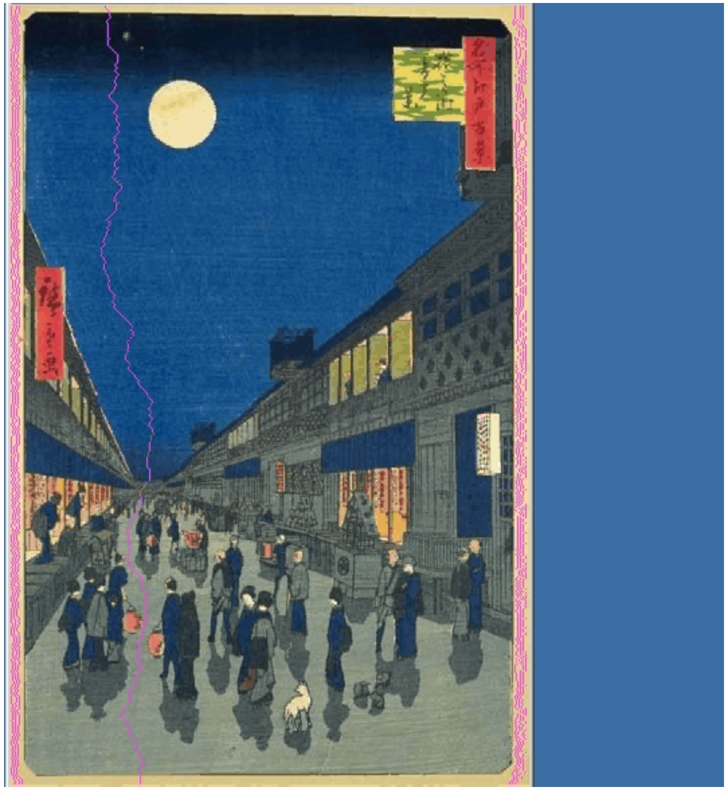
0	R	RR	…
C	RC	**RCR?**	
CC	…		
…			

Seam Insertion?

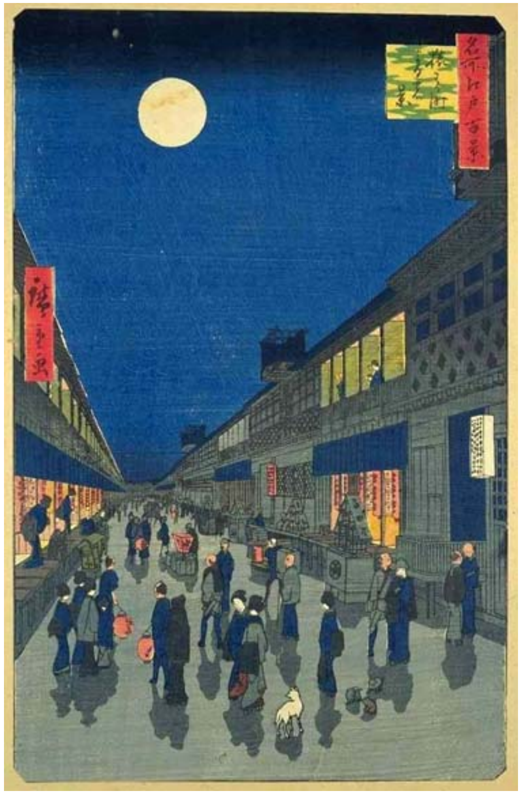
Seam Insertion

Duplicate seams in removal order





Seam Insertion



Two Ways to Change Aspect Ratio

Seam Removal

Seam Insertion

Two Ways to Change Aspect Ratio

Seam Removal

Seam Insertion

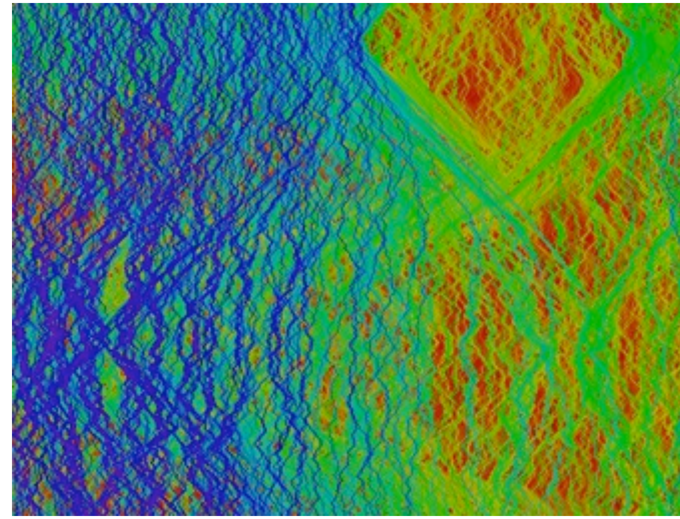
Combine Insert & Remove

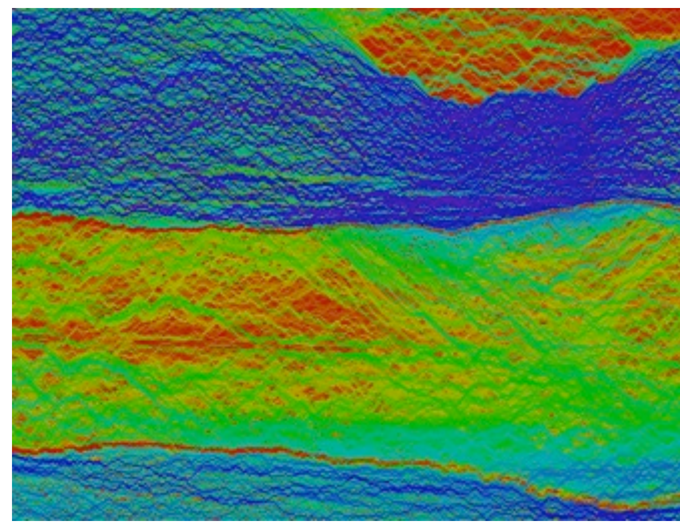
Insert & remove seams

Scaling

Enlarged or Reduced?

Multi-Size Images



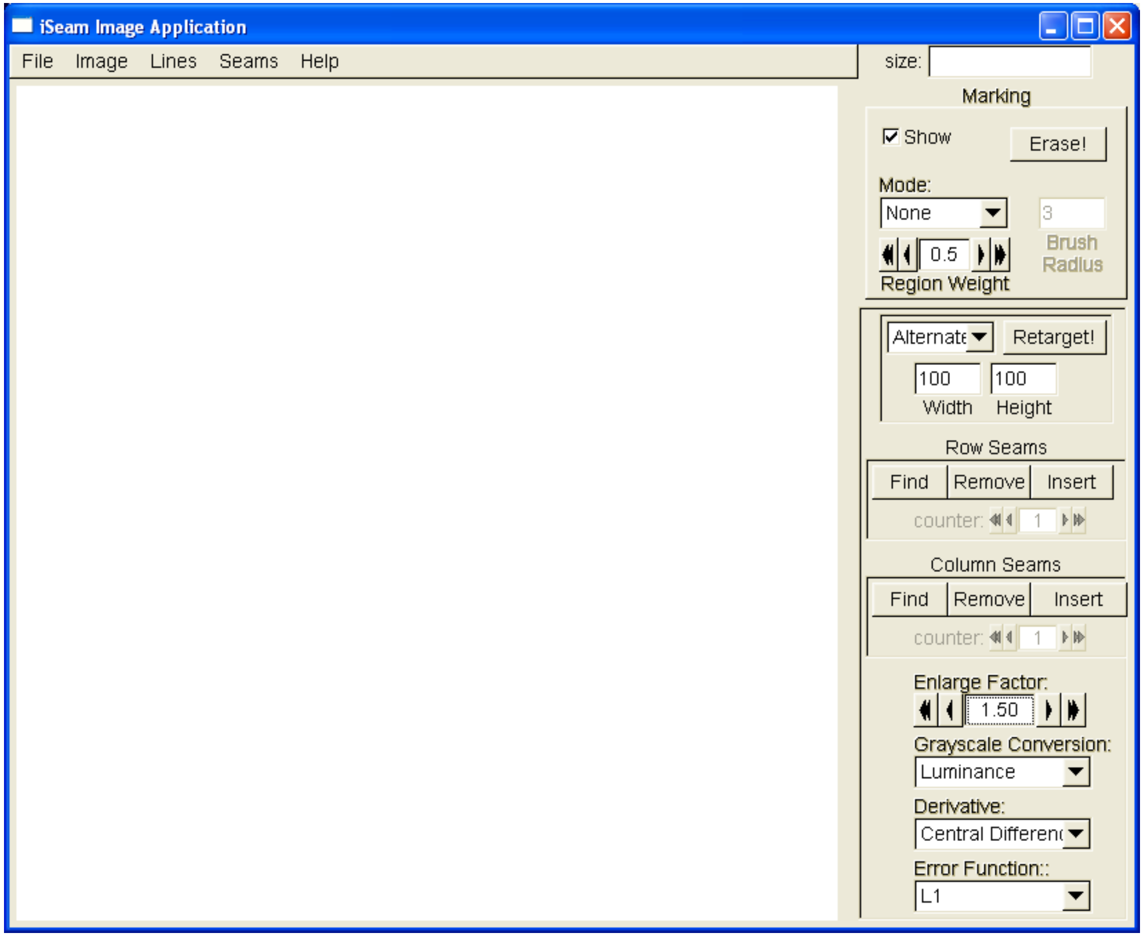


First to be removed

•
•
•

Last to be removed

Multi-Size Images & Demo



Not Always a Success

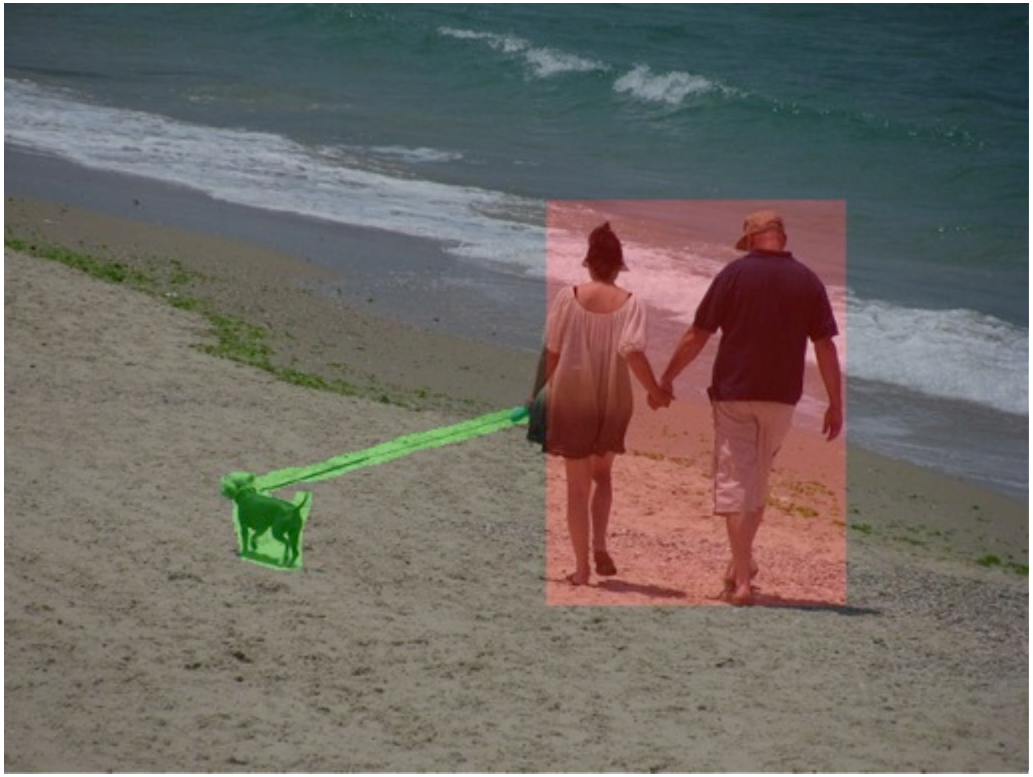


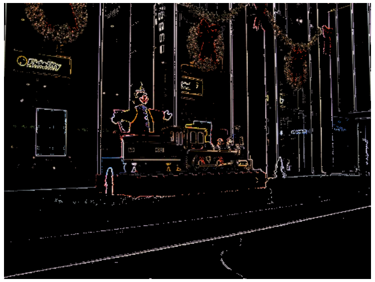
Image: Yehudit Garinkol

Find the Missing Shoe!

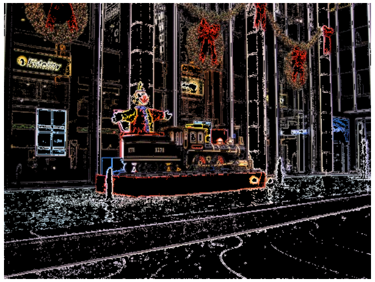
Solution

Pixel Energy Preservation

Energy



10%



30%

40%

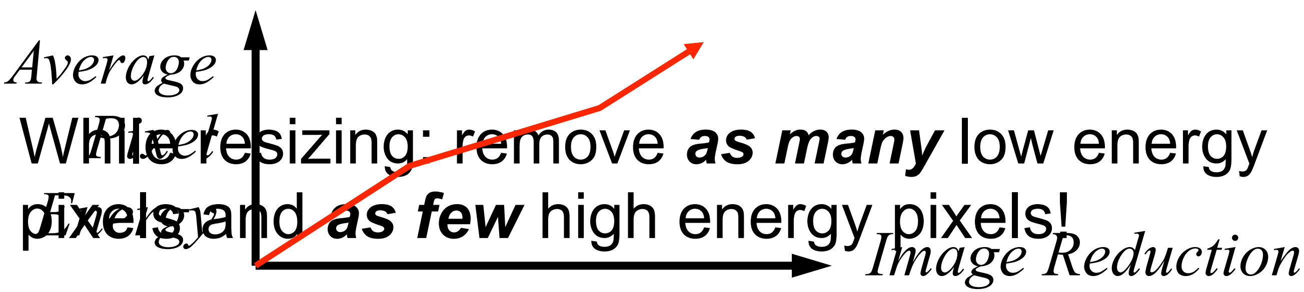
75%

While resizing: remove ***as many*** low energy pixels and ***as few*** high energy pixels!

Energy Preservation

If we measure the average energy of pixels in the image after applying a resizing operator…

…the average should increase!



While resizing: remove ***as many*** low energy pixels and ***as few*** high energy pixels!

Reduce Width

Average Pixel Energy

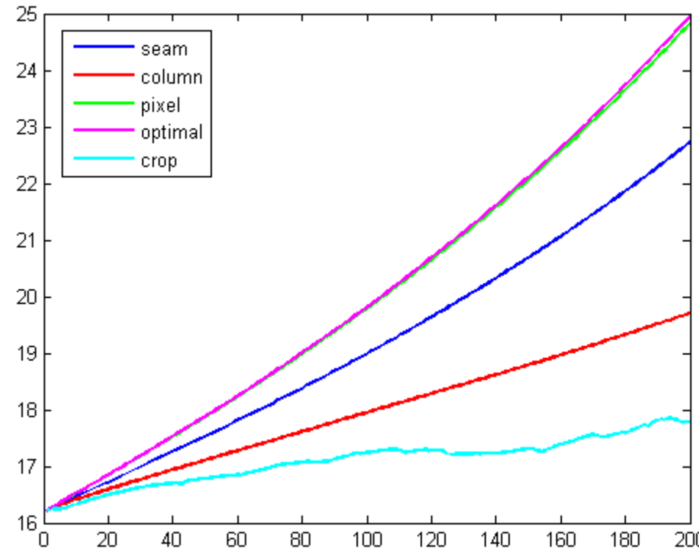
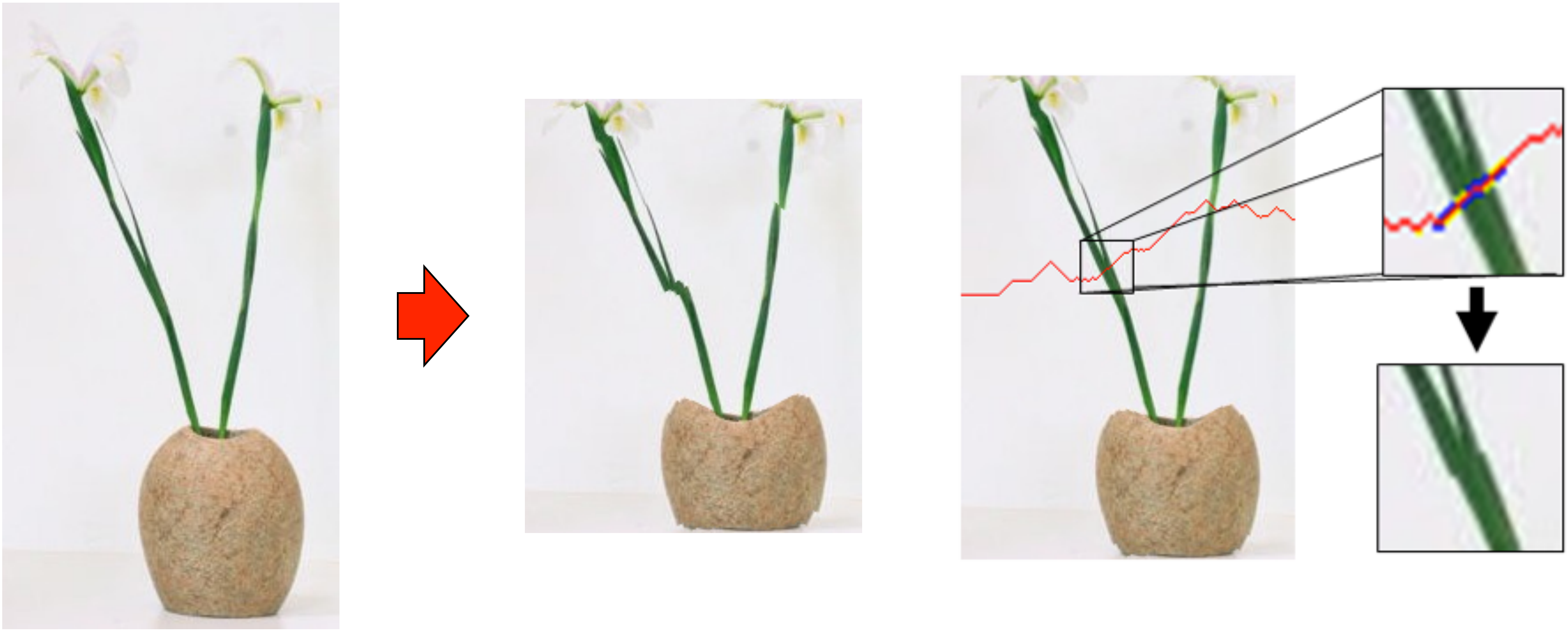


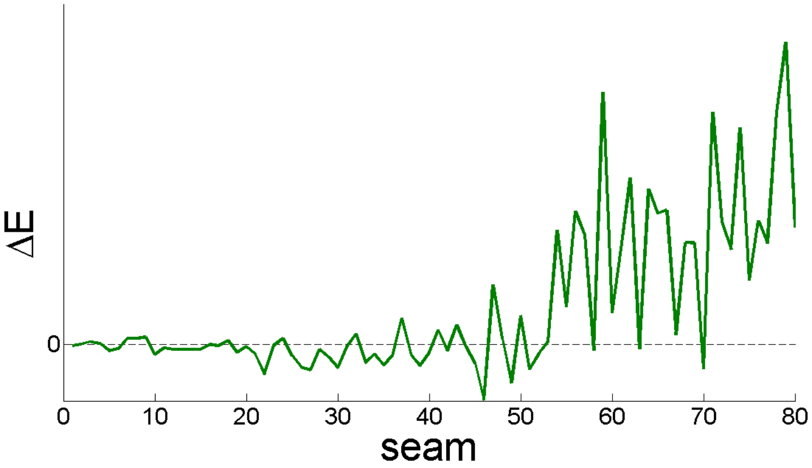
Image Reduction

crop **column** **pixel** **optimal**

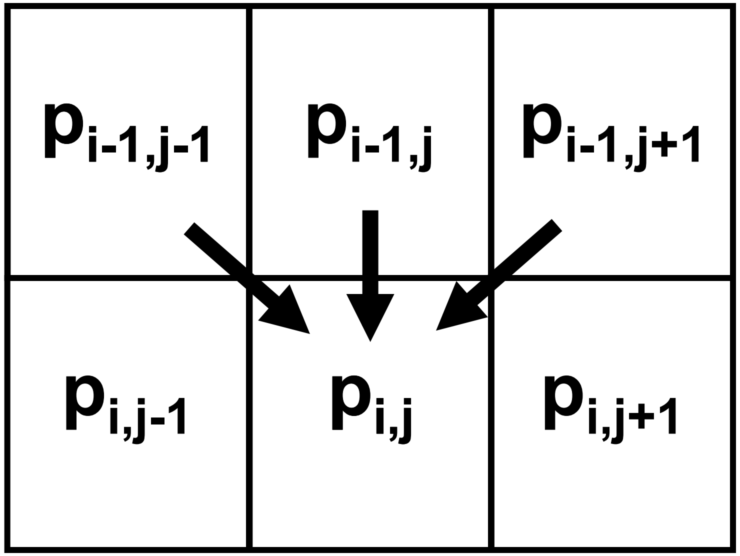
More Problems…



Change in Energy

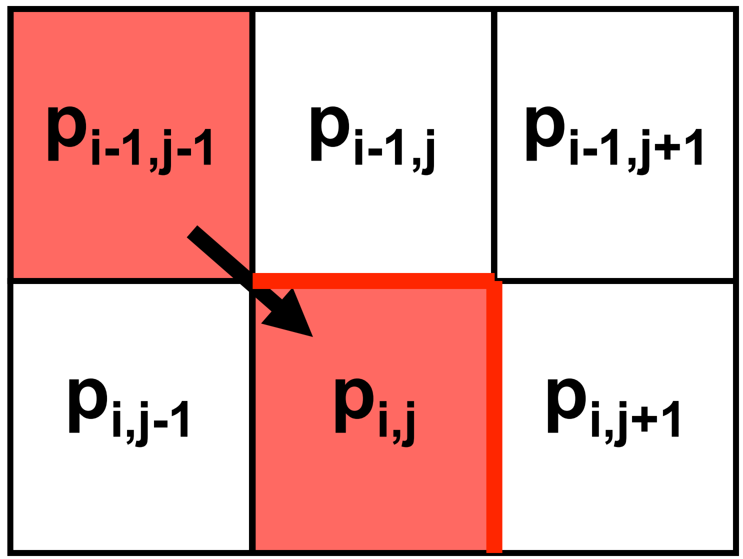


Tracking Inserted Energy



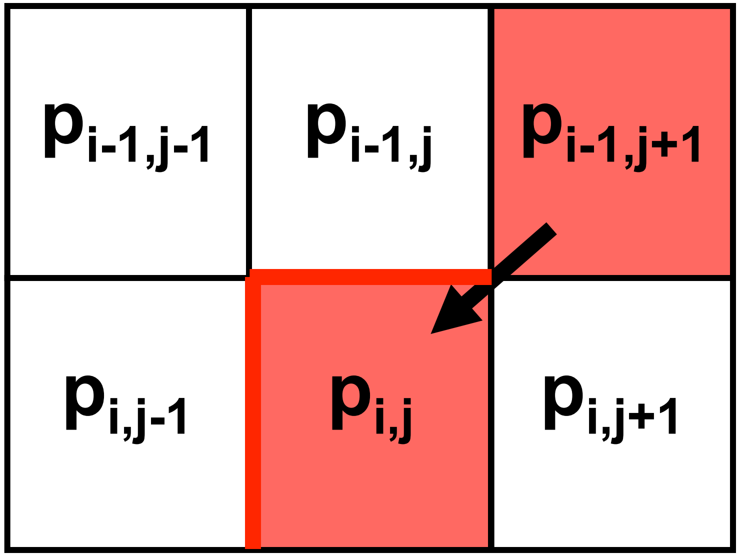
- Three possibilities when removing pixel $P_{i,j}$

Pixel $P_{i,j}$: Left Seam



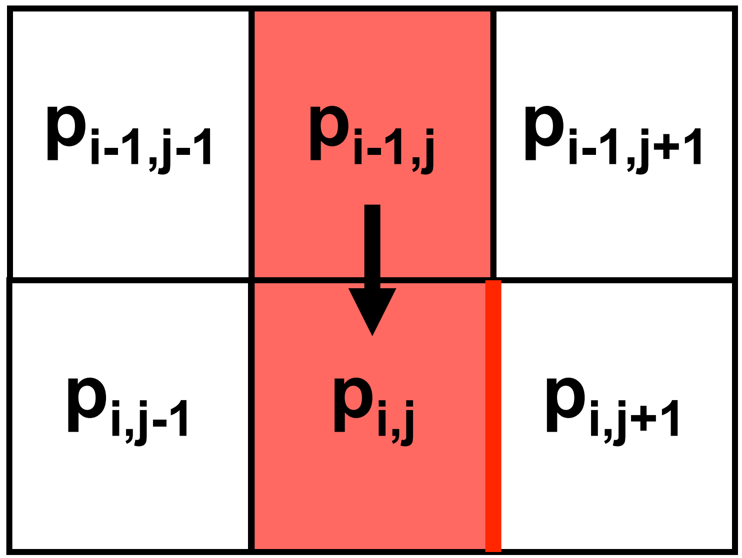
$$C_L(i,j) = |I(i,j+1) - I(i,j-1)|| + |I(i-1,j) - I(i,j-1)|$$

Pixel $P_{i,j}$: Right Seam



$$C_R(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j+1)|$$

Pixel $P_{i,j}$: Vertical Seam



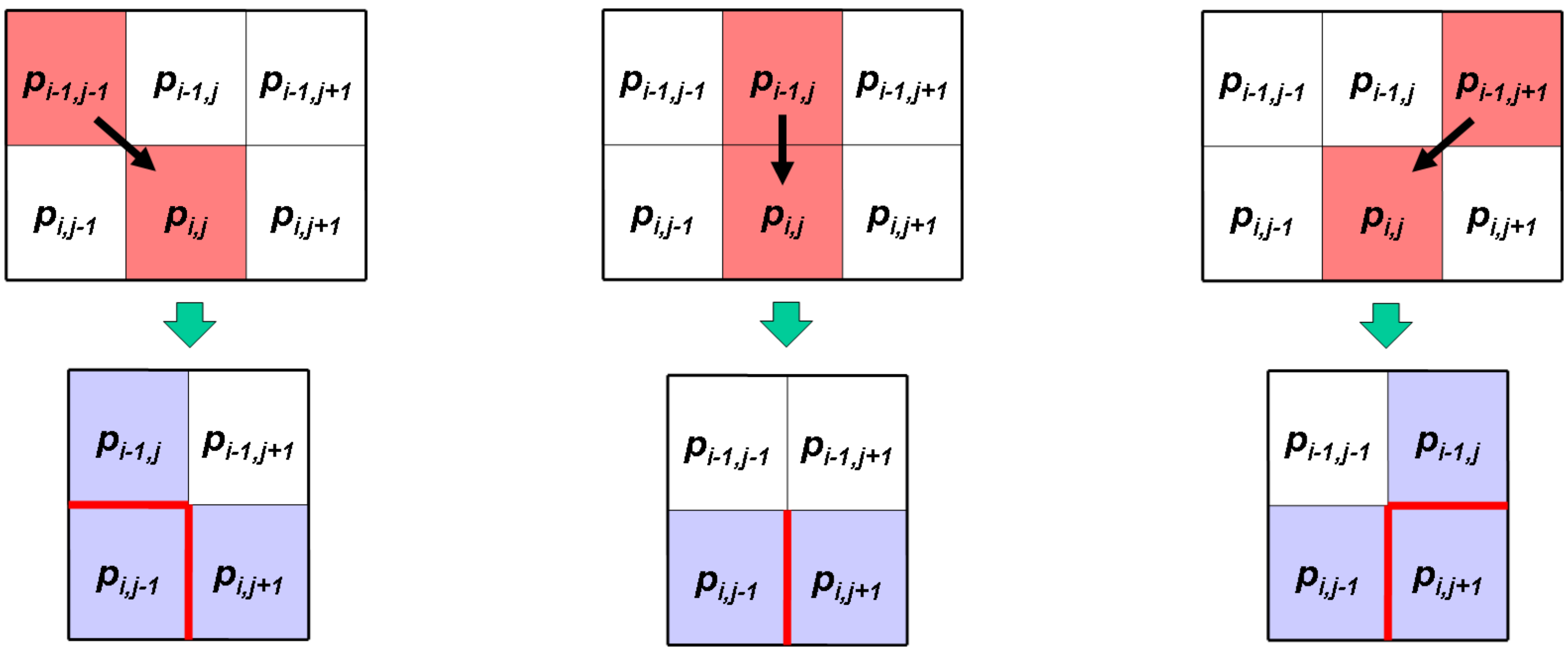
$$C_V(i,j) = |I(i,j+1) - I(i,j-1)|$$

Old "Backward" Energy Function

$$M(i,j) = E(i,j) + \min \begin{cases} M(i-1,j-1) \\ M(i-1,j) \\ M(i-1,j+1) \end{cases}$$

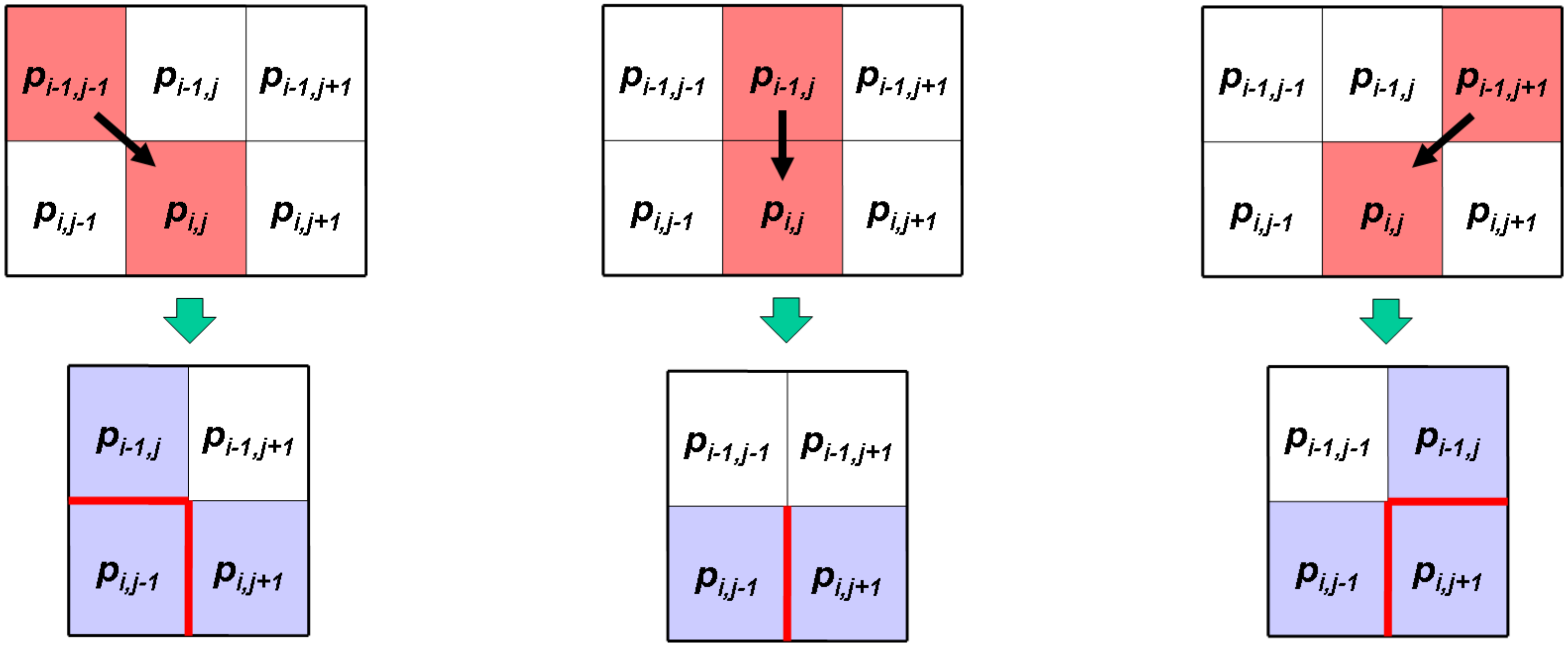
New <u>Forward</u> Looking Energy

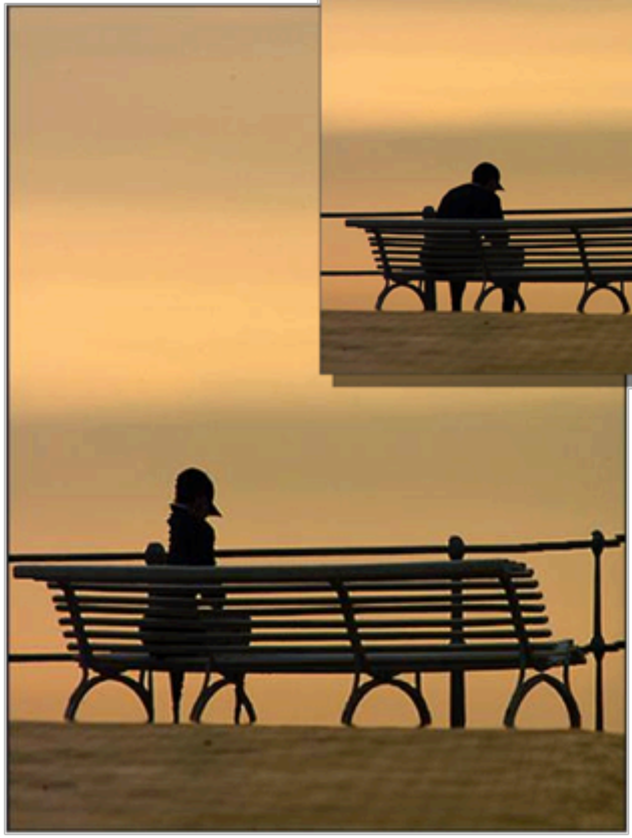
$$M(i,j) = \min \begin{cases} M(i-1,j-1) + C_L(i,j) \\ M(i-1,j) + C_U(i,j), \\ M(i-1,j+1) + C_R(i,j) \end{cases}$$

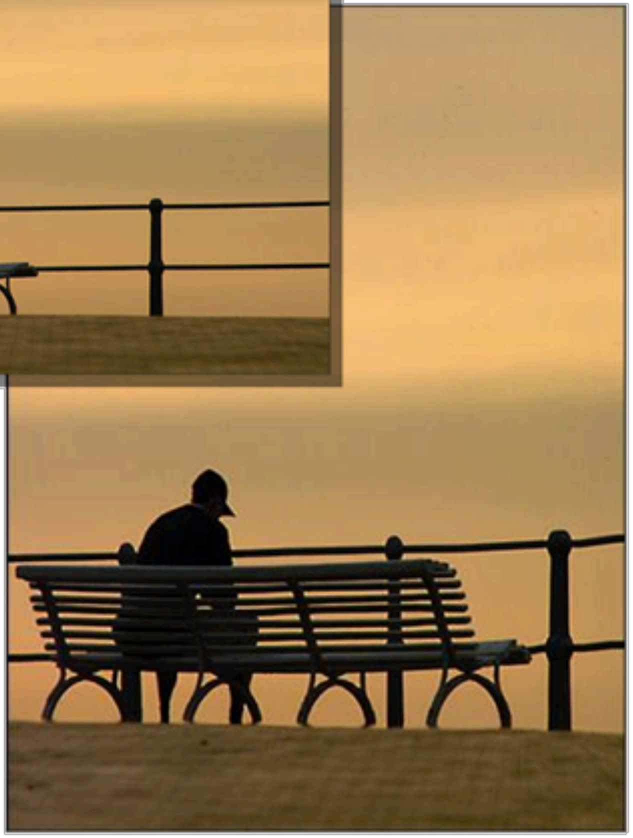


Adding "Pixel Energy"

$$M(i,j) = P(i,j) + \min \begin{cases} M(i-1,j-1) + C_L(i,j) \\ M(i-1,j) + C_U(i,j), \\ M(i-1,j+1) + C_R(i,j) \end{cases}$$



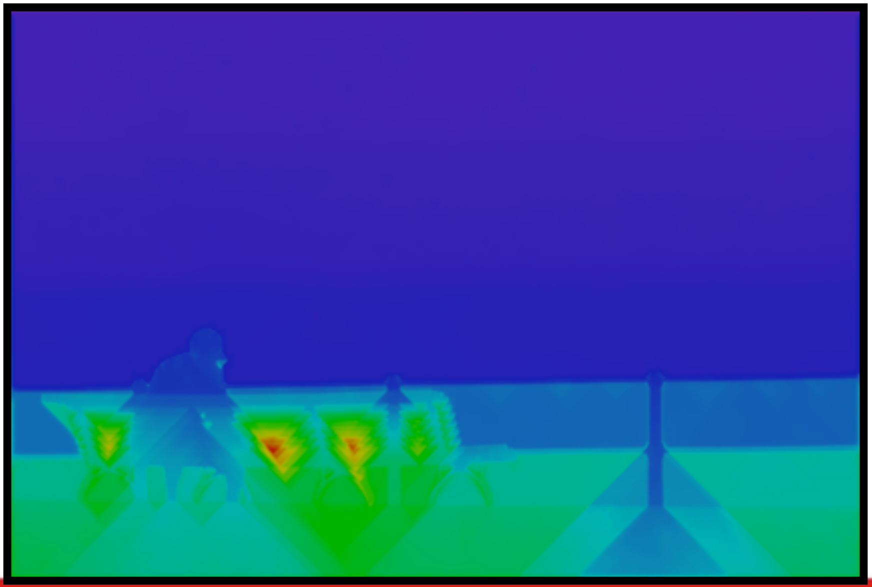




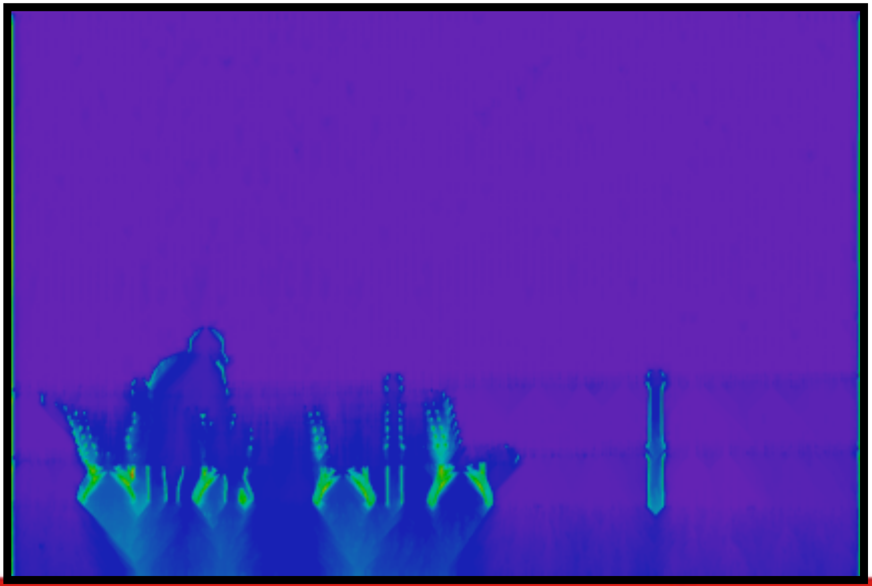
Backward Energy

Forward Energy

Backward Energy



Forward Energy



Backward Energy

Forward Energy

Backward

Forward

Backward

Forward

Backward

Forward

Expand

Expand

Video?

- Naive… every frame by itself

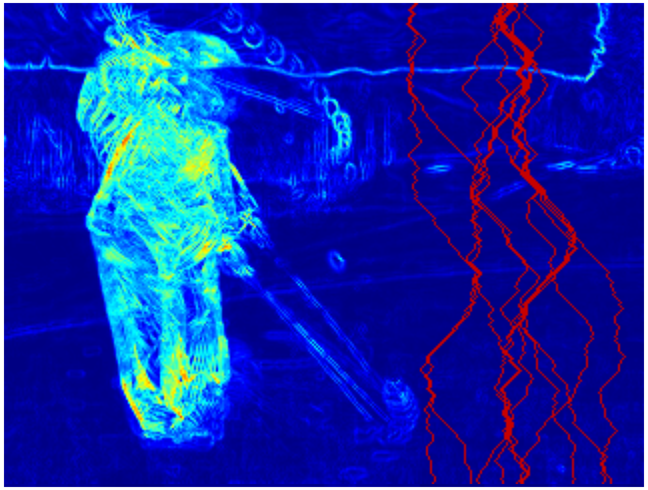
Jittery Results

Global Projection (Naïve #2)

- Reduction of the video problem to image seam carving by using projection of maximum energy through time:

Global Projection (Naïve #2)

- Reduction of the video problem to image seam carving by using projection of maximum energy through time:

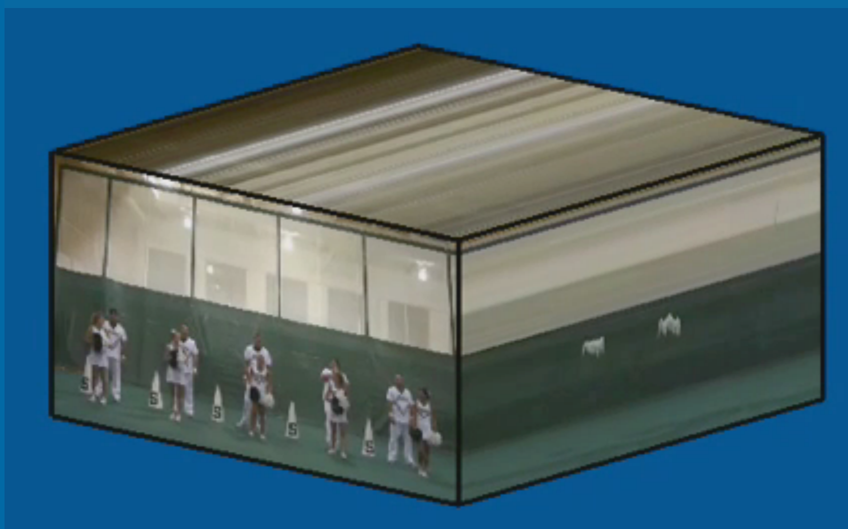


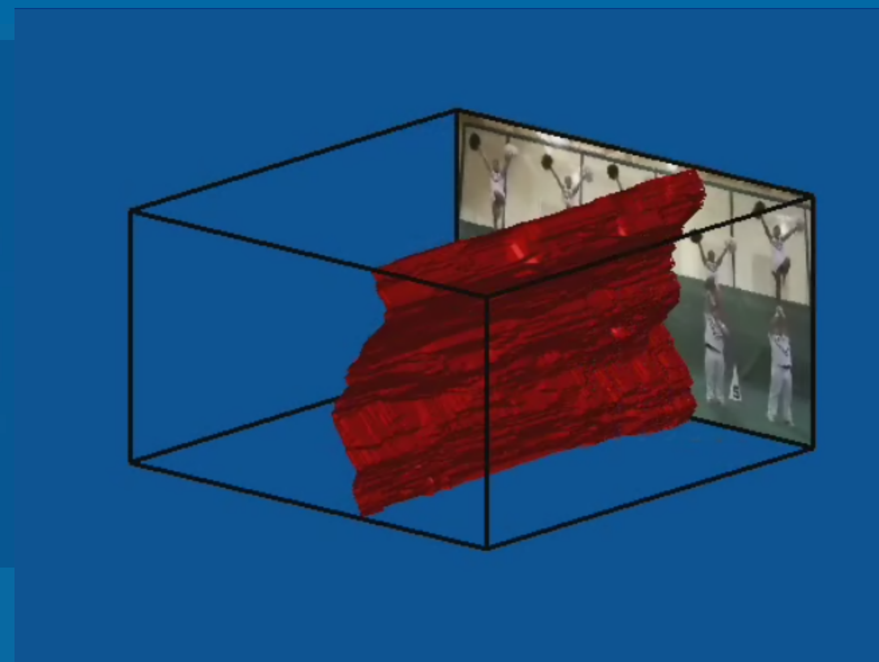
Problems

- Camera movement
- Object movement
- Seams should adapt and change through time!

→ Maybe adapt per frame?

Global Solution: 3D Seam Surface Inside Video Cube



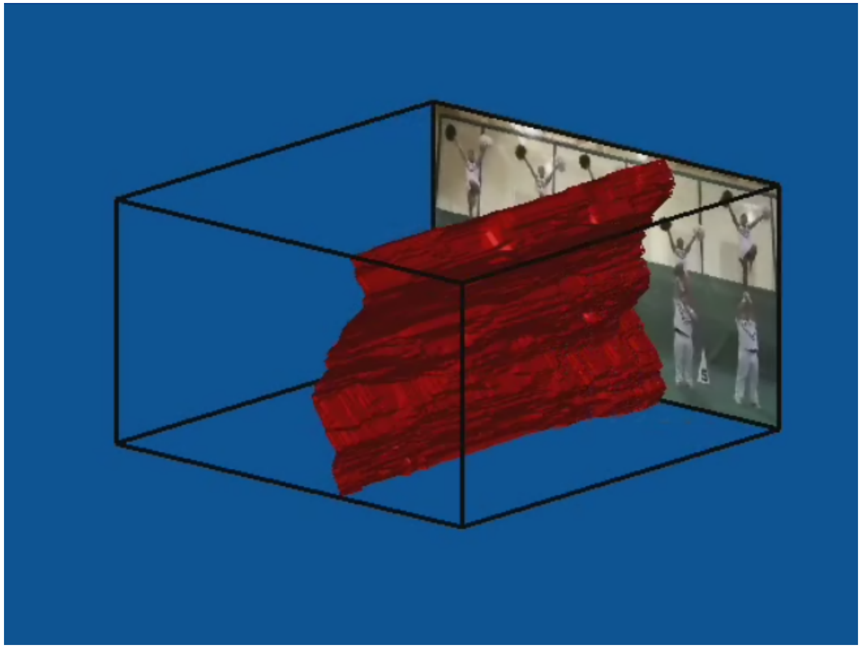


Video Cube

- **Video Cubes**:
  - Schodl et al. Siggraph 2000, *Video Textures*
  - Kwatra et al. Siggraph 2003, *Graph cut textures*
  - Rav-Acha et al. CVPR 2005, Video Mosaics
  - Wang et al. Siggraph 2005, *Interactive video cutout*
  - Chen and Sen EG 2008 (short papers), *Video Carving*

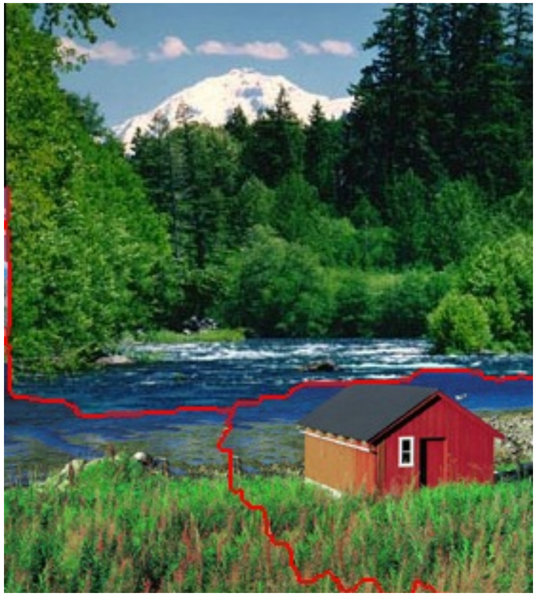
Problem:

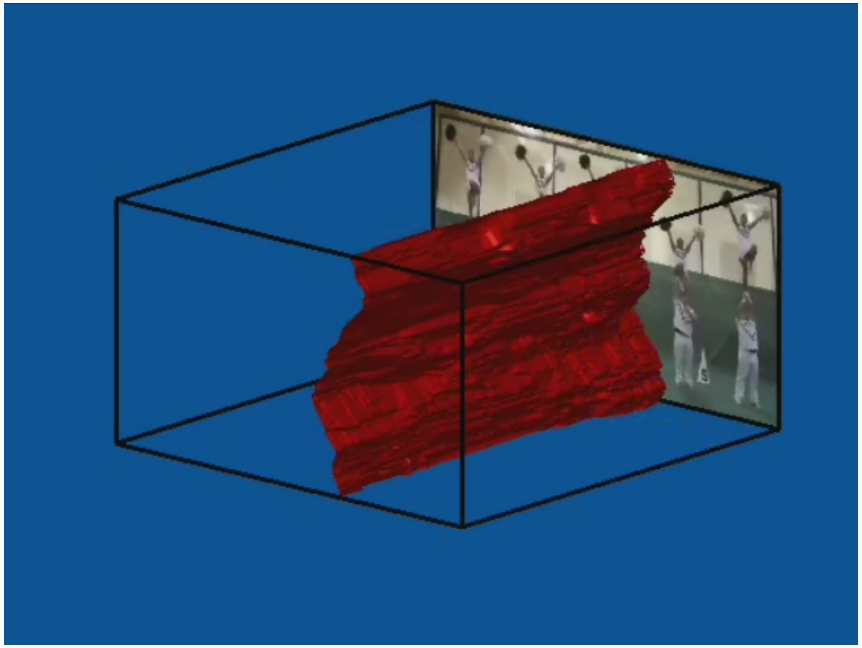
Dynamic Programming no longer works in 3D!



Use Graph Cut

- Kwatra et al. Siggraph 2003, *Graph cut textures:*





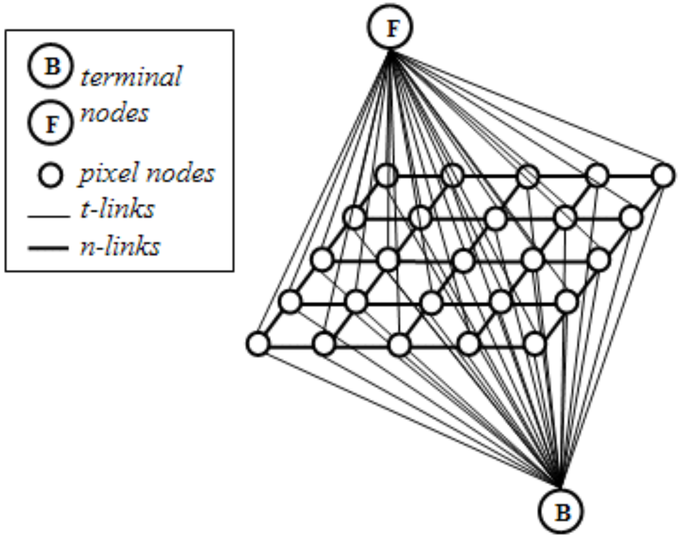
Labeling as Energy Minimization

$$E(L) = \alpha \underbrace{\sum_{p \in I} E_d(L(p))}_{\text{Data}} + \underbrace{\sum_{p,q \in N} E_s(L(p), L(q))}_{\text{Smoothness}}$$

$$E_d(L_f(p)) = P(p \in \text{forground})$$
$$E_d(L_b(p)) = P(p \in \text{background})$$

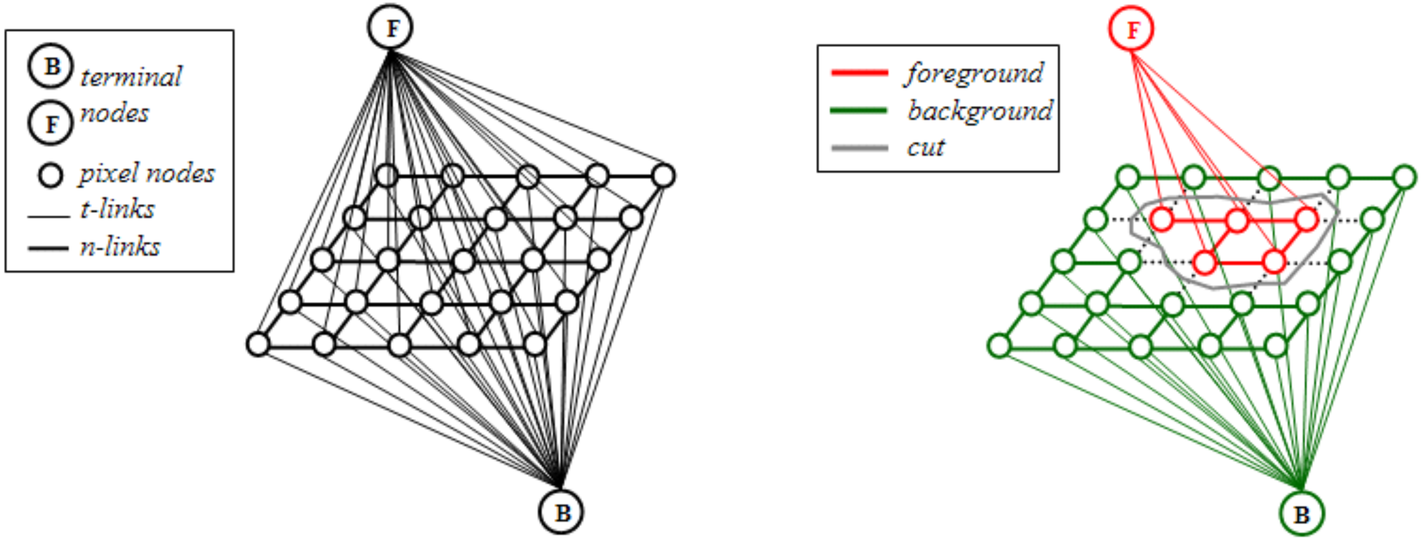
$$E_s(L(p), L(q)) = d(p,q)\delta(p,q)$$
$$\delta(p,q) = \begin{cases} 1 & \text{if } L(p) \neq L(q) \\ 0 & \text{otherwise} \end{cases}$$

Building a Graph



- Node weights = data term
- Edge weights = smoothness term

Solution as a Graph Cut



- Find the minimal cut
  - Cut is a set of edges disconnecting F from B
  - Minimum cut is the one with minimize sum of edge weight

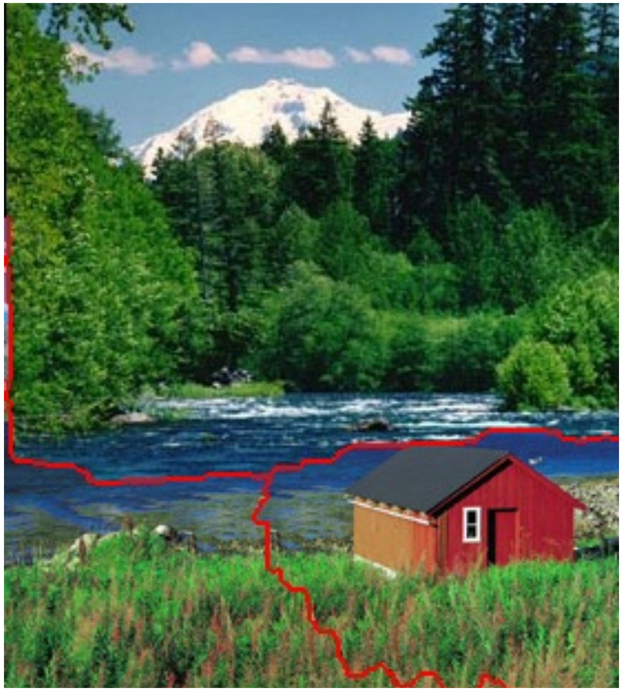
Algorithm

Max-Flow = Min Cut (Ford-Fulkerson)

```
Set flow to zero everywhere
Big loop
    compute residual graph (capacity - flow)
    Find path (shortest path) from source to
    sink in residual graph
        If path exist
                add corresponding flow
        Else
            Return Min cut = {vertices reachable
            from source; other vertices}
```

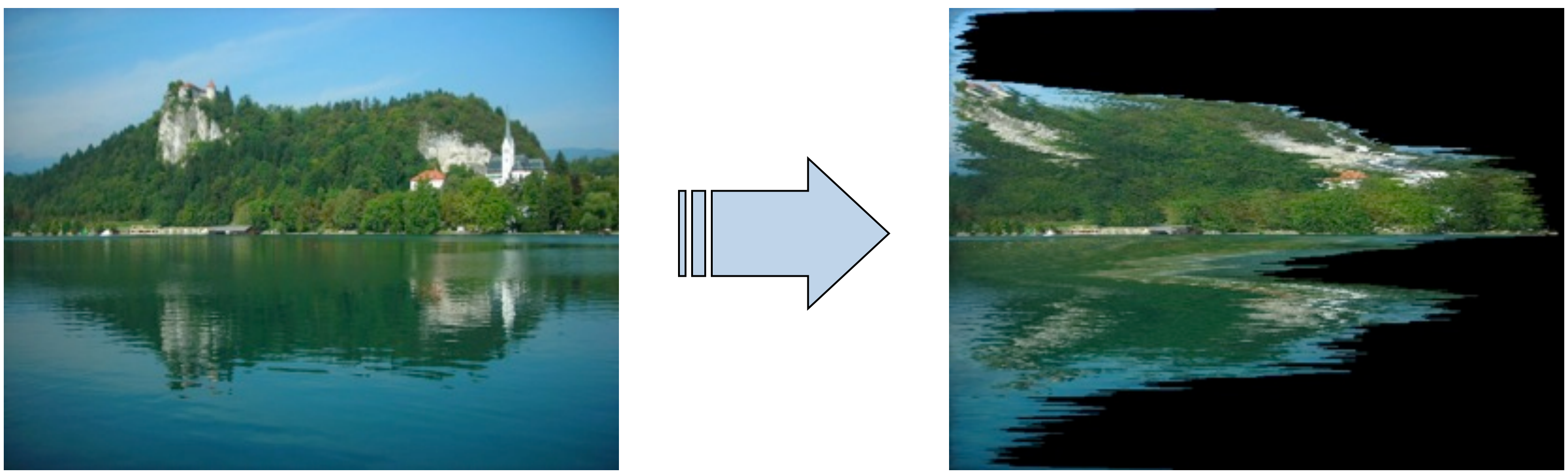
What Is the Challenge for Seams?

- How to Define a Seam from a Cut?



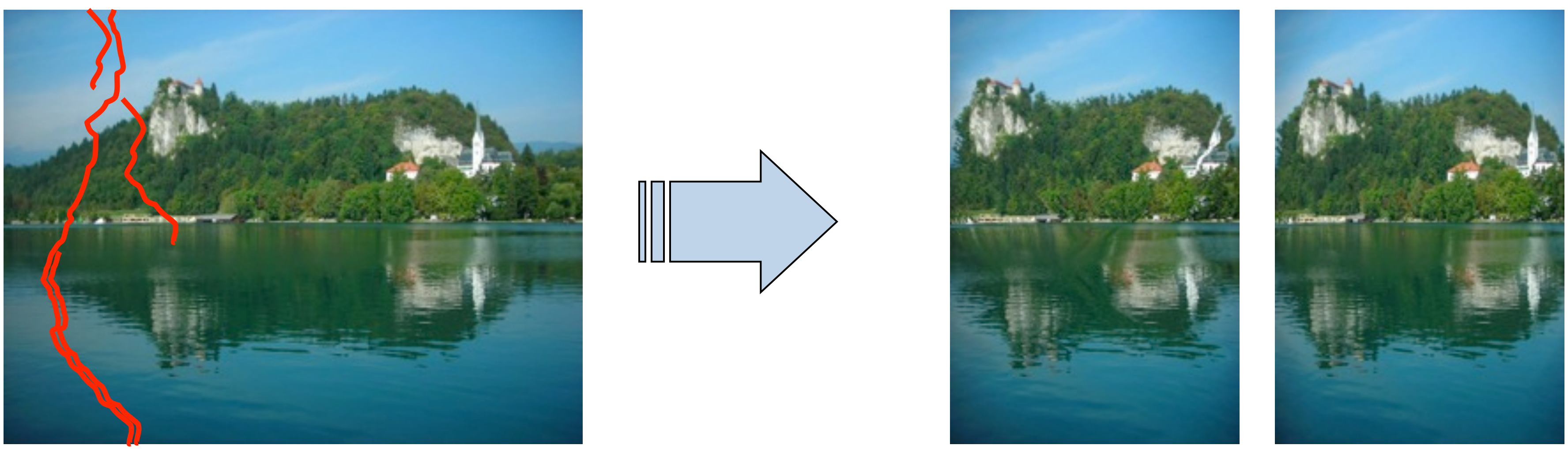
Constraints

1. Seams should be monotonic!
   (i.e. one pixel in each row)



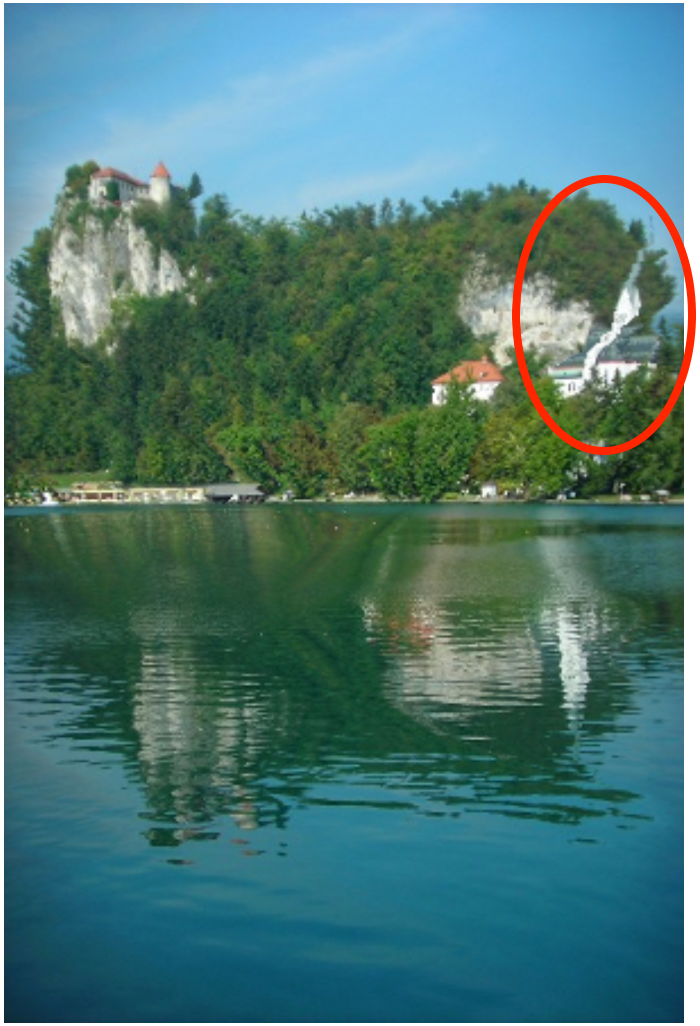
Constraints

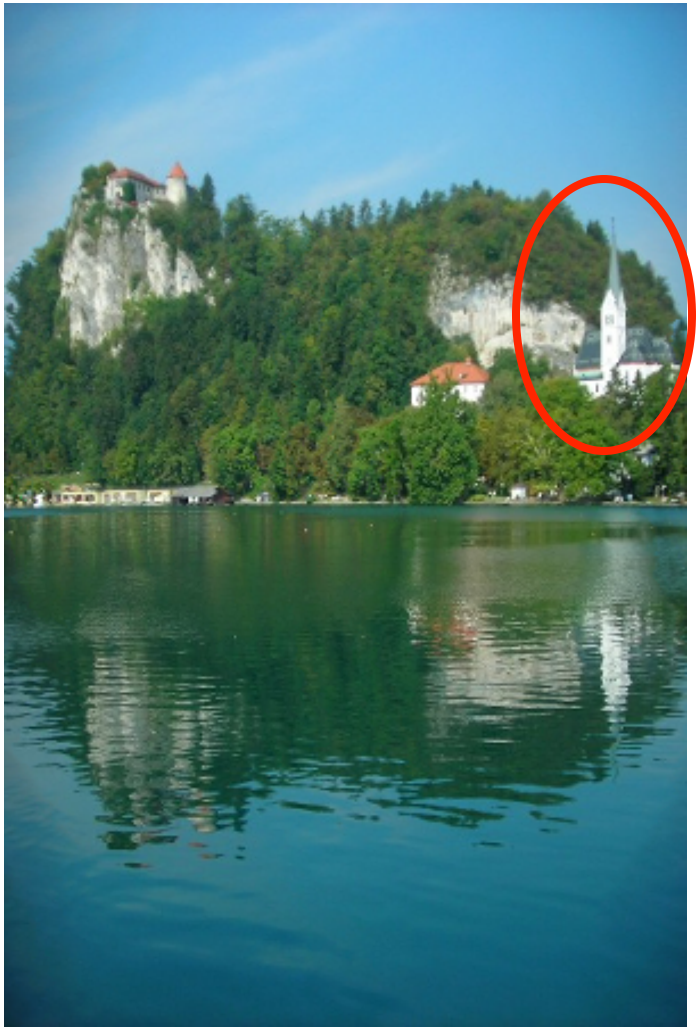
1. Seams should be monotonic!
2. Seams should be connected!



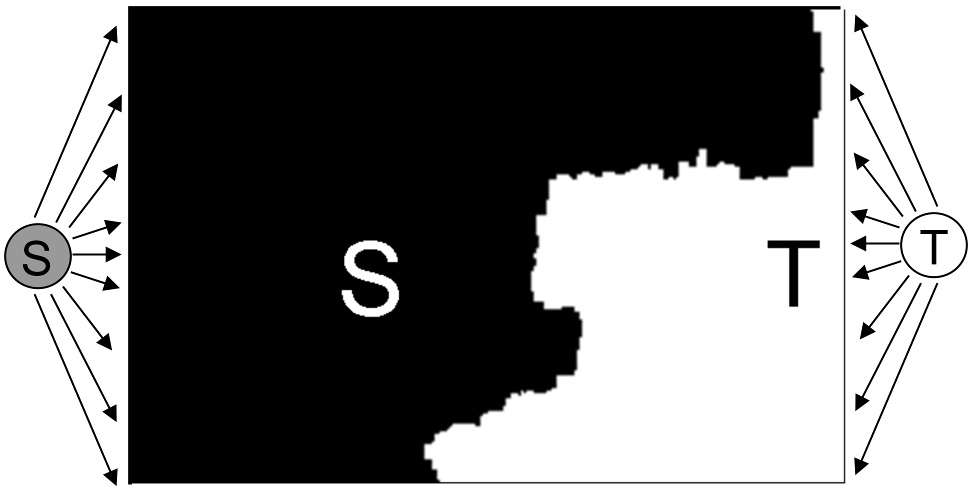
Piecewise Connected

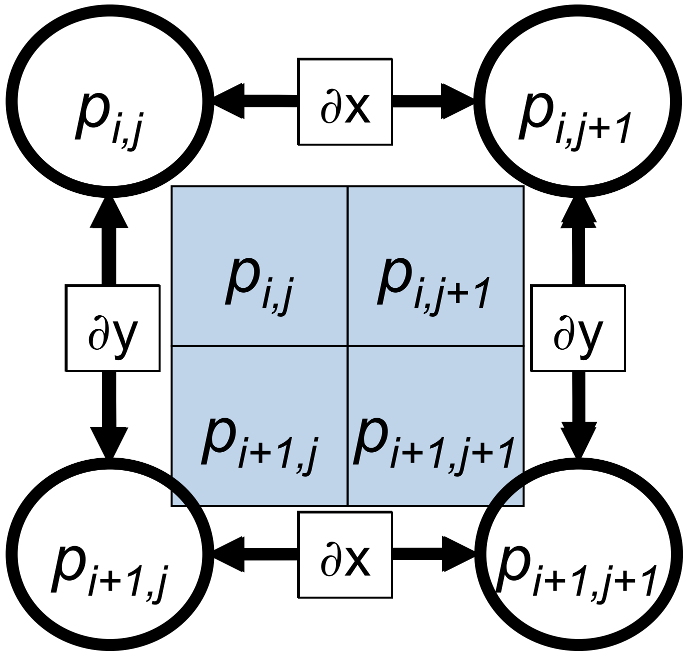
Piecewise vs. Connected



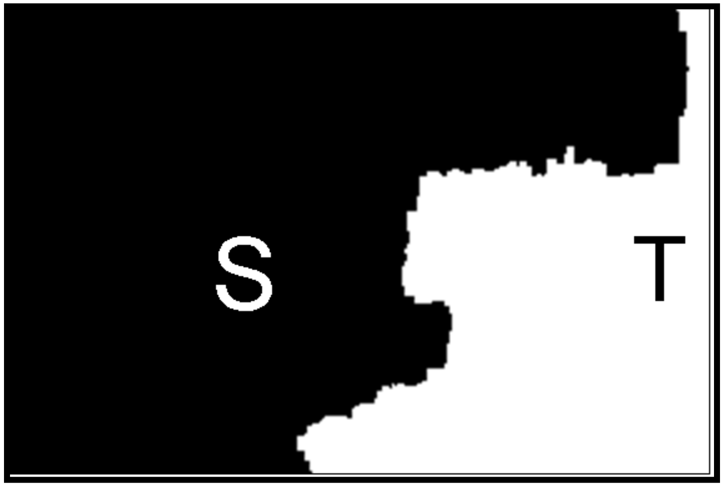


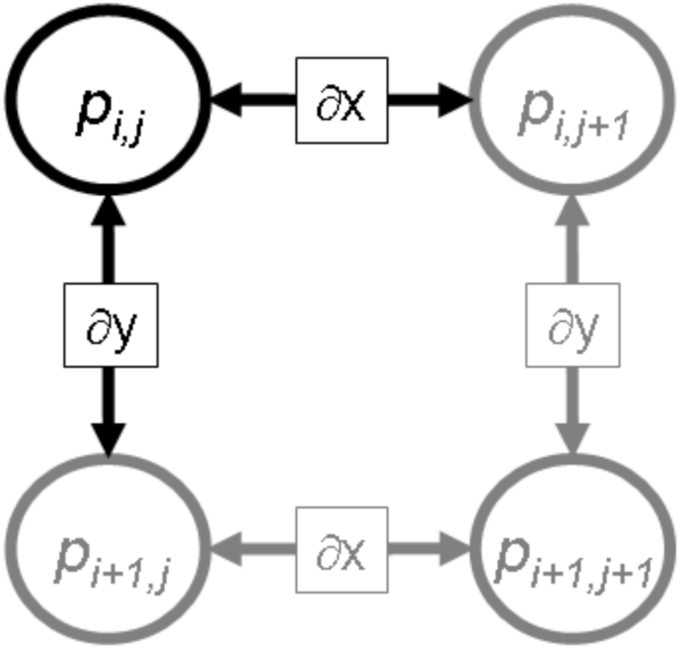
Standard Graph Construction?





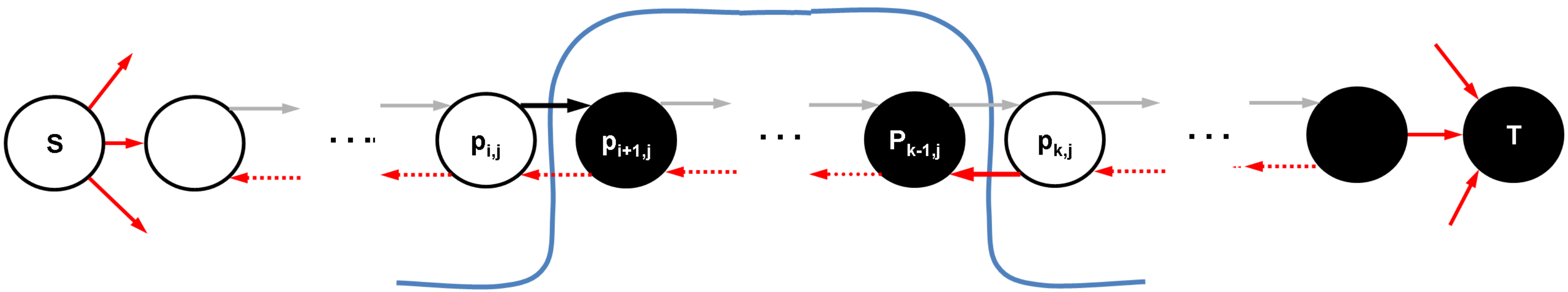
Simple Graph Cut



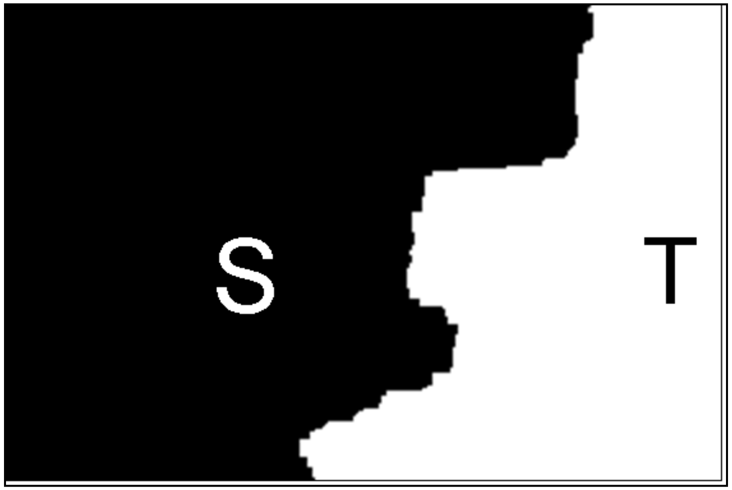


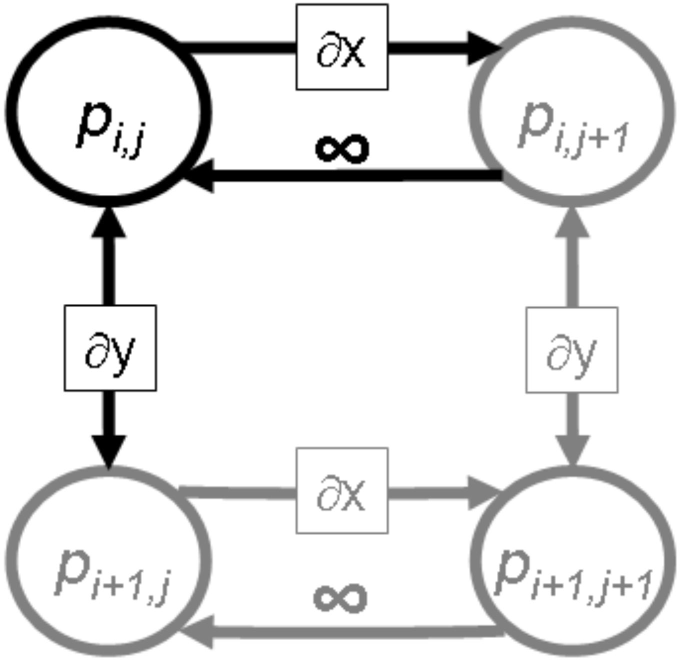
Monotonic (Function) Constraint

- Add "backward" infinity edges
- Proof:
  - All target nodes must be on the right of the cut
  - If a cut cuts more than once – it must cut an even number
  - Hence it must cut infinity edge – contradicting its minimal assumption



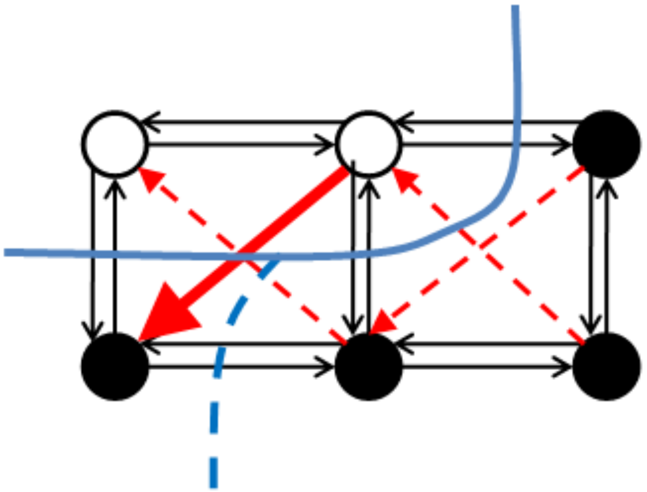
Monotonic (not connected)

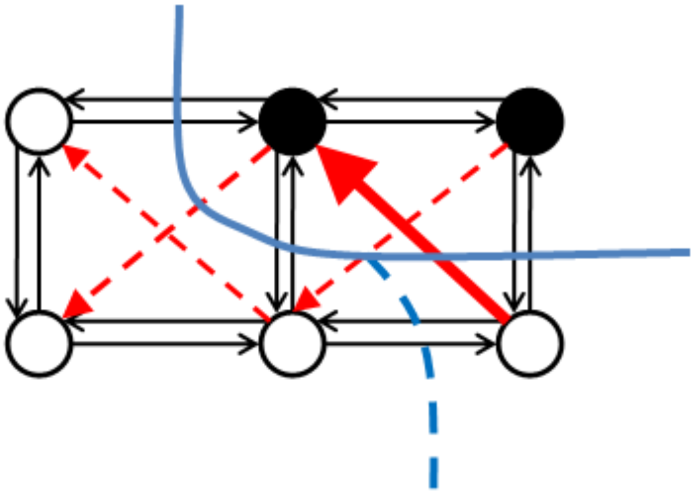




Add Diagonal Edges (No Jumps)

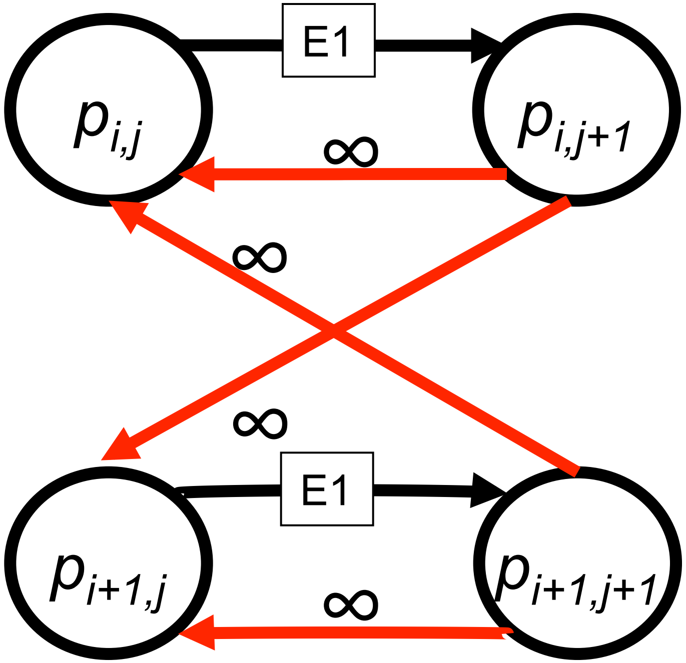
Proof:



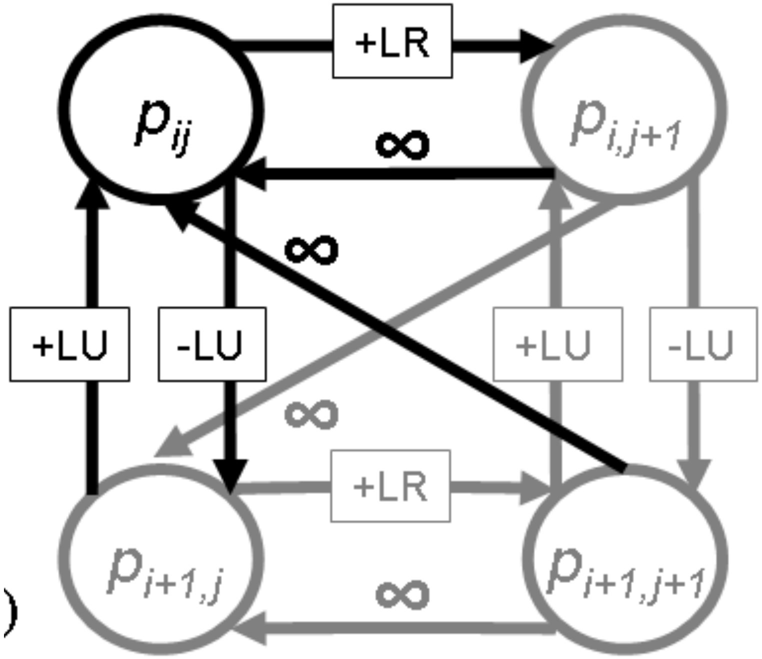


Backward Energy Construction

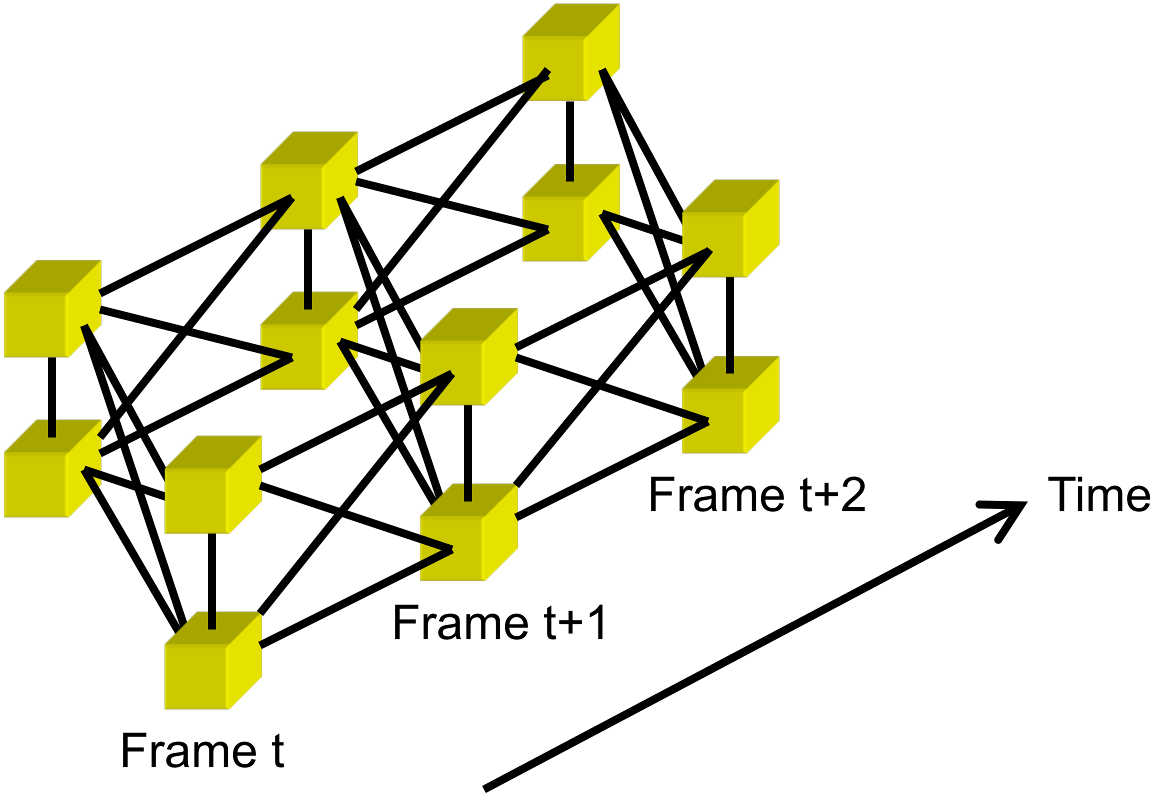
- This construction <u>guarantees</u> monotonic and connected seams
- This construction creates seams that are equivalent to the dynamic programming approach



Forward Energy Construction

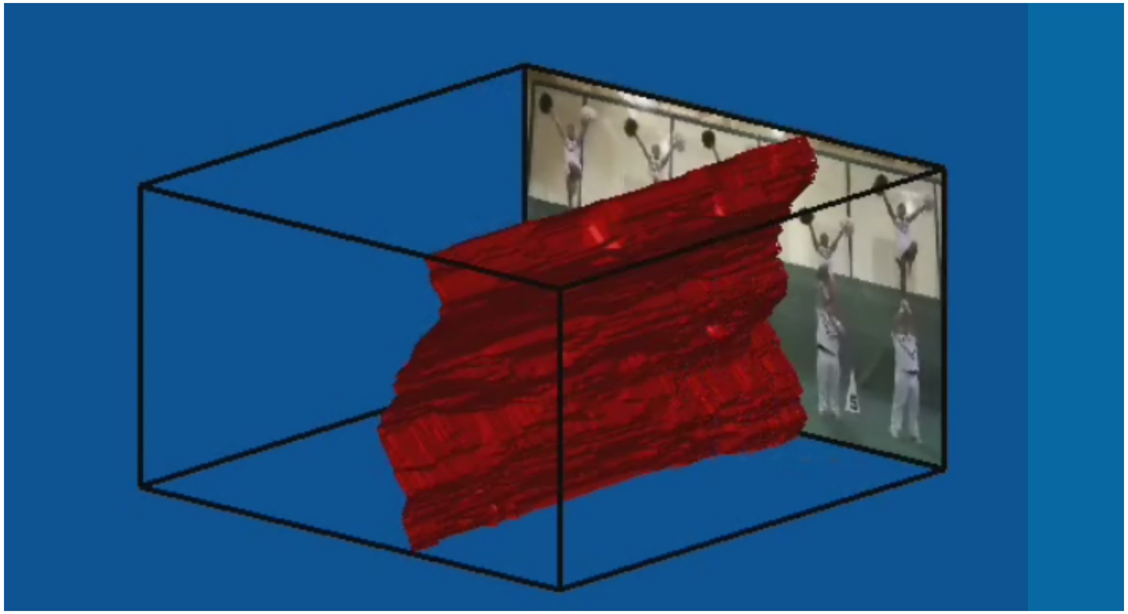


3D Graph Construction

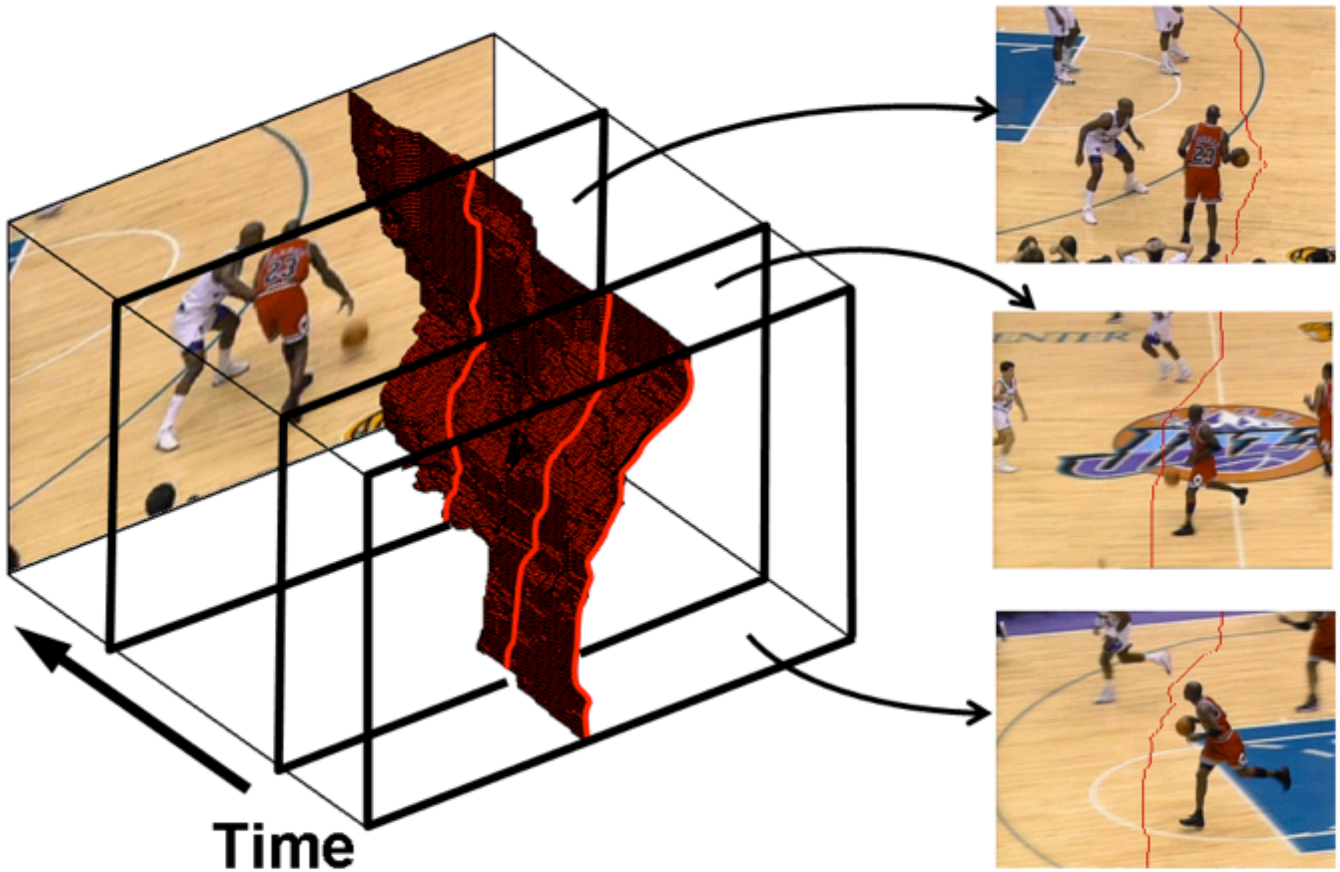


Video Cube

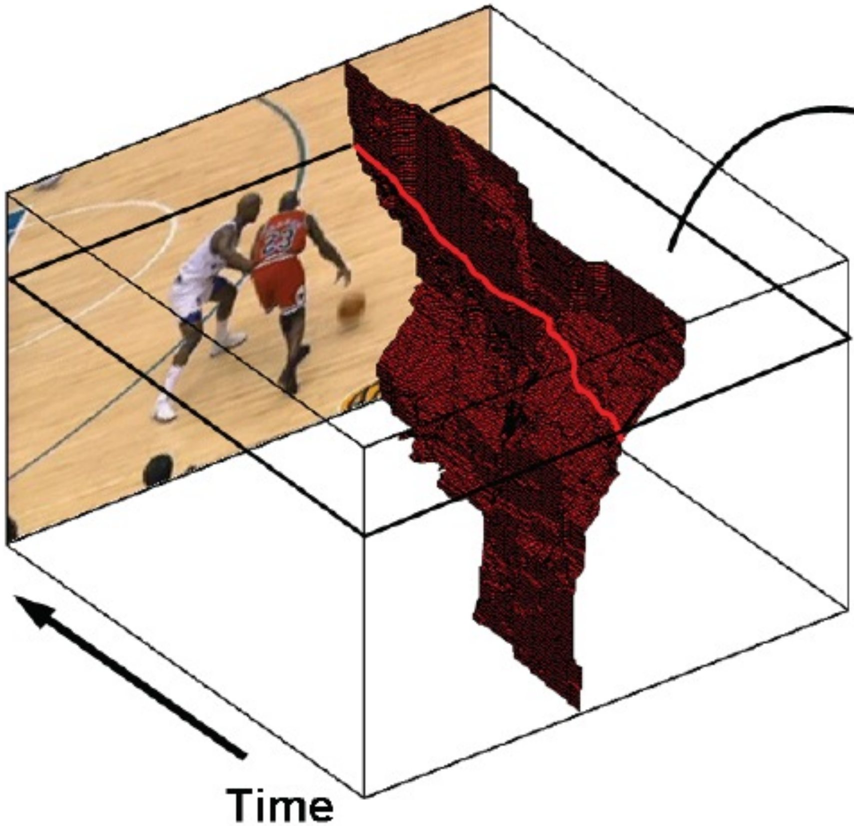
3D Graph Cut

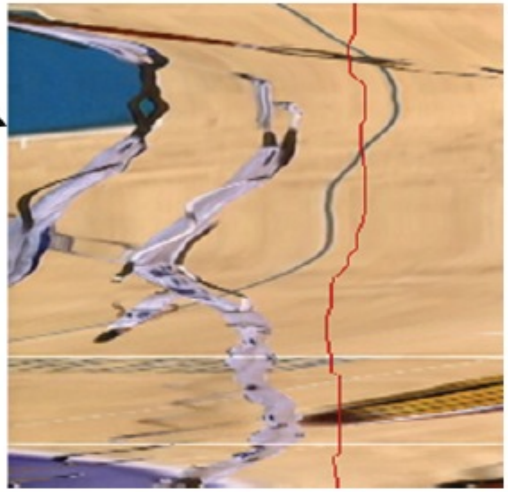


XY Plane

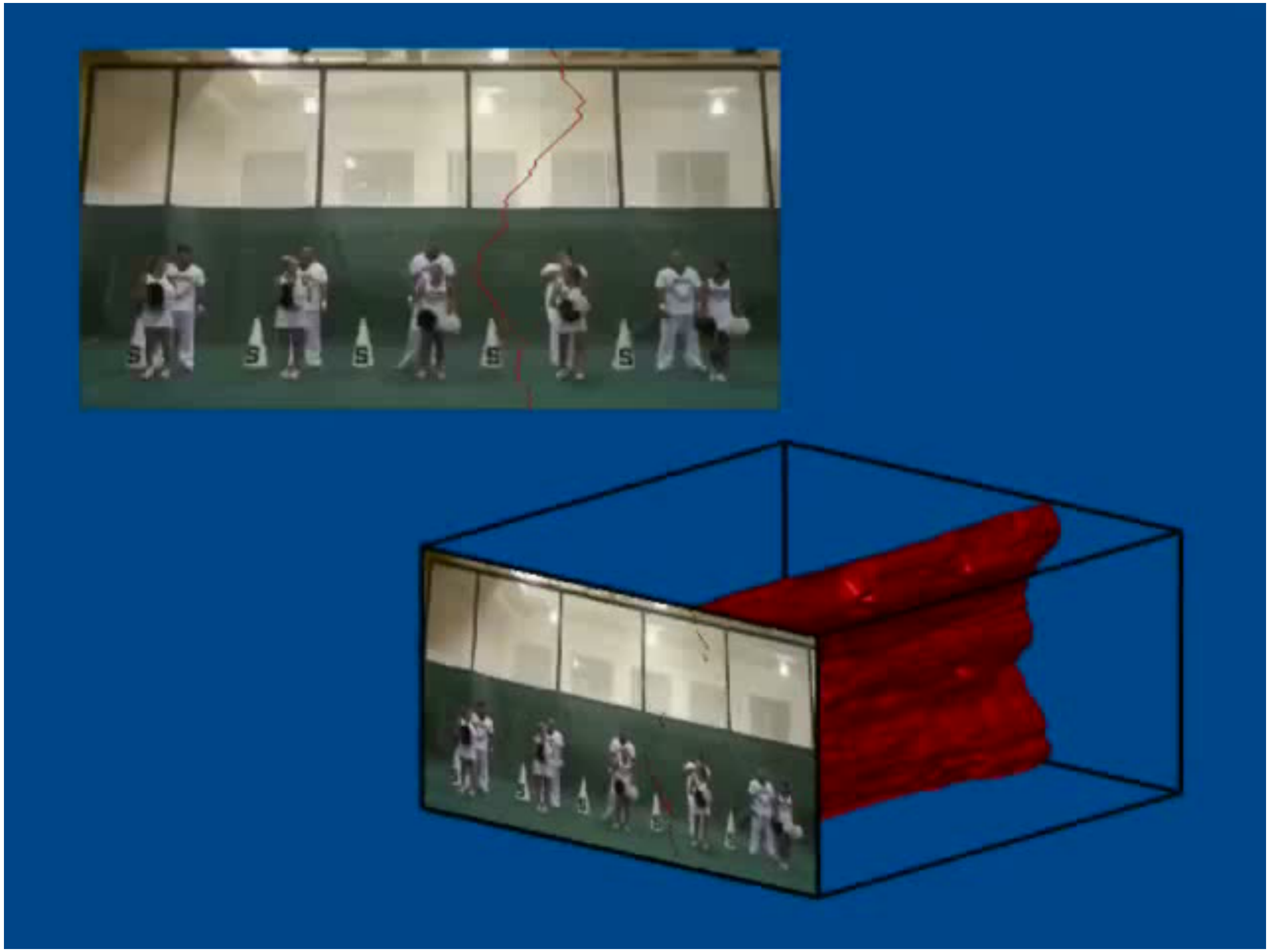


XT Plane





Dynamic Seams

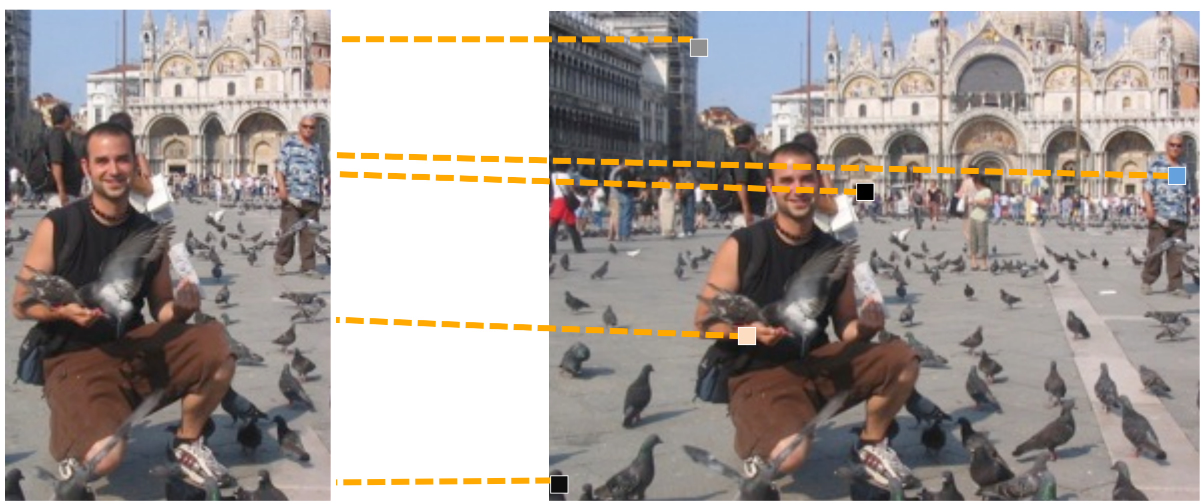


Results

Shift-Map

- Shift-Maps represent a ***mapping*** for each pixel in the output image into the input image

Output : R(u,v) $M(u,v) = (t_x, t_y)$ **Input : I(x,y)**



The color of the output pixel is copied from corresponding input pixel

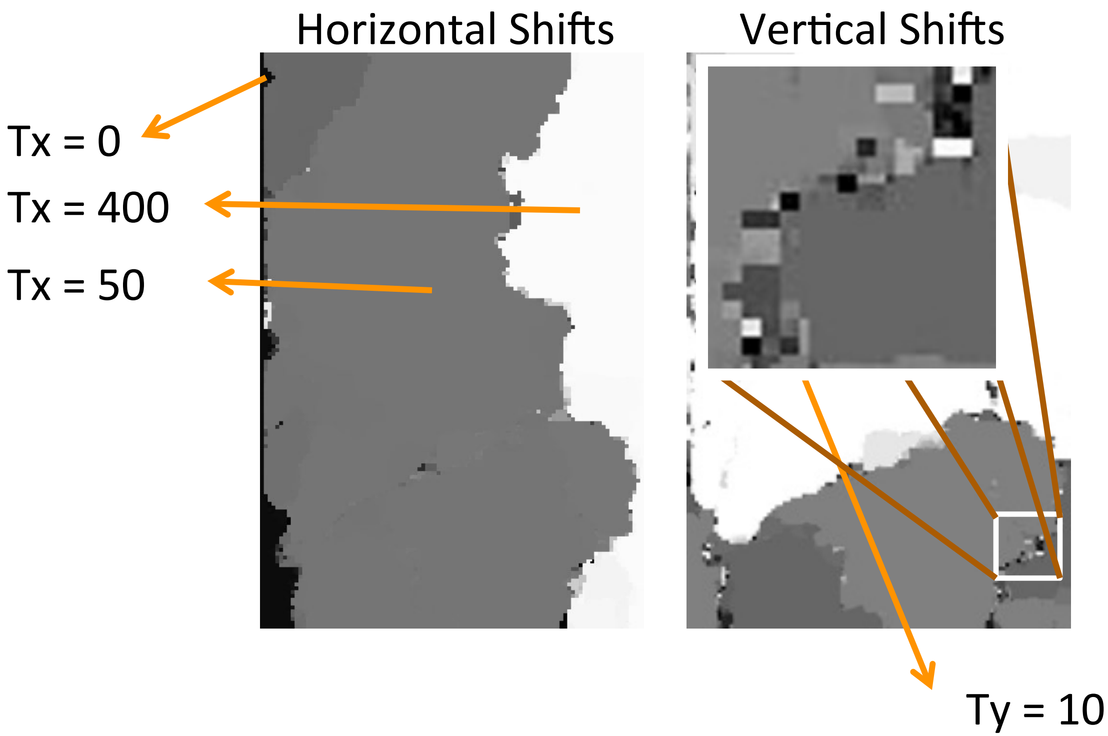
Output as a Composition of Input Parts

Output

Shift-Map
Output Image

Input

Shift-Map Approach



- Minimal distortion
- Adaptive boundaries
- Fast optimization

The Energy Minimization

The **optimal mapping** - can be described as an Energy Minimization problem

$$E(M) = \alpha \sum_{p \in R} E_d(M(p)) + \sum_{p,q \in N} E_s(M(p), M(q))$$

Data term :
External Editing Requirement

Smoothness term :
Avoid Stitching Artifacts

Compute For Each Pixel

Compute For Each Pair of Neighboring pixels

- Unified representation for geometric editing applications
- Solved using a graph labeling algorithm

The Smoothness Term

- Assigns a penalty to a discontinuity introduced to the output image by a discontinuity in the Shift-Map

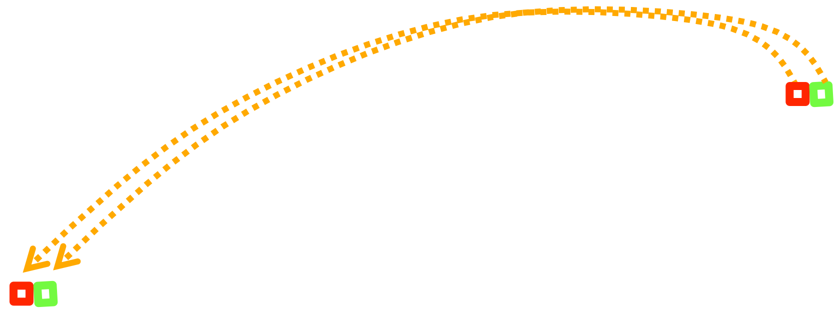
  This term will minimize editing artifacts and create good stitching in the output image

Discontinuities are computed based on ***color*** differences and ***gradient*** differences (preserve image structure)

The Smoothness Term

R - Output Image

I - Input Image



No discontinuity
in the shift-map

$$M(p) = M(q) \Rightarrow E_s(M(p), M(q)) = 0$$

The Smoothness Term

R - Output Image

I - Input Image

Discontinuity
in the shift-map

$$M(p) = M(q) \Rightarrow E_s(M(p), M(q)) = 0$$

$$M(p) \neq M(q) \Rightarrow E_s(M(p), M(q)) =$$

$$= \overset{\text{For } p}{\boxed{\begin{array}{l}(I(n_{p'}) - I(q'))^2 \\ (\nabla I(n_{p'}) - \nabla I(q'))^2\end{array}}} \begin{array}{c} + \\ + \end{array} \overset{\text{For } q}{\boxed{\begin{array}{l}(I(n_{q'}) - I(p'))^2 + \\ (\nabla I(n_{q'}) - \nabla I(p'))^2\end{array}}}$$

color

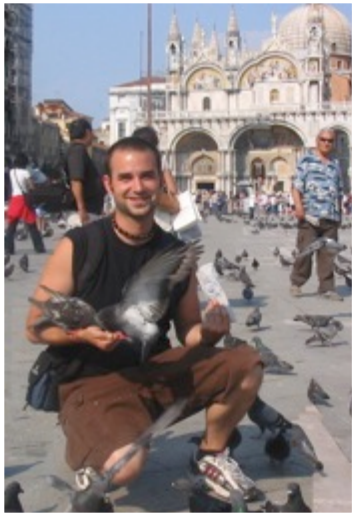
gradient

(Kwatra et al. 03, Agarwala et al. 05)

The Data Term: Retargeting

Data term varies between different application

- Use picture borders
- Can incorporate importance mask
  – Order constraint on mapping is applied to prevent duplications of important areas

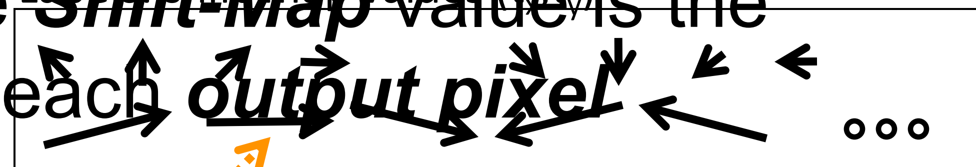


Shift-Map as Graph Labeling

- ***Labels***: relative shift

The minimal energy is solved by graph labeling where the ***Shift-Map*** value is the selected ***label*** for each ***output pixel***

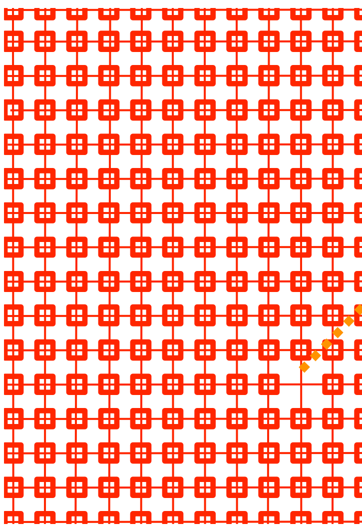
Labels: Shift-map values (t_x, t_y)



Output image pixels

Input image

Nodes:



Shift Map:
assign
a label to
each pixel

Shift-Map as Graph Labeling

- The minimal energy is solved by graph labeling where the ***Shift-Map*** value is the selected ***label*** for each ***output pixel***
- **In this case – a multi-way graph cut (many labels)**
- **Implementation:**
  - Boykov, Y., and Kolmogorov, V. An experimental comparison of min-cut/max-ow algorithms for energy minimization in vision. In Energy Minimization Methods in Computer Vision and Pattern Recognition, 359-374. 2001
  - Boykov, Y., and Veksler, O. Graph Cuts in Vision and Graphics: Theories and Applications. Handbook of Mathematical Models in Computer Vision, Springer, 2006.

Which Method to Use?

- Seam Carving
- Shift Map
- More Later…

Simple Scale is Better

Homo. scaling

Non-homo. warping

Scale&Stretch

Simple Crop is Better

Cropping

Seam-Carving

Scale&Stretch

The ‘Ugly Face’ of Content-aware Retargeting

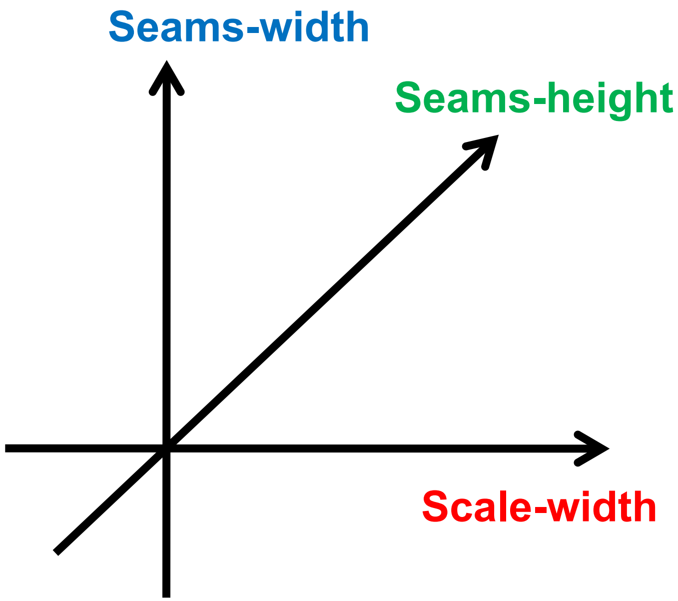
[Rubinstein et al. 2008] [Wang et al. 2008]

The Multi-operator Approach

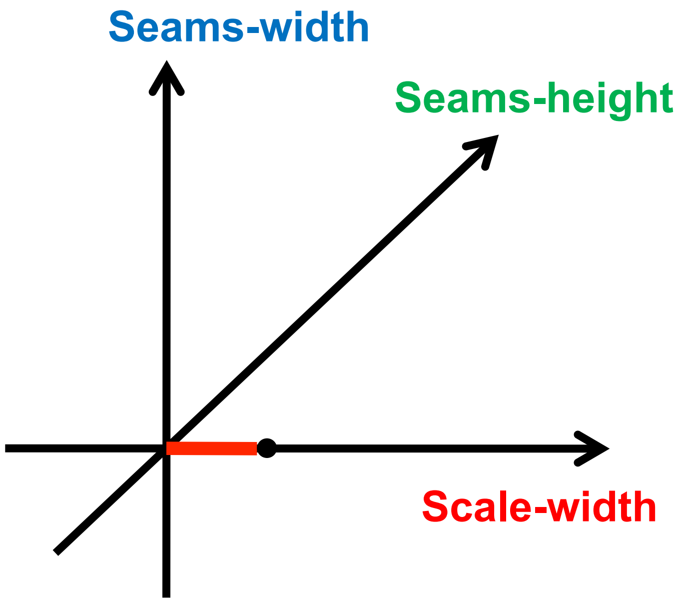
Multiple Operators Resizing Space

Seams-Width

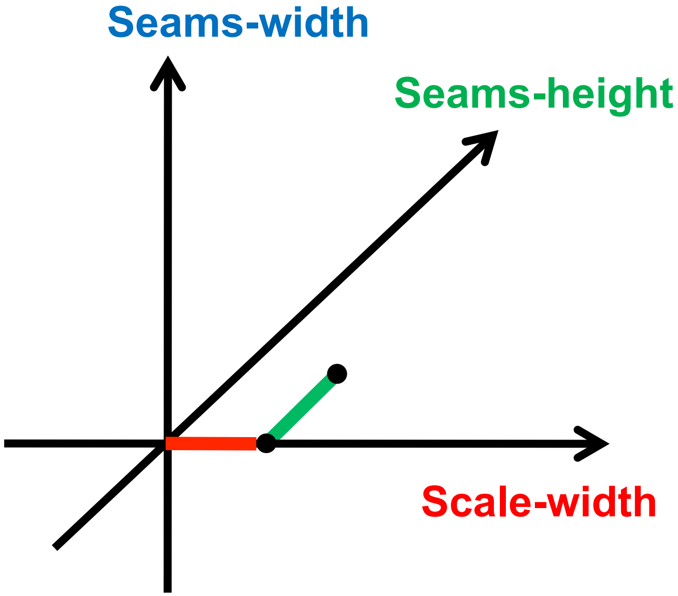
Multi-operator Sequence



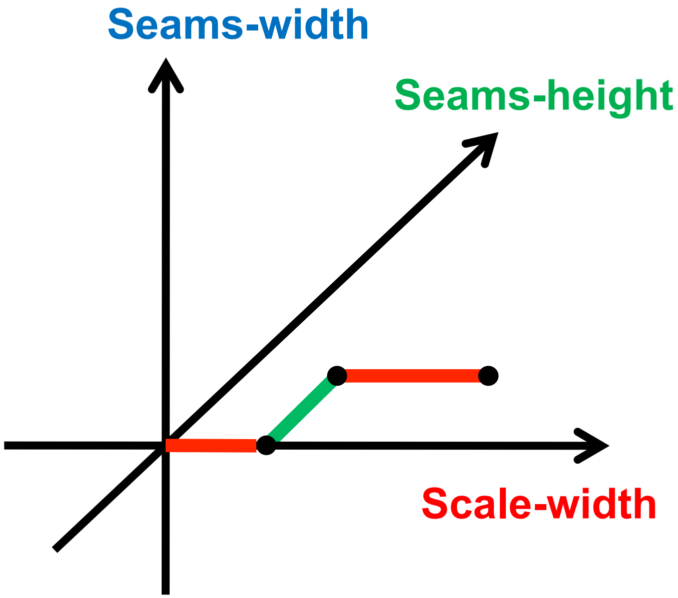
Multi-operator Sequence



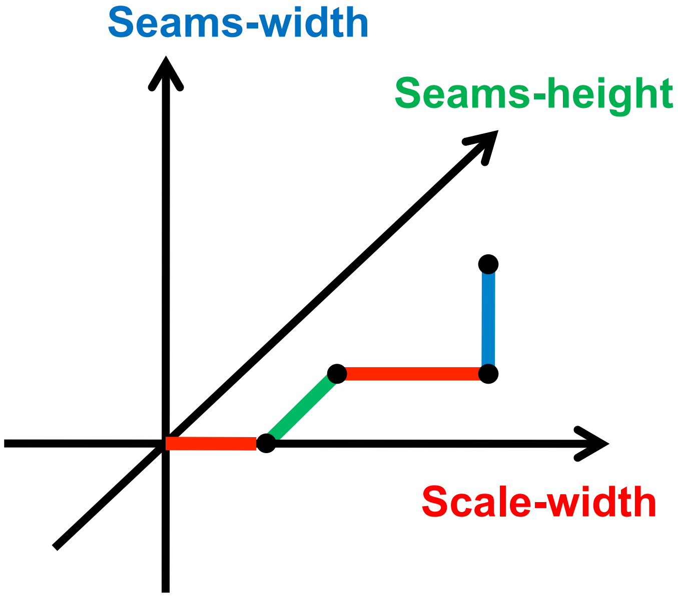
Multi-operator Sequence



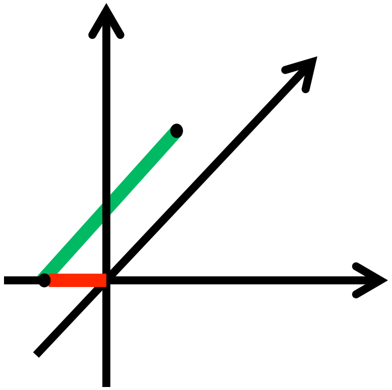
Multi-operator Sequence

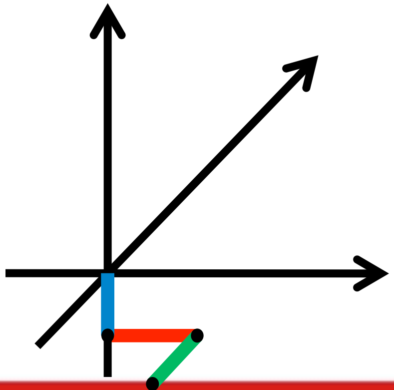


Multi-operator Sequence

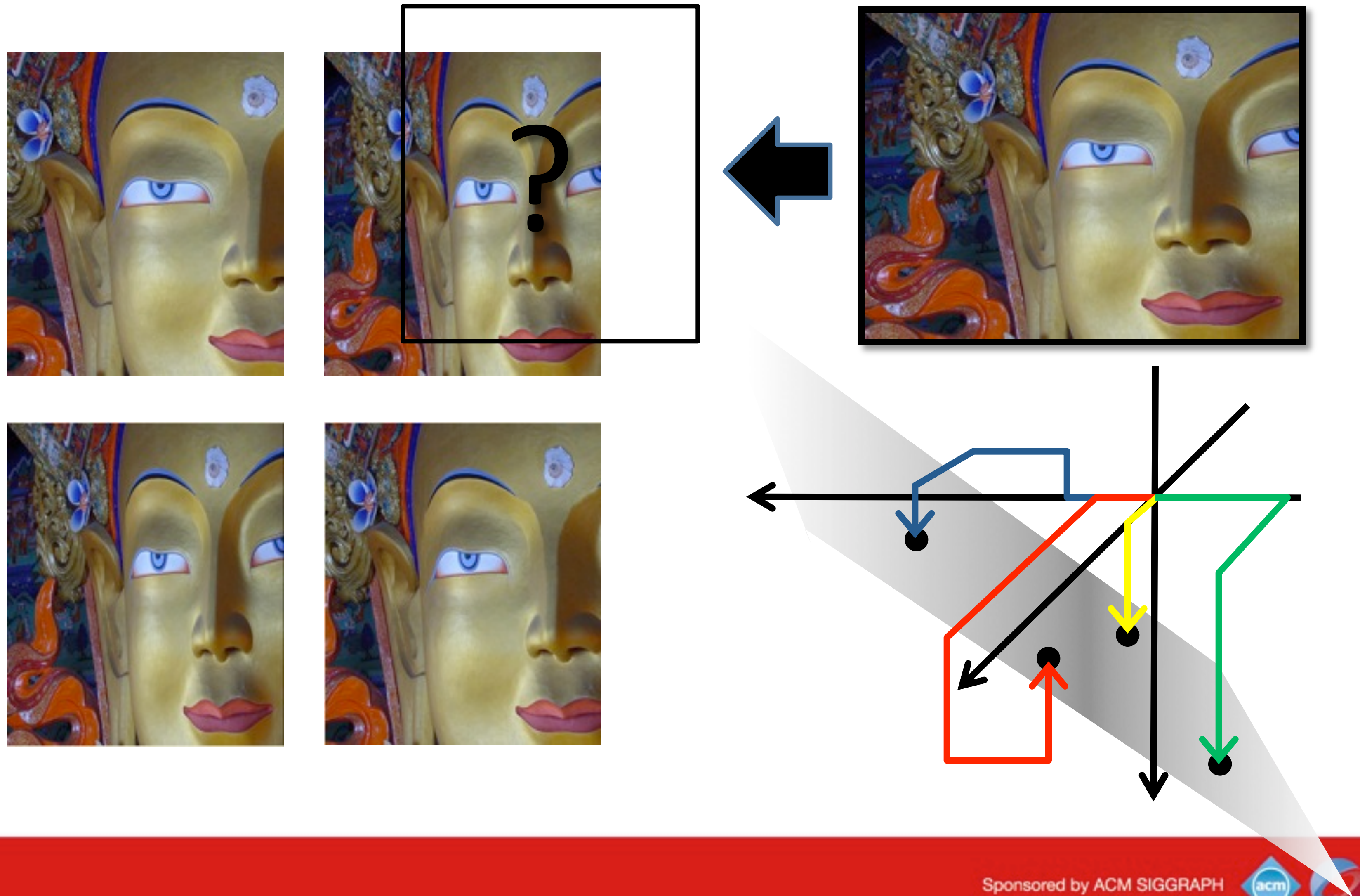


Different Paths → Different Results

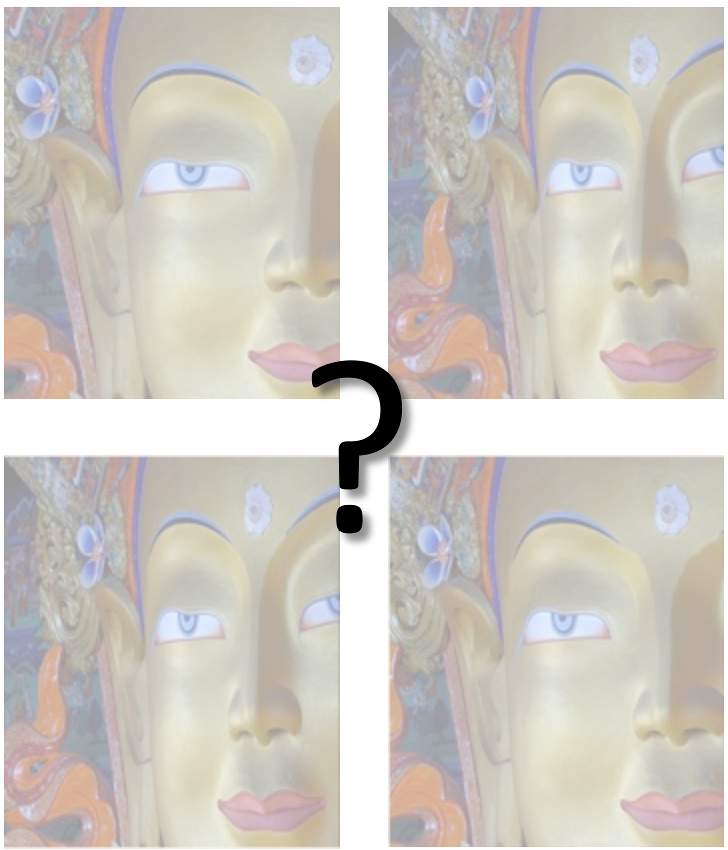


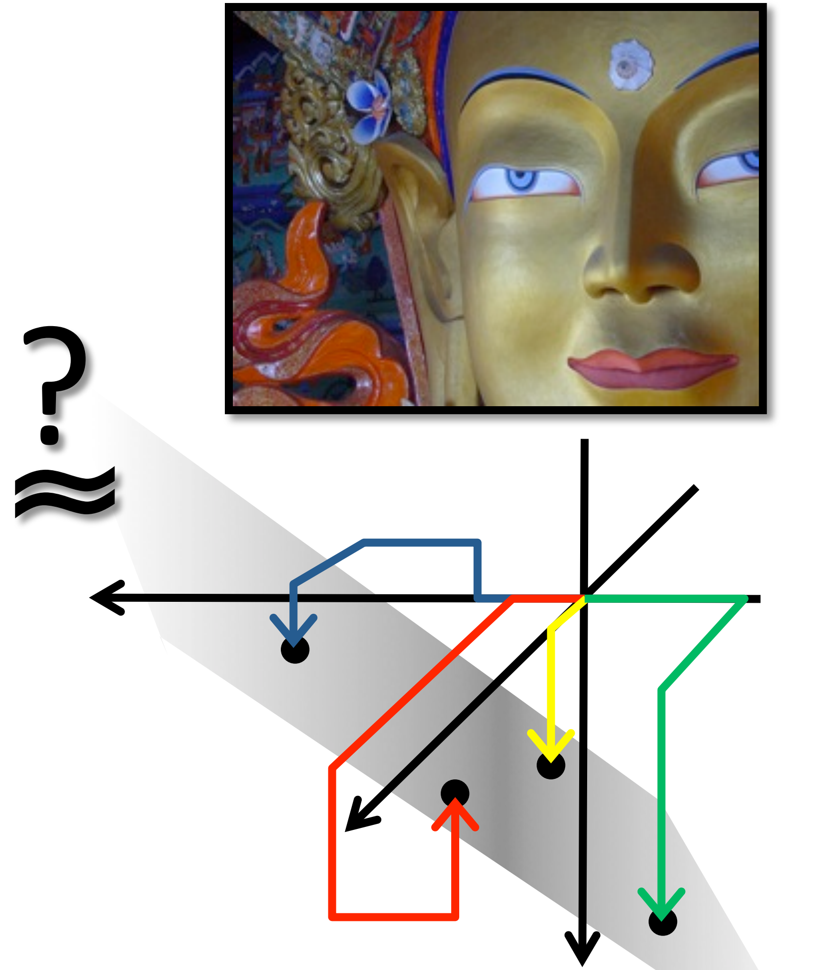


The Resizing Space



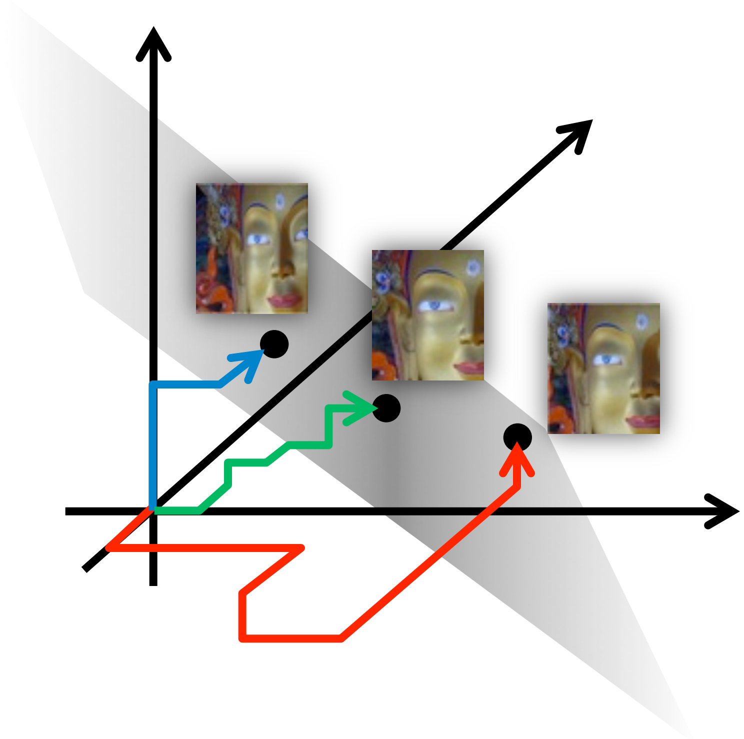
Which Path is Better?





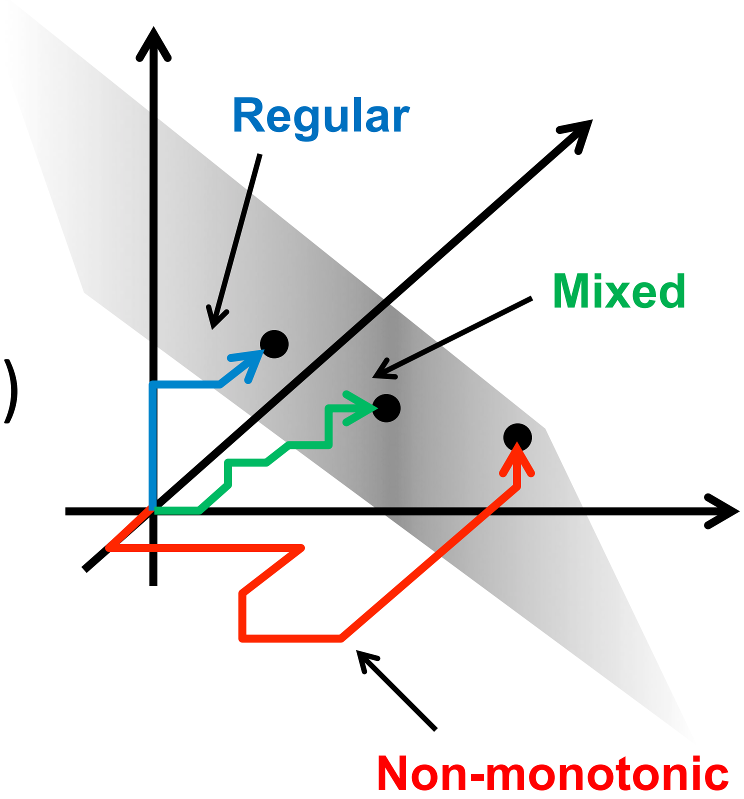
General Optimization Procedure

- Loop over paths
  - Measure distance between result and original image
- Choose best result
- Problem: infinite number of paths!



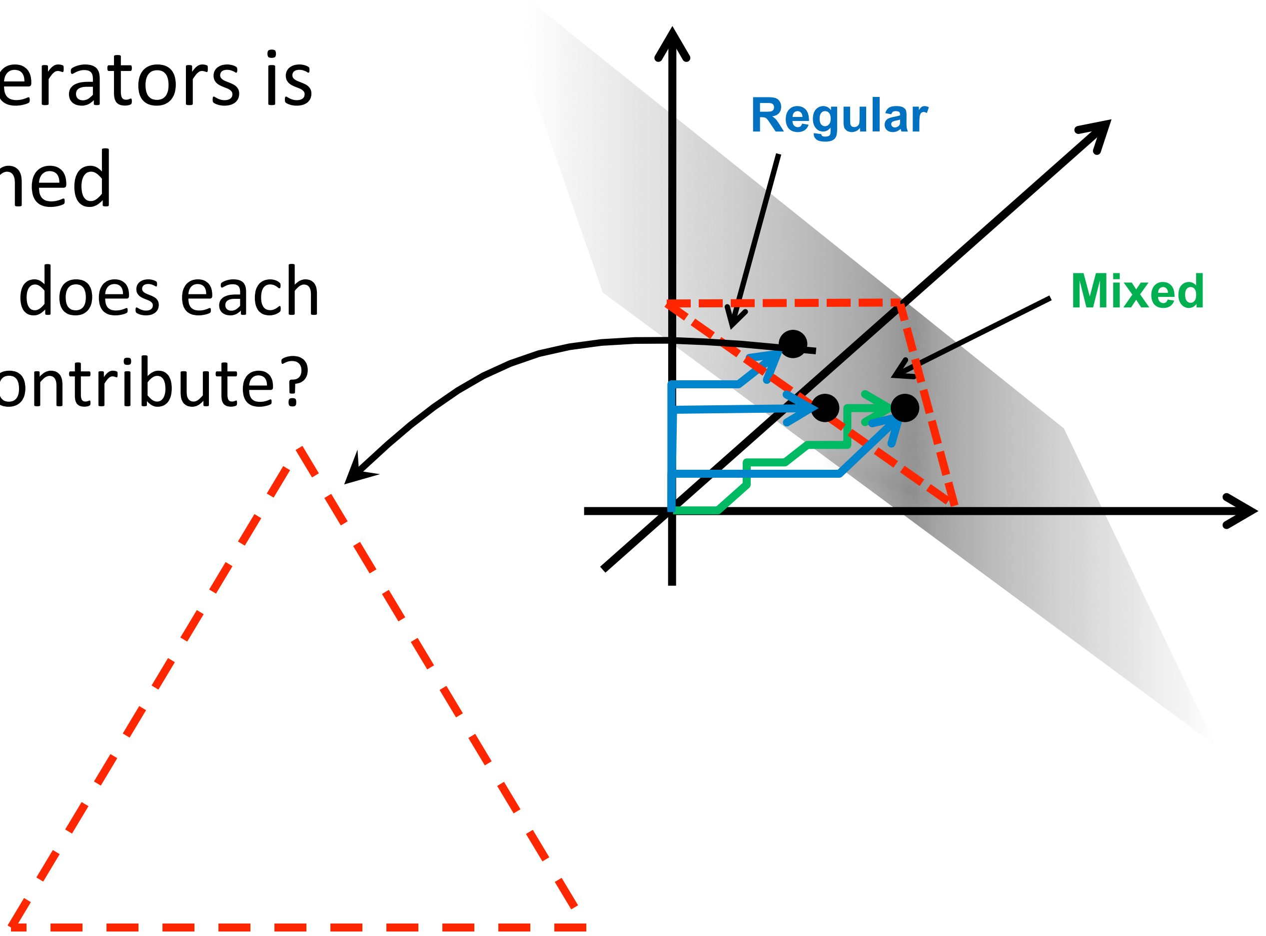
Limit the Search Space

- Restrictions:
  – Remove Non *Monotonic* (infinite)
  – *Remove Mixed* (exponential)
  – *Leave Regular* (polynomial)

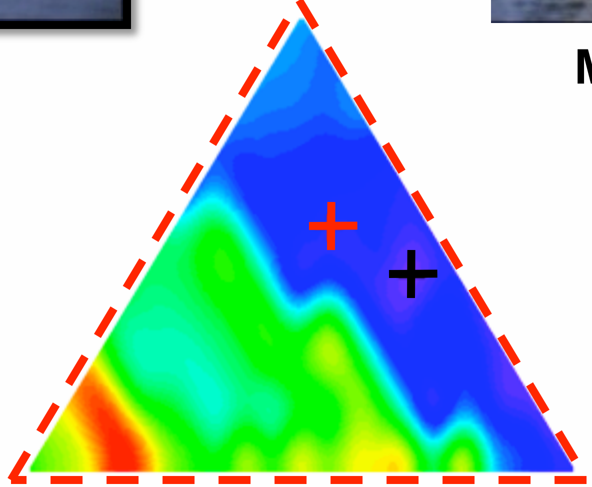


Optimal Regular Path

- Order of operators is predetermined
  - How much does each operator contribute?



Optimal 3-operator Regular Path



Color plot of BDW

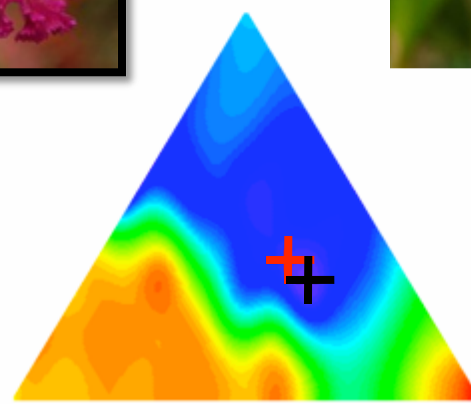
Multi-op

Mean of users results

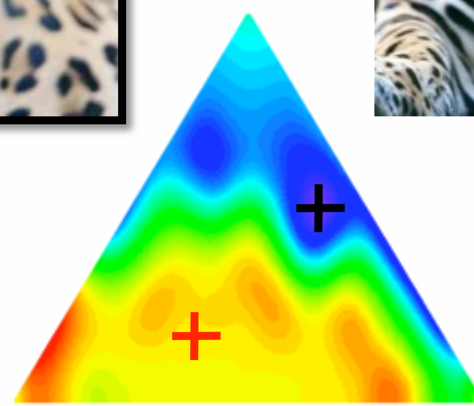
Optimal 3-operator Regular Path

Multi-op

Users



Optimal 3-operator Regular Paths



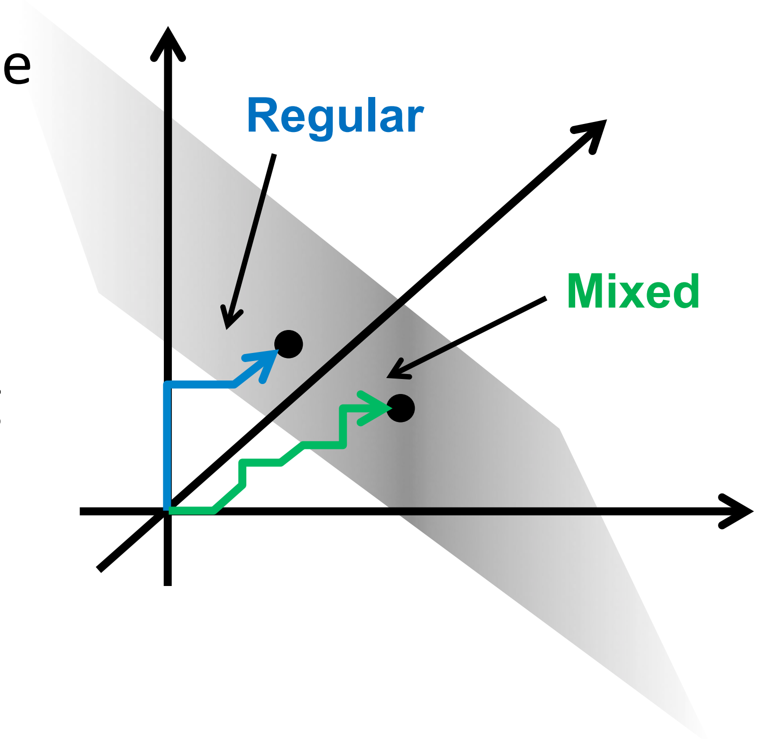
Multi-op

Users

Optimal mixed path

- Exponential (in size change) possible paths

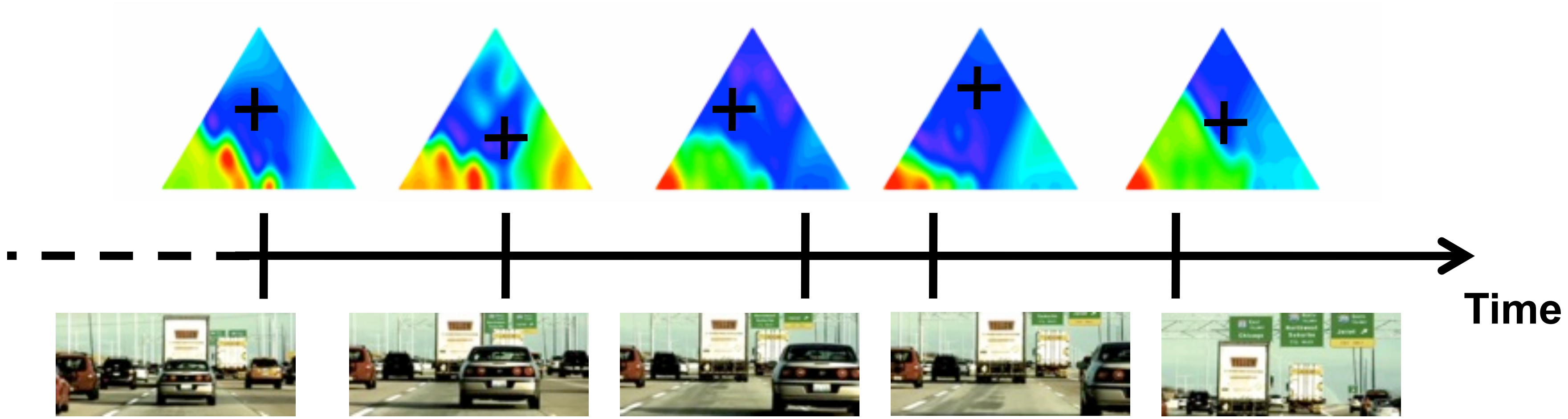
- Using an assumption, can be calculated in polynomial time using dynamic programming



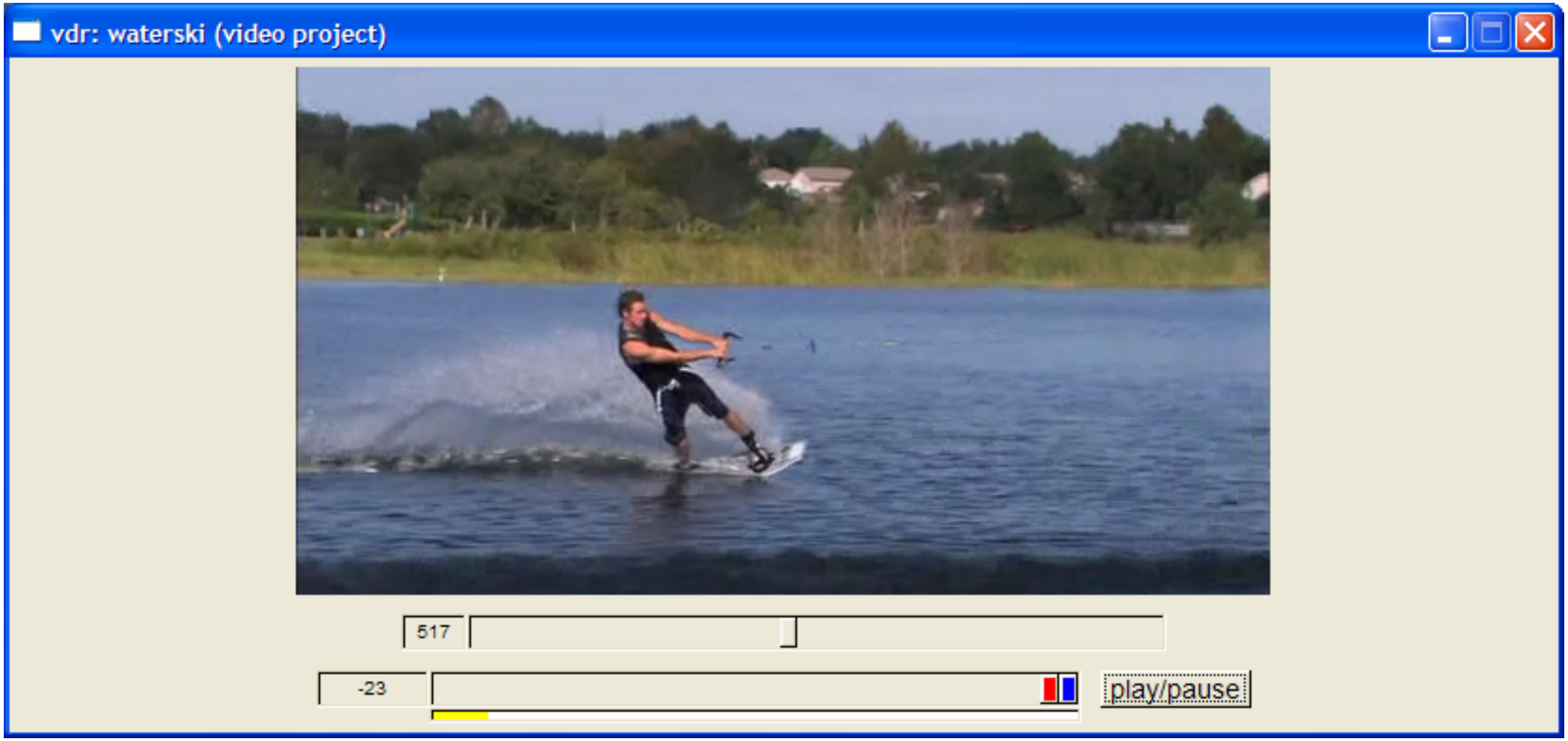
(Algorithm is detailed in the paper)

Multi-operator Video Retargeting

- Optimal multi-operator sequence in one frame need not be optimal in another
- Keyframes + interpolation



Multi-operator Video Retargeting



Summary

- Images as graphs
- Pixels are discrete entities
- Algorithms:
  - Dynamic programming
  - Graph cut
- Operators:
  - Scale
  - Crop
  - Seam carving
  - Shift map
- Multiple Operators

INTERACTIVE GEOMETRY LAB

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Retargeting by Warping

Olga Sorkine-Hornung
ETH Zurich

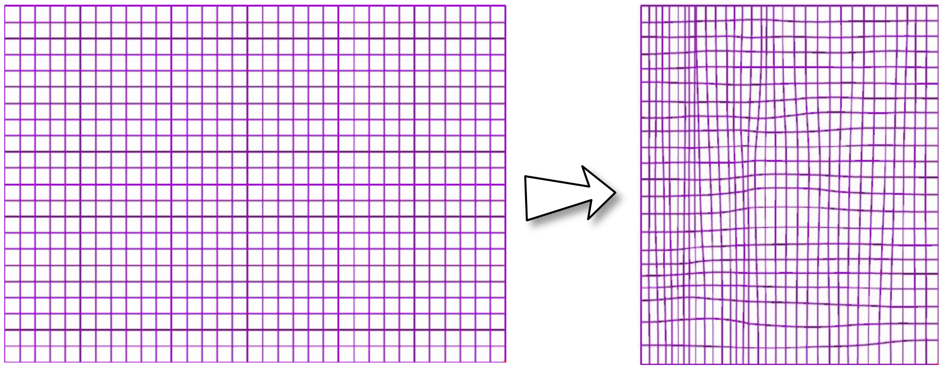
SIGGRAPH ASIA 2012

Problem definition

- Image resizing as a continuous problem:
  - Deform (a portion of) the 2D plane
- The deformation should respect:
  - Desired target resolution (image size)
  - Content preservation

Problem definition

- Image resizing as a continuous problem:
  - Deform (a portion of) the 2D plane
- The deformation should respect:
  - Desired target resolution (image size)
  - Content preservation



isoparametric lines
of a warping function

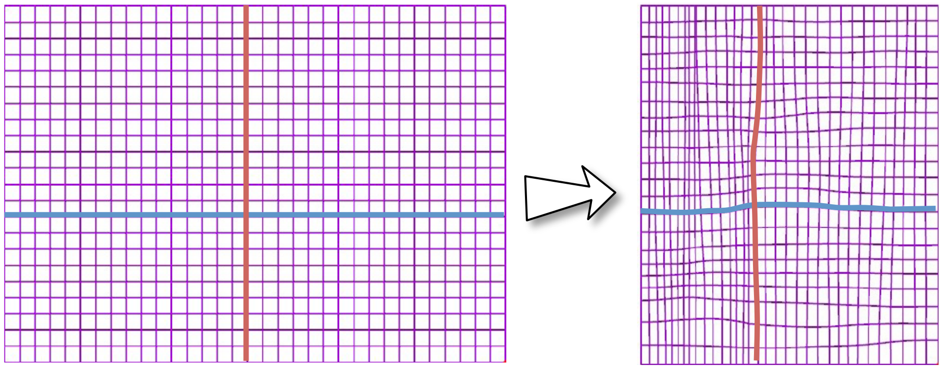
What is a warp?

- Function that deforms space

$$F : \mathbb{R}^2 \to \mathbb{R}^2$$

- Components of F :

$$F(x, y) = (F_x(x, y),\ F_y(x, y))$$

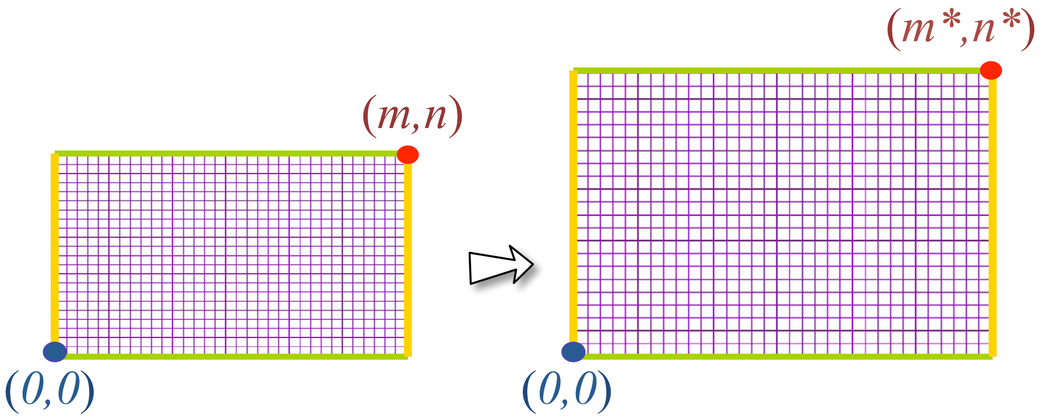


isoparametric lines
of a warping function

Warps for image retargeting

- F should be continuous and at least piecewise C^1-smooth
  - Prevents discontinuities and artifacts in the warped images
- Boundary conditions:
  - Target resolution
  - Keep rectangular shape

$$F_x(0,\cdot) = 0$$
$$F_x(m,\cdot) = m^*$$
$$F_y(\cdot,0) = 0$$
$$F_y(\cdot,n) = n^*$$



Designing content-aware warps

- Variational formulation – find F by optimization
  - Use derivatives of F to describe desired properties

$$\frac{\partial F}{\partial x}(x,y) = \left(\frac{\partial F_x}{\partial x} \; \frac{\partial F_y}{\partial x}\right)^T \qquad \frac{\partial F}{\partial y}(x,y) = \left(\frac{\partial F_x}{\partial y} \; \frac{\partial F_y}{\partial y}\right)^T$$

  - Use the importance map

$$S(x,y) \qquad S : \mathbf{I} \to [0,1]$$

- Define an energy functional and minimize it!

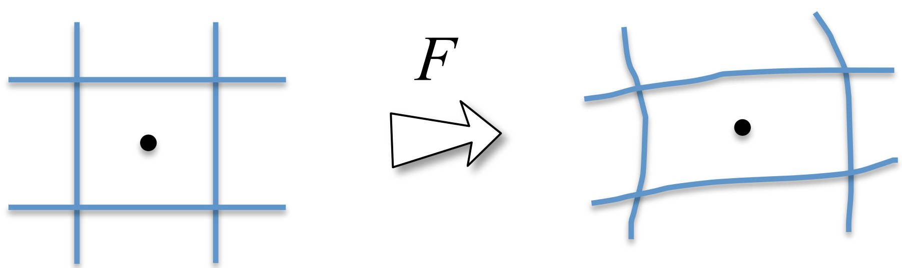
$$\boxed{F = \arg\min_F \; E(F)}$$

We don't care about the values of F themselves but about their local **relationships**. **Derivatives** characterize **local** behavior of F.

Partial derivatives and Jacobian

- Jacobian: best local linear approximation of F

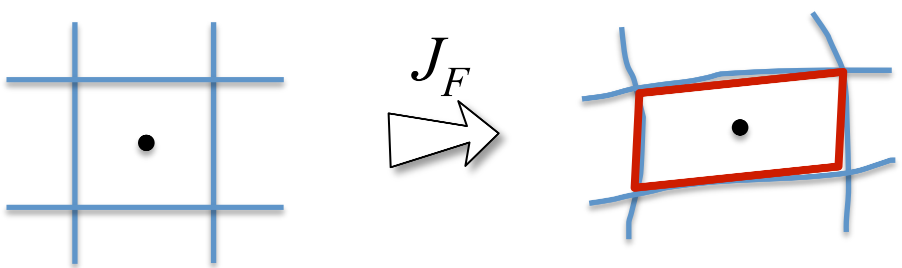
$$J_F(x,y) = \begin{pmatrix} \dfrac{\partial F}{\partial x} & \dfrac{\partial F}{\partial y} \end{pmatrix} = \begin{pmatrix} \partial F_x / \partial x & \partial F_x / \partial y \\ \partial F_y / \partial x & \partial F_y / \partial y \end{pmatrix}$$



Partial derivatives and Jacobian

- Jacobian: best local linear approximation of F

$$J_F(x,y) = \begin{pmatrix} \frac{\partial F}{\partial x} & \frac{\partial F}{\partial y} \end{pmatrix} = \begin{pmatrix} \partial F_x / \partial x & \partial F_x / \partial y \\ \partial F_y / \partial x & \partial F_y / \partial y \end{pmatrix}$$



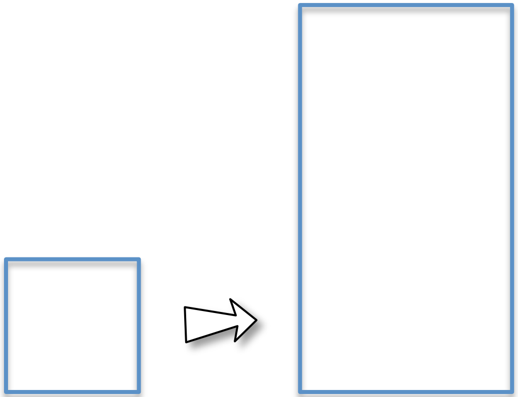
non-uniform scale and shear

Partial derivatives and Jacobian

- Example: if F is non-uniform scaling:

$$F(x, y) = (1.5x, 3y)$$

$$J_F(x, y) = \begin{pmatrix} 1.5 & 0 \\ 0 & 3 \end{pmatrix}$$



- If the Jacobian is shape-preserving then F is locally shape-preserving!
  - shape preserving = uniform scale only
  - rotations are not included

Example: trivial variational warp

- We wish that F is shape-preserving everywhere:

$$\int_{x=0}^{m}\int_{y=0}^{n} \|J_F - I\|^2 dx\, dy \quad \to \quad \min$$

$$\|J_F - I\|^2 = \left\|\partial F/\partial x - (1\ 0)^T\right\|^2 + \left\|\partial F/\partial y - (0\ 1)^T\right\|^2$$

- We also have boundary conditions:

$$F_x(0,\cdot) = 0, \quad F_x(m,\cdot) = m^*, \quad F_y(\cdot,0) = 0, \quad F_y(\cdot,n) = n^*$$

- Result: F is homogeneous scaling

$$F(x,y) = ((m^*/m)\,x,\ (n^*/n)\,y)$$

Employing the importance map

- Weight the energy by the importance map:

$$\int_{x=0}^{m} \int_{y=0}^{n} S(x,y)\, \|J_F - I\|^2 \, dx\, dy \quad \rightarrow \quad \min$$

  - Non-important parts are allowed to distort

Discretization

- To obtain the numerical solution we need to discretize

$$\int_{x=0}^{m}\int_{y=0}^{n} S(x,y)\,\|J_F - I\|^2\,dx\,dy$$

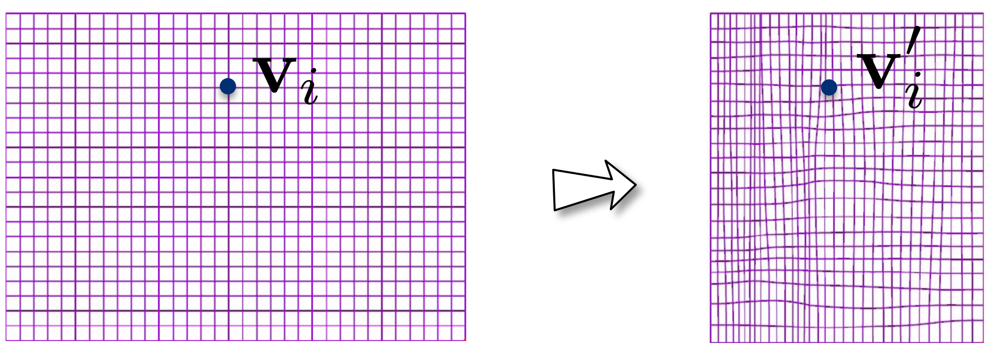
- Meaning, we will find discrete values of F on some grid mesh

Discretization

- To obtain the numerical solution we need to discretize

$$\int_{x=0}^{m} \int_{y=0}^{n} S(x,y) \, \|J_F - I\|^2 \, dx \, dy$$

- Meaning, we will find discrete values of F on some grid mesh



$$F(\mathbf{v}_i) = \mathbf{v}_i'$$

Discretization

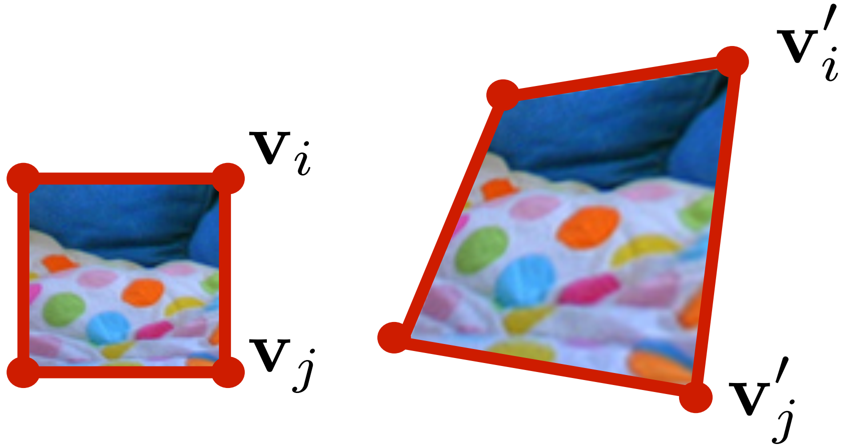
- To obtain the numerical solution we need to discretize

$$\int_{x=0}^{m}\int_{y=0}^{n} S(x,y)\,\|J_F - I\|^2\,dx\,dy$$

- Meaning, we will find discrete values of F on some grid mesh
  - The mesh can have pixel resolution, quad grid
  - Or coarser grid – for better efficiency
  - Adaptive meshes – hasn't been fully explored yet, but see [Laffont et al., Graphics Interface 2010]

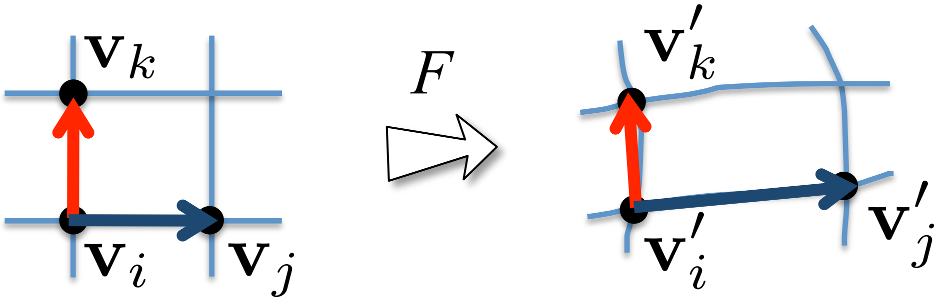
Discretization

- Between the computed discrete values of F, we will interpolate F
  - per-element interpolation
  - bi-linear (like texture mapping)
  - bi-cubic, splatting…



Discretization

- Discretization of the derivatives of F :



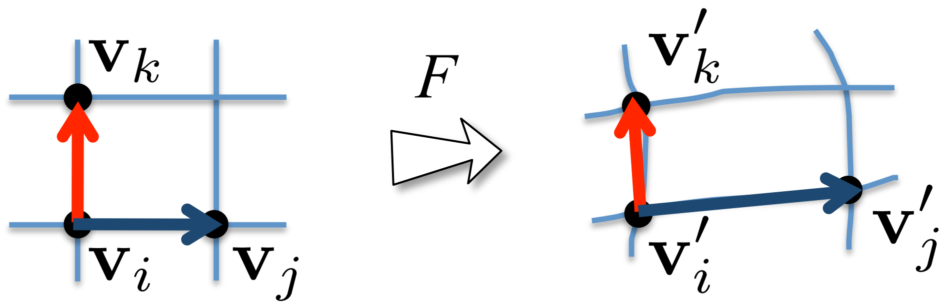
assuming each edge length in the original grid is 1:

$$\partial F / \partial x = \mathbf{v}'_j - \mathbf{v}'_i$$

$$\partial F / \partial y = \mathbf{v}'_k - \mathbf{v}'_i$$

Discretization

- Discretization of the derivatives of F :



$$\|J_F - I\|^2 = \left\|\partial F/\partial x - (1\ 0)^T\right\|^2 + \left\|\partial F/\partial y - (0\ 1)^T\right\|^2 \approx$$

$$\approx \left\|(\mathbf{v}_j' - \mathbf{v}_i') - (1\ 0)^T\right\|^2 + \left\|(\mathbf{v}_k' - \mathbf{v}_i') - (0\ 1)^T\right\|^2 =$$

$$= \left\|(\mathbf{v}_j' - \mathbf{v}_i') - (\mathbf{v}_j - \mathbf{v}_i)\right\|^2 + \left\|(\mathbf{v}_k' - \mathbf{v}_i') - (\mathbf{v}_k - \mathbf{v}_i)\right\|^2$$

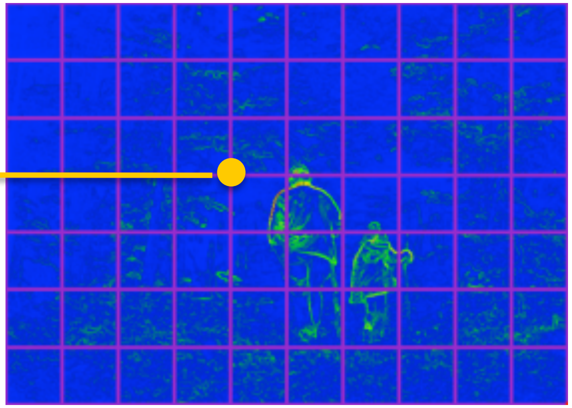
← Jacobian energy

← discretized term: look at edges

Discretization

- Discretization of the importance map – lump values at each grid mesh vertex

$S(\mathbf{v}_i)$ – average value



- The simple energy we saw earlier will look like this:

$$E(F) = \int_{x=0}^{m} \int_{y=0}^{n} S(x,y) \, \|J_F - I\|^2 \, dx \, dy \approx \sum_{(i,j)\in\mathcal{E}} S(\mathbf{v}_i) \, \left\| (\mathbf{v}'_j - \mathbf{v}'_i) - (\mathbf{v}_j - \mathbf{v}_i) \right\|^2$$

Solving the optimization problem

- After discretization we get a system of equations to solve

$$E(F) = \sum_{(i,j)\in\mathcal{E}} S(\mathbf{v}_i)\left\|(\mathbf{v}'_j - \mathbf{v}'_i) - (\mathbf{v}_j - \mathbf{v}_i)\right\|^2 \quad \to \quad \min$$

$$\frac{\partial}{\partial \mathbf{v}'_i} E(F) = \sum_{(i,j)\in\mathcal{E}} 2S(\mathbf{v}_i)\left((\mathbf{v}'_j - \mathbf{v}'_i) - (\mathbf{v}_j - \mathbf{v}_i)\right) \overset{!}{=} 0$$

unknowns

constants (depend only on the original image)

Solving the optimization problem

- After discretization we get a system of equations to solve

$$E(F) = \sum_{(i,j)\in\mathcal{E}} S(\mathbf{v}_i) \left\| (\mathbf{v}'_j - \mathbf{v}'_i) - (\mathbf{v}_j - \mathbf{v}_i) \right\|^2 \quad \rightarrow \quad \min$$

$$\frac{\partial}{\partial \mathbf{v}'_i} E(F) = \sum_{(i,j)\in\mathcal{E}} 2S(\mathbf{v}_i) \left((\mathbf{v}'_j - \mathbf{v}'_i) - (\mathbf{v}_j - \mathbf{v}_i) \right) \stackrel{!}{=} 0$$

- We get a sparse linear system:

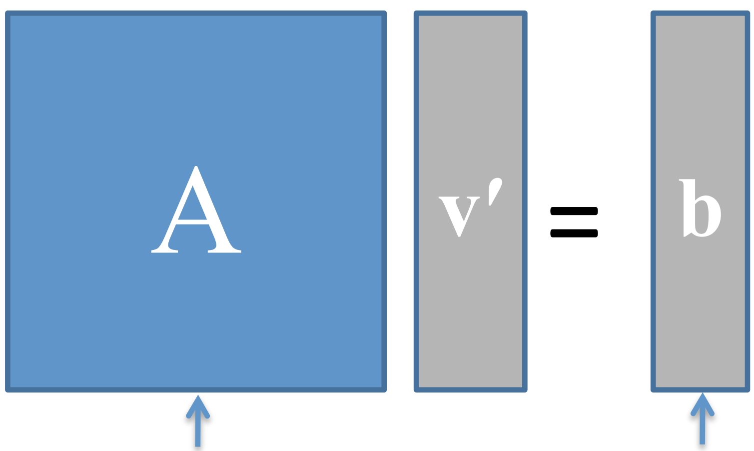
$$A\mathbf{v}' = \mathbf{b}$$

A few words about numerics

- Depending on the energy functional, the equations could be linear or nonlinear
- Linear equations are simple to solve
  - Sparsity
  - Direct solvers (libraries exist, plug-and-play)
  - Multigrid solvers – fast GPU implementation
- Nonlinear is harder to solve – need to be careful about the energy design

Solving sparse linear systems

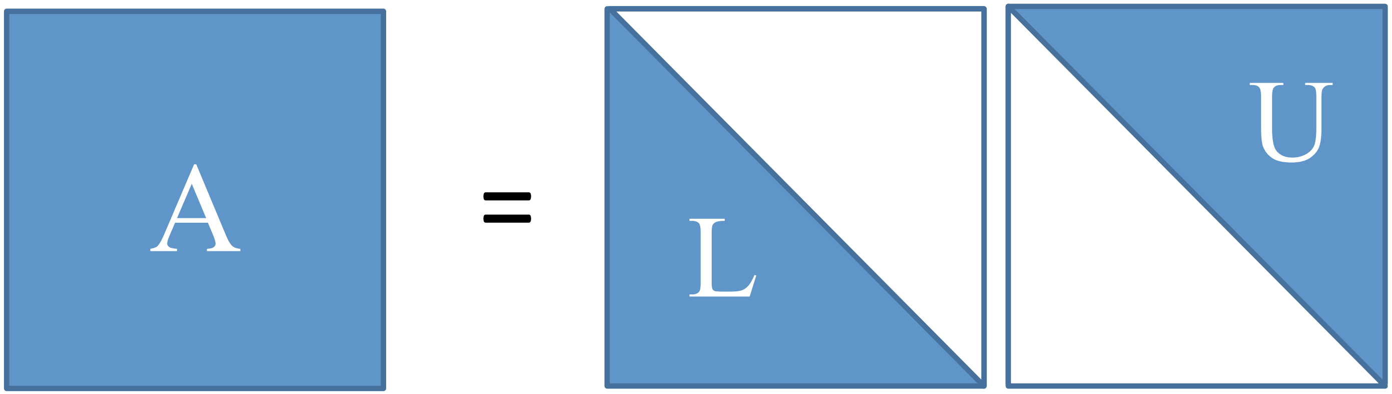
- Direct solvers can be used:
  - Easy to code – just use library, no parameters
  - Efficient especially when matrix is fixed, only right-hand side changes



depends on input image only: mesh and importance map

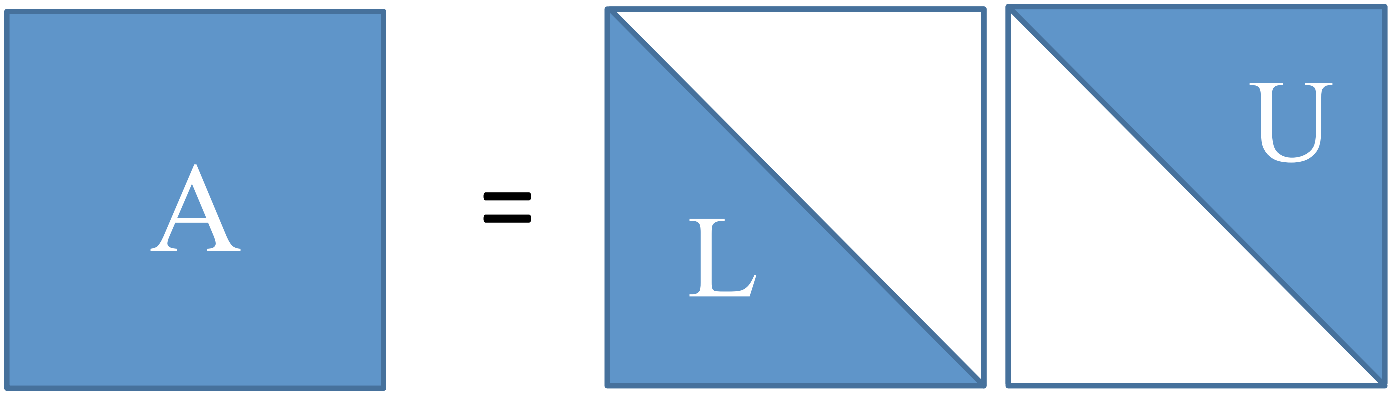
depends boundary conditions

Matrix factorization: LU decomposition



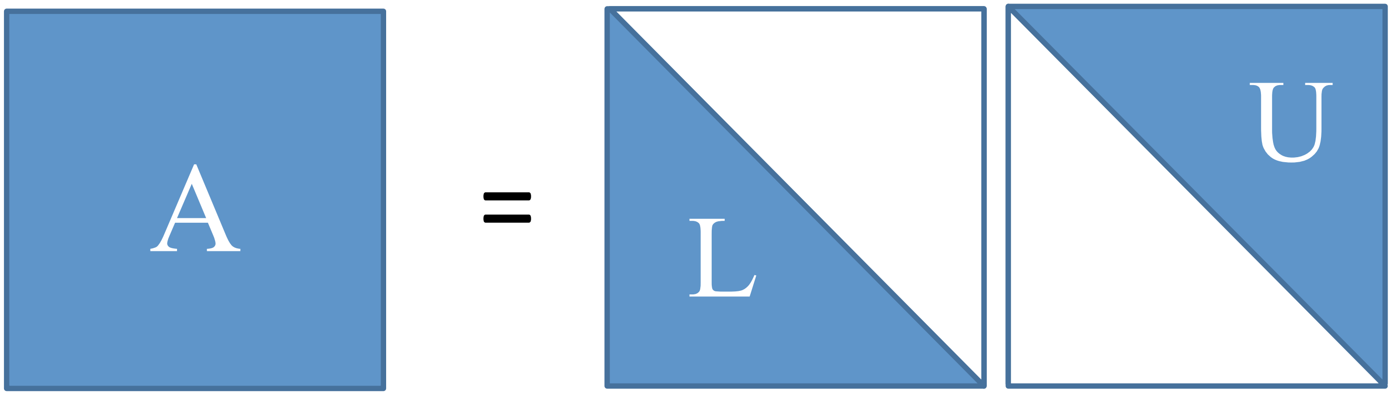
$$\mathrm{A}\mathbf{v}' = \mathbf{b}$$
$$\mathrm{LU}\mathbf{v}' = \mathbf{b}$$

Matrix factorization: LU decomposition



$$\mathrm{A}\mathbf{v}' = \mathbf{b}$$
$$\mathrm{L}(\mathrm{U}\mathbf{v}') = \mathbf{b}$$

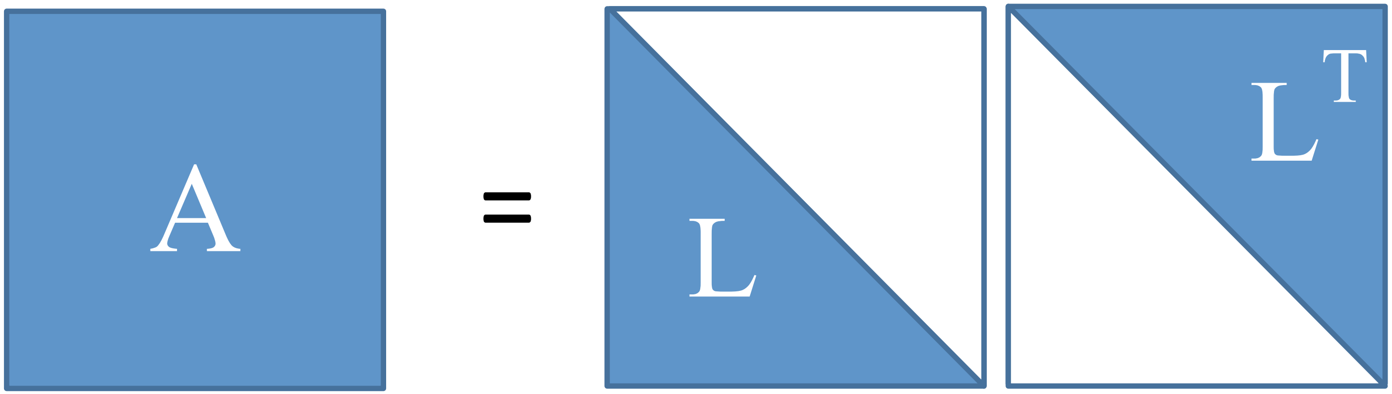
Matrix factorization: LU decomposition



$$\begin{aligned} A\mathbf{v}' &= \mathbf{b} \\ L(U\mathbf{v}') &= \mathbf{b} \end{aligned} \quad \Longrightarrow \quad \left.\begin{aligned} L\mathbf{w} &= \mathbf{b} \\ U\mathbf{v}' &= \mathbf{w} \end{aligned}\right\}$$

This is backsubstitution. If L, U are sparse, it is very fast. The hard work is computing L and U.

Matrix factorization: Cholesky decomposition



Cholesky factor exists if A is positive definite. It is even better than LU because we save memory.

A is positive definite in our case since we are solving least-squares problems.

Multigrid solvers

- Progressively coarsen the grid mesh
- Solve on the coarse level, then interpolate solution to the finer level
- Iterate till error $||A\mathbf{v}' - \mathbf{b}||$ is small enough

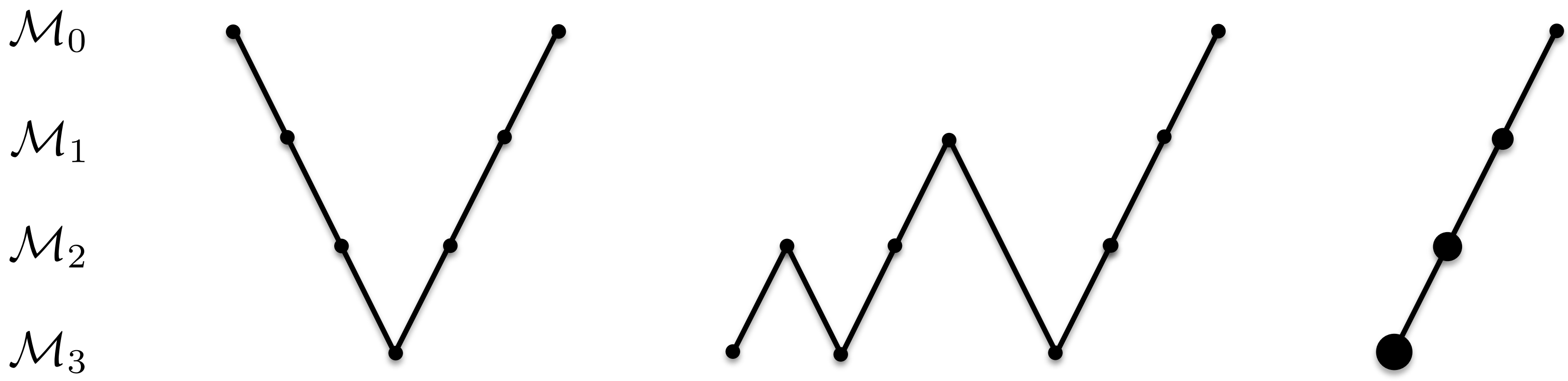


Image from [Botsch, Bommes, Kobbelt 2005]

Solvers – discussion

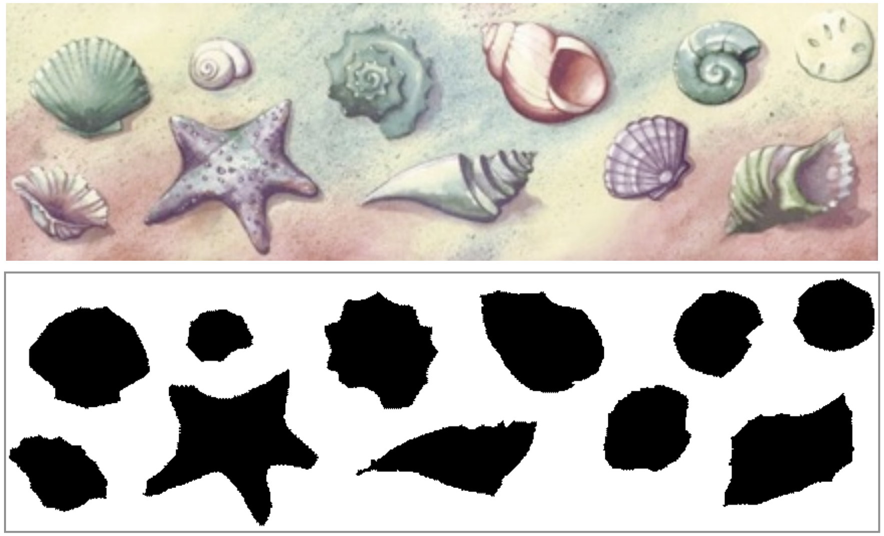
- Direct solvers – easy to implement (use existing library)
- Can factor a 1M × 1M matrix in seconds; solve takes milliseconds
- High memory cost (need to store the factor)

- Multigrid is very efficiently implemented on the GPU
- Low memory consumption
- However, requires setting problem-dependent parameters (number of iterations, hierarchy depth…)

[Gal et al. 06] – feature-aware warp

- Binary importance map
- s is a preset scaling factor, A is homogeneous scaling

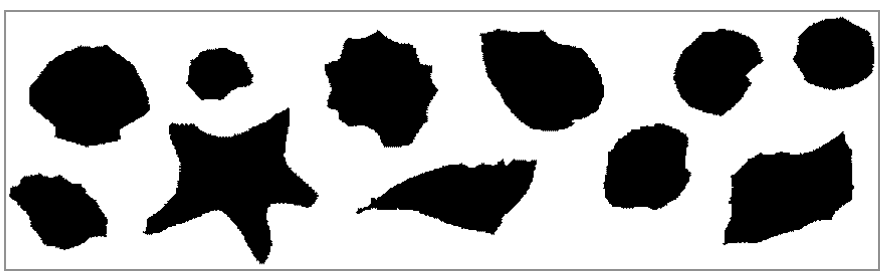
$$E(F) = \int_{x=0}^{m} \int_{y=0}^{n} S(x,y) \left\| J_F - sI \right\|^2 + (1 - S(x,y)) \left\| J_F - A \right\|^2 dx\, dy$$



[Gal et al. 06] – feature-aware warp

- Binary importance map
- s is a preset scaling factor, A is homogeneous scaling

$$E(F) = \int_{x=0}^{m} \int_{y=0}^{n} S(x,y)\, \|J_F - sI\|^2 + (1 - S(x,y))\, \|J_F - A\|^2 \, dx\, dy$$

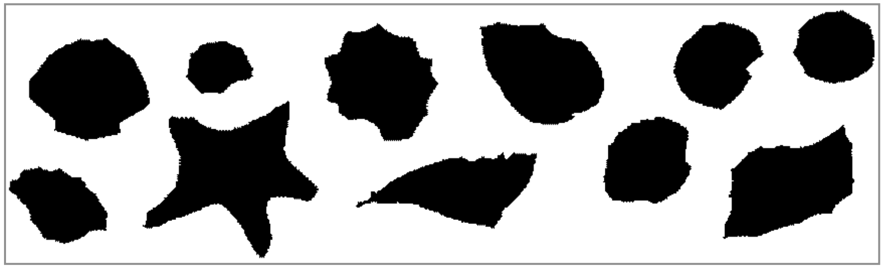


homogeneous scaling (just A)

[Gal et al. 06] – feature-aware warp

- Binary importance map
- s is a preset scaling factor, A is homogeneous scaling

$$E(F) = \int_{x=0}^{m} \int_{y=0}^{n} S(x,y)\,\|J_F - sI\|^2 + (1 - S(x,y))\,\|J_F - A\|^2\,dx\,dy$$



feature-aware warp (energy minimization)

Feature-aware warp: scale factor

$$E(F) = \int_{x=0}^{m} \int_{y=0}^{n} S(x,y) \left\| J_F - sI \right\|^2 + (1 - S(x,y)) \left\| J_F - A \right\|^2 dx\, dy$$

- [Gal et al. 06] defined s as

$$A(x,y) = \left(\frac{m^*}{m} x, \ \frac{n^*}{n} y \right)$$

$$s = \min \left\{ \frac{m^*}{m}, \ \frac{n^*}{n} \right\}$$

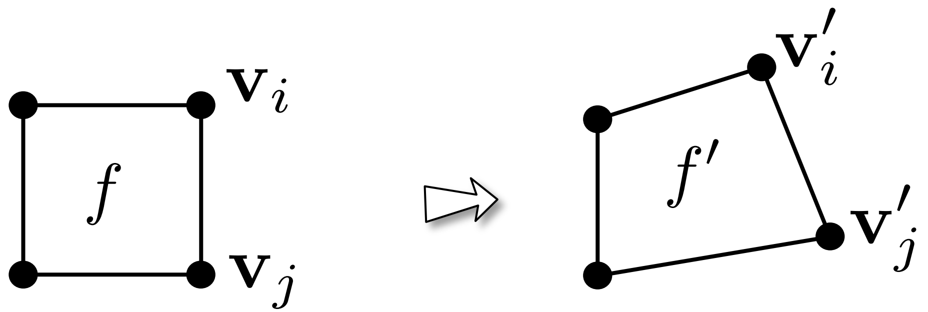
homogeneous scaling required to get the new image dimensions

Results [Gal et al. 2006]

See video at
http://igl.ethz.ch/projects/retargeting/feature-aware-texturing/

[Wang et al. 08]: optimizing the scale

- Scale s_f becomes variable per quad f



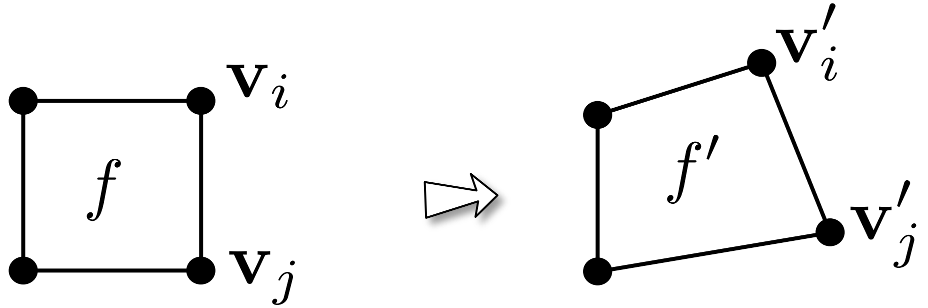
$$D_u(f) = \sum_{(i,j)\in\mathcal{E}(f)} \left\|(\mathbf{v}'_i - \mathbf{v}'_j) - s_f(\mathbf{v}_i - \mathbf{v}_j)\right\|^2$$

s_f – uniform scaling factor

$$\frac{\partial D_u(f)}{\partial s_f} \stackrel{!}{=} 0 \quad \Rightarrow \quad s_f = \frac{\sum_{(i,j)\in\mathcal{E}(f)} (\mathbf{v}'_i - \mathbf{v}'_j)^T(\mathbf{v}_i - \mathbf{v}_j)}{\sum_{(i,j)\in\mathcal{E}(f)} \|\mathbf{v}_i - \mathbf{v}_j\|^2}$$

[Wang et al. 08]: optimizing the scale

- Total energy $E(F)$ – sum up for all the quads



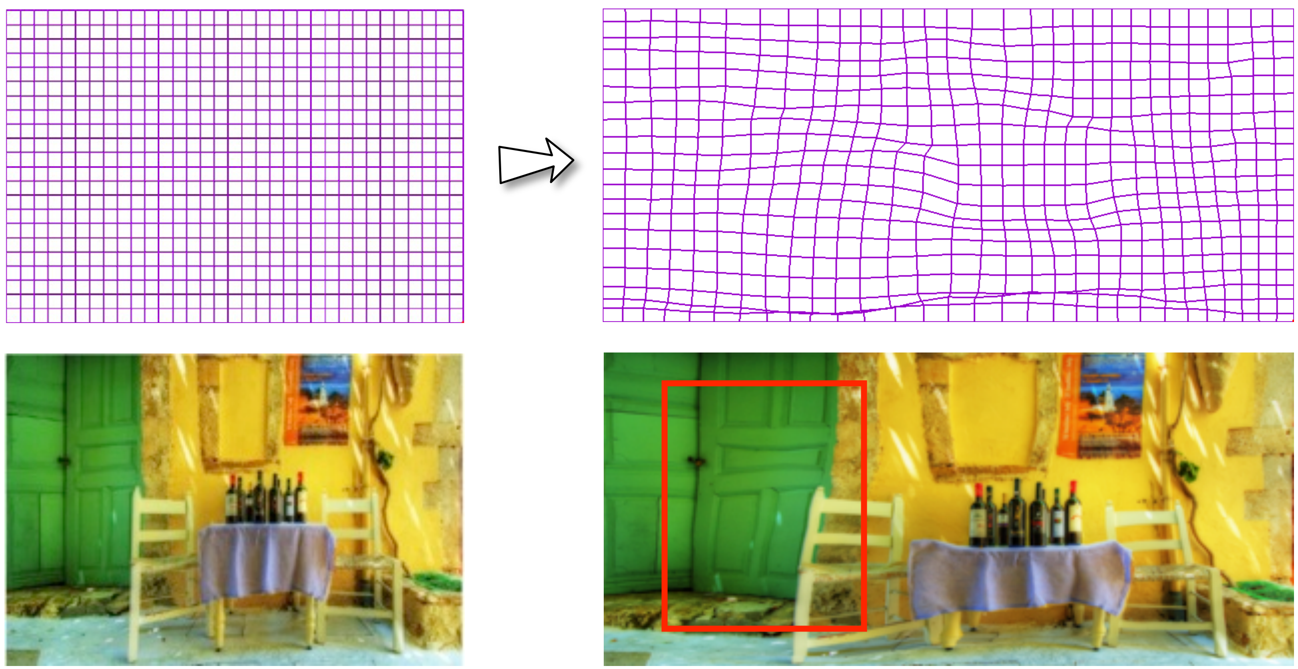
$$D_u(f) = \sum_{(i,j)\in\mathcal{E}(f)} \|(\mathbf{v}'_i - \mathbf{v}'_j) - s_f(\mathbf{v}_i - \mathbf{v}_j)\|^2$$

s_f – uniform scaling factor

$$\frac{\partial D_u(f)}{\partial s_f} \stackrel{!}{=} 0 \quad \Rightarrow \quad s_f = \frac{\sum_{(i,j)\in\mathcal{E}(f)} (\mathbf{v}'_i - \mathbf{v}'_j)^T(\mathbf{v}_i - \mathbf{v}_j)}{\sum_{(i,j)\in\mathcal{E}(f)} \|\mathbf{v}_i - \mathbf{v}_j\|^2}$$

[Wang et al. 08]: optimizing the scale

- Problem: scaling factors are independent of each other so they vary a lot. Grid lines bend as a result!



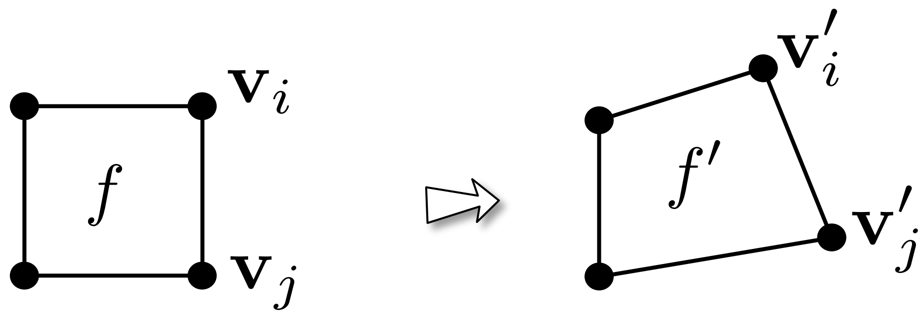
[Wang et al. 08]: line bending

- Add another energy term to prevent line bending:
- Keep original edge orientations but allow length scaling

$$D_l = \sum_{(i,j)\in\mathcal{E}} \|(\mathbf{v}'_i - \mathbf{v}'_j) - l_{ij}(\mathbf{v}_i - \mathbf{v}_j)\|^2$$

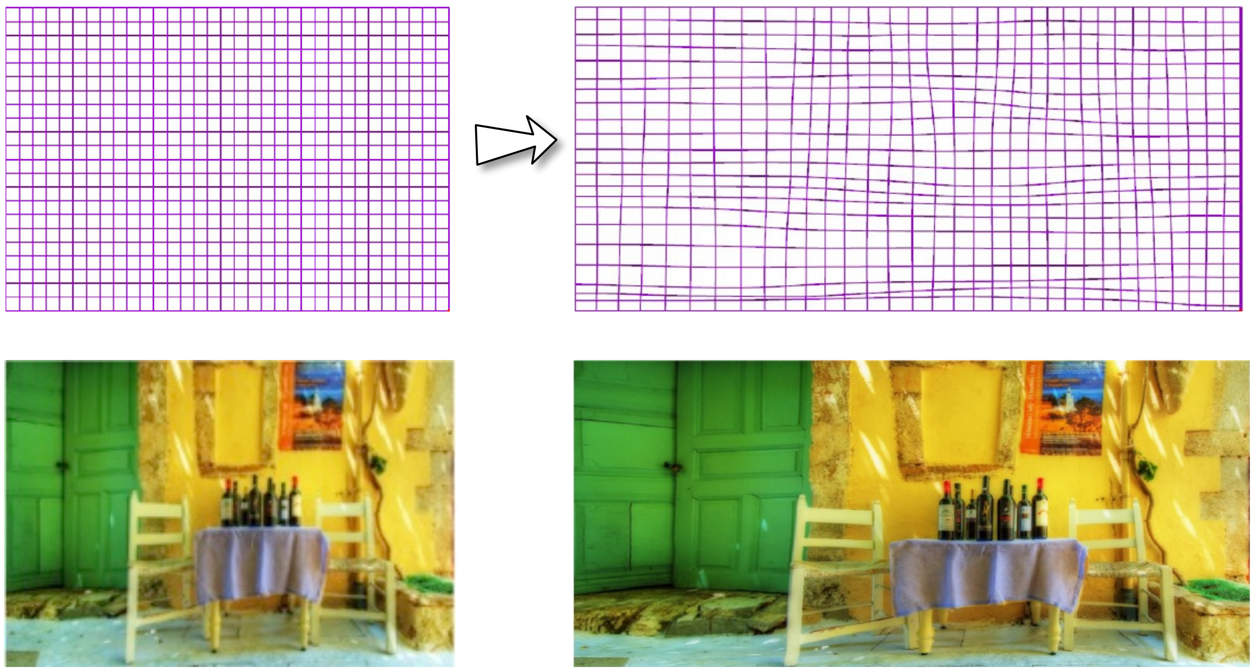
note the nonlinear factor

$$l_{ij} = \|\mathbf{v}'_i - \mathbf{v}'_j\| / \|\mathbf{v}_i - \mathbf{v}_j\|$$



[Wang et al. 08]: line bending

- Result with the bending term added:



[Wang et al. 08]: line bending

- Total energy is nonlinear in $\mathbf{v}'$

$$D_u(f) = \sum_{(i,j)\in\mathcal{E}(f)} \|(\mathbf{v}'_i - \mathbf{v}'_j) - s_f(\mathbf{v}_i - \mathbf{v}_j)\|^2$$

$$D_l = \sum_{(i,j)\in\mathcal{E}} \|(\mathbf{v}'_i - \mathbf{v}'_j) - l_{ij}(\mathbf{v}_i - \mathbf{v}_j)\|^2$$

- Iterative minimization with tricks:

$$l_{ij} = \|\mathbf{v}'_i - \mathbf{v}'_j\| / \|\mathbf{v}_i - \mathbf{v}_j\|$$

  - Keep s_f and l_{ij} as additional variables
  - Do alternating minimization steps (global-local)
    - Fix s_f and l_{ij} and optimize $\mathbf{v}'$
    - Compute new s_f and l_{ij}

- **Sparse direct solver** for the linear system and **reuse** the matrix factorization to gain speed.

Results [Wang et al. 08]

See video at
http://igl.ethz.ch/projects/retargeting/scale-and-stretch/ImageResizing_final.mp4

[Wolf et al. 07]: one-directional scaling

- One-directional energy
  - for resizing horizontally:

$$E(F) = \int_{x=0}^{m} \int_{y=0}^{n} S(x,y) \left(\frac{\partial F_x}{\partial x}\right)^2 + w \left(\frac{\partial F_y}{\partial x}\right)^2 dx\, dy \quad \rightarrow \quad \min$$

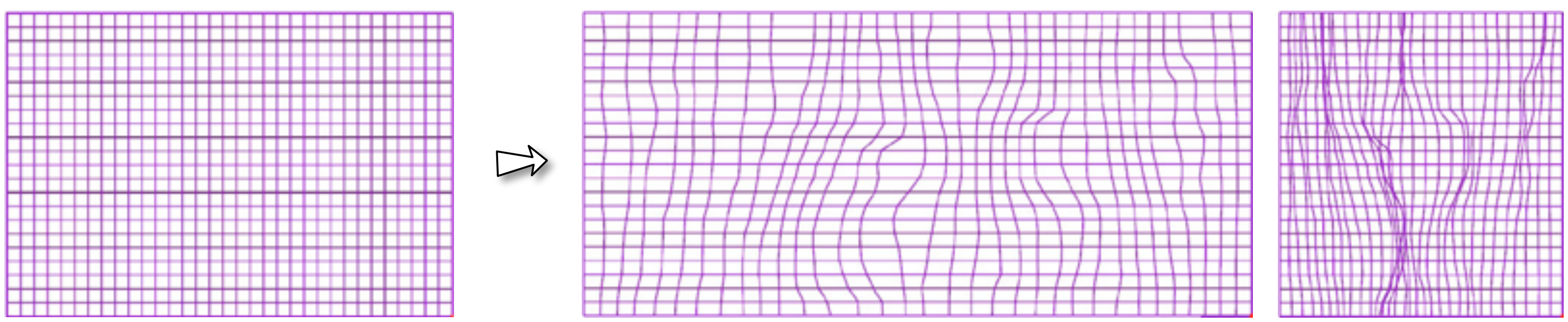
important pixels should keep horizontal distance of 1

columns should keep smooth mapping

[Wolf et al. 07]: one-directional scaling

- One-directional energy
  - for resizing horizontally:

$$E(F) = \int_{x=0}^{m} \int_{y=0}^{n} S(x,y) \left(\frac{\partial F_x}{\partial x} \right)^2 + w \left(\frac{\partial F_y}{\partial x} \right)^2 \, dx \, dy \quad \rightarrow \quad \min$$



[Wolf et al. 07]: one-directional scaling

- Crop or self-intersections may occur
- Space less effectively used due to single direction
- Advantage – linear solve only

original image

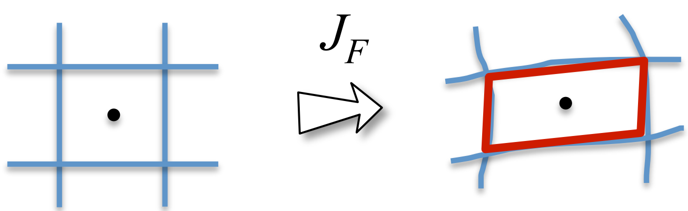
[Wolf et al. 07]

[Wang et al. 08]

[Zhang et al. 09]: conformal energy – "as similar as possible"

- Jacobian should be as close to similarity as possible

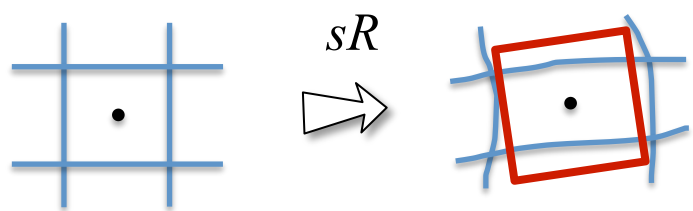
$$\iint S(x,y)\,\|J_F(x,y) - \underbrace{s(x,y)}_{\text{varying scaling factor}}\underbrace{R(x,y)}_{\text{2D rotation}}\|^2 \; dx\,dy \quad \rightarrow \quad \min$$



[Zhang et al. 09]: conformal energy – "as similar as possible"

- Jacobian should be as close to similarity as possible

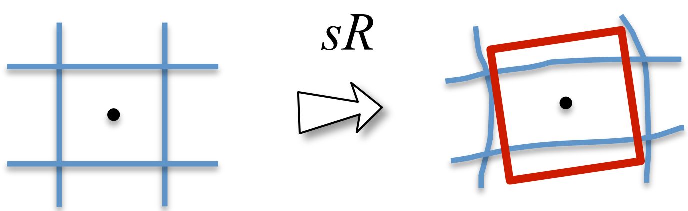
$$\iint S(x,y)\,\|J_F(x,y) - \underbrace{s(x,y)}_{\text{varying scaling factor}}\underbrace{R(x,y)}_{\text{2D rotation}}\|^2 \; dx\,dy \quad \to \quad \min$$



[Zhang et al. 09]: conformal energy – "as similar as possible"

- Jacobian should be as close to similarity as possible

$$\iint S(x,y)\,\left\|J_F(x,y) - \underbrace{s(x,y)}_{\text{varying scaling factor}}\underbrace{R(x,y)}_{\text{2D rotation}}\right\|^2 \,dx\,dy \quad \to \quad \min$$



- Linear formulation, coupling of x and y
  - system matrix size grows × 2 in both dimensions
- Less control over scaling

[Krähenbühl et al. 09]: single optimized scaling factor for entire image

- They note that spatially-varying scaling factor may unnaturally change proportions
- Use single scaling for all pixels and optimize it

$$E(F) = \int_{x=0}^{m} \int_{y=0}^{n} S(x,y)\,\|J_F - s_F I\|^2 + \|J_F - A\|^2 \,dx\,dy$$

important pixels scale uniformly

to obtain the right size, all pixels should scale according to boundary conditions (A is the homogeneous scaling function)

similar to [Gal et al. 06]

[Krähenbühl et al. 09]: single optimized scaling factor for entire image

- They note that spatially-varying scaling factor may unnaturally change proportions
- Use single scaling for all pixels and optimize it

$$E(F) = \int_{x=0}^{m} \int_{y=0}^{n} S(x,y)\, \|J_F - \underline{s_F} I\|^2 + \|J_F - A\|^2 \, dx\, dy$$

initialized as $s_F = 1$; iteratively updated after solving for F (again, global-local optimization)

Comparison: local vs. global scaling

[Wang et al. 08]

[Krähenbühl et al. 09]

Scaling – discussion

- Two extremes:
  - allow uniform scaling to vary everywhere; the scaling factor value is not directly object- or importance-dependent [Wang et al. 08, Zhang et al. 09]
  - same scaling factor for entire image [Krähenbühl et al. 09]

- Is there something in the middle?
  - one scaling factor per object?
  - scaling size depends on importance?

[Karni et al. 09]: variation on the local transformation set

- Remember [Gal et al. 06]? Vary the local transformation according to the importance map

$$E(F) = \int_{x=0}^{m} \int_{y=0}^{n} S(x,y)\,\| J_F - \underbrace{sI}_{\begin{pmatrix} s & \\ & s \end{pmatrix}} \|^2 + (1 - S(x,y))\,\| J_F - \underbrace{A}_{\begin{pmatrix} m^*/m & \\ & n^*/n \end{pmatrix}} \|^2 \, dx\, dy$$

- In [Gal et al. 06], importance $S(x,y)$ was binary

[Karni et al. 09]: variation on the local transformation set

- [Karni et al. 09] use arbitrary importance map and interpolate between sI and A accordingly

$$E(F) = \int_{x=0}^{m} \int_{y=0}^{n} \left\| J_F - \mathrm{interp}(sI, A, \underline{S(x,y)}) \right\|^2 \, dx \, dy$$

interp. parameter
larger → closer to uniform scaling

- More granular than a global weighted least-squares

Results [Karni et al. 09]

original

[Wang et al. 08]

[Karni et al. 09]

Results [Karni et al. 09]

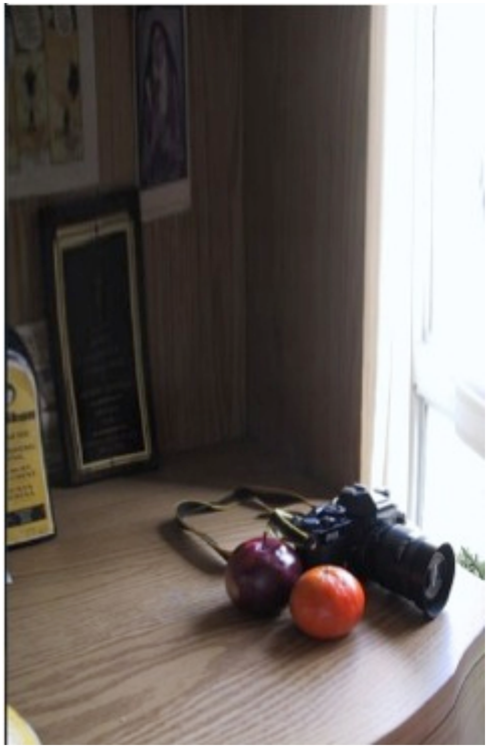
original

[Wang et al. 08]

[Karni et al. 09]

Results [Karni et al. 09]

original



[Wang et al. 08]

[Karni et al. 09]

Results [Karni et al. 09]

original

[Wang et al. 08]

[Karni et al. 09]

Some comparisons

original image [Rubinstein et al. 08] [Wolf et al. 07] [Wang et al. 08]

Some comparisons

original

SC

indirect SC

[Wang et al. 08]

Some comparisons

original

SC

indirect SC

[Wang et al. 08]

Discussion

- Scaling allows more flexibility in the warp
- Too much scaling freedom may lead to unintuitive changes in proportions
- The line problem
  - Edges and especially straight lines bend
  - Lines are prominent in images of man-made objects
- Automatic importance maps do not always work
  - No choice but to add user control

Line constraints [Krähenbühl et al. 09]

- Automatic edge detection
  - Sobol filters
  - Augment the importance map by edge importance $S_e(x,y)$
- Additional energy terms:

prevent bending $$\iint S_e(x,y)\left(\left(\frac{\partial F_x}{\partial y}\right)^2+\left(\frac{\partial F_y}{\partial x}\right)^2\right)dx\,dy$$

prevent smearing $$\iint S_e(x,y)\left(\left(\frac{\partial F_x}{\partial x}-1\right)^2+\left(\frac{\partial F_y}{\partial y}-1\right)^2\right)dx\,dy$$

Line constraints [Krähenbühl et al. 09]

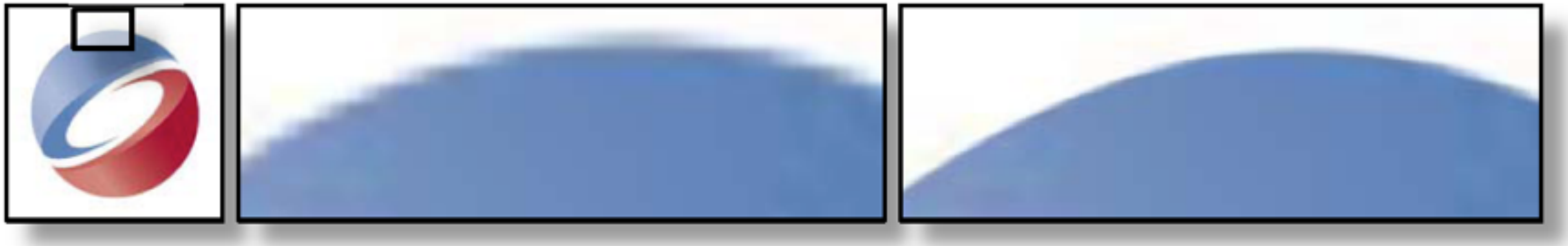
- Effect on the line bending

Line constraints [Krähenbühl et al. 09]

See video at
http://ahornung.net/files/pub/Hornung_SIGAsia09.mov

Line constraints [Krähenbühl et al. 09]

- Effect of the line smearing term:



$$\iint S_e(x,y)\left(\left(\frac{\partial F_x}{\partial x}-1\right)^2+\left(\frac{\partial F_y}{\partial y}-1\right)^2\right)dx\,dy$$

Line constraints [Krähenbühl et al. 09]

- Manual marking of prominent lines
- Placing global line constraints
  - original line:
    $$\cos(\alpha)\,x + \sin(\alpha)\,y + b = 0$$
  - energy term:

$$E(F) = \int_{x=0}^{m}\int_{y=0}^{n} \underline{c(x,y)}\,\left(\cos(\alpha')F_x(x,y) + \sin(\alpha')F_y(x,y) + b'\right)^2 dx\,dy$$

coverage of pixel (x,y) by
the line in the original image

Line constraints [Krähenbühl et al. 09]

- Manual marking of prominent lines
- Placing global line constraints
  - original line:
    $$\cos(\alpha)\,x + \sin(\alpha)\,y + b = 0$$
  - energy term:

$$E(F) = \int_{x=0}^{m}\int_{y=0}^{n} c(x,y)\left(\cos(\underline{\alpha'})F_x(x,y) + \sin(\underline{\alpha'})F_y(x,y) + \underline{b'}\right)^2 dx\,dy$$

At first set $\alpha' = \alpha,\ b' = b$, but then iteratively optimize

Line constraints [Krähenbühl et al. 09]

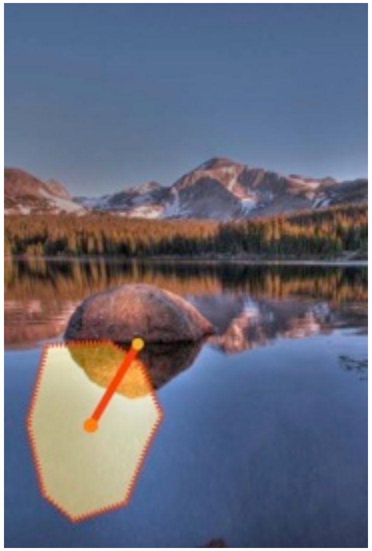
- Effect of global line constraints:

Line constraints [Krähenbühl et al. 09]

See video at
http://ahornung.net/files/pub/Hornung_SIGAsia09.mov

Position constraints [Krähenbühl et al. 09]

- User marks polygon around an object
- Constrain the position of center of mass using baricentric coordinates



Position constraints [Krähenbühl et al. 09]

See video at
http://ahornung.net/files/pub/Hornung_SIGAsia09.mov

[Panozzo et al. 12] Axis-aligned warping

- Minimize **any** warping energy in the **subspace** of **axis-aligned warps**
- Convert into optimization of column widths w_i and row heights h_i!

$$\min_{w,h} E(F)$$

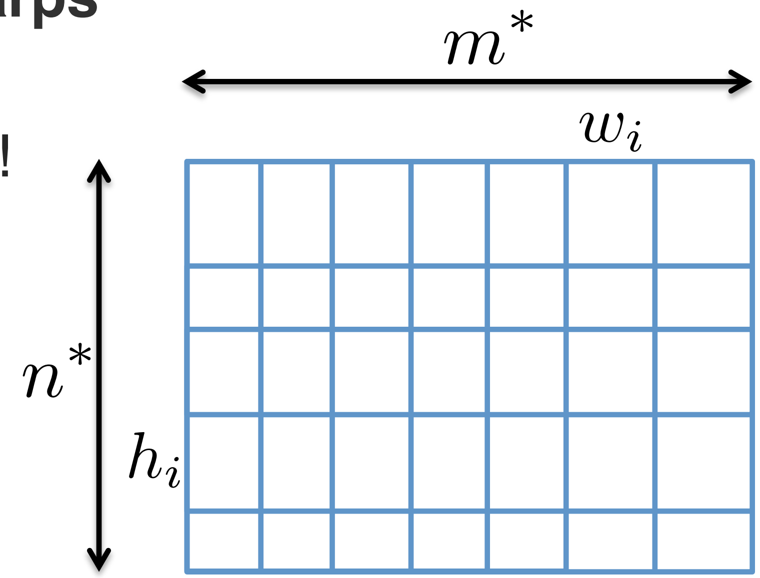
- Disallow self-intersections:

$$w_i, h_i > 0$$

- Boundary conditions:

$$\sum w_i = m^*, \ \sum h_i = n^*$$

- For quadratic $E(F)$ we end up with a small, dense QP



[Panozzo et al. 12] Axis-aligned warping

- Reduce problem dimensionality from $O(mn)$ to $O(m+n)$
- Discretization can be very coarse
  - we use 25×25
- Very fast, realtime saliency changing!
  - 3ms per solve
- Demo!

See video and demo software at
http://igl.ethz.ch/projects/retargeting/aa-retargeting/

Summary – image warping

	Importance map	Energy type	Solver
Gal et al. 06	binary	Linear LS, coarse grid	Sparse direct
Wolf et al. 07	L^2 gradients + face detection	Linear LS, pixel grid	Sparse direct
Wang et al. 08	L^2 gradients + Itty's saliency	Nonlinear LS, coarse grid	Sparse direct, local-global iterations
Krähenbühl et al. 09	Guo's saliency + line detection + user marking	Nonlinear LS, pixel grid	GPU multigrid, local-global iterations
Karni et al. 09	L^2 gradients	Nonlinear LS, coarse grid	Sparse direct, local-global iterations
Panozzo et al. 12	any, manual	Any, efficient for linear LS	QP solver (CVXGEN)

Summary – image warping

	Scaling	Line constraints	User manipulation
Gal et al. 06	Single global factor, fixed	–	Mark importance, position constraints
Wolf et al. 07	–	Grid bending, linear	–
Wang et al. 08	Local factors, optimized	Grid bending, nonlinear	Mark importance, position constraints
Krähenbühl et al. 09	Single global factor, optimized	Edge bending, edge blurring, nonlinear	Mark importance, mark global lines, position constraints
Karni et al. 09	Local factors, optimized	–	–
Panozzo et al. 12	Local factors, or forced to = 1	–	Mark importance (realtime response)

Evaluation framework [Rubinstein et al. 10]

http://people.csail.mit.edu/mrub/retargetme/

- Best energy to minimize should be based on user preference!
- Very hard to quantify, hence saliency and geometric energies
- Distance between images?
  - Automatic measures like bi-directional distance do not correlate with user preferences
- **Benchmark** and **user study framework** of retargeting results
  - A diverse set of images
  - Gathered results of many methods
  - Methodology for online user study (paired comparisons)
  - Now people compare!

Warping video

- Space-time cube – warping function has 3 variables

$$F(x, y, t)$$

- Boundary constraints per frame
  - fit target size – same as with image retargeting

- Time stays the same

$$F_t(x, y, t) = t$$

Temporal coherence

- Temporal coherence is key
- Earlier works [Wolf et al. 07], [Rubinstein et al. 08]:
  - Temporally adjacent pixels should change similarly

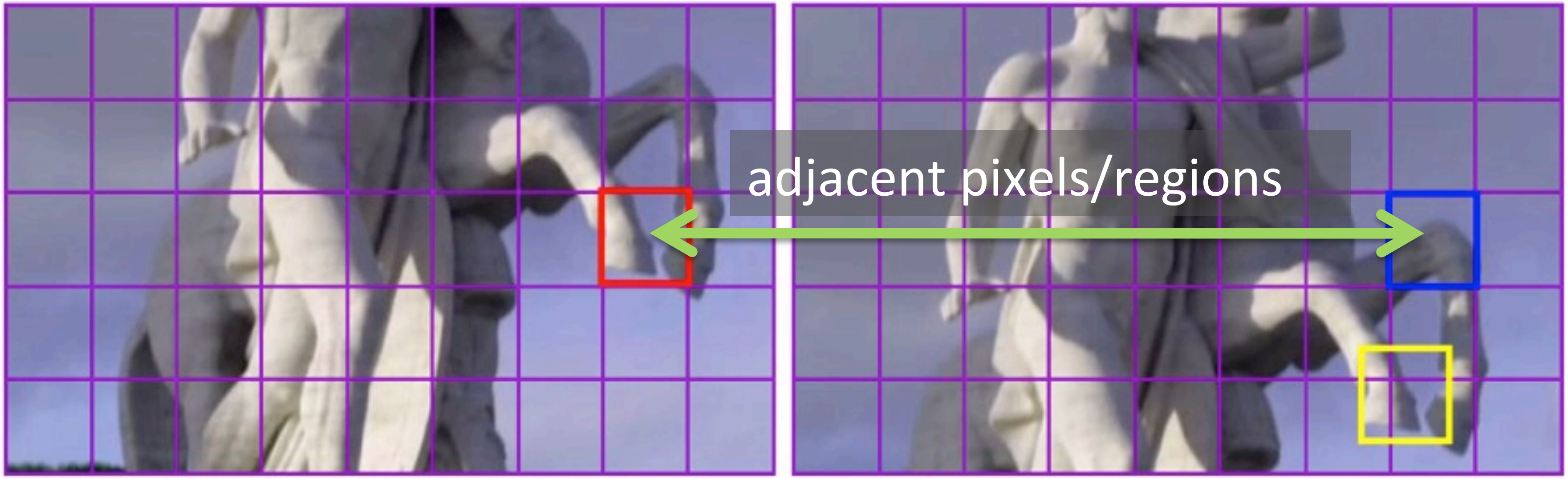
$$\iint \left\| \frac{\partial F}{\partial t} \right\|^2 \rightarrow \min$$

- Problem: this energy is motion-oblivious!

Temporal coherence

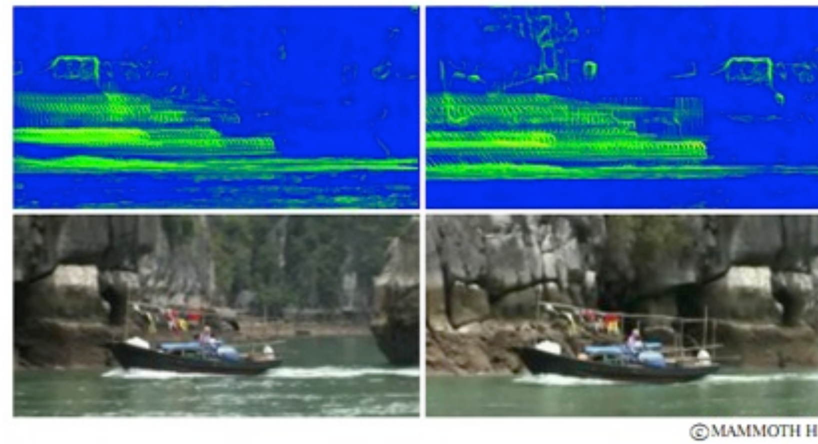
- Temporal coherence is key
- Earlier works [Wolf et al. 07], [Rubinstein et al. 08]:
  - Temporally adjacent pixels should change similarly

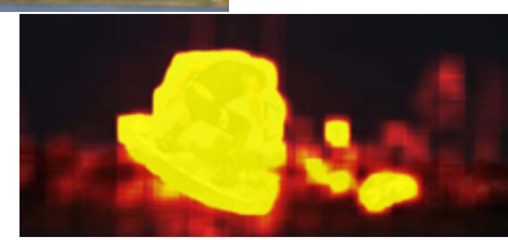
- Motion-oblivious!



Taking motion into account

- Use motion importance maps [Krähenbühl et al. 09], [Wang et al. 09-11]:
  - Average per-frame importance maps over several frames such that motion is taken into account
  - Add optical flow estimation, such that moving objects get higher importance





Taking motion into account

- [Krähenbühl et al. 09]: compute the warp per-frame
- Temporal coherence constraint between consecutive frames only (use previous frame as reference for current frame)

$$\iint \|F(x,y,t) - \underbrace{F(x,y,t-1)}_{\text{previously computed}}\|^2 \quad \rightarrow \quad \min$$

- Detect scene cuts
  – The above term is not used over scene cuts

Results [Krähenbühl et al. 09]

See video at
http://ahornung.net/files/pub/Hornung_SIGAsia09.mov

Taking motion into account [Wang et al. 09]

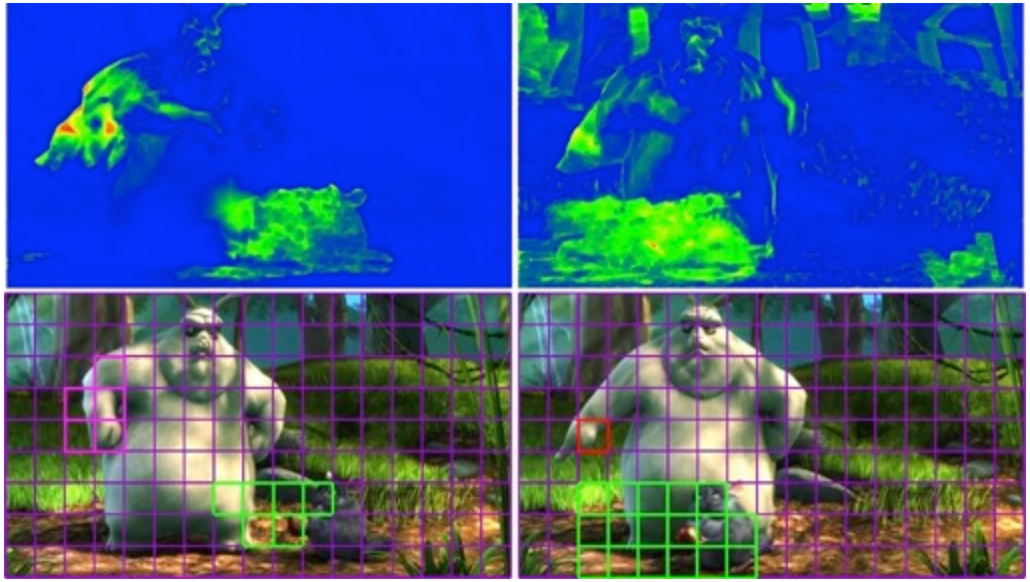
- Solve for the entire space-time cube simultaneously (up to scene cuts)
- Two types of motion of content:
  - Due to camera movement
  - Due to object movement

Taking motion into account [Wang et al. 09]

- [Wang et al. 09] pre-register all frames in one coordinate system to eliminate camera motion
  - detect SIFT features
  - estimate camera matrix

Taking motion into account [Wang et al. 09]

- Segment moving objects
- Each moving object is resized consistently (its scaling factor should be smooth)



Taking motion into account [Wang et al. 09]

See video at
http://igl.ethz.ch/projects/retargeting/motion-aware-video-retargeting/VideoResize09.mp4

Results and comparisons [Wang et al. 09]

See video at
http://igl.ethz.ch/projects/retargeting/motion-aware-video-retargeting/VideoResize09.mp4

Optimized crop-and-warp [Wang et al. 10]

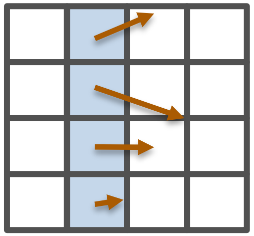
- [Wang et al. 09] cannot handle parallax
  - registration to global coordinate frame fails
- All methods degrade to homogeneous scaling when scene too crowded

- **Crop-and-warp**: introduce automated cropping to enable the warp to better utilize the available space

See video at
http://igl.ethz.ch/projects/retargeting/crop-and-warp/CropAndWarp_SIG2010.mp4

Optimized crop-and-warp [Wang et al. 10]

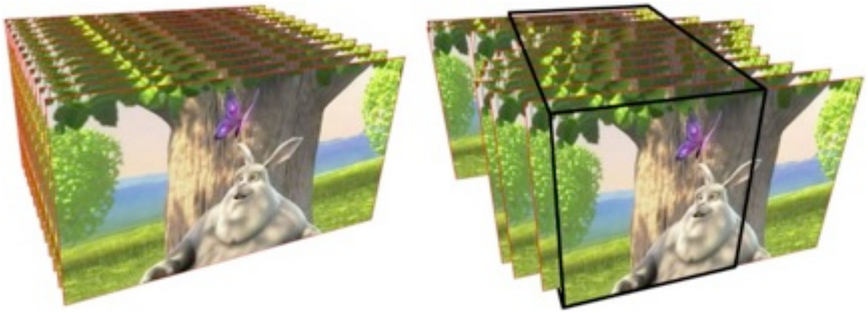
- Determine critical region in each frame
  - using optical flow [Werlberger et al. 09]



temporal persistence
(left side is about to disappear)

actively moving objects

- Warp with inequality constraints, such that critical regions stay inside the target cube



Optimized crop-and-warp [Wang et al. 10]

- Results and comparisons

See videos at
http://igl.ethz.ch/projects/retargeting/crop-and-warp/CropAndWarp_SIG2010.mp4
http://igl.ethz.ch/projects/retargeting/SVR/SVR_supp.wmv

Summary – video warping

	Temporal importance map	Energy type	Temporal coherence constraints
Wolf et al. 07	Motion saliency	Per-frame, linear	Temporally-adjacent pixels smoothness
Krähenbühl et al. 09	Motion saliency, image importance averaging	Per-frame, nonlinear	Temporally-adjacent pixels smoothness
Wang et al. 09	Motion saliency, image importance averaging	Entire video cube, nonlinear	Camera alignment, consistent resizing of moving objects
Wang et al. 10	Motion saliency, minimal temporal persistence	Entire video cube, nonlinear, inequalities	Consistent resizing of moving objects, smoothly varying crop

Discussion

- **Performance/quality tradeoff:**
  - Per-frame video retargeting can be done in real time, but has difficulty with temporal coherence
  - Temporal coherence requires processing of longer sequences – offline process
  - Some recent papers showed ***scalable*** performance, still not realtime as entire video cube needs to be processed at once. See e.g. [Wang et al. 11]

- Motion saliency relies on reliable optical flow estimation
  - Difficult when motion is fast or no trackable features (cartoons)

Conclusions

- Warp-based methods take a continuous view on the image retargeting problem.
- Generic variational approach: define an energy functional depending on importance map and find a warping function that optimizes it.

Conclusions

- **Advantages:**
  - Flexibility w.r.t. energy design
  - Tend to smoothly distort image content
  - Efficiency can be controlled by discretization resolution
- **Disadvantages:**
  - (sometimes) costly optimization
  - Local descriptors – hard to maintain global structures like symmetry, straight lines, proportions, perspective…
  - Weighting of different energy terms is content-dependent
  - In some cases cropping/carving makes more sense than squeezing
- **Bottom line ☺**
  - Multi-operator approaches are probably unavoidable!

Acknowledgements

- The authors of all the papers we discussed in the course!
- YuShuen Wang, Alexander Hornung, Manuel Lang, Daniele Panozzo for additional video material
- Daniel Graf for the AA-retargeting mobile demo
- ACM SIGGRAPH Asia
- SNF award 200021-137879
- ERC Starting Grant 306877 “iModel”

Thank You!