Autostitch

- *Recognizing Panoramas*, M. Brown and D. Lowe, *Proc. ICCV*, 2003
- Goal: Search a collection of photos for sets that can be stitched together completely automatically
- http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Autostitch: Example

Input:

Output:

[Brown 2003]

Method

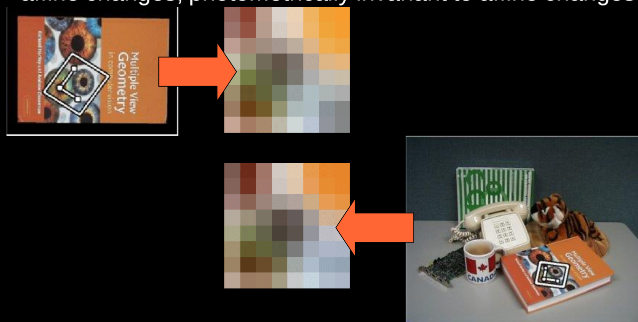
- Detect point features
- Match features between images
- Determine overlapping pairs of images
- Solve for homographies between all images
- Blend

Detect and Match Feature Points

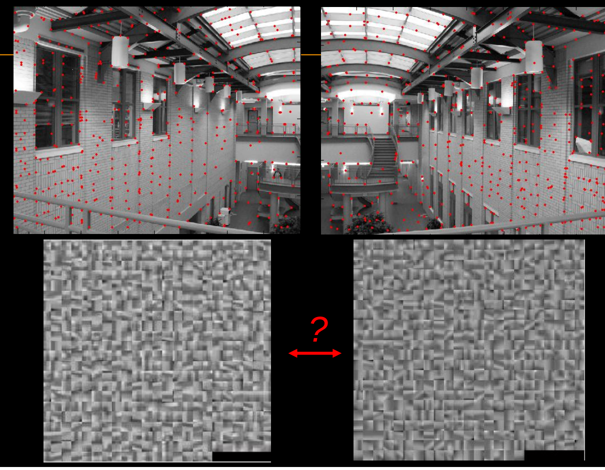
- In each image detect distinctive “interest points” (at multiple scales)
- Each point described by a feature vector (aka feature **descriptor**)
- For each feature point in each image, find most similar feature points in the other images (using hashing or k-d tree to find approximate nearest neighbors)

Invariant Features

- Geometrically invariant to similarity transforms and some affine changes; photometrically invariant to affine changes

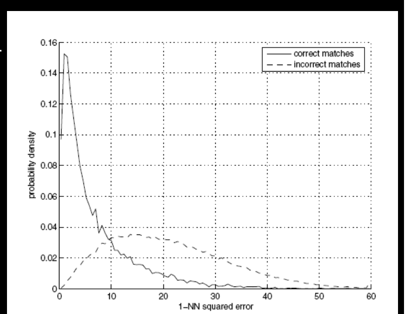


How to Find Matching Points?



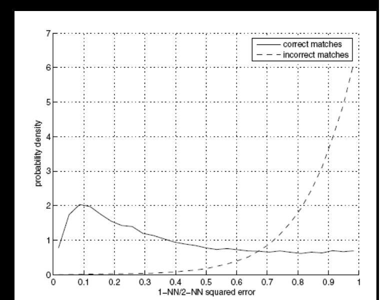
Feature-Space Outlier Rejection

- Let's not match *all* pairs of features, but only those that have "similar enough" matches
- How?
  - SSD(patch1, patch2) < T
  - How to set threshold?

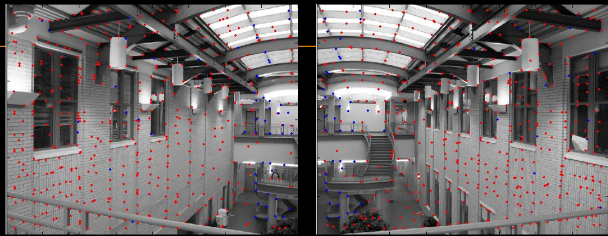


Feature-Space Outlier Rejection

- A better way [Lowe, 1999]:
  - 1-NN: SSD of the closest match
  - 2-NN: SSD of the second-closest match
  - Look at how much better 1-NN is than 2-NN, e.g., 1-NN/2-NN
  - That is, is our best match a lot better than the rest?

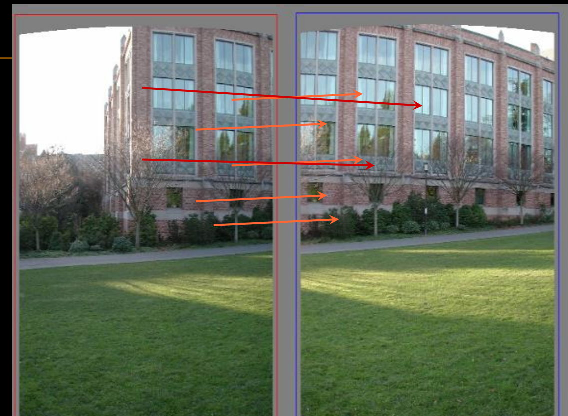


Feature-Space Outliner Rejection



- Can we now compute **H** from the blue points?
  - No! Still too many outliers
  - What can we do?

Matching Features

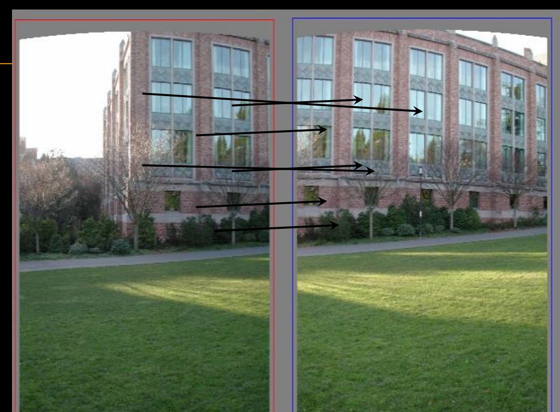


What do we do about the "bad" matches?

Image Matching

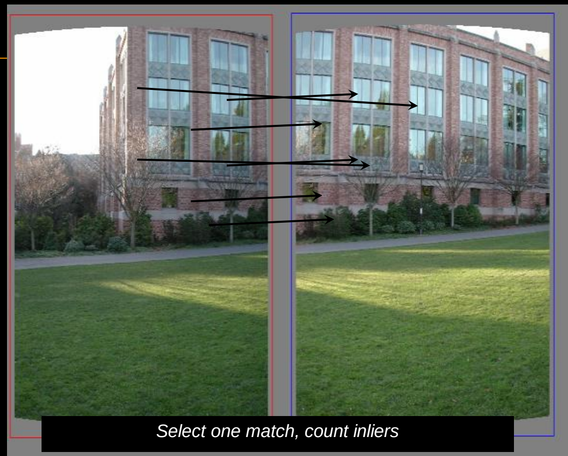
- For each image, find *m*=6 other images with greatest number of feature matches to current image
- For each pair of neighboring images, use RANSAC algorithm to find true matches (inliers), eliminate non-matching points (outliers), and compute homography

<u>RA</u>ndom <u>SA</u>mple <u>C</u>onsensus

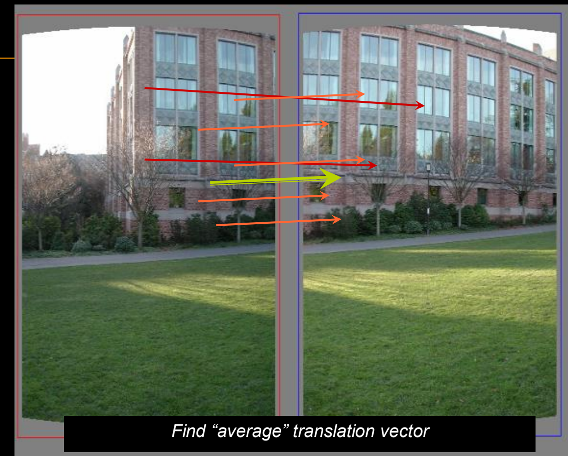


Select one match, count inliers

RAndom SAmple Consensus



Compute Least Squares Fit

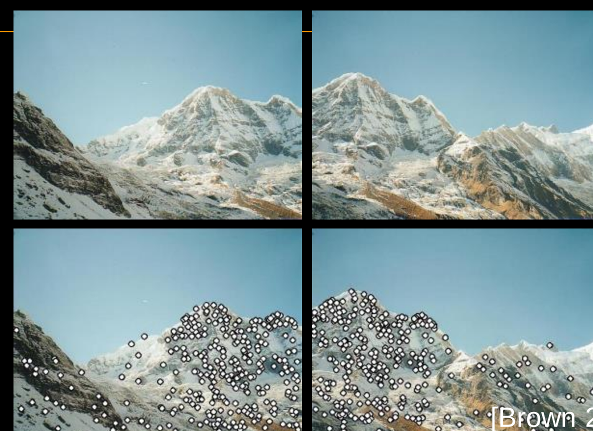


RANSAC Algorithm for Estimating Homography

- Loop many times:

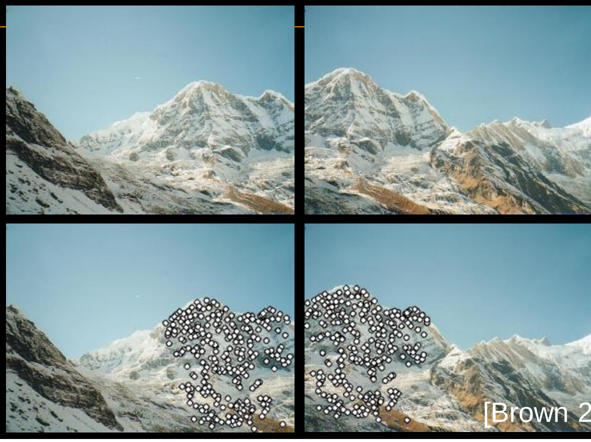
1. Select 4 feature pairs (at random)
2. Compute homography **H** (exact)
3. Compute *inliers*, *i.e.*, $SSD(p_i', \mathbf{H}\, p_i) < \varepsilon$
4. Keep largest set of inliers
5. Re-compute least-squares **H** estimate using *all* inliers

Use RANSAC to Compute Homography



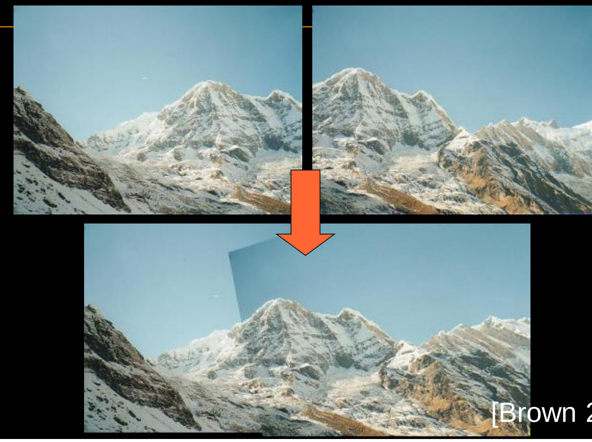
[Brown 2003]

Reject Outliers using RANSAC



[Brown 2003]

Use RANSAC to Compute Homography

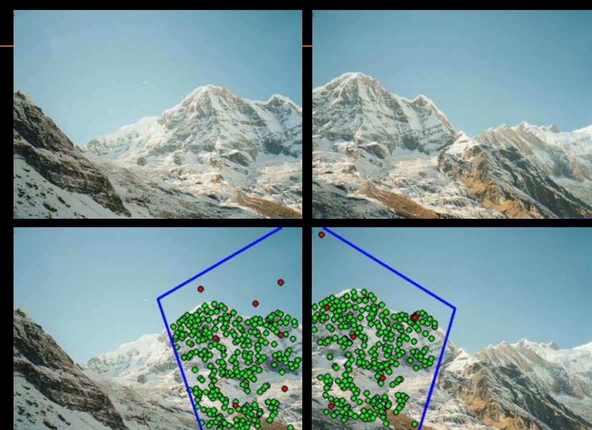


[Brown 2003]

Robustness

- RANSAC is just one of many “robust” methods that deals with issues related to
  - inadequate models
  - missing data (i.e., which data is noise, “outliers,” and which is not)
  - error measures that heavily penalize large errors (which lead to poor fit between data and model)
- **M-estimators** are another robust method

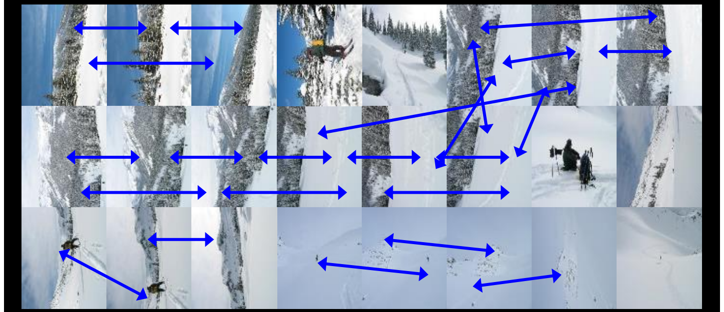
Probabilistic Model for Verification



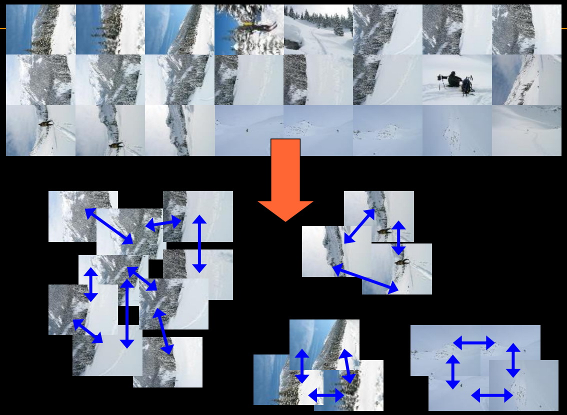
Probabilistic Model for Verification of Image Match

- Compare probability that this set of RANSAC inliers/outliers was generated by a correct/false image match
- Let n_f = # features in overlap area, n_i = # inliers, m a binary r.v. where m=1 means correct image match, and $f^{(i)} \in \{0,1\}$ mean the i^{th} feature is a match or not
- Image match if $p(m=1 \mid f^{(1:n_f)}) > p_{\min}$
- Solve using Bayes' rule, assuming binomial distribution on $f^{(i)}$, likelihood ratio test formulation, and parameters $\Rightarrow$
- Image match if $n_i > 5.9 + 0.22 n_f$

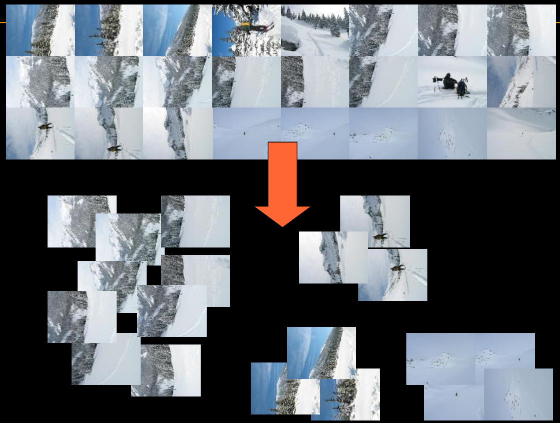
Results of Image Matching



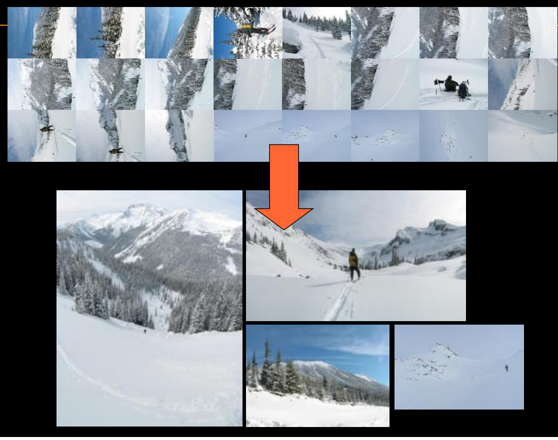
Finding the Panoramas (= Cliques)



Finding the Panoramas



Finding the Panoramas



Improving the Homographies: Bundle Adjustment

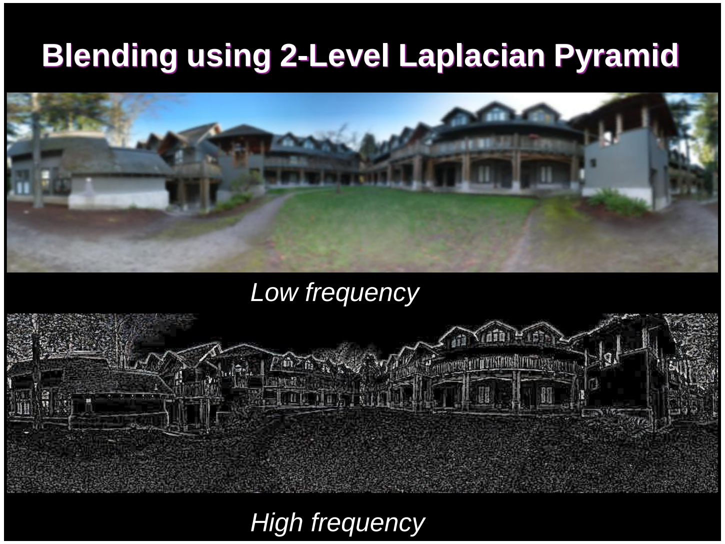
- Jointly solve for *all* homographies together to improve robustness
- Find the parameters of all homographies that minimize the sum of squared projection errors
- Solve optimization problem by adding images best to worst

Improving the Homographies: Bundle Adjustment

- New images initialized with rotation, focal length of best matching image

Bundle Adjustment

- New images initialized with rotation, focal length of best matching image


Blending using 2-Level Laplacian Pyramid
Low frequency
High frequency


2-Band Blending

Matching Mistakes: False Positives

Matching Mistakes: False Positives

Matching Mistake: False Negatives

- Moving objects: large areas of disagreement

Matching Mistakes

- Accidental alignment
  - repeated / similar regions
- Failed alignments
  - moving objects / parallax
  - low overlap
  - “feature-less” regions (more variety?)
- 5-10% failures; no 100% reliable algorithm

Autostitch

- Huge number of features to match
  - Uses efficient approx. nearest-neighbor search
  - $O(n \log n)$ where n = number of features
  - Uses priors to accelerate RANSAC
- Handles full space of rotations
- Estimates camera intrinsics for each photo
  - Bundle adjustment

Discovering Panoramas in Web Videos

- F. Liu, Y. Hu and M. Gleicher , *Proc. Multimedia*, 2005
- Videos often contain appropriate images to make panoramas

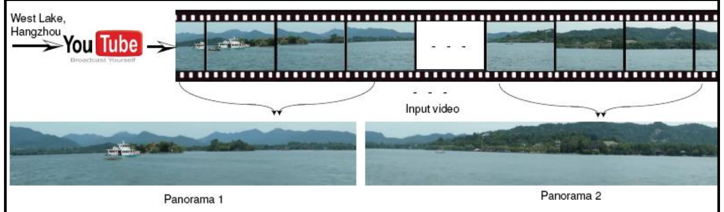


Figure 1: Working example. A user makes a query of "West Lake, Hangzhou" to *YouTube*, and feeds retrieved video clips into our system. Our system selects useful frames from the given videos and synthesizes panoramas using the selected source frames.

Automatic Panorama Creation

Recognizing Panoramas in Sets of Stills

- Not all images work
- Matching is Hard
  - Images unordered
  - Large differences in images
  - Different Orientations
- High Quality Images
- Relatively small image sets
- Assume sufficient coverage

Discovering Panoramas in Web Video

- Not all videos work
- Matching is not so hard
  - Images are ordered
  - Continuity limits differences
  - Orientations consistent
- Variable Image Quality
- Potentially large image sets
- Small motions uninteresting
- Dynamic Objects

When does a Segment of Video lead to a Good Panorama?

- Good homographies between frames
- Individual images have good quality
- Result has a wide field of view

Competing interests:

- More frames give wider field of view
- More frames give more accumulated error

Homography Quality

- Points should match (robust best fit)
- Measure residual distances
- Penalize large residuals

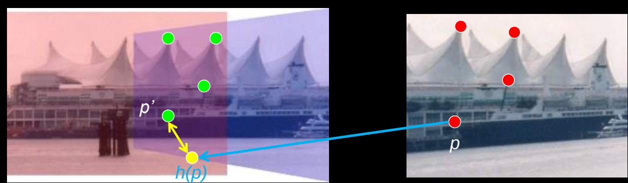
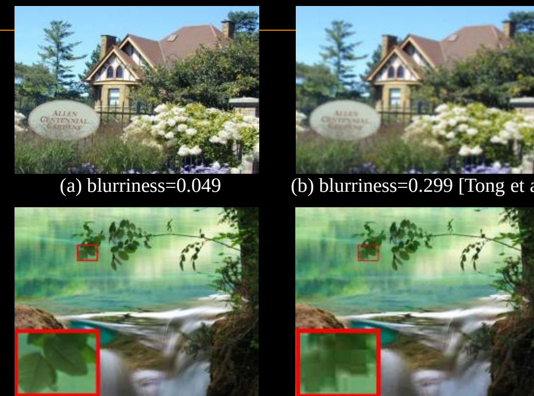


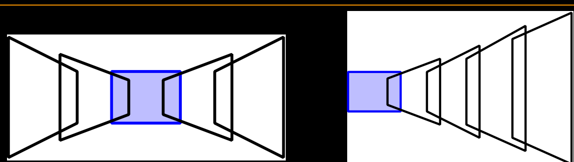
Image Quality Cost



(a) blurriness=0.049 (b) blurriness=0.299 [Tong et al. 04]

(c) blockness=0.204 (d) blockness=0.497 [Wang et al. 02]

Field of View (Coverage)



- Approximate field of view as scene coverage
- A panorama with a wide fov is preferred over 2 smaller panoramas
- Pick the base frame that minimizes area
  - And therefore has minimum distortion

An Optimization Problem

Given a video V

Find (non-overlapping) segments S_i that

- Have maximal field of view / coverage
- Have minimal penalties
  - Homography error
  - Image quality penalty

Reject segments that have

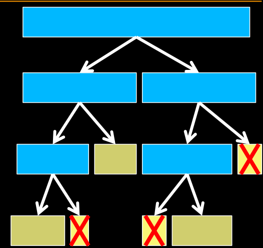
- Too little coverage
- Too much penalty

Greedy, Approximate Algorithm

Divide video as one segment

Keep dividing segments that have too much penalty until all are OK

Discard segments with too little coverage



Example Results

(a) West lake 1. Its source images are frame 0 to 86 from a 26-second, 10fps web video.

(b) West lake 2. Its source images are frame 89 to 251 from the same video as (a).

(c) Lake Louise. Its source images are frame 0 to 98 from a 21-second, 10fps web video.

Online Examples with Videos

- http://web.cecs.pdx.edu/~fliu/project/discover-pano.htm

More Examples

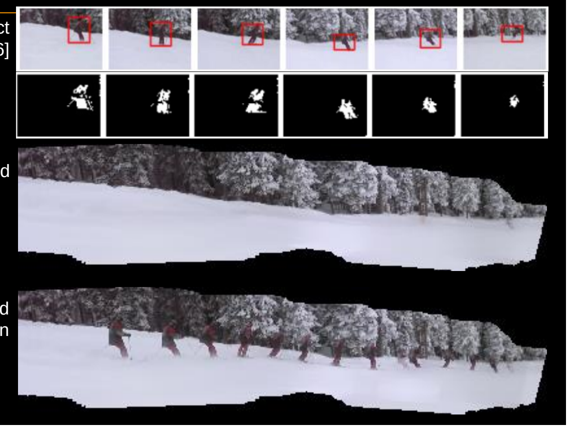
(d) Arches National Park. Its source images are frame 55 to 173 from a 15-second, 15fps web video.

(e) Arches National Park. Its source images are frame 142 to 265 from a 31-second, 15fps web video.

(f) Arches National Park. Its source images are frame 1168 to 1296 from a 83-second, 15fps web video.

More Examples

What about Moving Objects?



Activity Synopsis Examples

(a) Horse riding. Its source images are frame 297 to 495 from a 58-second, 15fps web video

(b) Biking. The source images of the two panoramas are frame 2729 to 2856, and frame 4298 to 4625 from a 194-second 30fps web video.

Evaluation

- Tried 6 queries (60 total videos)
- Created panoramas from most (87%)
- Compared auto-discovery with human expert
  - Expert only looks for camera motions
  - Algorithm looks for panorama sources
- Never found panoramas that expert did not
- Found 87% of those identified by expert

Panorama Cameras

Point Grey Ladybug3

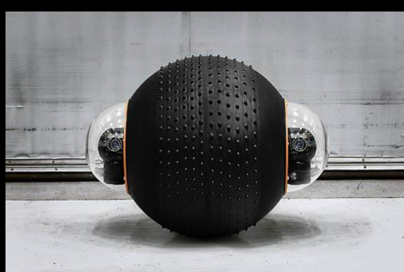
- 6 video cameras, stitched into 5400 x 2700 “spherical image” @ 15 fps

Throwable Panoramic Ball Camera

- http://jonaspfeil.de/ballcamera

GroundBot

- http://www.rotundus.se/



Panoramic Video Textures

Input Video:

[Agarwala et al., 2005]

Panoramic Video Textures

Gigapixel Panoramas

- Microsoft HD View
- Google Earth
- Gigapan
- Gigapxl
- 360 Cities

Gigapixel Panoramas

- Shanghai Skyline (10/2010): 12,000 images, 272 gigapixels (887K × 307K), 1.09 TB file size
- http://gigapan.org
- http://research.microsoft.com/en-us/um/redmond/groups/ivm/HDView/
- Global Connection Project
  - The Global Connection Project develops software tools and technologies to increase the power of images to connect, inform, and inspire people to become engaged and responsible global citizens

Unwrap Mosaics: A new representation for video editing

A. Rav-Acha, P. Kohli, C. Rother and A. Fitzgibbon, *Proc. SIGGRAPH*, 2008

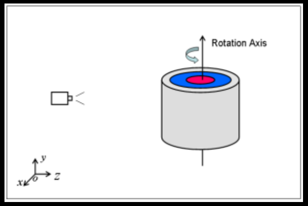
Goal: Given a video, recover for each moving, non-rigid object (1) its **texture** map modeling the object's appearance, (2) a 2D-to-2D mapping describing the texture map's **projection** to the images, and (3) a sequence of binary masks modeling **occlusion**

In other words, build an "**Object Panorama**"

Slides by P. Kohli

Viewing A Rotating Object

- Consider the class of single-axis rotations



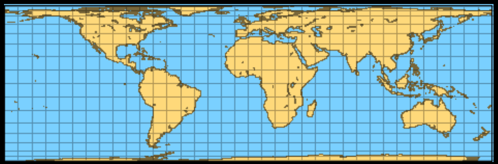
- Image sequence is a spatiotemporal volume showing all parts of surface

Removing Occlusions

Images of Mayan vases by Justin Kerr

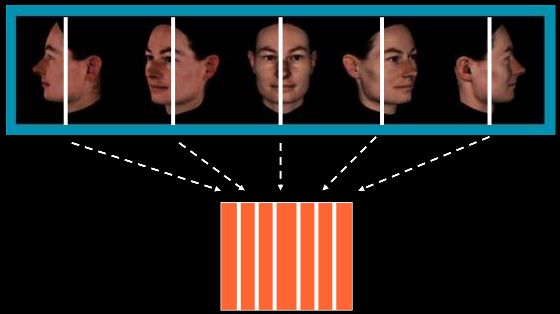
A **cyclograph** (aka **rollout photograph** or **peripheral photograph**) is a technique developed in photography; it is generated when an object rotates in front of a 1D camera or the camera moves around an object

Cylindrical Projection



Constructing a Cyclograph

Concatenate center column from each image in sequence



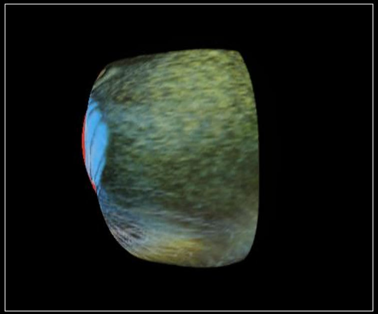
Cyclograph

http://grail.cs.washington.edu/projects/stereo/

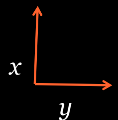
Cyclograph

- A cyclograph is a concise, multi-perspective representation of a video sequence looking *inward* as it moves around a rigid object
- Encodes approximately fronto-parallel views of the object
  - Little foreshortening distortion of surfaces
  - Limitation: Profile shape features are lost
  - Limitation: Occluded parts not represented

A Deforming 3D Object



$\mathbf{I}(\mathbf{x}, t)$

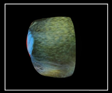


Note: $\boldsymbol{x} = (x, y)$

A Deforming 3D Object

$\mathbf{C}(\mathbf{u})$

Texture map = panoramic image

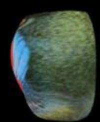


$\mathbf{I}(\mathbf{x}, t)$

Note: $\boldsymbol{u} = (u, v)$

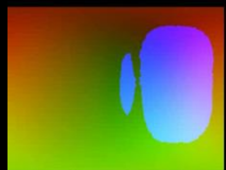
A Deforming 3D Object

$\mathbf{I}(\mathbf{x}, t)$

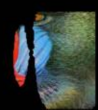


$w(u,t)$

Defines how visible pixels are warped to give *I*



Warping



$b(u,t)$

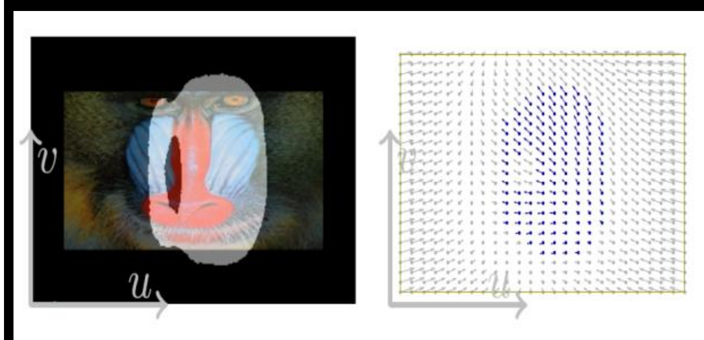
Visibility Mask

Video Generation Model

Texture map

$\mathbf{C}(\mathbf{u})$

Video Generation Model



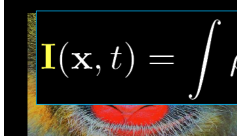
Visibility map defines which pixels are used to produce image *I*

2D-to-2D warping function ***w*** that describes how pixels in texture map C are projected into image *I*

Video Generation Model

Texture map

$\mathbf{C}(\mathbf{u})$



visibility mask $b(\mathbf{u}, t)$

warping $\mathbf{w}(\mathbf{u}, t)$

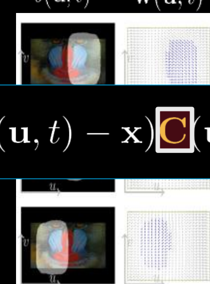
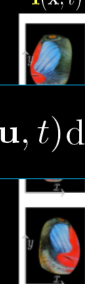


Image $\mathbf{I}(\mathbf{x}, t)$



$$\mathbf{I}(\mathbf{x}, t) = \int \rho(\boxed{\mathbf{w}}(\mathbf{u}, t) - \mathbf{x})\boxed{\mathbf{C}}(\mathbf{u})\boxed{b}(\mathbf{u}, t)\,d\mathbf{u}$$

Part 2: Model Recovery

$$\mathbf{I}(\mathbf{x}, t) = \int \rho(\boxed{\mathbf{w}}(\mathbf{u}, t) - \mathbf{x})\boxed{\mathbf{C}}(\mathbf{u})\boxed{b}(\mathbf{u}, t)\,d\mathbf{u}$$

- Given *I*, how can we recover ***w***, ***C***, ***b***?
- Formulate as an energy minimization problem
- 3 steps: Track, Embed & Stitch

Estimating w, b *Estimating C*

Summary of the Algorithm

Application: Video Editing

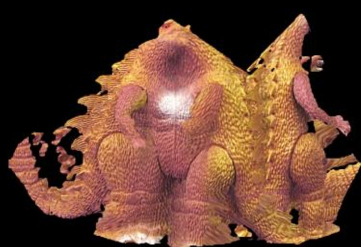
Again…

Giraffe Edits

More Results

Failures

- Non-smooth 3D
- Non-disc topology



Failures

- Non-smooth 3D
- Non-disc topology
- Lack of texture