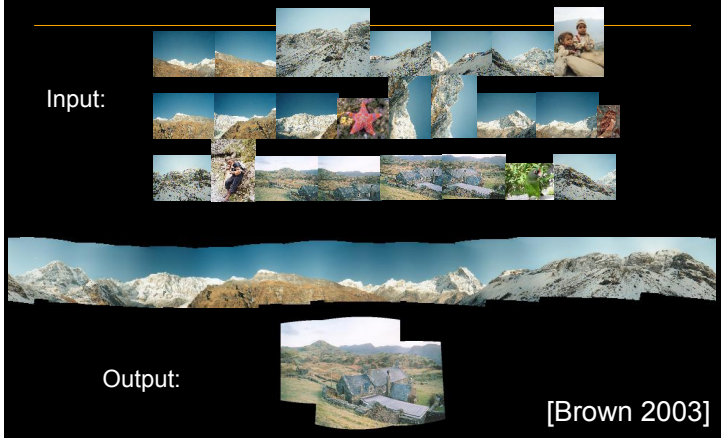
## Autostitch

- *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




[Brown 2003]

## Panorama Algorithm

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

## Solving for Homography: *n* Pairs of Corresponding Points

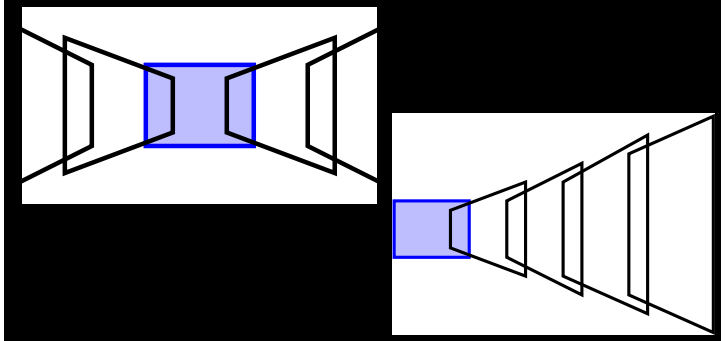
$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1'x_1 & -x_1'y_1 & -x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y_1'x_1 & -y_1'y_1 & -y_1' \\ & & & & \vdots & & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_n'x_n & -x_n'y_n & -x_n' \\ 0 & 0 & 0 & x_n & y_n & 1 & -y_n'x_n & -y_n'y_n & -y_n' \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

**A** ***2n × 9*** **h** ***9*** **0** ***2n***

**Linear least squares**

- Since **h** is only defined up to scale, solve for unit vector $\hat{\mathbf{h}}$
- Minimize $\| A\,h \|^2$
- Can be done in Matlab using "\" command (see "help lmdivide")

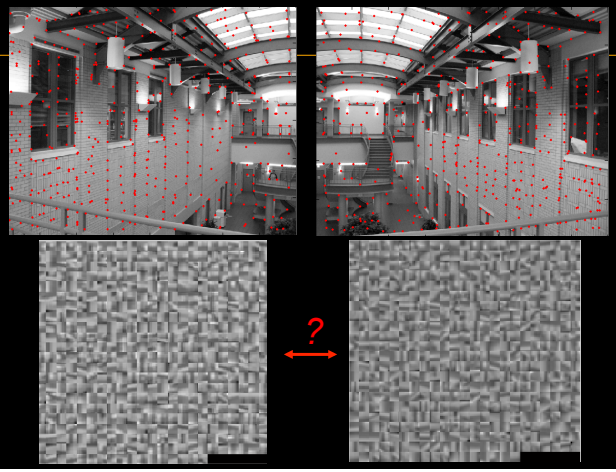
## Panorama "Shape" Depends on Output Image Plane



## Detect and Match Feature Points

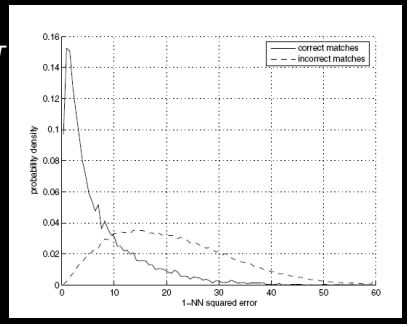
- In each image detect distinctive "feature points" (at multiple scales)
- Each point described by a feature vector (aka feature **descriptor**)
- For each feature point in each image, find most similar feature points in the other images (using hashing or k-d tree to find approximate nearest neighbors)

## How to Find Matching Points?



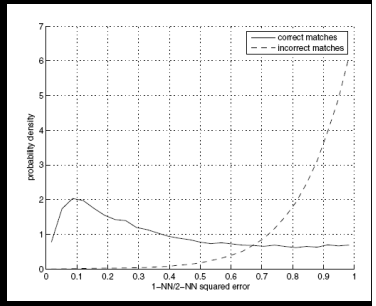

## Feature-Space Outlier Rejection

- Don't match *all* pairs of features, but only those that have similar feature vectors
- How?
  - SSD(patch1, patch2) < *T*
  - How to set threshold?

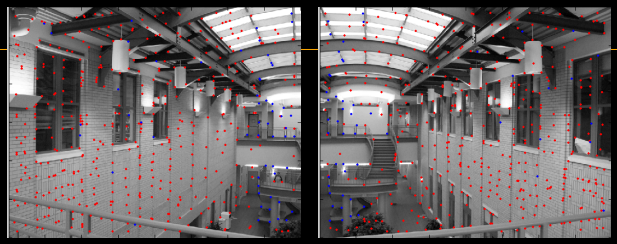


## Feature-Space Outlier Rejection

- 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?

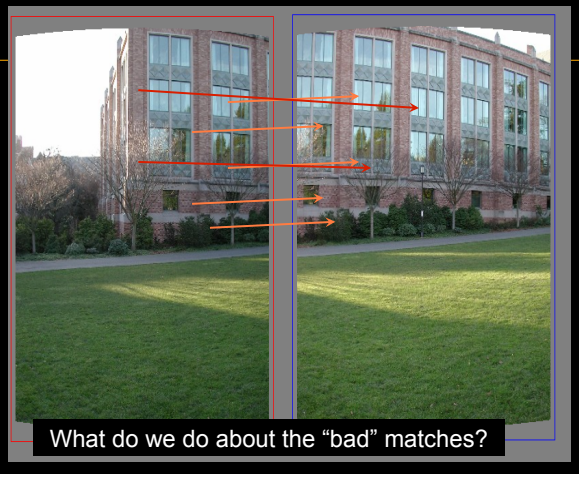



## Feature-Space Outliner Rejection



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

## Matching Features




## Image Matching

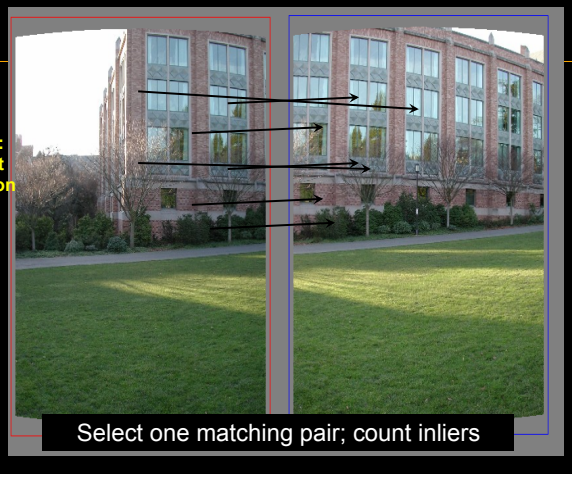
- For each image, find other images with greatest number of feature matches to current image
- For each pair of neighboring images, use **RANSAC algorithm** to find true matches (inliers), eliminate non-matching points (outliers), and compute best homography

## Dealing with Noisy Data: RANSAC

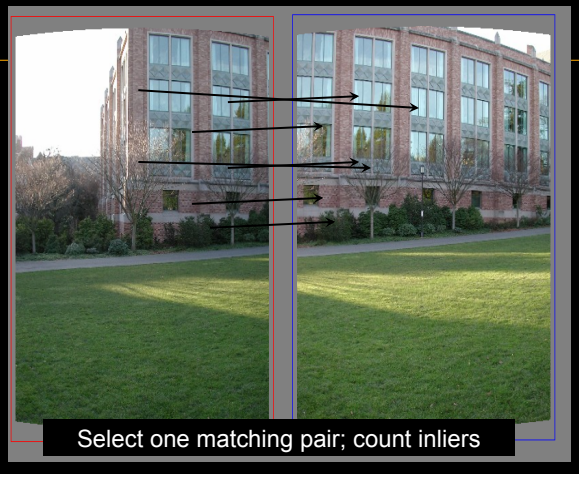
- How to find the best fitting data to a global model when the data are noisy – especially because of outliers, i.e., missing features and extraneous features?
- **RANSAC** (**Ran**dom **Sa**mple **C**onsensus) Algorithm
  - Iteratively select a small subset of data and fit model to that data
  - Then check all of data to see how many fit the model

## RANdom SAmple Consensus

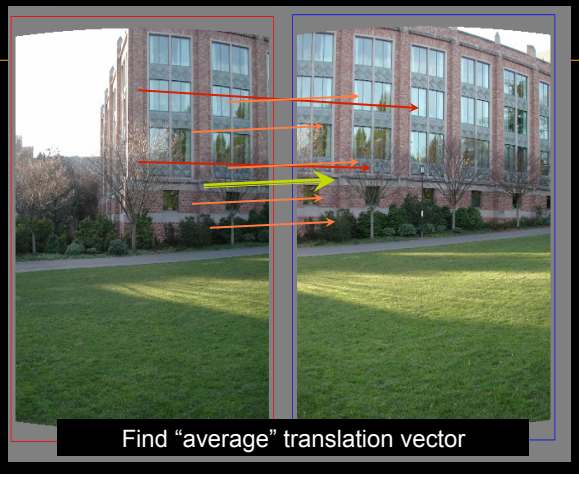
Example: Find best translation




## RANdom SAmple Consensus




## Compute Least Squares Fit



## RANSAC Algorithm for Estimating Homography

for k = 1:n

Select 4 feature pairs at random

Compute homography **H**

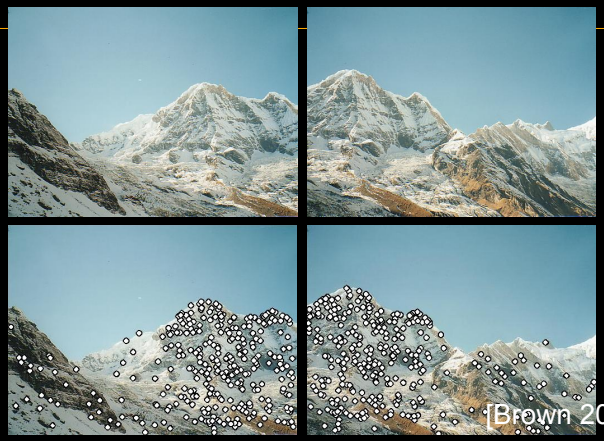
Count *inliers*, *i.e.*, where $SSD(p_i', \mathbf{H}\, p_i) < \varepsilon$

end

Keep the 4 pairs with largest number of inliers

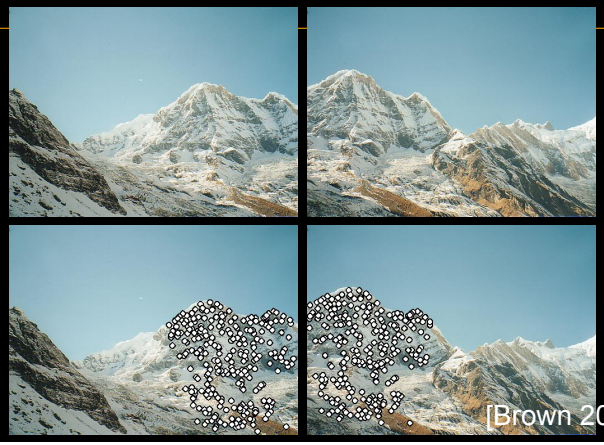
Re-compute least-squares **H** using *all* inliers

## Use RANSAC to Compute Homography



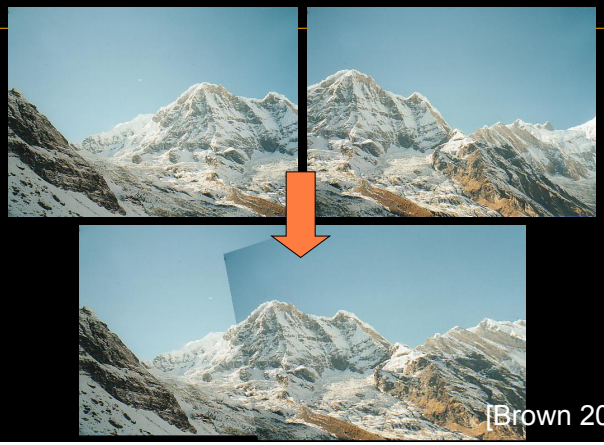
[Brown 2003]

## 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

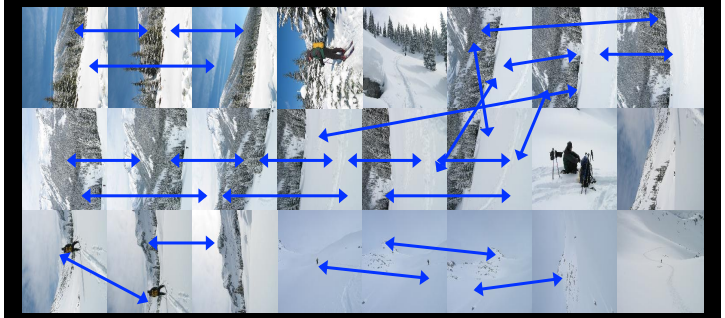
## How Many Iterations?

- If data contains $q$ percent outliers, $p$ points/ subset, and want probability $r$ that at least one subset is correct, then number of iterations needed is

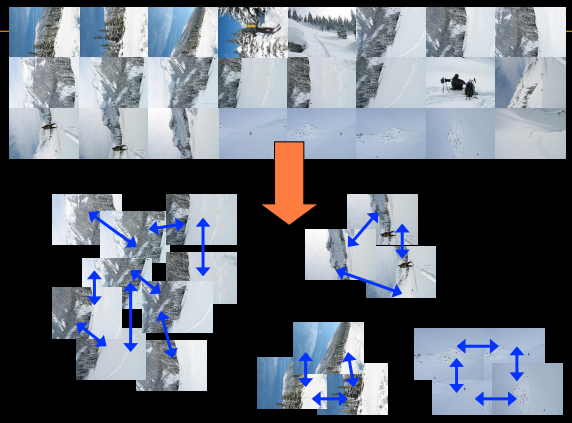
$$m = \frac{\log(1-r)}{\log(1-(1-q)^p)}$$

- Example: $p$ = 5, $q$ = 40%, $r$ = 0.99, then $m$ = 57

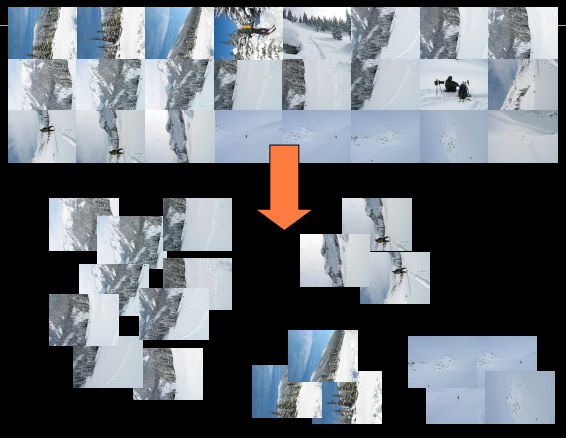
## Results of Image Matching



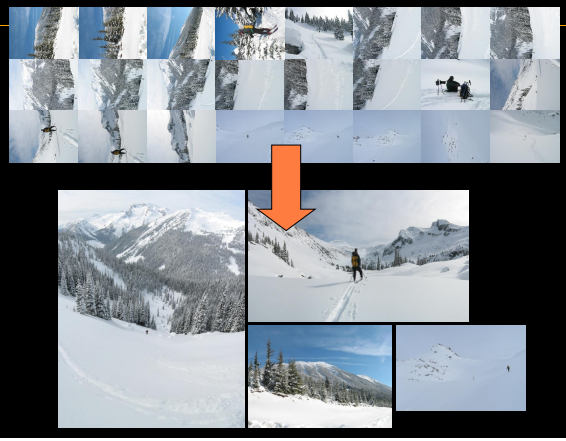
## Finding the Panoramas (= Cliques)



## 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 incrementally 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 Mistakes: 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

## Discovering Panoramas in Web Videos

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

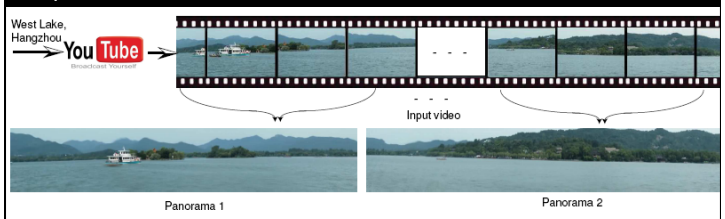



Figure 1: Working example. A user makes a query of "West Lake, Hangzhou" to *YouTube*, and feeds retrieved video clips into our system. Our system selects useful frames from the given videos and synthesizes panoramas using the selected source frames.

## An Optimization Problem

Given a video V

Find (non-overlapping) video segments $S_i$ that

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

Reject segments that have

- Too little coverage
- Too much penalty

## 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.

## More Examples

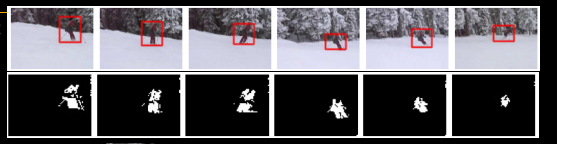
(d) Arches National Park. Its source images are frame 55 to 173 from a 15-second, 15fps web video.

(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.

## What about Moving Objects?

Detect [Liu&Gleicher06]



Discard

Selectively add back in

## 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.

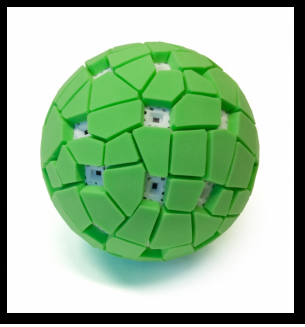
## Panorama Cameras

**Point Grey Ladybug3**

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

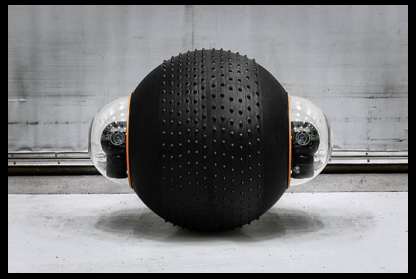
## Throwable Panoramic Ball Camera

- 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

## Unwrap Mosaics:
## A new representation for video editing

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

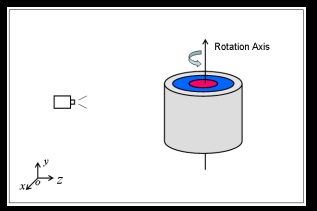
**Goal**: Given a video, recover for each moving, non-rigid object (1) its **texture** map modeling the object's appearance, (2) a 2D-to-2D mapping describing the texture map's **projection** to the images, and (3) a sequence of binary masks modeling **occlusion**

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

Slides by P. Kohli

## Viewing A Rotating Object

- Consider the class of single-axis rotations



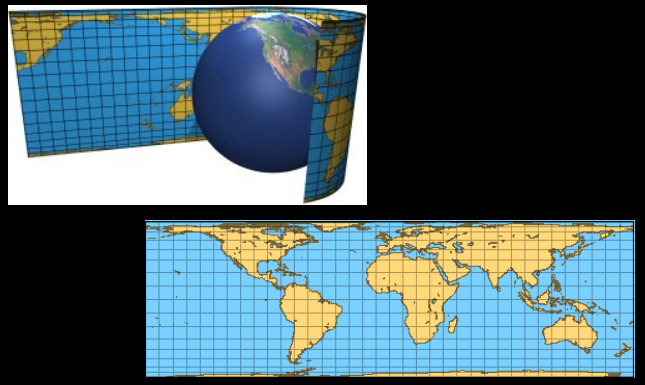

- Image sequence is a spatiotemporal volume showing all parts of surface

## Removing Occlusions


Images of Mayan vases by Justin Kerr

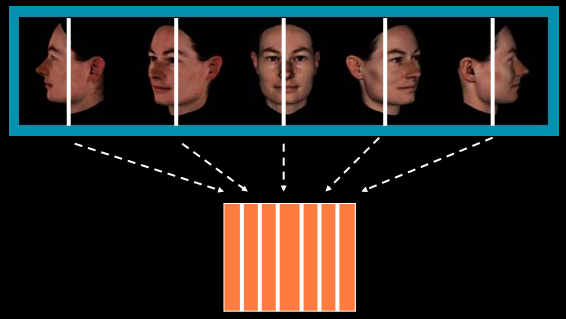
A **cyclograph** (aka **rollout photograph** or **peripheral photograph**) is a technique developed in photography; it is generated when an object rotates in front of a 1D camera or the camera moves around an object

## 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

## "Unwrap Mosaics" for Video Editing

Rav-Acha *et al., SIGGRAPH 2008*

## Unwrap Mosaics for Video Editing