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


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

- *Nokia Challenge: Where was this Photo Taken, and How?* Multimedia Grand Challenge 2010

Global Image Geolocation

- Quack, Leibe, Van Gool. *World-Scale Mining of Objects and Events from Community Photo Collections.* CIVR 2008

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

Minning Time-Lapse Videos from Internet Photos

Ricardo Martin-Brualla¹  David Gallup²  Steve Seitz³,²
¹University of Washington  ²Google

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

Line parallel to scene line and passing through optical center
Vanishing Points (2D)

- Image plane
- Eye
- Line on ground plane

Vanishing Points

- 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*

Leonardo da Vinci, “Last Supper” (c. 1497)
- Use of perspective to direct viewer’s eye
- Strong perspective lines to corners of image
**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

---

**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

---

**Photogrammetric Modeling**

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

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

Example

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

Photogrammetric Modeling

- The Campanile, UC Berkeley

Slide credit: D. Luebke
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

![Diagram of Photo Tourism Overview](image.png)
Photo Tourism

Photo Tourism Overview

Input Photos

Scene Reconstruction

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

  Feature detection
  Pairwise feature matching
  Correspondence estimation
  Incremental “structure from motion”
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 \( d_1/d_2 < 0.6 \)

**SIFT Descriptor**

![SIFT descriptor diagram]

**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 \( \pi \) project onto epipolar lines \( l \) and \( l' \)
Epipolar Geometry

Family of planes $\pi$ and lines $l$ and $l'$ intersect in epipoles $e$ and $e'$

- **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

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

Example: Converging Cameras
**Example: Parallel Cameras**

![Parallel Cameras Diagram]

**Example: Forward Motion**

![Forward Motion Diagram]

---

**Fundamental Matrix $F$**

- The fundamental matrix is the algebraic representation of the epipolar geometry.

- The fundamental matrix is the unique $3 \times 3$, rank 2 matrix that satisfies the condition that for any pair of corresponding points $x \leftrightarrow x'$ in the two images:

  $$x'^{T}Fx = 0$$

- $F$ 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 $\mathbf{F}$: The “8-Point Algorithm”

- The fundamental matrix $\mathbf{F}$ is defined by
  \[
  \mathbf{x}'^T \mathbf{F} \mathbf{x} = 0
  \]
  for a pair of matching points $\mathbf{x}$ and $\mathbf{x}'$ in two images

- Let $\mathbf{x} = (u, v, 1)^T$, $\mathbf{x}' = (u', v', 1)^T$, $\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$

  So, each match gives 1 linear equation:
  
  
  \[
  uu'f_{11} + vu'f_{12} + uf_{13} + uv'f_{21} + vv'f_{22} + vf_{23} + uf_{31} + vf_{32} + f_{33} = 0
  \]

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 $\mathbf{F}$ (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 $\mathbf{F}$ by $\mathbf{F}'$, the closest singular matrix to $\mathbf{F}$

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

8-Point Algorithm

\[
\begin{bmatrix}
  u_1 u'_1 & v_1 u'_1 & u'_1 \\
  u_2 u'_2 & v_2 u'_2 & u'_2 \\
  \vdots & \vdots & \vdots \\
  u_n u'_n & v_n u'_n & u'_n \\
\end{bmatrix}
\begin{bmatrix}
  f_{11} \\
  f_{12} \\
  f_{13} \\
  f_{21} \\
  f_{22} \\
  f_{23} \\
  f_{31} \\
  f_{32} \\
  f_{33} \\
\end{bmatrix}
= 0
\]

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

Pairwise Feature Matching

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

Fundamental matrix:

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

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

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

The Power of Transitivity

Image Connectivity Post-Track Generation

Raw image matches
Image matches after track generation
The Story so far...

Input images

Feature detection

Matching + track generation

Images with feature correspondence

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)

minimize $f(R, T, P)$

Camera 1
$R_1, t_1$

Camera 2
$R_2, t_2$

Camera 3
$R_3, t_3$
### 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

$$
\begin{bmatrix}
\pi x \\
\pi y \\
\pi z
\end{bmatrix} =
\begin{bmatrix}
-X \\
-Y \\
-Z
\end{bmatrix} =
\begin{bmatrix}
R_{xz} & R_{yz} & R_{zx} & T_x \\
R_{xy} & R_{yz} & R_{yx} & T_y \\
R_{xz} & R_{xy} & R_{zx} & T_z
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
$$

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

$$
\Pi =
\begin{bmatrix}
-f_x & 0 & x'_c & 0 \\
0 & -f_y & y'_c & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
R_{xz} & R_{yz} & R_{zx} & T_x \\
R_{xy} & R_{yz} & R_{yx} & T_y \\
R_{xz} & R_{xy} & R_{zx} & T_z
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
$$

- **Projective camera model**

### Structure from Motion

#### The SfM Problem
- Reconstruct scene geometry and camera “motion” from two or more images

#### 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

#### 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

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: \((q_{fp}, q_{fp}), f: \text{ frame}, p: \text{ point}\)
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”: \((x, y, z) \rightarrow (x, y)\)

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

SFM under Orthographic Projection

\[q = \Pi p + t\]
2\(\times 1\) \(2\times3\) \(3\times1\) \(2\times1\)

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:

\[q = \Pi p\]

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

projection of \(n\) features in one image:

\[
\begin{bmatrix}
q_1 & q_2 & \cdots & q_n
\end{bmatrix}
\begin{bmatrix}
2\times n
\end{bmatrix} = \prod_i
\begin{bmatrix}
p_i \\
2 \times 3
\end{bmatrix}
\begin{bmatrix}
p_i \\
3 \times n
\end{bmatrix}
\]

projection of \(n\) features in \(m\) images:

\[
\begin{bmatrix}
q_{11} & q_{12} & \cdots & q_{1n} \\
q_{21} & q_{22} & \cdots & q_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
q_{m1} & q_{m2} & \cdots & q_{mn}
\end{bmatrix}
\begin{bmatrix}
2m \times n
\end{bmatrix} = \prod_i
\begin{bmatrix}
p_i \\
2m \times 3
\end{bmatrix}
\begin{bmatrix}
p_i \\
3 \times n
\end{bmatrix}
\]

\[
W \text{ measurement} \\
M \text{ motion} \\
S \text{ shape}
\]

Shape by Factorization [Tomasi & Kanade, 92]

\[
\begin{bmatrix}
W
\end{bmatrix}_{2f \times n} = \begin{bmatrix}
M & S
\end{bmatrix}_{2f \times 3 \times n}
\]

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

\[
W = M' S'
\]

\[
\begin{bmatrix}
2f \times n \\
2f \times 3 \\
3 \times n
\end{bmatrix}
\]
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

- Initialize Motion ($P_1, P_2$ compatible with $F$)
- Initialize Structure (minimize reprojection error)
- Extend motion (compute pose through matches seen in 2 or more previous views)
- Extend structure (initialize new structure, refine existing structure)
Incremental Structure from Motion

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

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

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

Example Uses of Reconstruction: Navigation Controls

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

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

Saint Basil's Cathedral

Trafalgar Square

Rockefeller Center
Finding Paths Through the World’s Photos
N. Snavely, R. Garg, S. Seitz, R. Szeliski
SIGGRAPH 2008
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
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?

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

What can you say about where these photos were taken?
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

![](image)

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.
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

Algorithm Overview

<table>
<thead>
<tr>
<th>Input</th>
<th>Segment and label regions</th>
<th>Cut 'n fold</th>
<th>3D model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Ground</td>
<td>Vertical</td>
<td>Sky</td>
</tr>
</tbody>
</table>
Results

Input Image

Cut and Fold

Automatic Photo Pop-up

Failures

Labeling Errors

Failures

Foreground Objects
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

Measurements within a Plane

- 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 $H^{-1}$

3D Modeling from a Single Photograph

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

Vanishing point

Vertical vanishing point (at infinity)

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

RANSAC algorithm

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)

\[
p = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}
\]

\[
\frac{P_1 - P_2}{P_3 - P_4} = \frac{P_1 - P_2}{P_3 - P_4}
\]

- 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

\[
\frac{v}{b} \times (b \times b_0) \times (v_x \times v_y)
\]

\[
v = \frac{H}{R}
\]

\[
v = \frac{H}{R}
\]
Measuring Height

- The distance $|| t, - b, ||$ 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

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

Results

Vermeer's Music Lesson
The Virtual Museum
A. Criminisi @ Microsoft, 2002