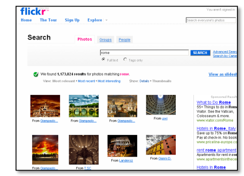
Seeing Places

Some slides by T. Berg, A. Criminisi, J. Hays, D. Hoiem, S. Lazebnik, S. Seitz, and N. Snavely

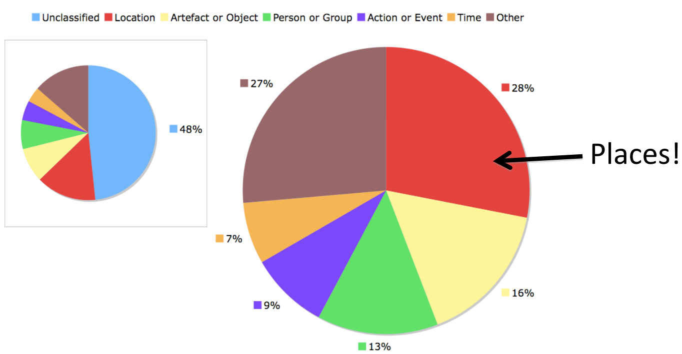
The World in Photos

- There are ***billions*** of photos online
- Photographic record of the surface of the earth
- Photo sharing on a massive scale





Importance of Places



Tag classification into broad WordNet categories

"Flickr Tag Recommendation based on Collective Knowledge", Borkur Sigurbjörnsson, Roelof van Zwol, www 2008

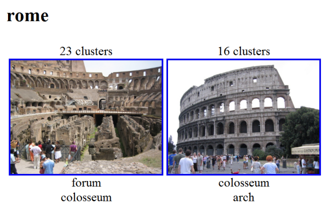
Types of Places

- Specific – Eiffel Tower, Golden Gate Bridge, Taj Mahal, …
- General – bakery, store, street, park, …
- Methods for dealing with each may be different – exact appearance matching vs. generalization

Applications of Photo Collections of Places

Hays and Efros, **Scene completion** using millions of photographs

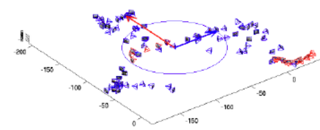
Crandall *et al.*, **Mapping** the world's photos



Simon *et al.*, **Scene summarization**

Image Localization

- *Where Am I?* Contest. ICCV 2005. Organized by Richard Szeliski.
- *Nokia Challenge: Where was this Photo Taken, and How?* Multimedia Grand Challenge 2010



Global Image Geolocation

- Hays and Efros. *Im2gps: estimating geographic information from a single image*. CVPR 2008.
- Quack, Leibe, Van Gool. *World-Scale Mining of Objects and Events from Community Photo Collections*. CIVR 2008
- Crandall, Backstrom, Huttenlocher, and Kleinberg. *Mapping the World's Photos*. WWW 2009.

Landmark Recognition

- What's that I'm looking at?
  - Google Goggles
  - Microsoft Bing for iPhone
  - Layar
  - Hyperlinking reality via camera phones (Univ. Ljubljana)
    - Take photo of a place and hyperlinks to information on the place and possible actions to take pop up

Place Time-Lapse

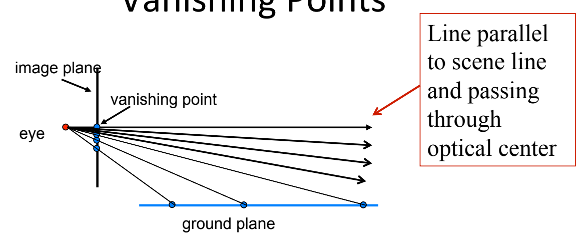
Scene Reconstruction and Visualization

- **Tour into the Picture** – General places from a single view
- **Façade** – Specific places from multiple views
- **Photo Tourism / Photosynth** – Specific places from multiple views
- **Finding Paths through World's Photos**
- **Tour the World** – Landmark recognition
- **Photo Pop-Up** – General places from a single view
- **Single View Metrology** – Reconstruct general places from a single view
- **Im2GPS** – General or specific places

What Information Can be Recovered?

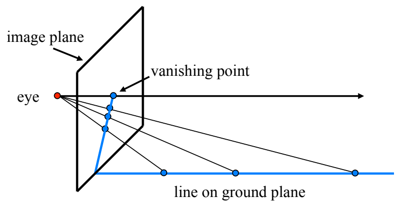
- From 1 image of a place (static, 3D)
- From multiple images of a place

Vanishing Points

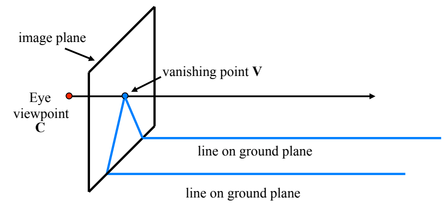


Vanishing point
- projection of a point "at infinity"
- Point in image beyond which projection of straight line cannot extend

Vanishing Points (2D)

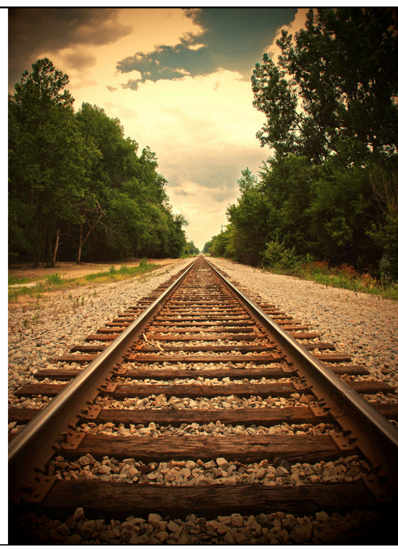


Vanishing Points

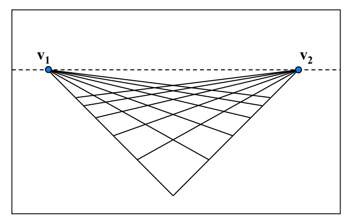


Properties

- Any two parallel lines have the same vanishing point **v**
- The ray from **C** through **v** is parallel to the lines
- An image may have more than one vanishing point

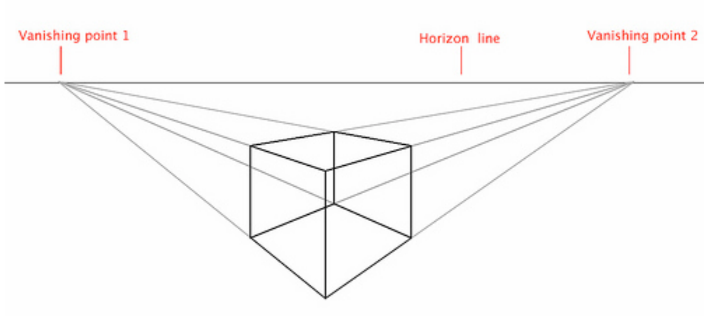


Vanishing Lines

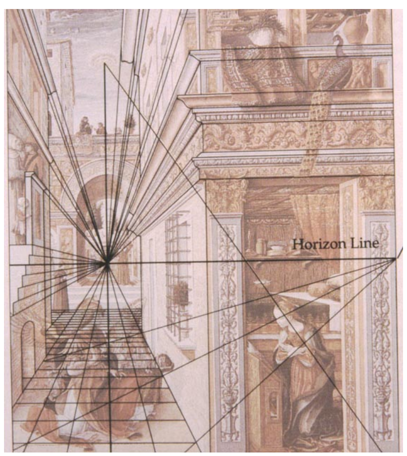


Multiple Vanishing Points

- Any set of parallel lines on a plane define a vanishing point
- Union of all of these vanishing points is the ***horizon line*** or ***vanishing line***
- Note that *different planes define different vanishing lines*



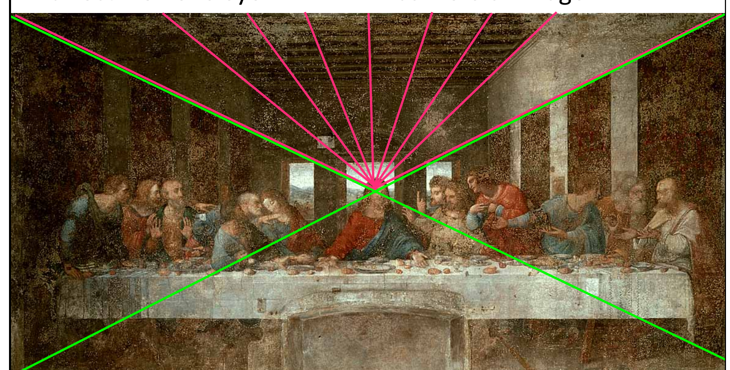
Carlo Crivelli (1486) *The Annunciation, with St. Emidius*



Perspective analysis of Crivelli's *Annunciation*

Leonardo da Vinci, "Last Supper" (c. 1497)

- Use of perspective to direct viewer's eye
- Strong perspective lines to corners of image



FAÇADE: Modeling and Rendering Architecture from Photographs

P. Debevec, C. J. Taylor, J. Malik

SIGGRAPH 1997

Façade Overview

- Take a few widely-separated photographs
- Build a simple 3D geometric model of scene
- Use correspondences between photos to adjust scene parameters
- Paste photos back onto simple geometry of scene for realistic façade

Slide credit: D. Luebke

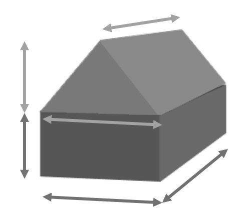
Photogrammetric Modeling

- User builds a simple geometric model using *blocks*: primitive solid shapes
  - Boxes, wedges, prisms, frusta, surfaces of revolution
- User marks correspondences between images and model
- System fits model to images

Slide credit: D. Luebke

Photogrammetric Modeling

- The system needs to solve for the parameters of blocks
  - Height, width, translation, rotation, etc.



Slide credit: D. Luebke

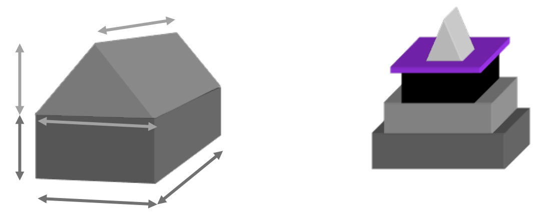
Photogrammetric Modeling

- Known: image segments to block edge correspondences
- Unknown: block parameters, camera position and orientation
- Architectural constraints reduce the number of unknowns

Slide credit: D. Luebke

Photogrammetric Modeling

- Architectural constraints
  - Roof prism lies flush on building block
  - Stacked tower blocks share center axis

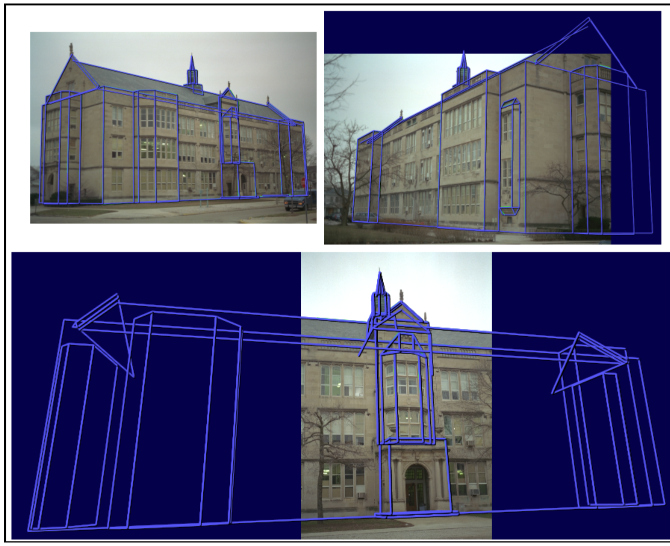


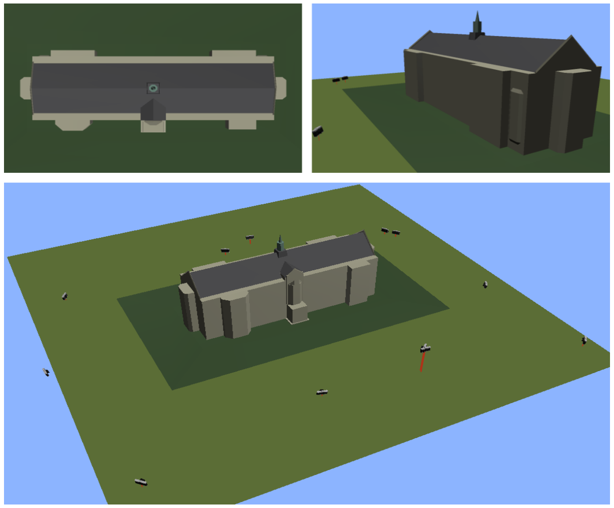
Slide credit: D. Luebke

Example

- Three of 12 input photographs of University High School, Urbana, IL

Slide credit: D. Luebke



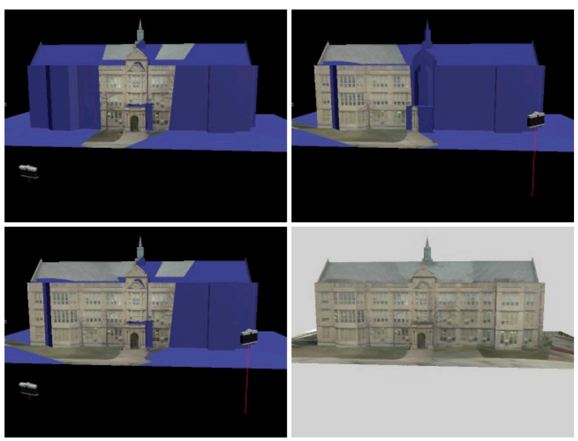


View-Dependent Texture Mapping

- Given the model, treat each camera position as a "projector"
- But some images overlap
  - Idea: pick image taken from viewpoint *closest* to desired rendering viewpoint
  - Better: use weighted average or do texture mapping on a per-pixel basis

Slide credit: D. Luebke

View-Dependent Texture Mapping



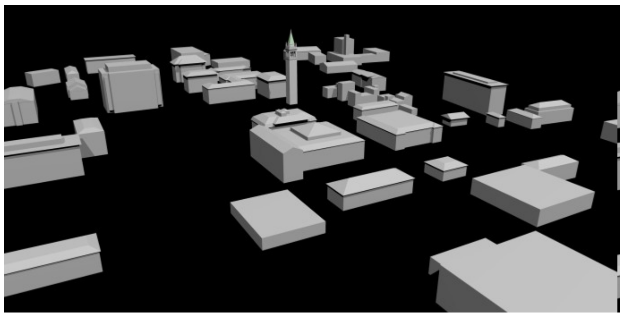
Photogrammetric Modeling

- The Campanile, UC Berkeley

Original photograph with marked edges | Recovered model | Model edges projected onto photograph | Synthetic rendering

Photogrammetric Modeling

- Model of UC Berkeley campus constructed from 15 photographs



The Campanile Movie

- Technique used in many movies, including
  - *The Matrix*
  - *The Matrix Reloaded*
  - *Mission Impossible*

Automatic 3D Scene Modeling

- Many products have been developed for visualizing, recognizing, and navigating **3D scenes** from a set of photos
  - Microsoft Photosynth
  - Autodesk 123D Catch
  - Google Goggles
  - Nokia Image Space

Photo Tourism / Photosynth

Noah Snavely
Steve Seitz
Rick Szeliski

Proc. SIGGRAPH, 2006
Int. J. Computer Vision, 2008

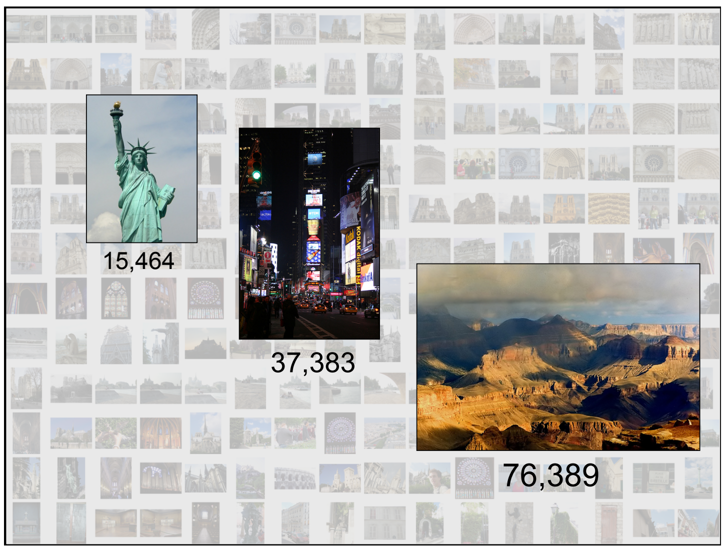


Photo Tourism Overview

- System for interactive browsing and exploring large collections of photos of a scene
- Computes viewpoint of each photo as well as a sparse 3D model of the scene

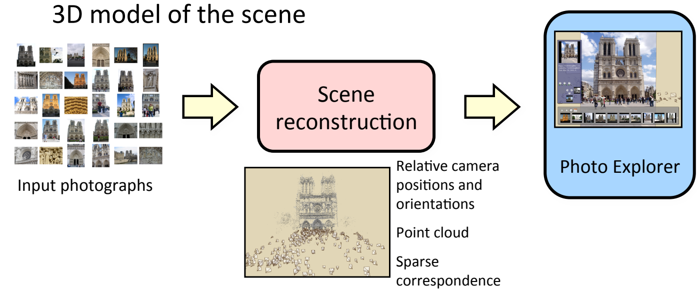


Photo Tourism

Input Photos

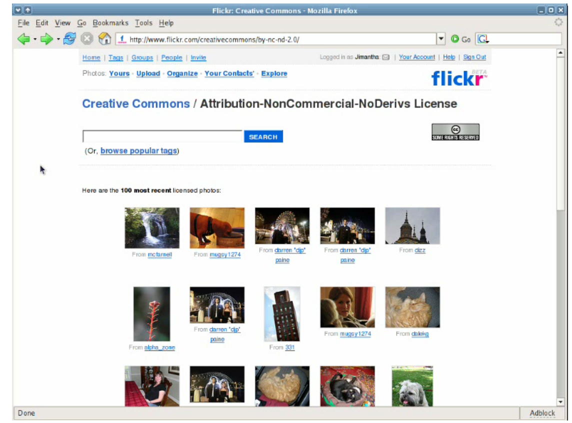
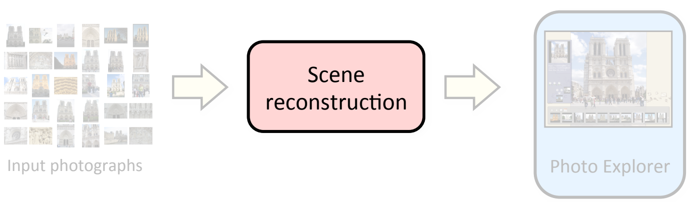
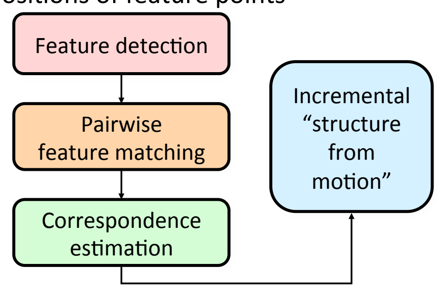


Photo Tourism Overview



Scene Reconstruction

- Automatically estimate
  - position, orientation, and focal lengths of cameras
  - 3D positions of feature points



Feature Detection

Detect feature points using SIFT detector

Feature Detection

Detect feature points using SIFT [Lowe, 1999]

Feature Detection

Detect feature points using SIFT

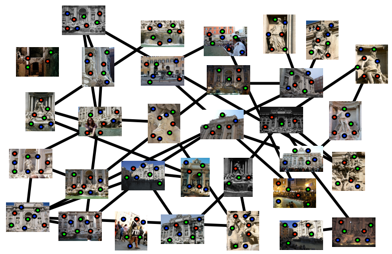
Feature Detection

Detect features using SIFT detector [Lowe, IJCV 2004]

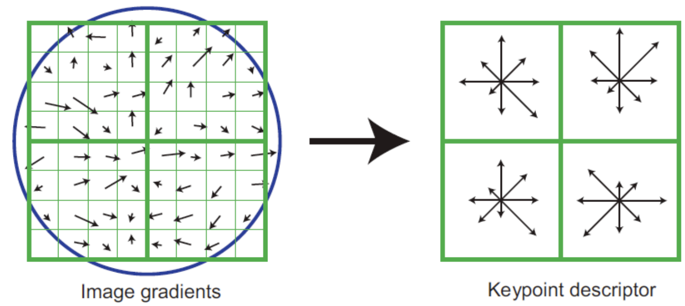
Pairwise Feature Matching

Match features between each pair of images

- Use SIFT descriptor
- For each feature point in image *I*, find 2 closest points in each other image, *J*, and accept if *d1/d2* < 0.6

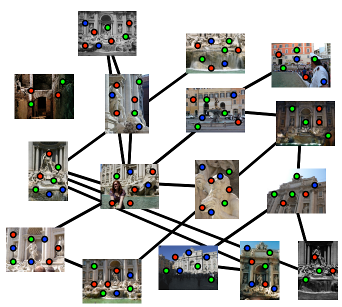


SIFT Descriptor



Wide-Baseline Feature Matching

Refine matching using **RANSAC + "8-point algorithm"** to estimate **fundamental matrices** between pairs of images

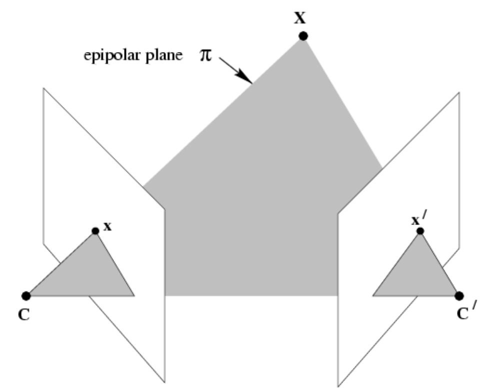


Epipolar Geometry and the Fundamental Matrix

Multi-View Geometry

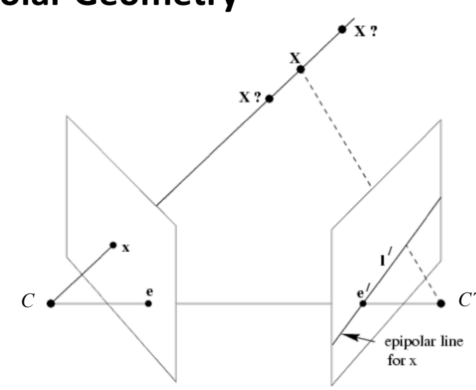
- Different views of a scene are *not* unrelated
- Relationships exist between two, three and more cameras
- Given an image point in one image, how does this restrict the position of the corresponding image point in another image?

Epipolar Geometry



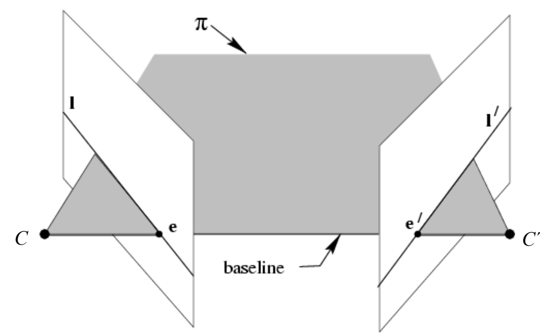
Co-Planarity Constraint: C, C', x, x' and X are co-planar

Epipolar Geometry



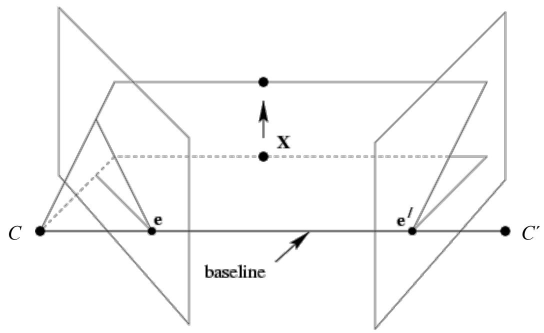
What if only C, C', and x are known?
Answer: x' constrained to lie on **epipolar line** l'

Epipolar Geometry



All points on plane π project onto **epipolar lines** l and l'

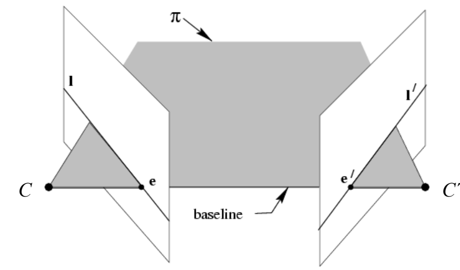
Epipolar Geometry



Family of planes π and lines l and l' intersect in **epipoles** e and e'

Epipolar Geometry

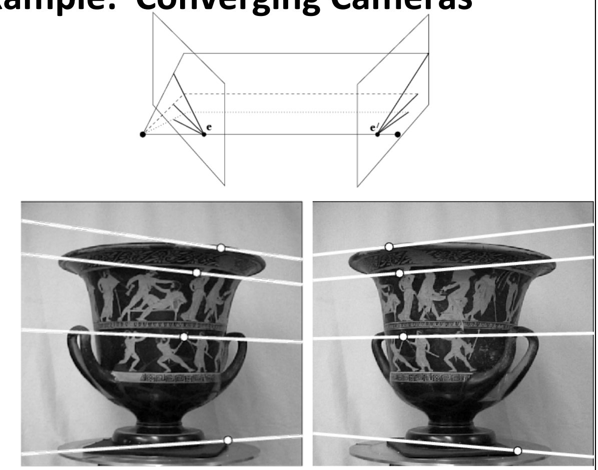
epipolar pole
= intersection of baseline with image plane
= projection of optical center from other image



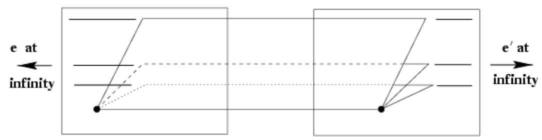
epipolar plane = plane containing baseline
epipolar line = intersection of epipolar plane with image

- **Correspondence geometry:** Given an image point *x* in the first view, how does this constrain the position of the corresponding point *x*′ in the second image?
- **Epipolar geometry constrains search for x′ from 2D to 1D**

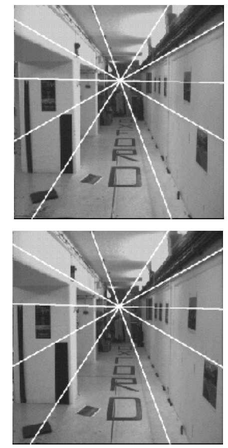
Example: Converging Cameras

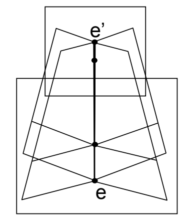


Example: Parallel Cameras



Example: Forward Motion

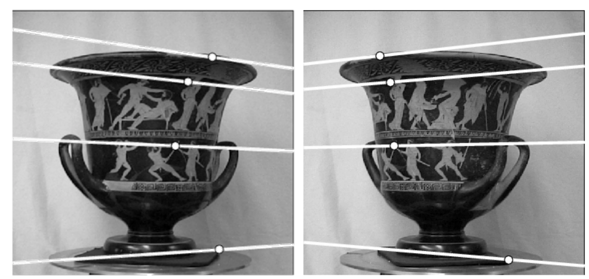




Fundamental Matrix F

- The fundamental matrix is the algebraic representation of the epipolar geometry
- The fundamental matrix is the unique 3 x 3, rank 2 matrix that satisfies the condition that for any pair of corresponding points **x**↔**x**´ in the two images: $x'^{\mathrm{T}}\mathbf{F}x = 0$
- **F** has 7 dof's since only known up to scale

Fundamental Matrix F



It can be used for
- Simplifying matching (1D search)
- Verifying candidate SIFT feature point matches

Computing F: The "8-Point Algorithm"

- The fundamental matrix F is defined by

$$\mathbf{x}'^{\mathrm{T}}\mathbf{F}\mathbf{x} = 0$$

for a pair of matching points **x** and **x**´ in two images

- Let ***x***=(*u*, *v*, 1)$^{\mathrm{T}}$, ***x***´ =(*u*´, *v*´, 1)$^{\mathrm{T}}$, $\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$

So, each match gives 1 linear equation:

$$uu'f_{11} + vu'f_{12} + u'f_{13} + uv'f_{21} + vv'f_{22} + v'f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

8-Point Algorithm

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

In practice, instead of solving $\mathbf{A}\mathbf{f} = 0$ we seek $\mathbf{f}$ to minimize $\|\mathbf{A}\mathbf{f}\|$ subj. to $\|\mathbf{f}\| = 1$ ⇒ Find the vector corresponding to the least singular value

RANSAC

Iterative method to estimate model parameters from observed data with outliers

Algorithm:

1. Pick 8 pairs of points, and assume they are in correct correspondence
2. Estimate **F** (using SVD to get best linear solution)
3. Calculate percentage of inliers
4. Repeat Steps 1-3 many times
5. Output the best model found
6. Replace **F** by **F**´, the closest singular matrix to **F**

Result is a set of geometrically consistent matches between each pair of images

Pairwise Feature Matching

Refine matching using RANSAC to estimate "fundamental matrices" between pairs of images

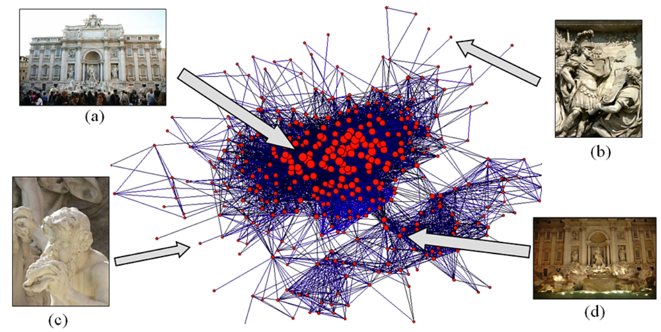
Fundamental matrix:

F is a 3x3 matrix with rank 2 such that for corresponding points $\mathbf{y_1}$ and $\mathbf{y_2}$

$$\mathbf{y_2^T F y_1} = 0.$$

The Power of SIFT

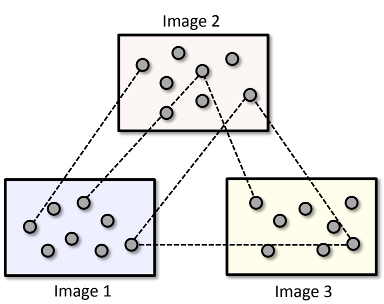
Construct Image Connectivity Graph



Each image is a node; edge if there are matching feature points between image pair

From Pairwise Matches to Tracks

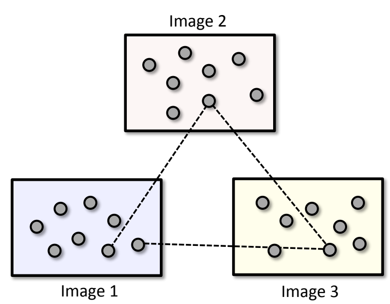
- Given pairwise matches, next link up matches to form "*tracks*"



- Each track is a connected component of the pairwise feature match graph
- Each track will eventually grow up to become a 3D point

From Pairwise Matches to Tracks

- Given pairwise matches, next link up matches to form "*tracks*"



- Some tracks might be *inconsistent*
- We remove these features from the troublesome images

Correspondence Estimation

Link up pairwise matches to form connected "tracks" of matching feature points across several images

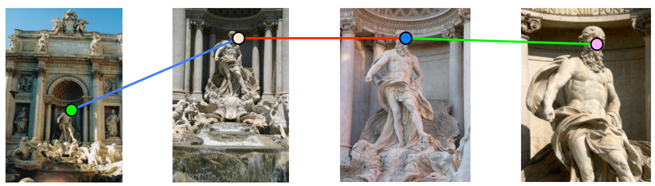


Image 1 Image 2 Image 3 Image 4

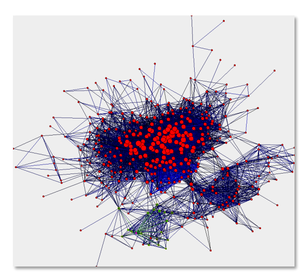
The Power of Transitivity

Most Tracks are Short

Example image collection with 3,000 images:

- 1,546,612 total tracks
- 79% have length 2
- 90% have length ≤ 3
- 98% have length ≤ 10
- Longest track: 385 features

Image Connectivity Post-Track Generation



Raw image matches

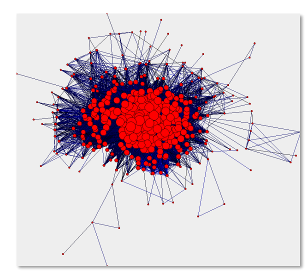
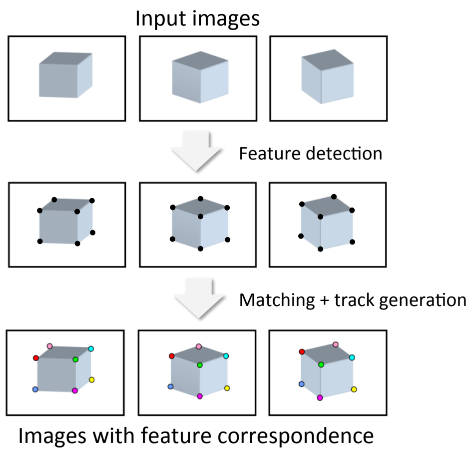
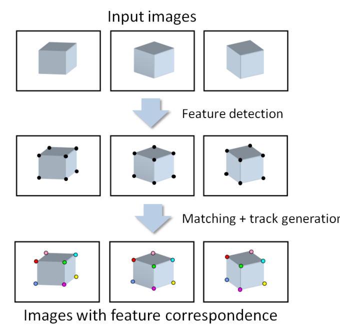


Image matches after track generation

The Story so far...



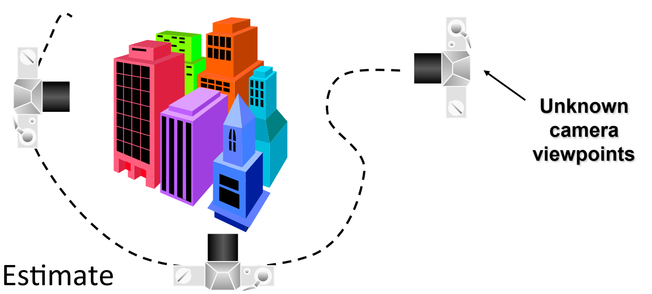
The Story so far...



Next step:

Use "**structure from motion**" algorithm to solve for geometry (cameras and 3D points)

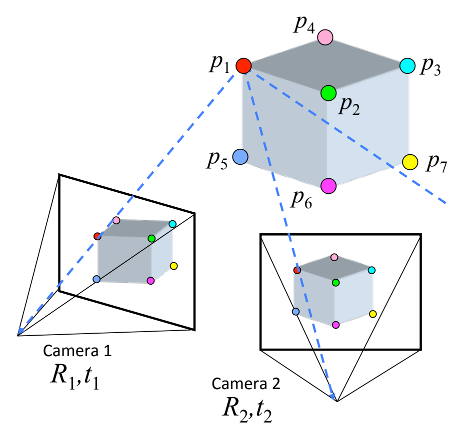
Structure from Motion (SfM)



Estimate

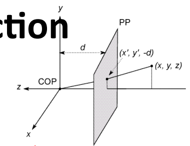
- Scene geometry (3D coordinates for each point)
- Camera extrinsic and intrinsic parameters (3D relative position and orientation, focal length, lens distortion)

Recover Structure from Motion



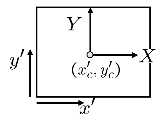
$$\textit{minimize}\quad f(\mathbf{R}, \mathbf{T}, \mathbf{P})$$

Modeling Camera Projection



- A camera is described by several parameters
  - Translation T of the optical center
  - Rotation R of the image plane
  - focal length f, principle point (x'_c, y'_c), pixel size (s_x, s_y)
- Camera projection equation

$$\mathbf{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$



- The projection matrix models the cumulative effect of all parameters:

$$\mathbf{\Pi} = \underbrace{\begin{bmatrix} -fs_x & 0 & x'_c \\ 0 & -fs_y & y'_c \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsics}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R}_{3x3} & \mathbf{0}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{T}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{translation}}$$

- **Projective camera model**

Structure from Motion

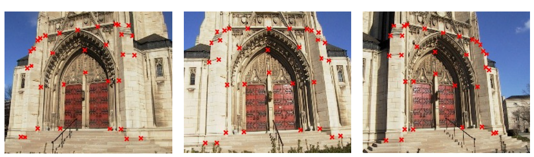
- The SfM Problem
  - Reconstruct scene geometry ***and*** camera "motion" from two or more images

Match 2D Features → **Estimate 3D parameters** → **Optimize (Bundle Adjust)** → **Fit Surfaces**

SfM Pipeline

- 9 parameters per camera (3D position, 3D orientation, focal length, 2 radial distortion parameters (assumes square pixels and known CoP)
- 3 parameters for each scene point (3D position)

Structure from Motion



Step 1: Match features

- Detect feature points using SIFT
- Find correspondences between frames using SIFT descriptor similarity
- Find best subset of correspondences that are consistent with a single fundamental matrix

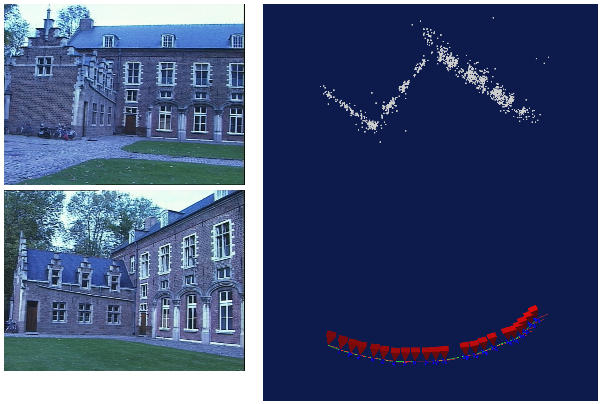
Structure from Motion

$$\underbrace{\begin{bmatrix} \mathbf{I}_1 \\ \mathbf{I}_2 \\ \vdots \\ \mathbf{I}_f \end{bmatrix}}_{\text{Images}} = \underbrace{\begin{bmatrix} \mathbf{\Pi}_1 \\ \mathbf{\Pi}_2 \\ \vdots \\ \mathbf{\Pi}_f \end{bmatrix}}_{\text{Motion}} \underbrace{\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}}_{\text{Structure}}$$

Step 2: Estimate "motion" *and* structure

- Simplified projection model, e.g., [Tomasi 92]
- 2 or 3 views at a time [Hartley 00]

Example: Computed "structure" is white scene points, and "motion" is red camera poses

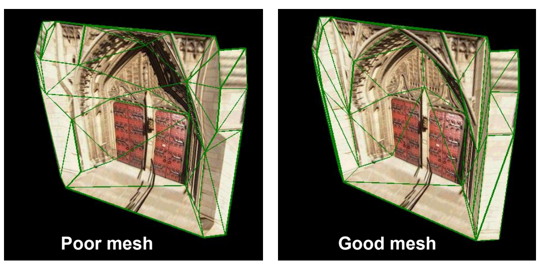


Structure from Motion

Step 3: Refine estimates of all parameters simultaneously

- Called "Bundle adjustment" in photogrammetry
- Non-linear, least-squares optimization problem

Structure from Motion



Morris and Kanade, 2000

Step 4: Recover surfaces from point cloud

- Image-based triangulation

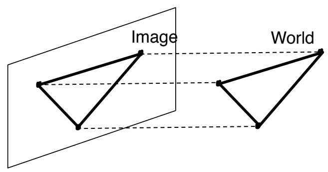
Factorization Method for Solving SfM

- Given a set of matching feature points, estimate the 3D structure and 3D motion (camera poses)
- Assumption: **Orthographic Projection** camera model
- Matched points: (q_{fp}, q_{fp}), f: frame, p: point

Orthographic Projection

Special case of perspective projection

– Distance from optical center to the image plane is *infinite*



– Good approximation for telephoto lenses

– Also called "parallel projection": (x, y, z) → (x, y)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

SFM under Orthographic Projection

2D inhomogeneous image point

orthographic projection matrix (w/ rotation)

3D inhomogeneous scene point

Camera translation

$$\underset{2\times 1}{\mathbf{q}} = \underset{2\times 3}{\mathbf{\Pi}} \underset{3\times 1}{\mathbf{p}} + \underset{2\times 1}{\mathbf{t}}$$

Observation

– Choose scene origin to be centroid of 3D points

– Choose image origins to be centroid of 2D points

– Allows us to drop the camera translation term:

$$\mathbf{q} = \mathbf{\Pi p}$$

Factorization Method for SfM under Orthographic Projection (Tomasi & Kanade)

projection of *n* features in one image:

$$\underset{2\times n}{\begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{bmatrix}} = \underset{2\times 3}{\prod} \underset{3\times n}{\begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \end{bmatrix}}$$

projection of *n* features in *m* images:

$$\underset{2m\times n}{\begin{bmatrix} \mathbf{q}_{11} & \mathbf{q}_{12} & \cdots & \mathbf{q}_{1n} \\ \mathbf{q}_{21} & \mathbf{q}_{22} & \cdots & \mathbf{q}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{q}_{m1} & \mathbf{q}_{m2} & \cdots & \mathbf{q}_{mn} \end{bmatrix}} = \underset{2m\times 3}{\begin{bmatrix} \mathbf{\Pi}_1 \\ \mathbf{\Pi}_2 \\ \vdots \\ \mathbf{\Pi}_m \end{bmatrix}} \underset{3\times n}{\begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \end{bmatrix}}$$

W measurement **M** motion **S** shape

Shape by Factorization [Tomasi & Kanade, 92]

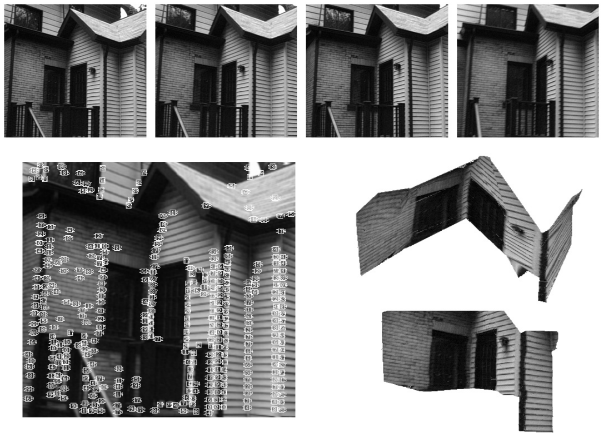
known — $\underset{2f\times n}{\mathbf{W}} = \boxed{\underset{2f\times 3}{\mathbf{M}} \; \underset{3\times n}{\mathbf{S}}}$ — solve for

Factorization Technique

- **W** is at most rank 3 (assuming no noise)
- We can use *singular value decomposition* to factor **W**:

$$\underset{2f\times n}{\mathbf{W}} = \underset{2f\times 3}{\mathbf{M'}} \; \underset{3\times n}{\mathbf{S'}}$$

Results

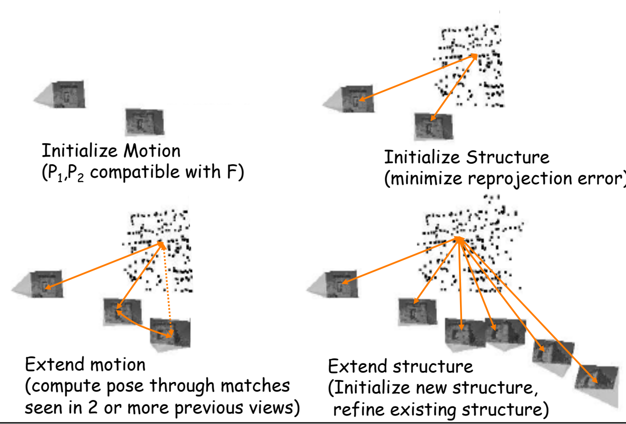


M. Pollefeys and L. Van Gool

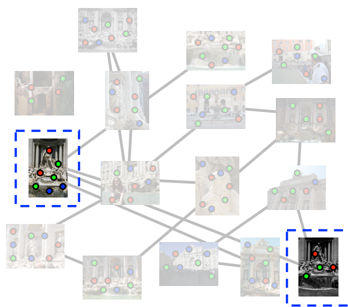
Incremental Structure from Motion

- Optimize parameters for two cameras and matching points
- Find new image with most matches to existing points
- Initialize new camera using pose estimation
- Bundle adjust
- Add new points
- Bundle adjust
- Etc.

Incremental Structure from Motion Computation



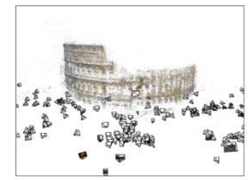
Incremental Structure from Motion



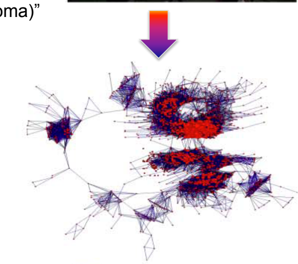
Pick initial pair of cameras with (1) large number of matches, ***and*** (2) large baseline

Example

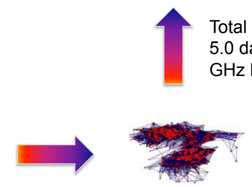
1,994 images from Flickr search on "colosseum AND (rome OR roma)"



Total runtime: 5.0 days on 3.8 GHz Pentium 4



Connectivity graph



390 registered photos (containing 188,306 matched points)

Reconstruction Performance

- For photo sets from the Internet, 20% to 75% of the photos were registered
- Most unregistered photos belonged to different connected components

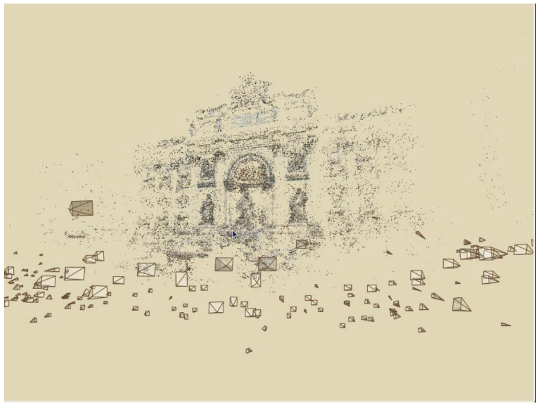
- Running time: < 1 hour for 80 photos

  > 1 week for 2,600 photos (of Notre Dame processed and matched, and 600 reconstructed)

Example Uses of Reconstruction: Navigation Controls

- Free-flight navigation
- Object-based browsing
- Relation-based browsing
- Overhead map

Free-Flight Navigation

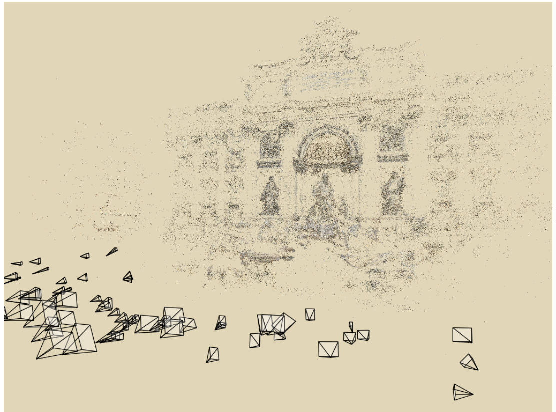


Linearly interpolate camera poses; morph between images

Object-based Browsing

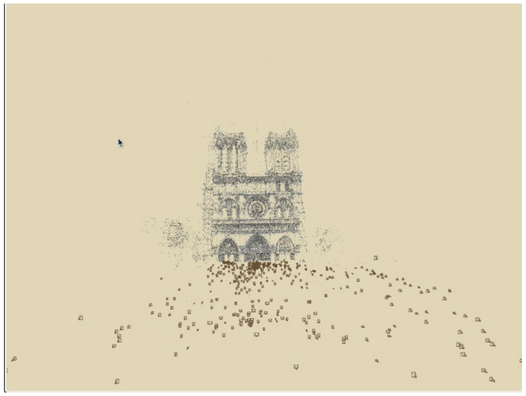
"Show me photos of this object" finds images with matching features and smaller field of view

Rendering



Rendering

Annotations

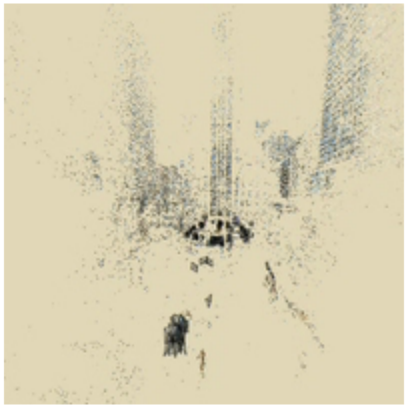


Annotations automatically transferred from one image to all others that contain matching points

Saint Basil's Cathedral

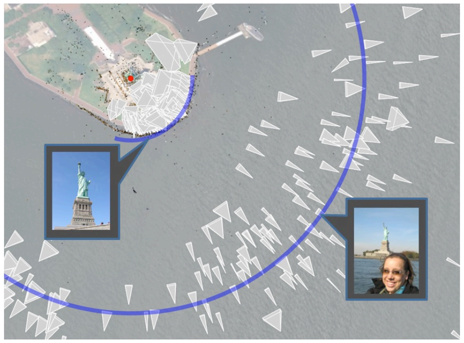
Trafalgar Square

Rockefellar Center



Mount Rushmore

Great Wall Fly-By

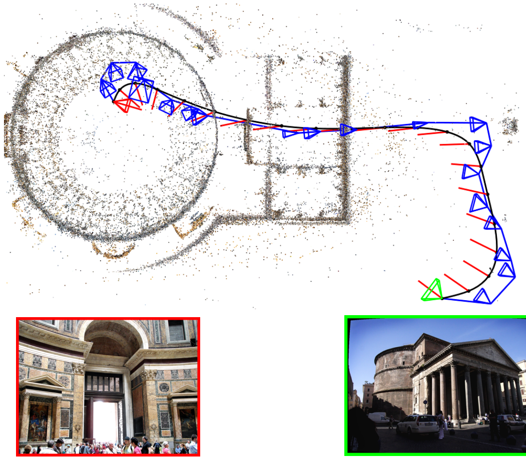


Finding Paths Through the World's Photos

N. Snavely, R. Garg, S. Seitz, R. Szeliski

SIGGRAPH 2008

Path Planning Example



Scaling Up: Building Rome in a Day

- City-scale Structure-from-Motion
- Rome: 150,000 images, 21 hours processing time using 496 cores
- Venice: 250,000 images, 65 hours using 496 cores
- 2.7 million images on Flickr from search on "Rome"

Scene Summarization for Image Collections

- Goal: Select a set of images that efficiently represents the visual content of a given scene. The ideal summary presents the most interesting and important aspects of the scene with minimal redundancy
- Find clusters of images based on SIFT points, then pick 1 "canonical" image for each cluster

Scene Summarization for Online Image Collections

Supplementary Video #1

(this video contains audio)

Scene Summaries for Internet Photo Collections

Supplementary Video

IM2GPS: Estimating Geographic Information from a Single Image

James Hays and Alexei A. Efros
Carnegie Mellon University

Where am I?

- Nokia Multimedia Conference Grand Challenge: Where was this photo taken?
- Is geolocation just instance-level landmark recognition or can you reason about location and geography from non-specific scene properties?
- Can image similarity be a proxy for geographic proximity?

Where am I?

What can you say about where these photos were taken?

How?

Collect a large collection of geo-tagged photos

6.5 million images with both GPS coordinates and geographic keywords, removing images with keywords like birthday, concert, abstract, …

Test set – 400 randomly sampled images from this collection. Manually removed abstract photos and photos with recognizable people – 237 test photos

Im2gps Image Features

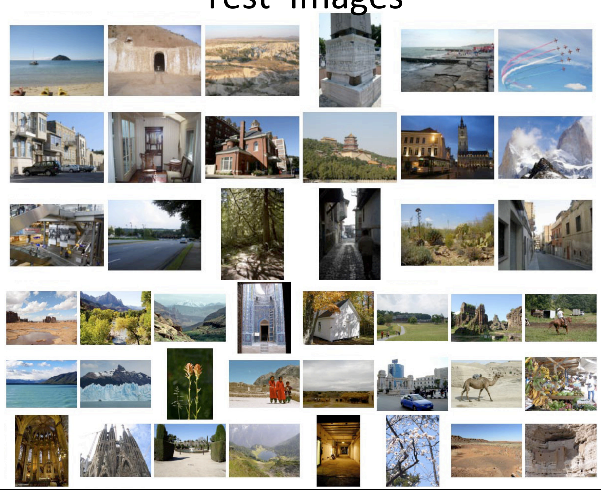
- **Gist descriptor** – 5x5 spatial resolution, 4 scales, 8 orientations. Code
- **Tiny Color Image** – 5x5 and 16x16 spatial resolutions.
- **Color Histogram** – L*A*B* 4x14x14 histograms.
- **Texton Histogram** – 512 entry, filter bank based. Code
- **Line Features** – Histograms of straight line lengths and angles.
- **Geometric Context** – 8x8 probability of geometric class (e.g. Ground, Sky, Vertical, Porous). Code
- Histograms are compared with Chi Squared measure, other features with L1 distance.

How?

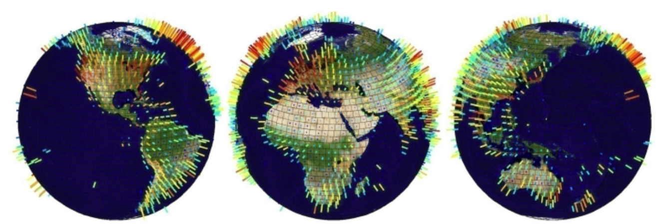
Data-driven geolocation:

1. For each input image, compute features
2. Compute distance in feature space to all 6 million images in the database (each feature contributes equally)
3. Label the image with GPS coordinates of:
   a. 1-nearest neighbor
   b. k = 120 nearest neighbors – probability map over entire globe

Test Images



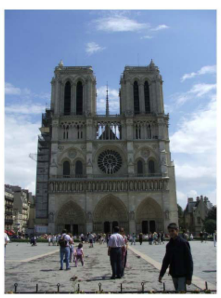
im2gps Geographic Photo Density

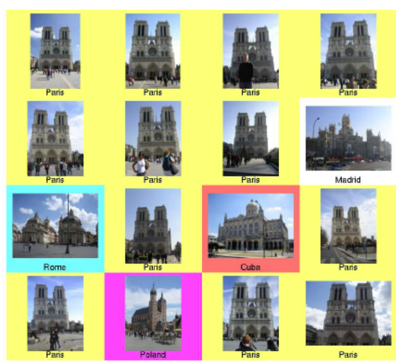


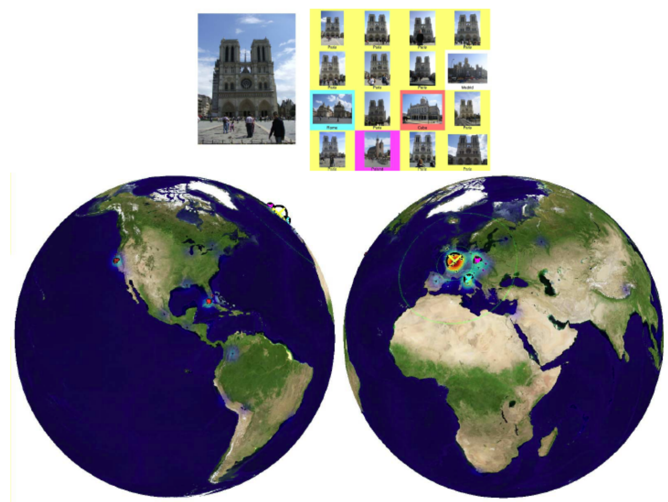
6.4 mil. photos by 110K photographers.
1 TB of visual data.
Photographs had at least one place keyword.
Photos average ~1 content descriptive keyword.

Scene Matches

Scene Matches







Modeling Places

Idea: Pop-up book world modeling of places

Automatic Photo Pop-Up

- Three classes of surface: “ground,” “sky,” “vertical”
- Not just a box: can model more kinds of scenes
- Automatic segmentation, classification, and reconstruction

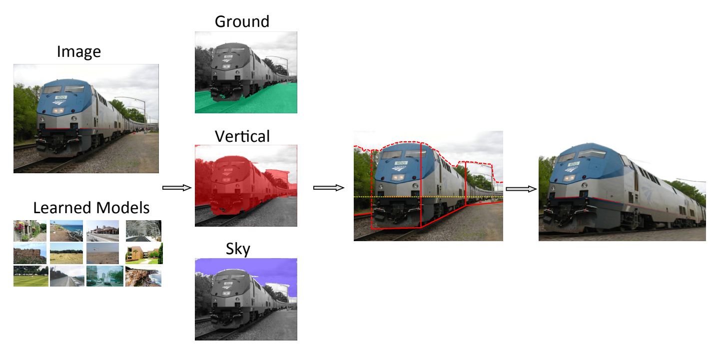
Goals

- Simple, piecewise planar objects oriented vertically
- Outdoor scenes with flat ground plane
- Sky is background

Pop-up book scene modeling

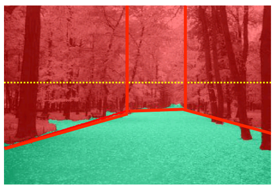
Algorithm Overview

Input	Segment and label regions	Cut ’n fold	3D model



Results

Input Image



Cut and Fold

Automatic Photo Pop-up

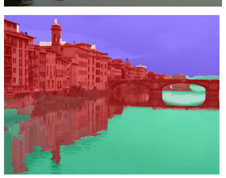
Results

Input Image

Automatic Photo Pop-up

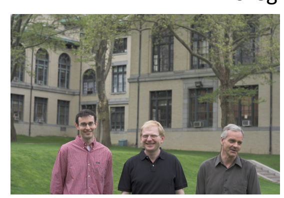
Failures

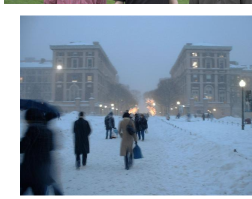
Labeling Errors

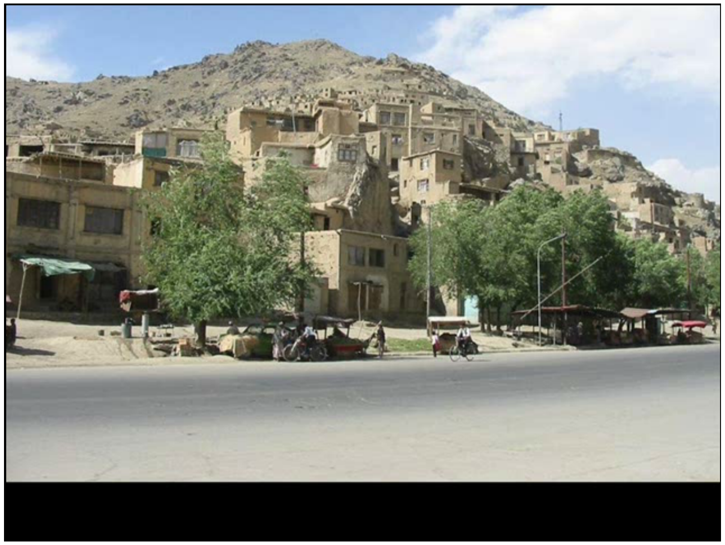


Failures

Foreground Objects







3D Modeling from a Single Photograph

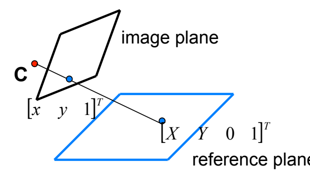
Is it possible to extract 3D geometric information from a ***single*** image?

Single View Metrology

- Goal: Obtain **measurements** of scene structure (e.g., lengths, areas) from a single image
- Idea: Use constraints imposed by *parallel* lines and planes, and *orthogonality* of lines and planes
  - Only **up to scale** — i.e., we only know ***ratios***
  - "Upgrade" to metric (absolute) with a few known **reference** measurements

from Criminisi *et al.*

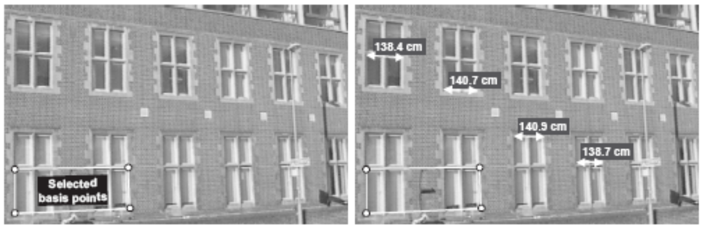
Measurements within a Plane



$X = \mathbf{H}x$

- Solve for homography **H** relating reference plane to image plane
  - **H** maps reference plane (*X*,*Y*) coords to image plane (*x*,*y*) coords
  - Fully determined from 4 known points on ground plane
    - Option A: physically measure 4 points on ground
    - Option B: find a square, guess the dimensions
  - Given (*x*, *y*), compute (*X*, *Y*) using $\mathbf{H}^{-1}$

Planar Measurements



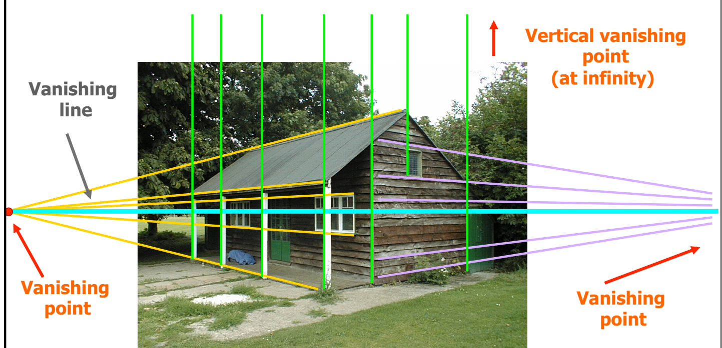
Known distances between 4 points on left allows measurements of other distances (right)

What Else can be Computed?

- Assume that images are obtained by perspective projection
- Uncalibrated cameras
- Assume that, from the image, a:
  - **vanishing line** of a *reference plane*
  - **vanishing point** of another *reference direction*

  can be determined

Geometric Cues

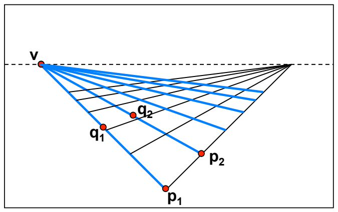


What can be Computed?

1. Measurements of the distance between any planes that are parallel to the reference plane
2. Measurements on these planes
3. The camera's position relative to the reference plane and direction

Results are sufficient for a partial or complete 3D reconstruction of the observed scene

Computing Vanishing Points from Lines



- Intersect p_1q_1 with p_2q_2

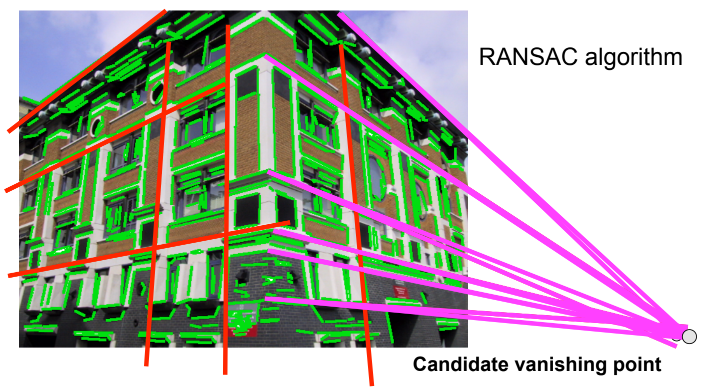
$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

- Least squares version
  - Better to use more than two lines and compute the "closest" point of intersection

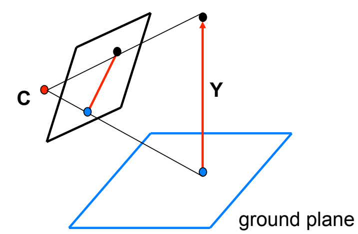
A Vanishing Point Detection Algorithm

1) Edge detection and straight line fitting to obtain the set of straight line segments, S_A
2) RANSAC: Repeat
   a) Randomly select two segments $s_1, s_2 \in S_A$ and intersect them to obtain point **p**
   b) The support set S_p is the set of straight lines in S_A going through point **p**
3) Set the dominant vanishing point as the point **p** with the largest support set S_p
4) Remove all edges in S_p from S_A and repeat Step 2

Automatic Estimation of Vanishing Points and Lines



Measuring Height without a Ruler

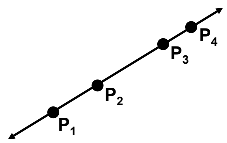


Goal: Compute *Y* from image measurements

Cross Ratio

The cross-ratio of 4 collinear points

- A **Projective Invariant**
  - That is, something that **does *not* change under projective transformations** (including perspective projection)



$$\frac{\|P_3 - P_1\| \, \|P_4 - P_2\|}{\|P_3 - P_2\| \, \|P_4 - P_1\|} \qquad \mathbf{P}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

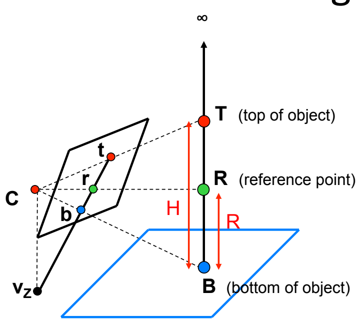
$$\frac{\|P_1 - P_3\| \, \|P_4 - P_2\|}{\|P_1 - P_2\| \, \|P_4 - P_3\|}$$

- Can permute the point ordering
  - 4! = 24 different orders (but only 6 distinct values)
- This is the fundamental invariant of projective geometry

Cross Ratio

- The **cross ratio** between points provides an affine length ratio
  - The value of the cross ratio determines a ratio of distances between planes in the world
- Thus, if we know the length of an object in the scene, we can use it as a reference to calculate the lengths of other objects

Measuring Height



$$\frac{\|T - B\| \, \|\infty - R\|}{\|R - B\| \, \|\infty - T\|} = \frac{H}{R}$$

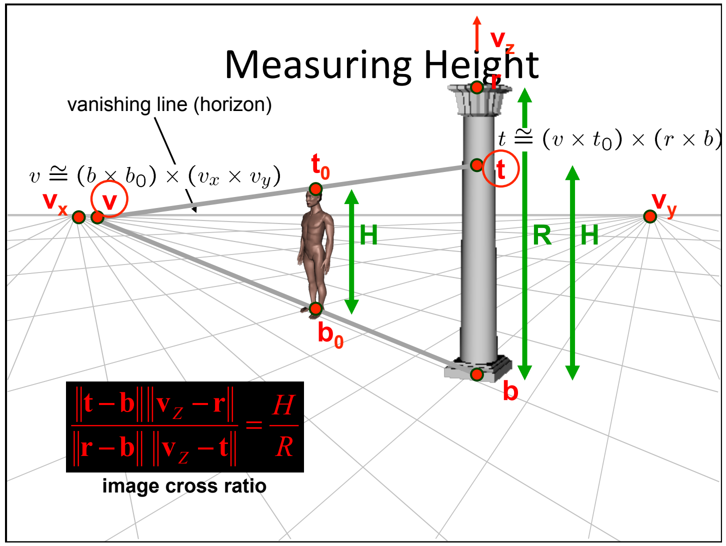
scene cross ratio

$$\frac{\|t - b\| \, \|v_Z - r\|}{\|r - b\| \, \|v_Z - t\|} = \frac{H}{R}$$

image cross ratio

scene points represented as $\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$ image points as $\mathbf{p} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Measuring Height



$$\frac{\|t - b\| \, \|v_Z - r\|}{\|r - b\| \, \|v_Z - t\|} = \frac{H}{R}$$

image cross ratio

Measuring Height

- The distance $|| t_r – b_r ||$ is known
- Used to estimate the height of the man in the scene

Example: Vanishing Line of Reference Plane & Reference Line

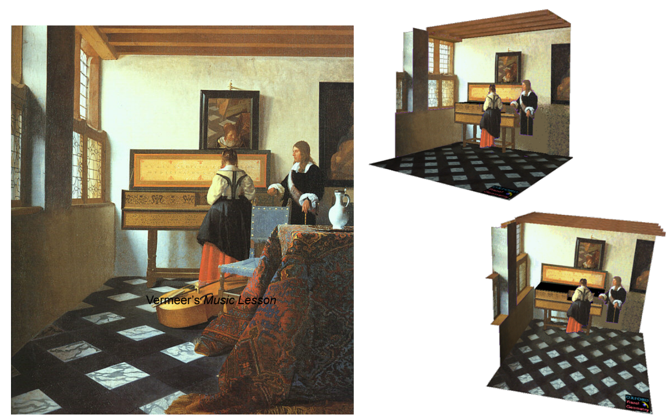
Reference plane here is the ground plane

Application: Forensic Science

from Criminisi *et al.*

Knowing the height of the phone booth (the reference measurement), can determine the height of the person

Results




The Virtual
Museum
A. Criminisi @
Microsoft, 2002