Computer Vision - Lecture 15

Camera Calibration & 3D Reconstruction

15.01.2015

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de

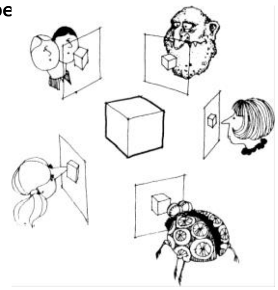
leibe@vision.rwth-aachen.de

Course Outline

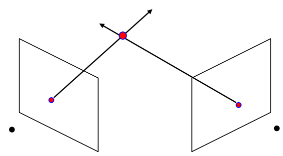
- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
  - Epipolar Geometry and Stereo Basics
  - Camera calibration & Uncalibrated Reconstruction
  - Structure-from-Motion
- Motion and Tracking

Recap: What Is Stereo Vision?

- Generic problem formulation: given several images of the same object or scene, compute a representation of its 3D shape



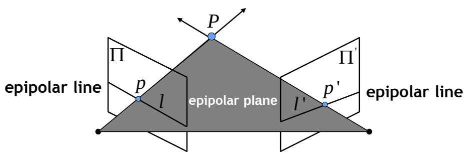
Recap: Depth with Stereo - Basic Idea



- Basic Principle: Triangulation
  - Gives reconstruction as intersection of two rays
  - Requires
    - Camera pose (calibration)
    - Point correspondence

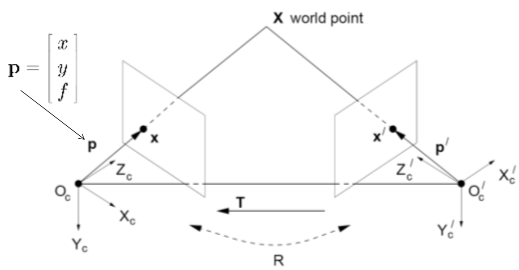
Recap: Epipolar Geometry

- Geometry of two views allows us to constrain where the corresponding pixel for some image point in the first view must occur in the second view.



- Epipolar constraint:
  - Correspondence for point p in Π must lie on the epipolar line l' in Π' (and vice versa).
  - Reduces correspondence problem to 1D search along conjugate epipolar lines.

Recap: Stereo Geometry With Calibrated Cameras



- Camera-centered coordinate systems are related by known rotation R and translation T:

$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T}$$

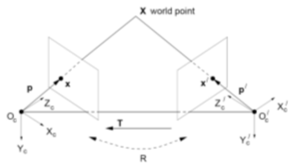
Recap: Essential Matrix

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X}) = 0$$

$$\mathbf{X}' \cdot (\mathbf{T}_x \; \mathbf{R}\mathbf{X}) = 0$$

Let $\mathbf{E} = \mathbf{T}_x\mathbf{R}$

$$\mathbf{X}'^T \mathbf{E} \mathbf{X} = 0$$



- This holds for the rays p and p' that are parallel to the camera-centered position vectors X and X', so we have: $\mathbf{p}'^T \mathbf{E}\mathbf{p} = 0$
- $\mathbf{E}$ is called the essential matrix, which relates corresponding image points [Longuet-Higgins 1981]

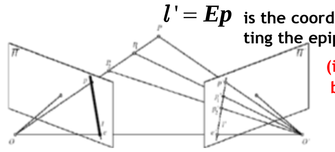
Recap: Essential Matrix and Epipolar Lines

$$\mathbf{p}'^T \mathbf{E}\mathbf{p} = 0$$

Epipolar constraint: if we observe point p in one image, then its position p' in second image must satisfy this equation.

$l' = Ep$ is the coordinate vector representing the epipolar line for point p

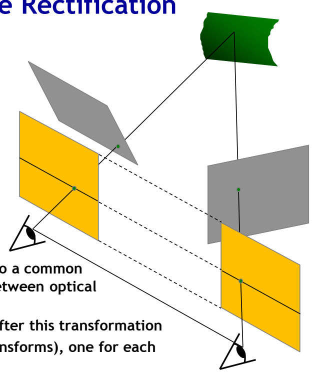
(i.e., the line is given by: $l'^T \mathbf{x} = 0$)



$l = E^T p'$ is the coordinate vector representing the epipolar line for point p'

Recap: Stereo Image Rectification

- In practice, it is convenient if image scanlines are the epipolar lines.

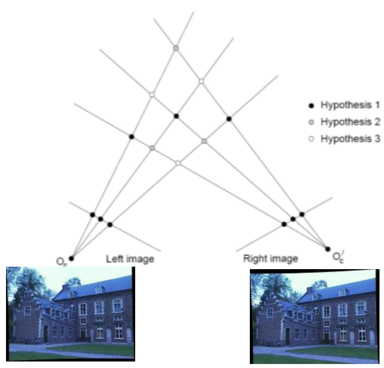


- Algorithm
  - Reproject image planes onto a common plane parallel to the line between optical centers
  - Pixel motion is horizontal after this transformation
  - Two homographies (3x3 transforms), one for each input image reprojection

Recap: Stereo Reconstruction

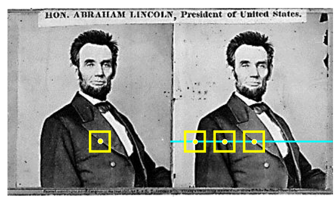
- Main Steps
  - Calibrate cameras
  - Rectify images
  - Compute disparity
  - Estimate depth

Correspondence Problem



Multiple match hypotheses satisfy epipolar constraint, but which is correct?

Dense Correspondence Search



- For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Examine all pixels on the epipolar line and pick the best match (e.g. SSD, correlation)
  - Triangulate the matches to get depth information
- This is easiest when epipolar lines are scanlines
  $\Rightarrow$ Rectify images first

Example: Window Search

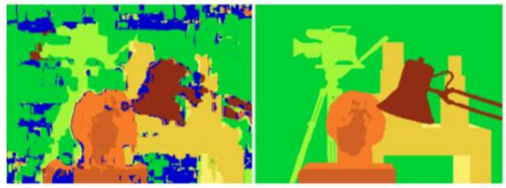
- Data from University of Tsukuba


Scene


Ground truth

Example: Window Search

- Data from University of Tsukuba

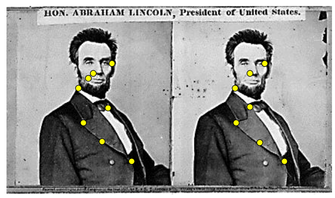

Window-based matching (best window size)
Ground truth

Effect of Window Size


$W = 3$ $W = 20$

Want window large enough to have sufficient intensity variation, yet small enough to contain only pixels with about the same disparity.

Alternative: Sparse Correspondence Search



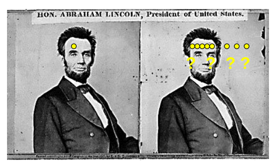
- Idea:
  - Restrict search to sparse set of detected features
  - Rather than pixel values (or lists of pixel values) use *feature descriptor* and an associated *feature distance*
  - Still narrow search further by epipolar geometry

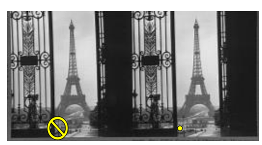
What would make good features?

Dense vs. Sparse

- Sparse
  - Efficiency
  - Can have more reliable feature matches, less sensitive to illumination than raw pixels
  - But…
    - Have to know enough to pick good features
    - Sparse information
- Dense
  - Simple process
  - More depth estimates, can be useful for surface reconstruction
  - But…
    - Breaks down in textureless regions anyway
    - Raw pixel distances can be brittle
    - Not good with very different viewpoints

Difficulties in Similarity Constraint


Untextured surfaces


Occlusions

Possible Sources of Error?

- Low-contrast / textureless image regions
- Occlusions
- Camera calibration errors
- Violations of *brightness constancy* (e.g., specular reflections)
- Large motions

Summary: Stereo Reconstruction

- Main Steps
  - Calibrate cameras
  - Rectify images
  - Compute disparity
  - Estimate depth

- So far, we have only considered calibrated cameras…

- Today
  - Uncalibrated cameras
  - Camera parameters
  - Revisiting epipolar geometry
  - Robust fitting

Recap: A General Point

- Equations of the form

$$\mathrm{Ax} = 0$$

- How do we solve them? (always!)
  - Apply SVD

SVD

$$\boldsymbol{A} = \boldsymbol{UDV}^T = \boldsymbol{U}\begin{bmatrix} d_{11} & & \\ & \ddots & \\ & & d_{NN} \end{bmatrix}\begin{bmatrix} v_{11} & \cdots & v_{1N} \\ \vdots & \ddots & \vdots \\ v_{N1} & \cdots & v_{NN} \end{bmatrix}^T$$

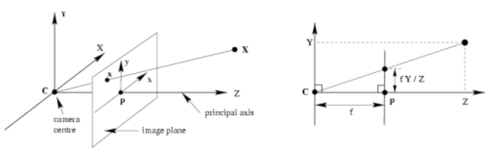
Singular values Singular vectors

  - Singular values of A = square roots of the eigenvalues of A^TA.
  - The solution of Ax=0 is the *nullspace* vector of A.
  - This corresponds to the *smallest singular vector* of A.

Topics of This Lecture

- Camera Calibration
  - Camera parameters
  - Calibration procedure
- Revisiting Epipolar Geometry
  - Triangulation
  - Calibrated case: Essential matrix
  - Uncalibrated case: Fundamental matrix
  - Weak calibration
  - Epipolar Transfer
- Active Stereo
  - Laser scanning
  - Kinect sensor

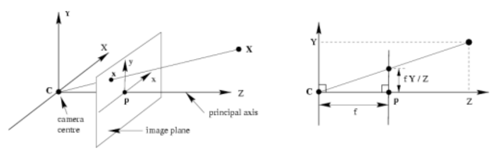
Recall: Pinhole Camera Model



$$(X, Y, Z) \mapsto (f\,X/Z,\ f\,Y/Z)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \qquad \mathrm{x} = \mathrm{PX}$$

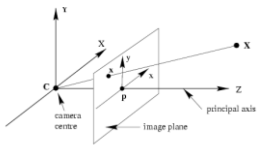
Pinhole Camera Model



$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix}\begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

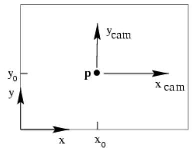
$$\mathrm{x} = \mathrm{PX} \qquad \mathrm{P} = \mathrm{diag}(f, f, 1)[\mathrm{I} \mid 0]$$

Camera Coordinate System



- *Principal axis*:
  - Line from the camera center perpendicular to the image plane
- *Normalized (camera) coordinate system*:
  - Camera center is at the origin and the principal axis is the z-axis
- *Principal point* (**p**):
  - Point where principal axis intersects the image plane (origin of normalized coordinate system)

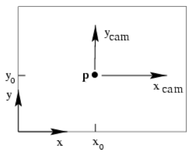
Principal Point Offset



principal point: (p_x, p_y)

- Camera coordinate system: origin at the principal point
- Image coordinate system: origin is in the corner

Principal Point Offset

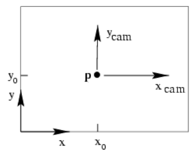


principal point: (p_x, p_y)

$$(X, Y, Z) \mapsto (f\,X/Z + p_x,\; f\,Y/Z + p_y)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\,X + Z\,p_x \\ f\,Y + Z\,p_x \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Principal Point Offset

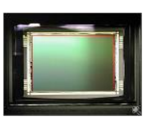


principal point: (p_x, p_y)

$$\begin{pmatrix} f\,X + Zp_x \\ f\,Y + Zp_x \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}$$ calibration matrix $$\mathrm{P} = \mathrm{K}[\mathrm{I}\,|\,0]$$

Pixel Coordinates: Non-Square Pixels



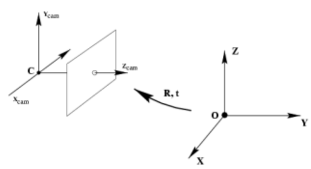
Pixel size: $\frac{1}{m_x} \times \frac{1}{m_y}$

***m_x* pixels per meter in horizontal direction,**
***m_y* pixels per meter in vertical direction**

$$K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

pixels/m **m** **pixels**

Camera Rotation and Translation



- In general, the camera coordinate frame will be related to the world coordinate frame by a rotation and a translation

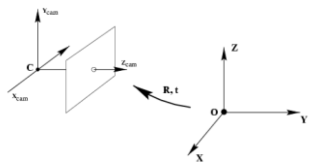
$$\tilde{\mathrm{X}}_{cam} = \mathrm{R}(\tilde{\mathrm{X}} - \tilde{\mathrm{C}})$$

coords. of point in camera frame

coords. of a point in world frame (nonhomogeneous)

coords. of camera center in world frame

Camera Rotation and Translation



In non-homogeneous coordinates:

$$\tilde{X}_{cam} = R(\tilde{X} - \tilde{C})$$

$$X_{cam} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \tilde{X} \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} X$$

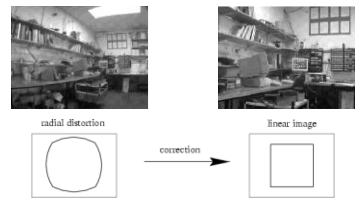
$$x = K[I\,|\,0]X_{cam} = K[R\,|\,-R\tilde{C}]X \qquad P = K[R\,|\,t], \quad t = -R\tilde{C}$$

Note: $\mathbf{C}$ is the null space of the camera projection matrix ($\mathbf{PC}=0$)

Summary: Camera Parameters

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

$$K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & s & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$



Summary: Camera Parameters

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

$$K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & s & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

- Extrinsic parameters
  - Rotation $\mathbf{R}$
  - Translation $\mathbf{t}$
    (both relative to world coordinate system)
- Camera projection matrix

$$P = K[R\,|\,t] = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix}$$

How many degrees of freedom does P have?

Camera Parameters: Degrees of Freedom

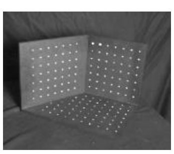
	DoF
• Intrinsic parameters	
Principal point coordinates	2
Focal length	1
Pixel magnification factors	1
Skew (non-rectangular pixels)	1
Radial distortion	
• Extrinsic parameters	
Rotation R	3
Translation t (both relative to world coordinate system)	3

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

- Camera projection matrix $P = K[R\,|\,t]$
  - ⇒ General pinhole camera: 9 DoF
  - ⇒ CCD Camera with square pixels: 10 DoF
  - ⇒ General camera: 11 DoF

Calibrating a Camera

- Compute intrinsic and extrinsic parameters using observed camera data.

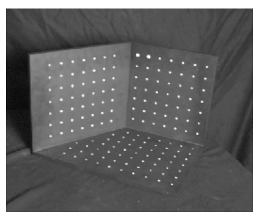


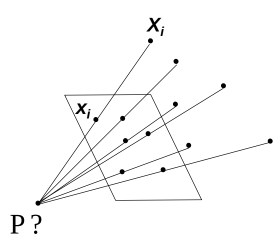
Main idea

- Place "calibration object" with known geometry in the scene
- Get correspondences
- Solve for mapping from scene to image: estimate $P=P_{int}P_{ext}$

Camera Calibration

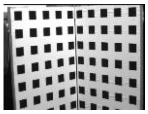
- Given n points with known 3D coordinates X_i and known image projections x_i, estimate the camera parameters





Camera Calibration: Obtaining the Points

- For best results, it is important that the calibration points are measured with *subpixel accuracy*.
- How this can be done depends on the exact pattern.
- Algorithm for checkerboard pattern
  1. Perform Canny edge detection.
  2. Fit straight lines to detected linked edges.
  3. Intersect lines to obtain corners.
  - If sufficient care is taken, the points can then be obtained with localization accuracy < 1/10 pixel.



- Rule of thumb
  - Number of constraints should exceed number of unknowns by a factor of five.
  - ⇒ For 11 parameters of P, at least 28 points should be used.

Camera Calibration: DLT Algorithm

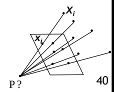
(DLT = "Direct Linear Transform")

$$\lambda \mathrm{x}_i = \mathrm{P}\mathrm{X}_i \qquad \lambda \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} \mathrm{X}_{i,1} \\ \mathrm{X}_{i,2} \\ \mathrm{X}_{i,3} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{P}_1^T \\ \mathrm{P}_2^T \\ \mathrm{P}_3^T \end{bmatrix} \mathrm{X}_i$$

$$\mathrm{x}_i \times \mathrm{P}\mathrm{X}_i = 0$$

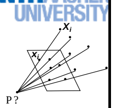
$$\begin{bmatrix} 0 & -\mathrm{X}_i^T & y_i\mathrm{X}_i^T \\ \mathrm{X}_i^T & 0 & -x_i\mathrm{X}_i^T \\ -y_i\mathrm{X}_i^T & x_i\mathrm{X}_i^T & 0 \end{bmatrix} \begin{pmatrix} \mathrm{P}_1 \\ \mathrm{P}_2 \\ \mathrm{P}_3 \end{pmatrix} = 0$$

Only two linearly independent equations



Slide adapted from Svetlana Lazebnik

Camera Calibration: DLT Algorithm



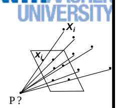
$$\begin{bmatrix} 0^T & \mathrm{X}_1^T & -y_1\mathrm{X}_1^T \\ \mathrm{X}_1^T & 0^T & -x_1\mathrm{X}_1^T \\ \cdots & \cdots & \cdots \\ 0^T & \mathrm{X}_n^T & -y_n\mathrm{X}_n^T \\ \mathrm{X}_n^T & 0^T & -x_n\mathrm{X}_n^T \end{bmatrix} \begin{pmatrix} \mathrm{P}_1 \\ \mathrm{P}_2 \\ \mathrm{P}_3 \end{pmatrix} = 0 \qquad \mathrm{Ap} = 0$$

Solve using... SVD!

- Notes
  - P has 11 degrees of freedom (12 parameters, but scale is arbitrary).
  - One 2D/3D correspondence gives us two linearly independent equations.
  - Homogeneous least squares (similar to homography est.)
  - ⇒ 5 ½ correspondences needed for a minimal solution.

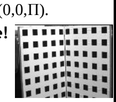
Slide adapted from Svetlana Lazebnik

Camera Calibration: DLT Algorithm



$$\begin{bmatrix} 0^T & \mathrm{X}_1^T & -y_1\mathrm{X}_1^T \\ \mathrm{X}_1^T & 0^T & -x_1\mathrm{X}_1^T \\ \cdots & \cdots & \cdots \\ 0^T & \mathrm{X}_n^T & -y_n\mathrm{X}_n^T \\ \mathrm{X}_n^T & 0^T & -x_n\mathrm{X}_n^T \end{bmatrix} \begin{pmatrix} \mathrm{P}_1 \\ \mathrm{P}_2 \\ \mathrm{P}_3 \end{pmatrix} = 0 \qquad \mathrm{Ap} = 0$$

- Notes
  - For coplanar points that satisfy $\Pi^T\mathrm{X}=0$, we will get degenerate solutions $(\Pi,0,0)$, $(0,\Pi,0)$, or $(0,0,\Pi)$.
  - ⇒ We need calibration points in more than one plane!



Slide credit: Svetlana Lazebnik

Camera Calibration

- Once we've recovered the numerical form of the camera matrix, we still have to figure out the intrinsic and extrinsic parameters
- This is a matrix decomposition problem, not an estimation problem (see F&P sec. 3.2, 3.3)

Slide credit: Svetlana Lazebnik

Camera Calibration: Some Practical Tips

- For numerical reasons, it is important to carry out some data normalization.
  - Translate the image points x_i to the (image) origin and scale them such that their RMS distance to the origin is $\sqrt{2}$.
  - Translate the 3D points X_i to the (world) origin and scale them such that their RMS distance to the origin is $\sqrt{3}$.
  - (This is valid for compact point distributions on calibration objects).
- The DLT algorithm presented here is easy to implement, but there are some more accurate algorithms available (see H&Z sec. 7.2).
- For practical applications, it is also often needed to correct for radial distortion. Algorithms for this can be found in H&Z sec. 7.4, or F&P sec. 3.3.

Topics of This Lecture

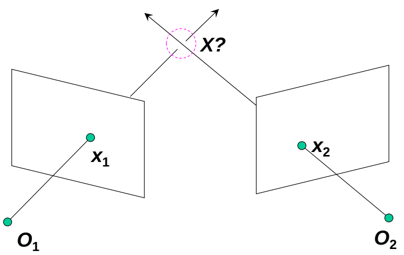
- Camera Calibration
  - Camera parameters
  - Calibration procedure
- Revisiting Epipolar Geometry
  - Triangulation
  - Calibrated case: Essential matrix
  - Uncalibrated case: Fundamental matrix
  - Weak calibration
  - Epipolar Transfer
- Active Stereo
  - Laser scanning
  - Kinect sensor

Two-View Geometry

- Scene geometry (structure):
  - Given corresponding points in two or more images, where is the pre-image of these points in 3D?
- Correspondence (stereo matching):
  - Given a point in just one image, how does it constrain the position of the corresponding point x' in another image?
- Camera geometry (motion):
  - Given a set of corresponding points in two images, what are the cameras for the two views?

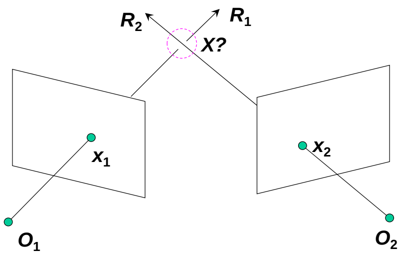
Revisiting Triangulation

- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point



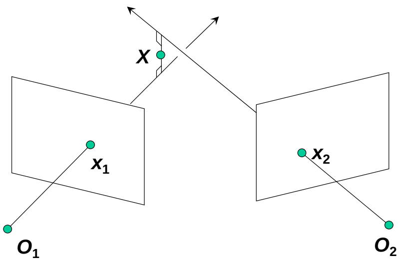
Revisiting Triangulation

- We want to intersect the two visual rays corresponding to x_1 and x_2, but because of noise and numerical errors, they will never meet exactly. How can this be done?



Triangulation: 1) Geometric Approach

- Find shortest segment connecting the two viewing rays and let X be the midpoint of that segment.



Triangulation: 2)Linear Algebraic Approach

$$\lambda_1 x_1 = P_1 X \qquad x_1 \times P_1 X = 0 \qquad [x_{1\times}] P_1 X = 0$$

$$\lambda_2 x_2 = P_2 X \qquad x_2 \times P_2 X = 0 \qquad [x_{2\times}] P_2 X = 0$$

Cross product as matrix multiplication:

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_\times]\mathbf{b}$$

Triangulation: 2) Linear Algebraic Approach

$$\lambda_1 x_1 = P_1 X \qquad x_1 \times P_1 X = 0 \qquad [x_{1\times}]P_1 X = 0$$
$$\lambda_2 x_2 = P_2 X \qquad x_2 \times P_2 X = 0 \qquad [x_{2\times}]P_2 X = 0$$

Two independent equations each in terms of three unknown entries of X

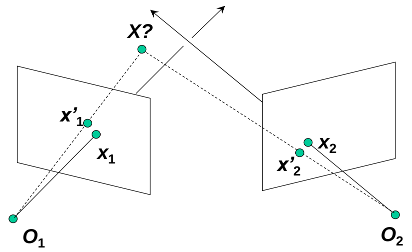
⇒ Stack them and solve using SVD!

- This approach is often preferable to the geometric approach, since it nicely generalizes to multiple cameras.

Triangulation: 3) Nonlinear Approach

- Find X that minimizes

$$d^2(x_1, P_1 X) + d^2(x_2, P_2 X)$$



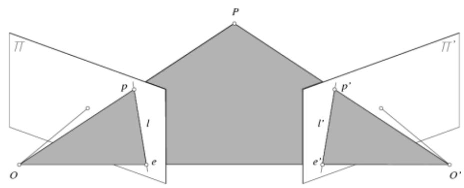
Triangulation: 3) Nonlinear Approach

- Find X that minimizes

$$d^2(x_1, P_1 X) + d^2(x_2, P_2 X)$$

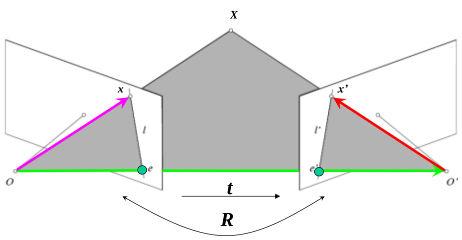
- This approach is the most accurate, but unlike the other two methods, it doesn't have a closed-form solution.
- Iterative algorithm
  - Initialize with linear estimate.
  - Optimize with Gauss-Newton or Levenberg-Marquardt (see F&P sec. 3.1.2 or H&Z Appendix 6).

Revisiting Epipolar Geometry



- Let's look again at the epipolar constraint
  - For the calibrated case (but in homogenous coordinates)
  - For the uncalibrated case

Epipolar Geometry: Calibrated Case

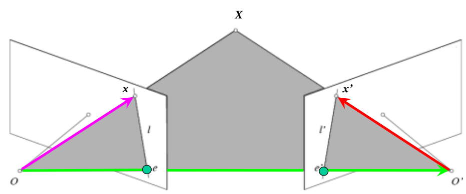


Camera matrix: $[I|0]$
$X = (u, v, w, 1)^T$
$x = (u, v, w)^T$

Camera matrix: $[R^T | -R^T t]$
Vector x' in second coord. system has coordinates Rx' in the first one.

The vectors x, t, and Rx' are coplanar

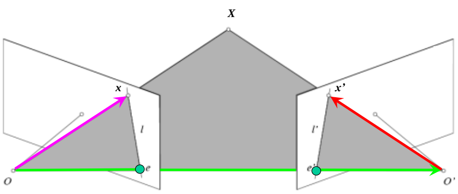
Epipolar Geometry: Calibrated Case



$$x \cdot [t \times (Rx')] = 0 \quad \Rightarrow \quad x^T E x' = 0 \quad \text{with} \quad E = [t_\times]R$$

Essential Matrix
(Longuet-Higgins, 1981)

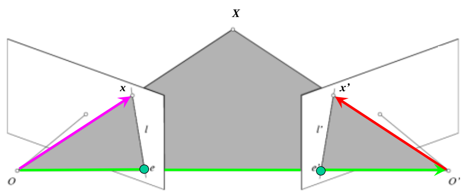
Epipolar Geometry: Calibrated Case



$$x \cdot [t \times (Rx')] = 0 \quad \Rightarrow \quad x^T E x' = 0 \quad \text{with} \quad E = [t_\times]R$$

- ***E x'*** is the epipolar line associated with ***x'*** (***l*** = ***E x'***)
- ***E***^{T}***x*** is the epipolar line associated with ***x*** (***l'*** = ***E***^{T}***x***)
- ***E e'* = 0** and ***E***^{T}***e* = 0**
- ***E*** is singular (rank two)
- ***E*** has five degrees of freedom (up to scale)

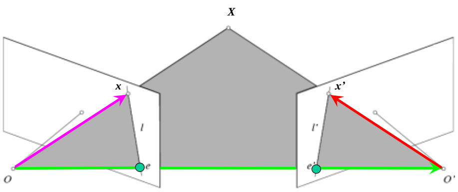
Epipolar Geometry: Uncalibrated Case



- The calibration matrices *K* and *K'* of the two cameras are unknown
- We can write the epipolar constraint in terms of *unknown* normalized coordinates:

$$\hat{x}^T E \hat{x}' = 0 \qquad x = K\hat{x}, \quad x' = K'\hat{x}'$$

Epipolar Geometry: Uncalibrated Case



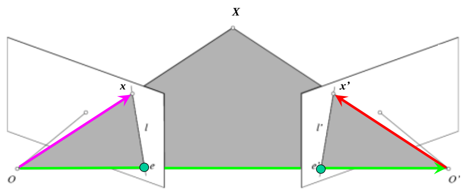
$$\hat{x}^T E \hat{x}' = 0 \quad \Rightarrow \quad x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

$$x = K\hat{x}$$

$$x' = K'\hat{x}'$$

Fundamental Matrix
(Faugeras and Luong, 1992)

Epipolar Geometry: Uncalibrated Case

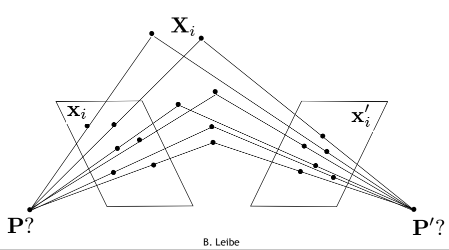


$$\hat{x}^T E \hat{x}' = 0 \quad \Rightarrow \quad x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

- ***F x'*** is the epipolar line associated with ***x'*** (***l*** = ***F x'***)
- ***F***^{T}***x*** is the epipolar line associated with ***x*** (***l'*** = ***F***^{T}***x***)
- ***F e'* = 0** and ***F***^{T}***e* = 0**
- ***F*** is singular (rank two)
- ***F*** has seven degrees of freedom

Estimating the Fundamental Matrix

- The Fundamental matrix defines the epipolar geometry between two uncalibrated cameras.
- How can we estimate F from an image pair?
  - We need correspondences…



The Eight-Point Algorithm

$x = (u, v, 1)^T, \quad x' = (u', v', 1)^T$

$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \quad \Rightarrow \quad [u'u, u'v, u', uv', vv', v', u, v, 1]\begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

- **Taking 8 correspondences:**

$$\begin{bmatrix} u'_1u_1 & u'_1v_1 & u'_1 & u_1v'_1 & v_1v'_1 & v'_1 & u_1 & v_1 & 1 \\ u'_2u_2 & u'_2v_2 & u'_2 & u_2v'_2 & v_2v'_2 & v'_2 & u_2 & v_2 & 1 \\ u'_3u_3 & u'_3v_3 & u'_3 & u_3v'_3 & v_3v'_3 & v'_3 & u_3 & v_3 & 1 \\ u'_4u_4 & u'_4v_4 & u'_4 & u_4v'_4 & v_4v'_4 & v'_4 & u_4 & v_4 & 1 \\ u'_5u_5 & u'_5v_5 & u'_5 & u_5v'_5 & v_5v'_5 & v'_5 & u_5 & v_5 & 1 \\ u'_6u_6 & u'_6v_6 & u'_6 & u_6v'_6 & v_6v'_6 & v'_6 & u_6 & v_6 & 1 \\ u'_7u_7 & u'_7v_7 & u'_7 & u_7v'_7 & v_7v'_7 & v'_7 & u_7 & v_7 & 1 \\ u'_8u_8 & u'_8v_8 & u'_8 & u_8v'_8 & v_8v'_8 & v'_8 & u_8 & v_8 & 1 \end{bmatrix}\begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{Af} = \mathbf{0}$$

Solve using… SVD!

This minimizes:

$$\sum_{i=1}^{N} (x_i^T F x'_i)^2$$

Excursion: Properties of SVD

- **Frobenius norm**
  - Generalization of the Euclidean norm to matrices

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\sum_{i=1}^{\min(m,n)} \sigma_i^2}$$

- **Partial reconstruction property of SVD**
  - Let σ_i $i=1,\ldots,N$ be the singular values of A.
  - Let $A_p = U_p D_p V_p^T$ be the reconstruction of A when we set $\sigma_{p+1}, \ldots, \sigma_N$ to zero.
  - Then $A_p = U_p D_p V_p^T$ is the best rank-p approximation of A in the sense of the Frobenius norm (i.e. the best least-squares approximation).

The Eight-Point Algorithm

- **Problem with noisy data**
  - The solution will usually not fulfill the constraint that F only has rank 2.
  - ⇒ *There will be no epipoles through which all epipolar lines pass!*
- **Enforce the rank-2 constraint using SVD**

$$F \overset{\text{SVD}}{=} UDV^T = U \begin{bmatrix} d_{11} & & \\ & d_{22} & \\ & & d_{33} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{13} \\ \vdots & \ddots & \vdots \\ v_{31} & \cdots & v_{33} \end{bmatrix}^T$$

Set d_{33} to zero and reconstruct F

- As we have just seen, this provides the best least-squares approximation to the rank-2 solution.

Problem with the Eight-Point Algorithm

- In practice, this often looks as follows:

$$\begin{bmatrix} u'_1u_1 & u'_1v_1 & u'_1 & u_1v'_1 & v_1v'_1 & v'_1 & u_1 & v_1 & 1 \\ u'_2u_2 & u'_2v_2 & u'_2 & u_2v'_2 & v_2v'_2 & v'_2 & u_2 & v_2 & 1 \\ u'_3u_3 & u'_3v_3 & u'_3 & u_3v'_3 & v_3v'_3 & v'_3 & u_3 & v_3 & 1 \\ u'_4u_4 & u'_4v_4 & u'_4 & u_4v'_4 & v_4v'_4 & v'_4 & u_4 & v_4 & 1 \\ u'_5u_5 & u'_5v_5 & u'_5 & u_5v'_5 & v_5v'_5 & v'_5 & u_5 & v_5 & 1 \\ u'_6u_6 & u'_6v_6 & u'_6 & u_6v'_6 & v_6v'_6 & v'_6 & u_6 & v_6 & 1 \\ u'_7u_7 & u'_7v_7 & u'_7 & u_7v'_7 & v_7v'_7 & v'_7 & u_7 & v_7 & 1 \\ u'_8u_8 & u'_8v_8 & u'_8 & u_8v'_8 & v_8v'_8 & v'_8 & u_8 & v_8 & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Slide adapted from Svetlana Lazebnik

Problem with the Eight-Point Algorithm

- In practice, this often looks as follows:

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81	1
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79	1
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81	1
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65	1
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15	1
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14	1
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64	1
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48	1

$$\begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

⇒ Poor numerical conditioning

⇒ Can be fixed by rescaling the data

Slide adapted from Svetlana Lazebnik

The Normalized Eight-Point Algorithm

1. Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels.
2. Use the eight-point algorithm to compute *F* from the normalized points.
3. Enforce the rank-2 constraint using SVD.

$$F \overset{\text{SVD}}{=} UDV^T = U \begin{bmatrix} d_{11} & & \\ & d_{22} & \\ & & d_{33} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{13} \\ \vdots & \ddots & \vdots \\ v_{31} & \cdots & v_{33} \end{bmatrix}^T$$

Set d_{33} to zero and reconstruct F

4. Transform fundamental matrix back to original units: if T and T' are the normalizing transformations in the two images, than the fundamental matrix in original coordinates is $T^T\, F\, T'$.

Slide credit: Svetlana Lazebnik

[Hartley, 1995]

The Eight-Point Algorithm

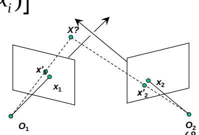
- **Meaning of error** $\sum_{i=1}^{N}(x_i^T F x'_i)^2$:

  Sum of Euclidean distances between points x_i and epipolar lines Fx'_i (or points x'_i and epipolar lines $F^T x_i$), multiplied by a scale factor

- **Nonlinear approach: minimize**

$$\sum_{i=1}^{N}\left[d^2(x_i, F x'_i) + d^2(x'_i, F^T x_i)\right]$$

  - Similar to nonlinear minimization approach for triangulation.
  - Iterative approach (Gauss-Newton, Levenberg-Marquardt,…)



Slide credit: Svetlana Lazebnik

Comparison of Estimation Algorithms

	8-point	Normalized 8-point	Nonlinear least squares
Av. Dist. 1	2.33 pixels	0.92 pixel	0.86 pixel
Av. Dist. 2	2.18 pixels	0.85 pixel	0.80 pixel

3D Reconstruction with Weak Calibration

- Want to estimate world geometry without requiring calibrated cameras.
- Many applications:
  - Archival videos
  - Photos from multiple unrelated users
  - Dynamic camera system
- Main idea:
  - Estimate epipolar geometry from a (redundant) set of point correspondences between two uncalibrated cameras.

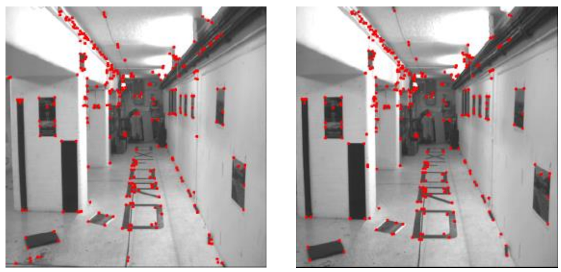
Stereo Pipeline with Weak Calibration

- So, where to start with uncalibrated cameras?
  - Need to find fundamental matrix F *and* the correspondences (pairs of points (u',v') ↔ (u,v)).

- Procedure
  1. Find interest points in both images
  2. Compute correspondences
  3. Compute epipolar geometry
  4. Refine

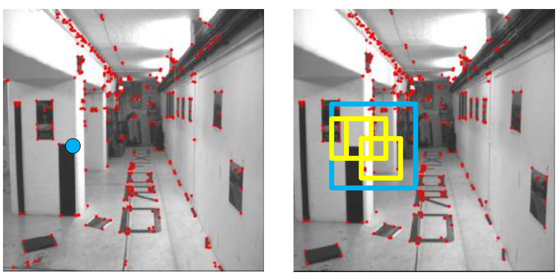
Stereo Pipeline with Weak Calibration

1. Find interest points (e.g. Harris corners)

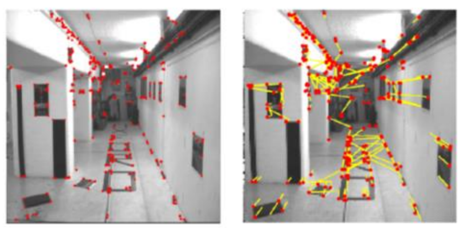


Stereo Pipeline with Weak Calibration

2. Match points using only proximity



Putative Matches based on Correlation Search

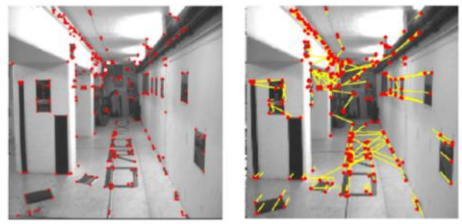


- Many wrong matches (10-50%), but enough to compute F

RANSAC for Robust Estimation of F

- Select random sample of correspondences
- Compute F using them
  - This determines epipolar constraint
- Evaluate amount of support – number of inliers within threshold distance of epipolar line
- Choose F with most support (#inliers)

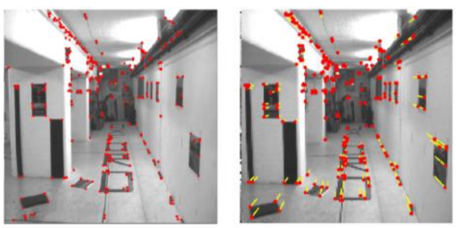
Putative Matches based on Correlation Search



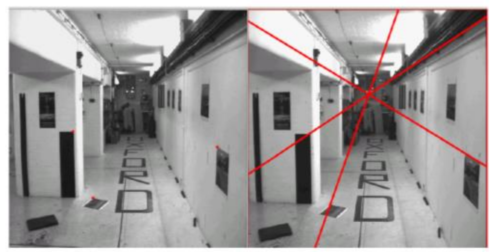
• Many wrong matches (10-50%), but enough to compute F

Pruned Matches

- Correspondences consistent with epipolar geometry

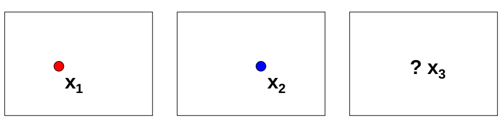


Resulting Epipolar Geometry



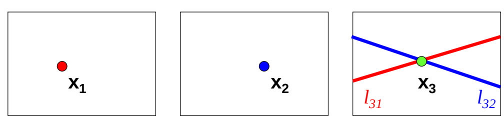
Epipolar Transfer

- Assume the epipolar geometry is known
- Given projections of the same point in two images, how can we compute the projection of that point in a third image?



Extension: Epipolar Transfer

- Assume the epipolar geometry is known
- Given projections of the same point in two images, how can we compute the projection of that point in a third image?



$$l_{31} = F^T_{13} x_1$$
$$l_{32} = F^T_{23} x_2$$

When does epipolar transfer fail?

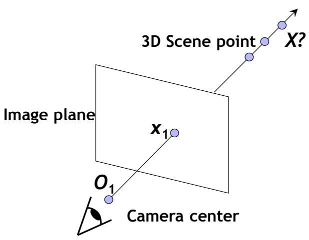
Topics of This Lecture

- Camera Calibration
  - Camera parameters
  - Calibration procedure
- Revisiting Epipolar Geometry
  - Triangulation
  - Calibrated case: Essential matrix
  - Uncalibrated case: Fundamental matrix
  - Weak calibration
  - Epipolar Transfer
- Active Stereo
  - Laser scanning
  - Kinect sensor

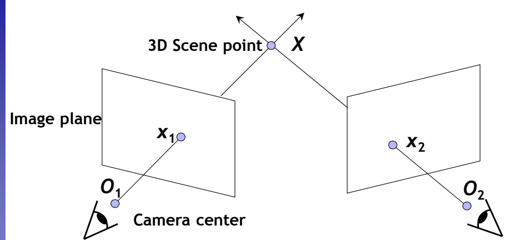
Microsoft Kinect - How Does It Work?

- Built-in IR projector
- IR camera for depth
- Regular camera for color

Recall: Optical Triangulation

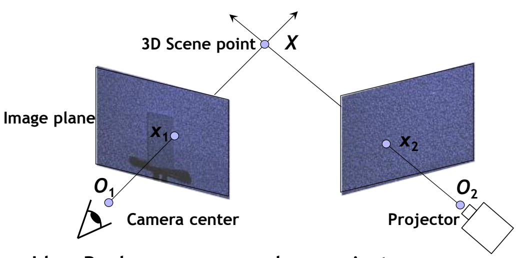


Recall: Optical Triangulation



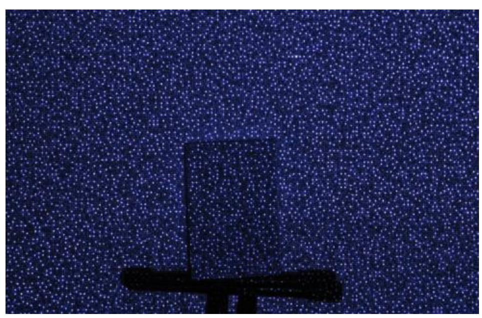
- Principle: 3D point given by intersection of two rays.
  - Crucial information: point correspondence
  - Most expensive and error-prone step in the pipeline…

Active Stereo with Structured Light



- Idea: Replace one camera by a projector.
  - Project "structured" light patterns onto the object
  - Simplifies the correspondence problem

What the Kinect Sees…



3D Reconstruction with the Kinect


SIGGRAPH Talks 2011

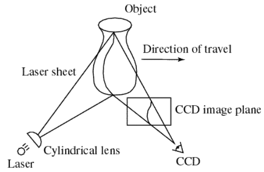
KinectFusion:

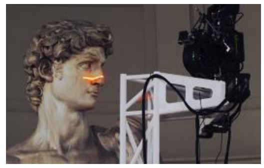
Real-Time Dynamic 3D Surface Reconstruction and Interaction

Shahram Izadi 1, Richard Newcombe 2, David Kim 1,3, Otmar Hilliges 1, David Molyneaux 1,4, Pushmeet Kohli 1, Jamie Shotton 1, Steve Hodges 1, Dustin Freeman 5, Andrew Davison 2, Andrew Fitzgibbon 1

1 Microsoft Research Cambridge 2 Imperial College London 3 Newcastle University 4 Lancaster University 5 University of Toronto

Laser Scanning





Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/

- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

Slide credit: Steve Seitz

Laser Scanned Models

The Digital Michelangelo Project, Levoy et al.

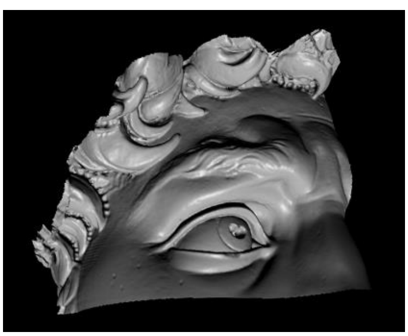
Slide credit: Steve Seitz

Laser Scanned Models

The Digital Michelangelo Project, Levoy et al.

Slide credit: Steve Seitz

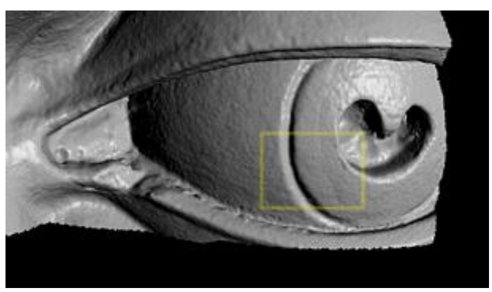
Laser Scanned Models



The Digital Michelangelo Project, Levoy et al.

Slide credit: Steve Seitz

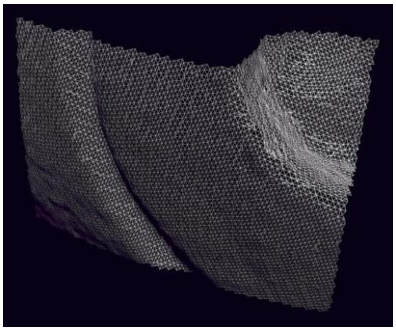
Laser Scanned Models



The Digital Michelangelo Project, Levoy et al.

Slide credit: Steve Seitz

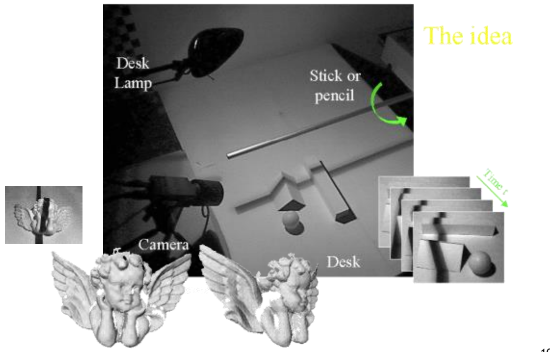
Laser Scanned Models



The Digital Michelangelo Project, Levoy et al.

Slide credit: Steve Seitz

Poor Man's Scanner



Bouget and Perona, ICCV'98

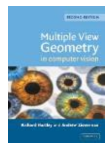
Slightly More Elaborate (But Still Cheap)

Software freely available from Robotics Institute TU Braunschweig
http://www.david-laserscanner.com/

References and Further Reading

- Background information on camera models and calibration algorithms can be found in Chapters 6 and 7 of

  R. Hartley, A. Zisserman
  Multiple View Geometry in Computer Vision
  2nd Ed., Cambridge Univ. Press, 2004



- Also recommended: Chapter 9 of the same book on Epipolar geometry and the Fundamental Matrix and Chapter 11.1-11.6 on automatic computation of F.