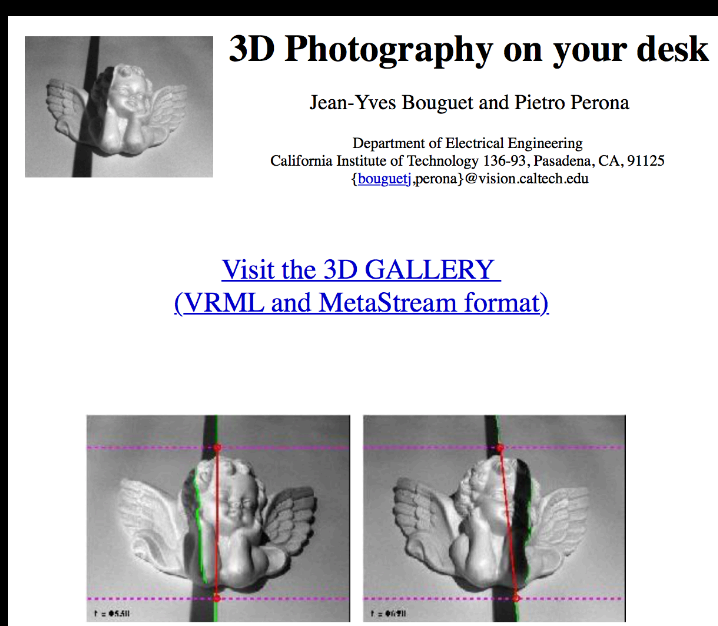
Spring 2016 ENGN2502 --- 3D Photography
Lecture 7

Gabriel Taubin
Brown University

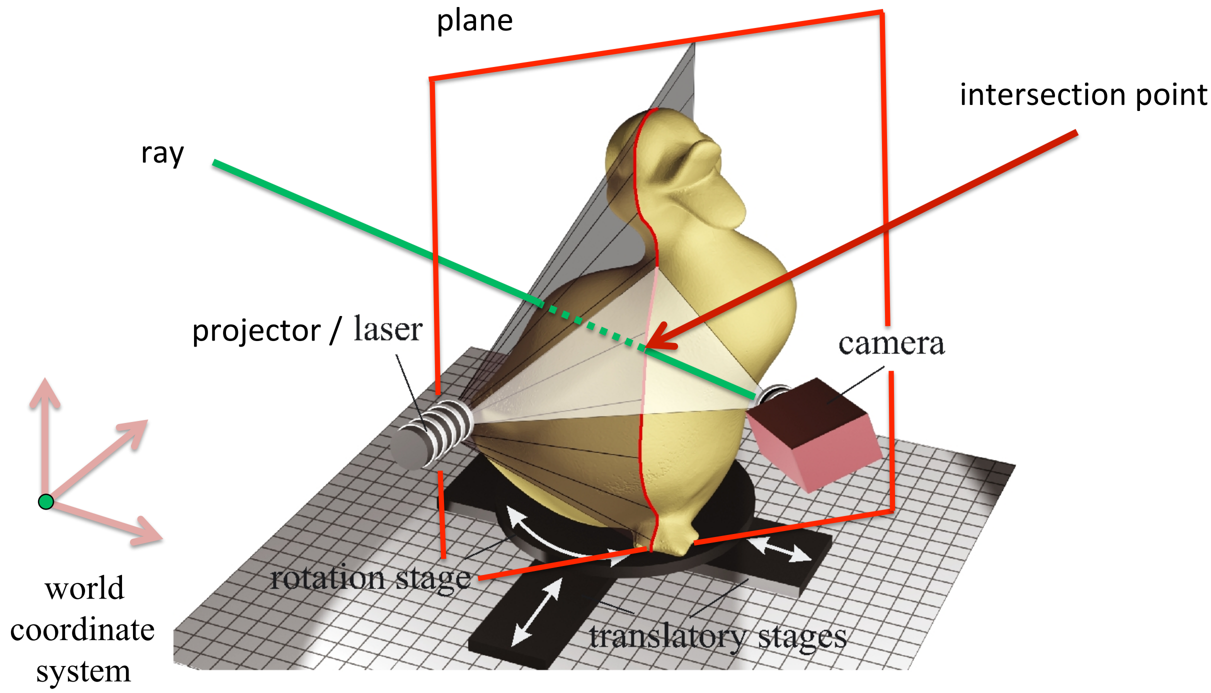
BROWN

Desktop 3D Photography

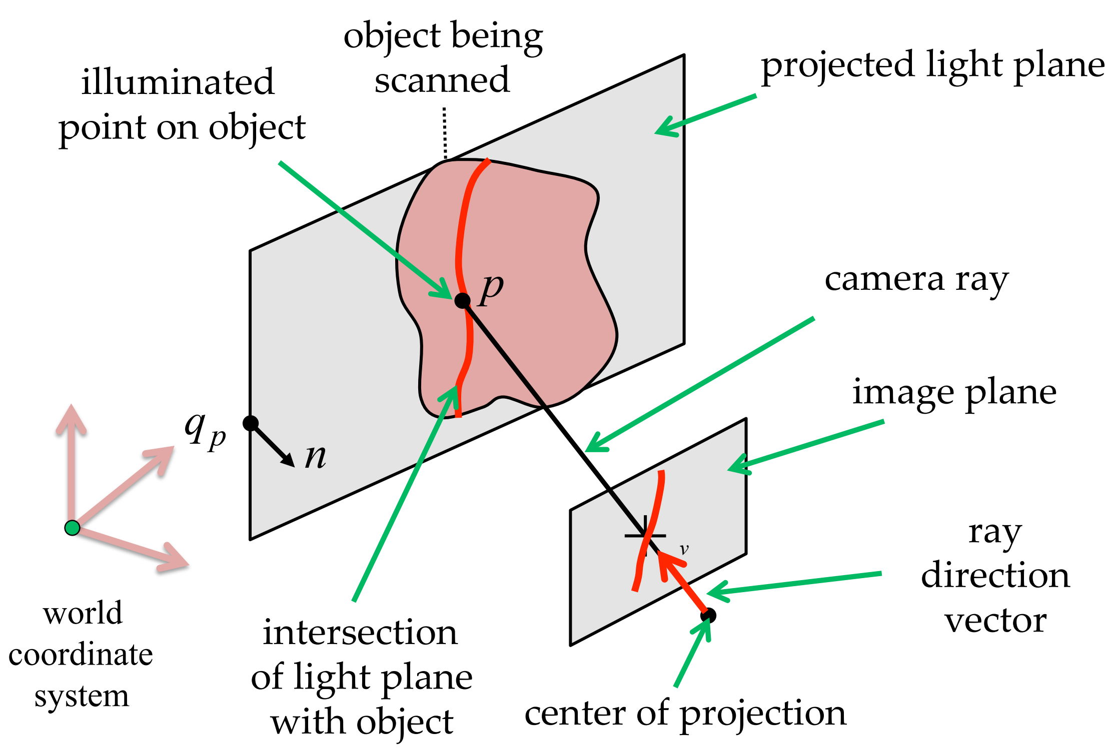
- http://www.vision.caltech.edu/bouguetj/ICCV98/.index.html



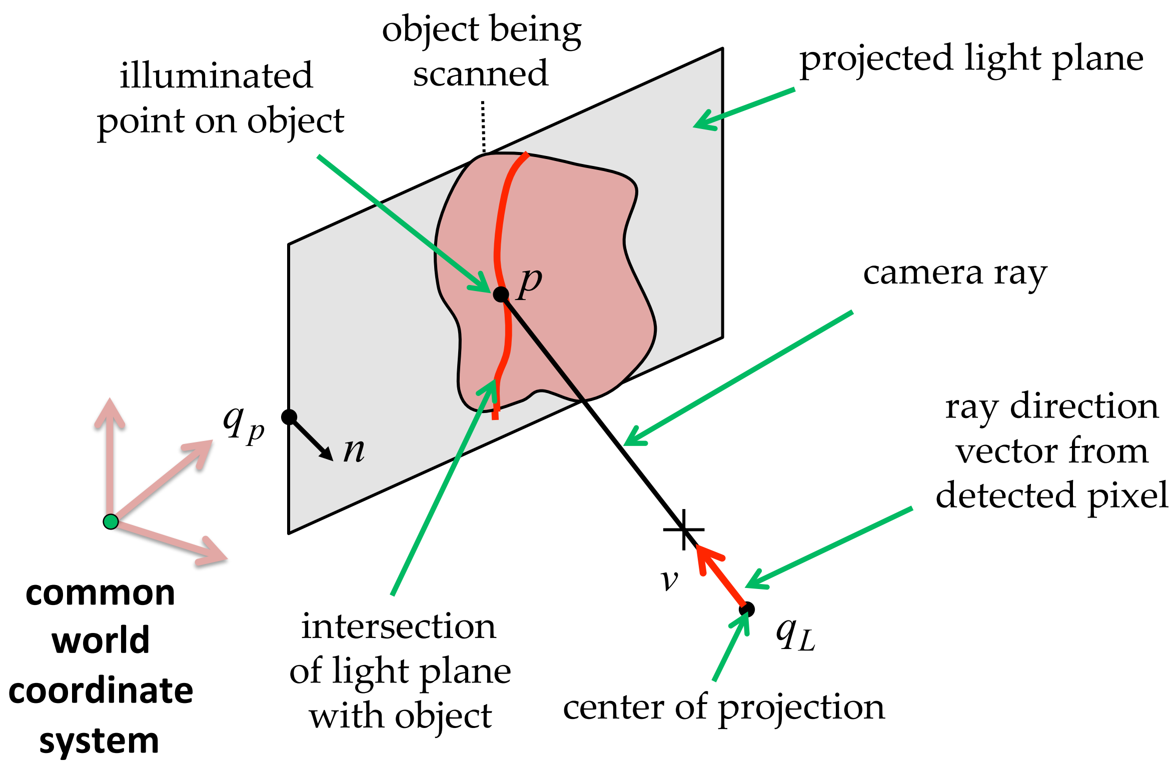
3D triangulation: ray-plane Intersection



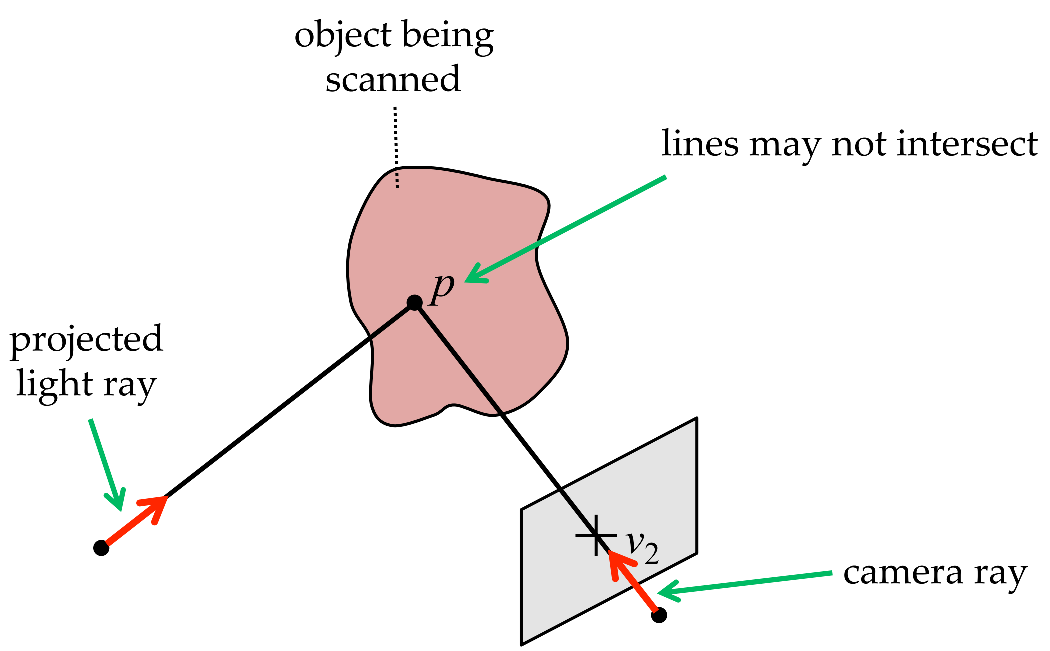
3D Triangulation by Ray-Plane intersection



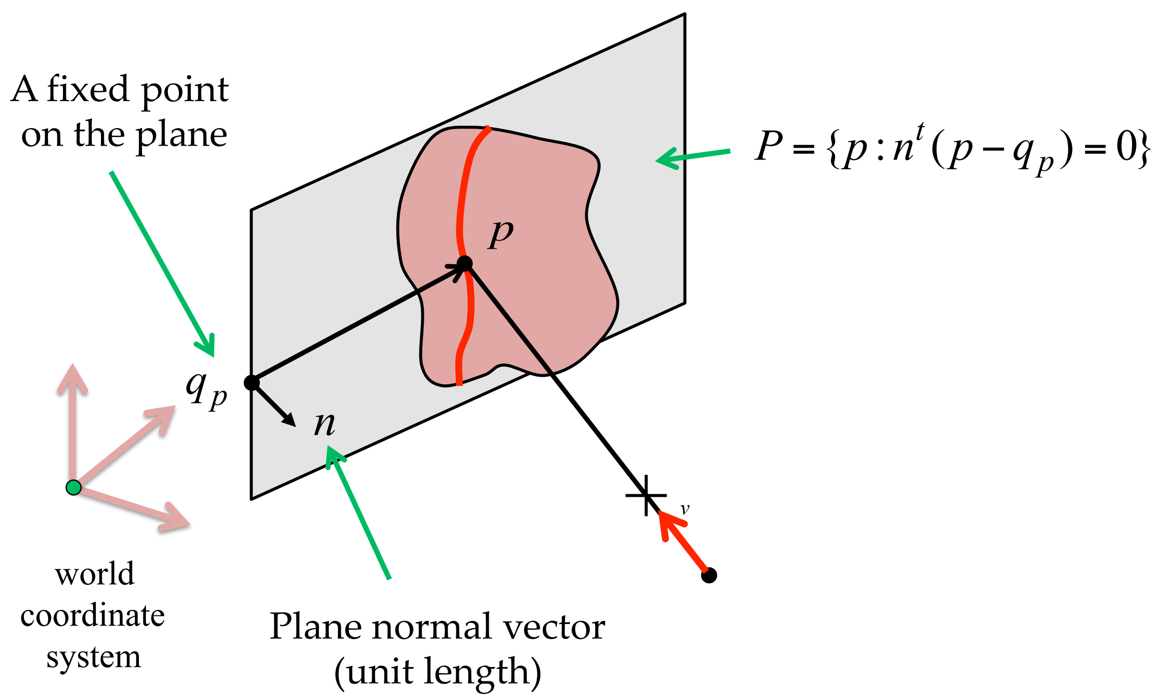
If camera and projector are calibrated



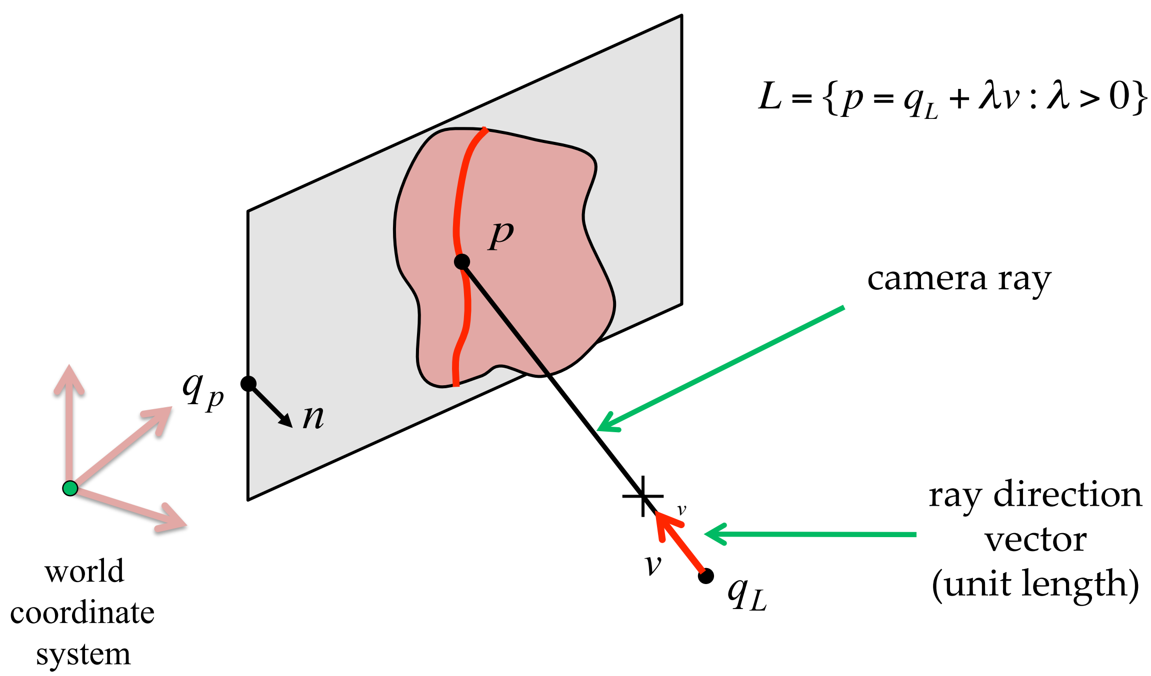
3D Triangulation by Ray-Ray Intersection



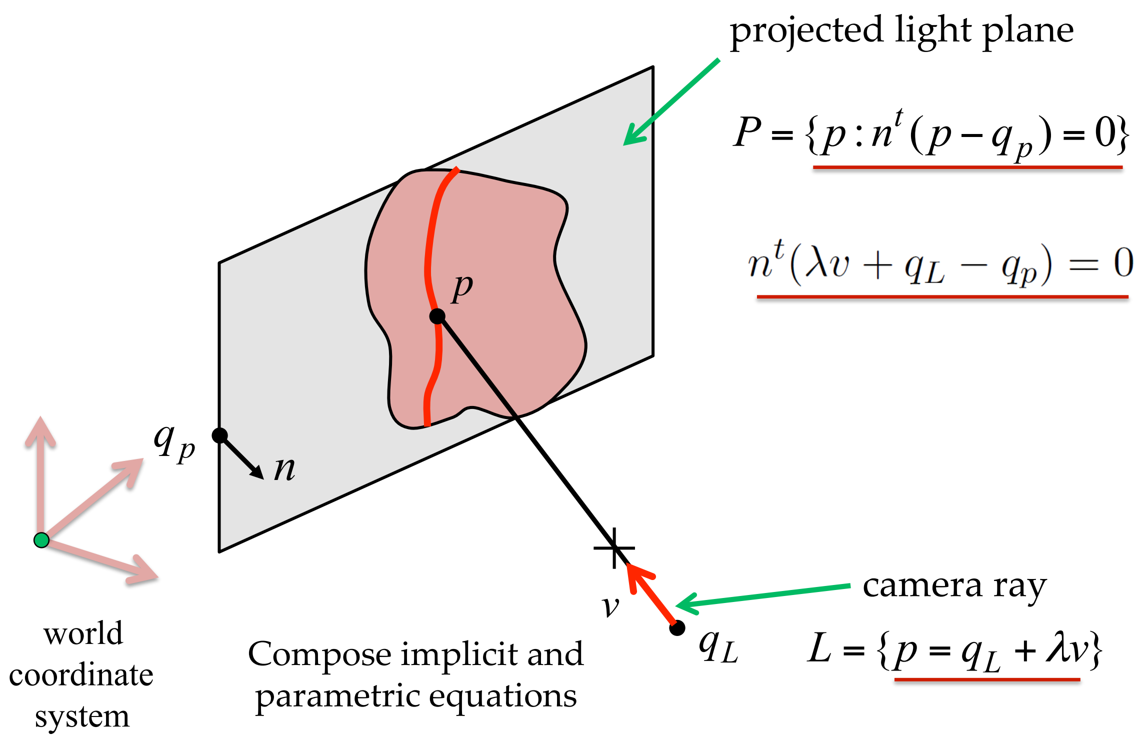
Implicit equation of the plane



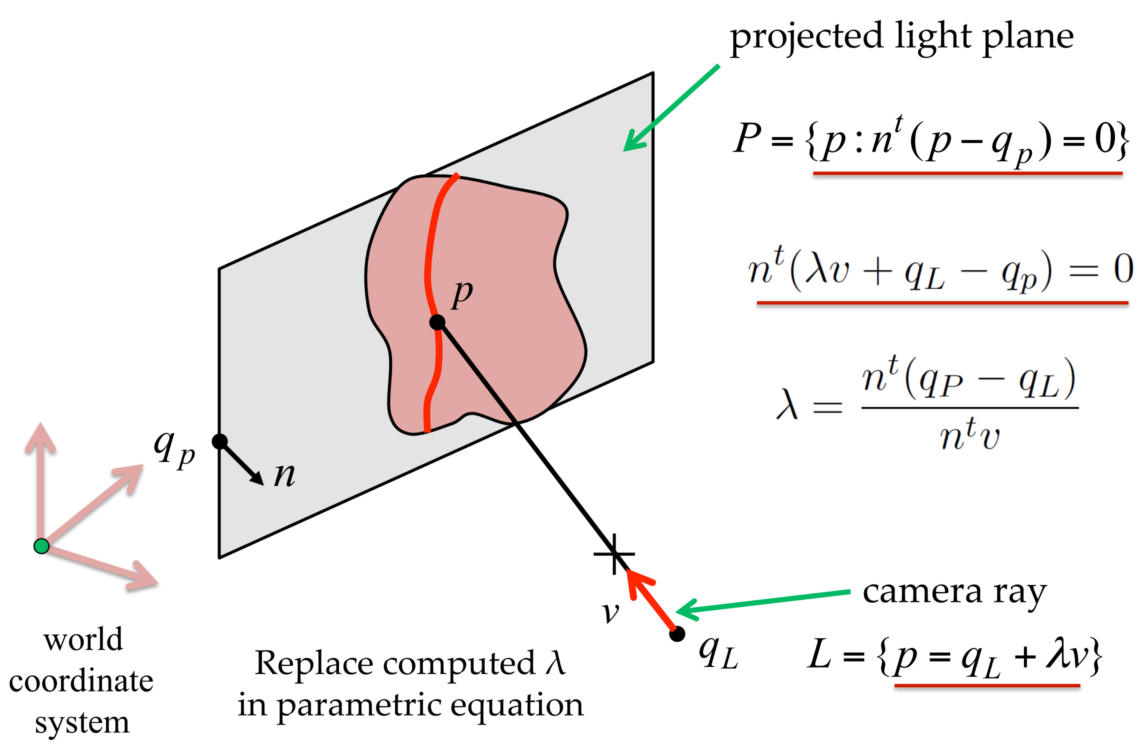
Parametric equation of the ray



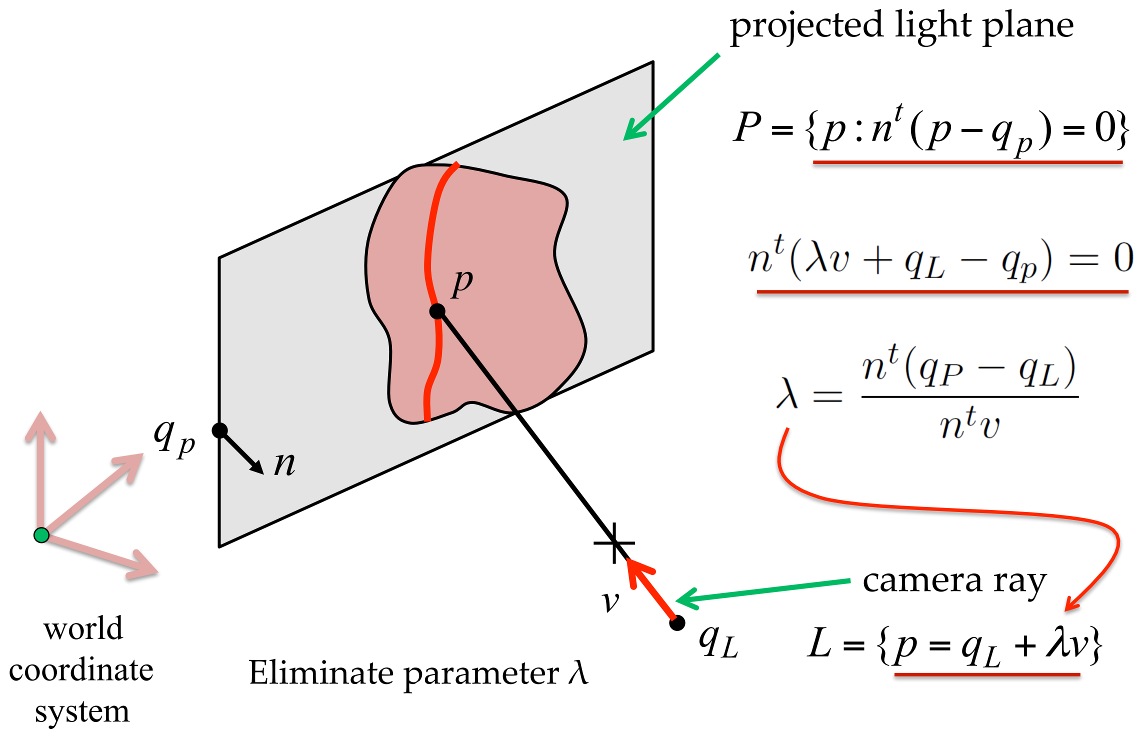
Triangulation by Line-Plane Intersection



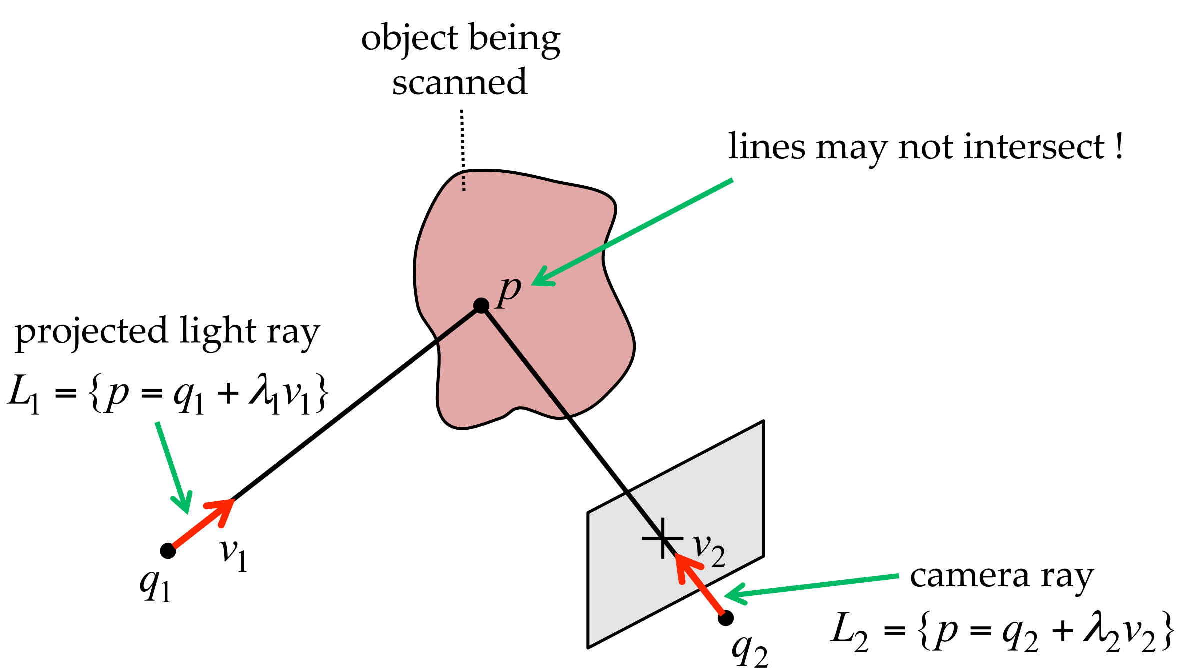
Triangulation by Line-Plane Intersection



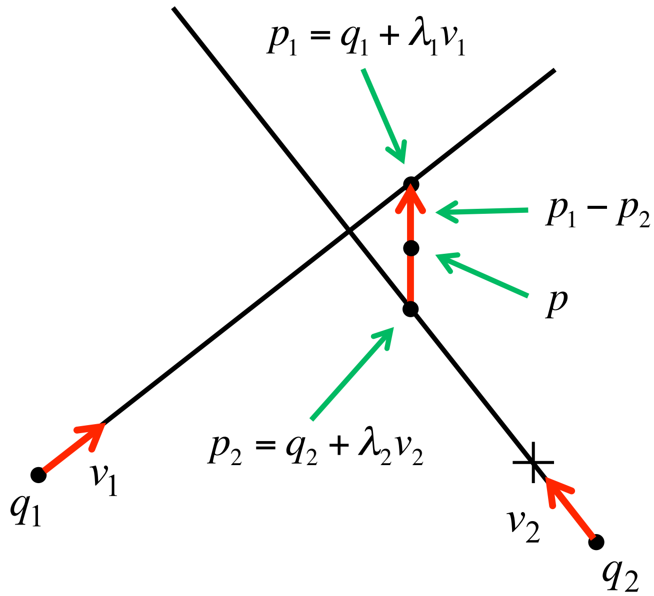
Triangulation by Line-Plane Intersection



Triangulation by Line-Line Intersection



Triangulation by Line-Line Intersection



$$L_1 = \{p_1 = q_1 + \lambda_1 v_1\}$$
$$L_2 = \{p_2 = q_2 + \lambda_2 v_2\}$$

Minimize

$$E(\lambda_1, \lambda_2) = dist(p_2 - p_1)^2$$

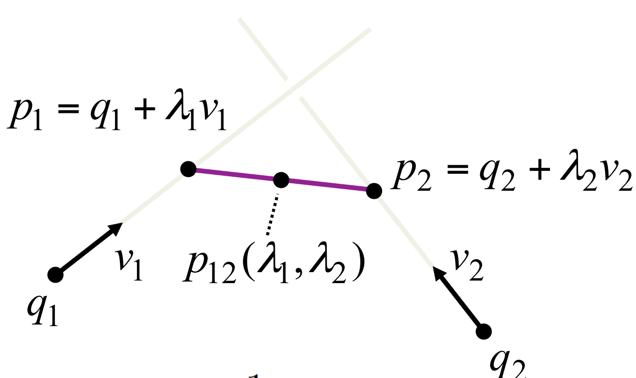
Necessary conditions

$$v_1^t (p_1 - p_2) = 0$$
$$v_2^t (p_2 - p_1) = 0$$

$$p = (p_1 + p_2)/2$$

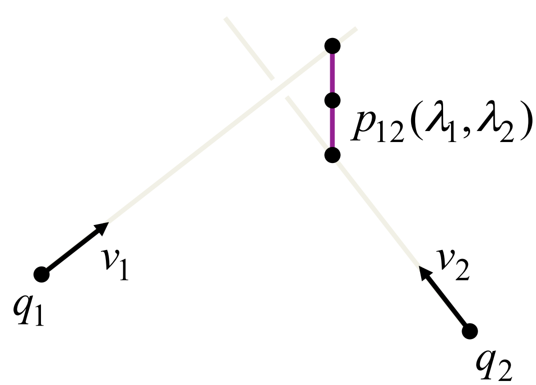
Approximate Line-Line Intersection

Midpoint of segment joining arbitrary points in the two lines



$$p_{12} = p_1 + \frac{1}{2}(p_2 - p_1)$$

Least-squares approach



Find parameters which minimize

$$\|(q_2 + \lambda_2 v_2) - (q_1 + \lambda_1 v_1)\|^2$$

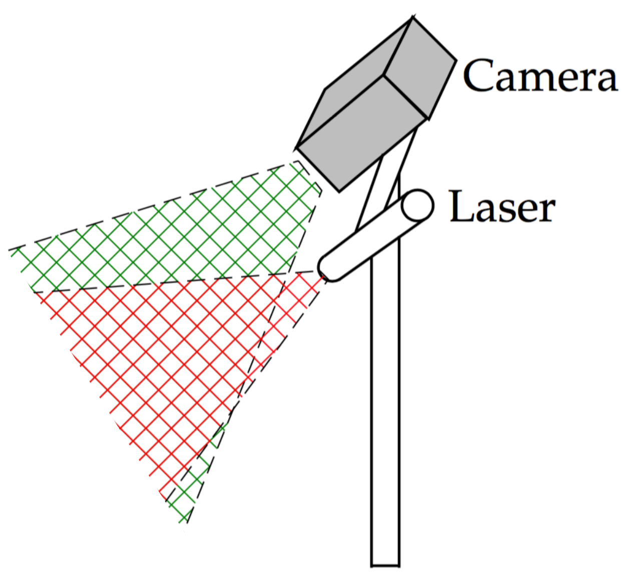
Approximate Line-Line Intersection

$$p_{12} = p_1 + \frac{1}{2}(p_2 - p_1)$$

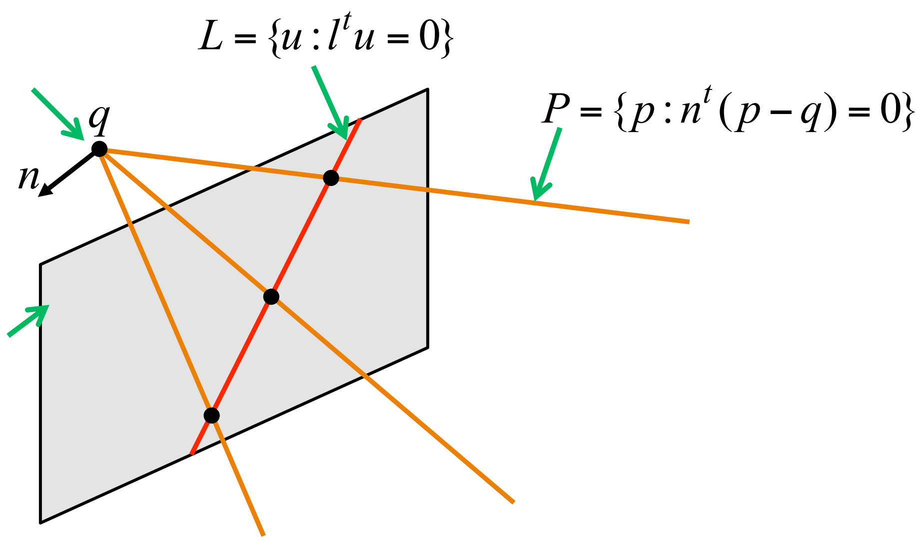
$$p_1 = q_1 + \lambda_1 v_1$$

$$p_2 = q_2 + \lambda_2 v_2$$

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \|v_1\|^2 & -v_1^t v_2 \\ -v_2^t v_1 & \|v_2\|^2 \end{pmatrix}^{-1} \begin{pmatrix} v_1^t(q_2 - q_1) \\ v_2^t(q_1 - q_2) \end{pmatrix}$$



Plane defined by image line and center of projection



$$0 = \lambda l^t u = l^t(R\, p_W + T) = (R^t l)^t\,(p_W - (-R^t T))\;.$$

Triangulation by Laser Striping

- Manually or mechanically translated laser stripe
- Per-pixel depth by ray-plane triangulation
- Requires accurate camera and laser plane calibration
- Popular solution for commercial and DIY 3D scanners

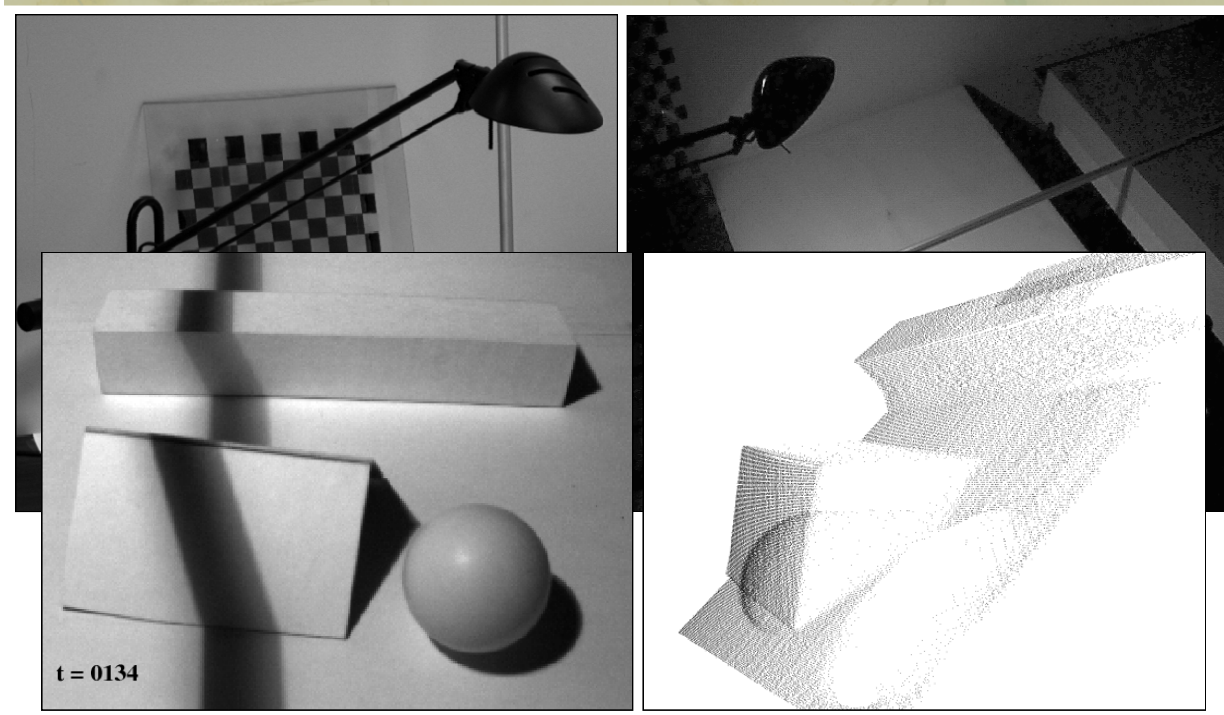
M. J. Leotta, A. Vandergon, and G. Taubin. 3D Slit Scanning With Planar Constraints. *Computer Graphics Forum*, 2008

3D Photography on Your Desk:
Bouguet and Perona [ICCV 1998]

- DIY scanner using only a camera, a halogen lamp, and a stick
- Per-pixel depth by ray-plane triangulation
- Requires accurate camera and shadow plane calibration

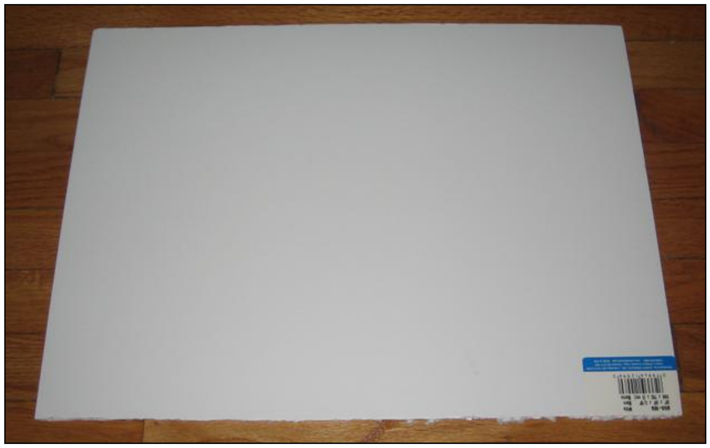
J.-Y. Bouguet and P. Perona. 3D photography on your desk.
Intl. Conf. Comp. Vision, 1998

3D Photography on Your Desk:
Bouguet and Perona [ICCV 1998]



J.-Y. Bouguet and P. Perona. 3D photography on your desk.
Intl. Conf. Comp. Vision, 1998

Assembling Your Own Scanner

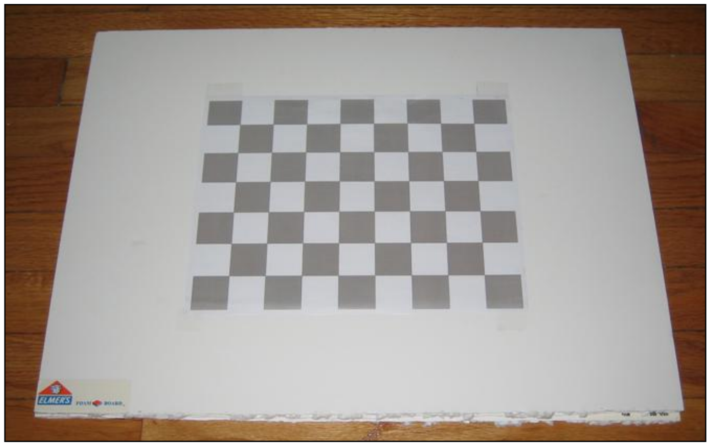


- Parts: camera (QuickCam 9000), lamp, stick, two planar objects [~$100]
- Step 1: Build the calibration boards (include fiducials and chessboard)
- Step 2: Build the point light source (remove reflector and place in scene)
- Step 3: Arrange the camera, light source, and calibration boards

Assembling Your Own Scanner

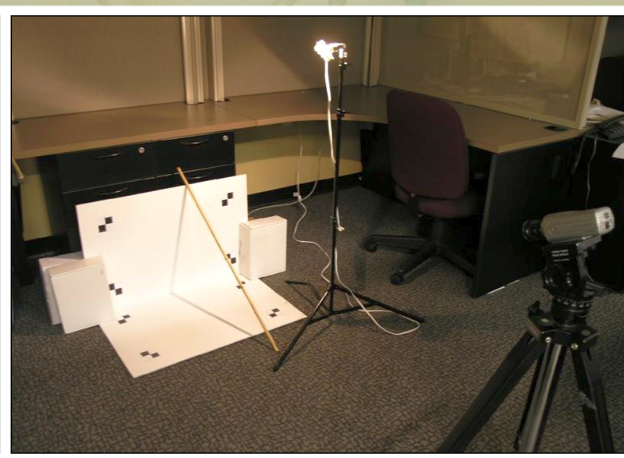
- Parts: camera (QuickCam 9000), lamp, stick, two planar objects [~$100]
- Step 1: Build the calibration boards (include fiducials and chessboard)
- Step 2: Build the point light source (remove reflector and place in scene)
- Step 3: Arrange the camera, light source, and calibration boards

Assembling Your Own Scanner



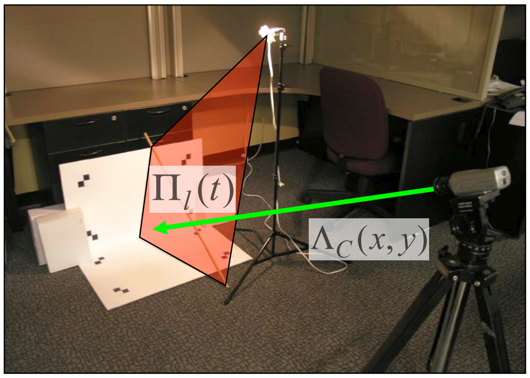
- Parts: camera (QuickCam 9000), lamp, stick, two planar objects [~$100]
- Step 1: Build the calibration boards (include fiducials and chessboard)
- Step 2: Build the point light source (remove reflector and place in scene)
- Step 3: Arrange the camera, light source, and calibration boards

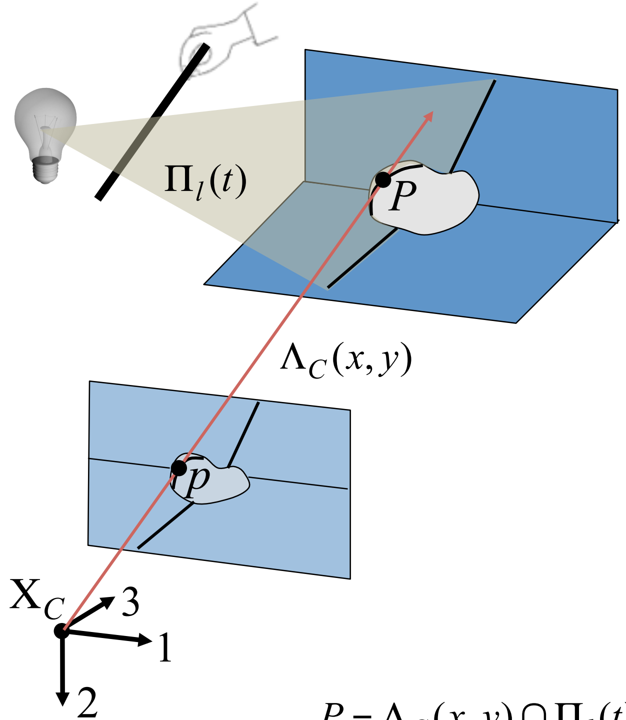
Assembling Your Own Scanner



- Parts: camera (QuickCam 9000), lamp, stick, two planar objects [~$100]
- Step 1: Build the calibration boards (include fiducials and chessboard)
- Step 2: Build the point light source (remove reflector and place in scene)
- Step 3: Arrange the camera, light source, and calibration boards

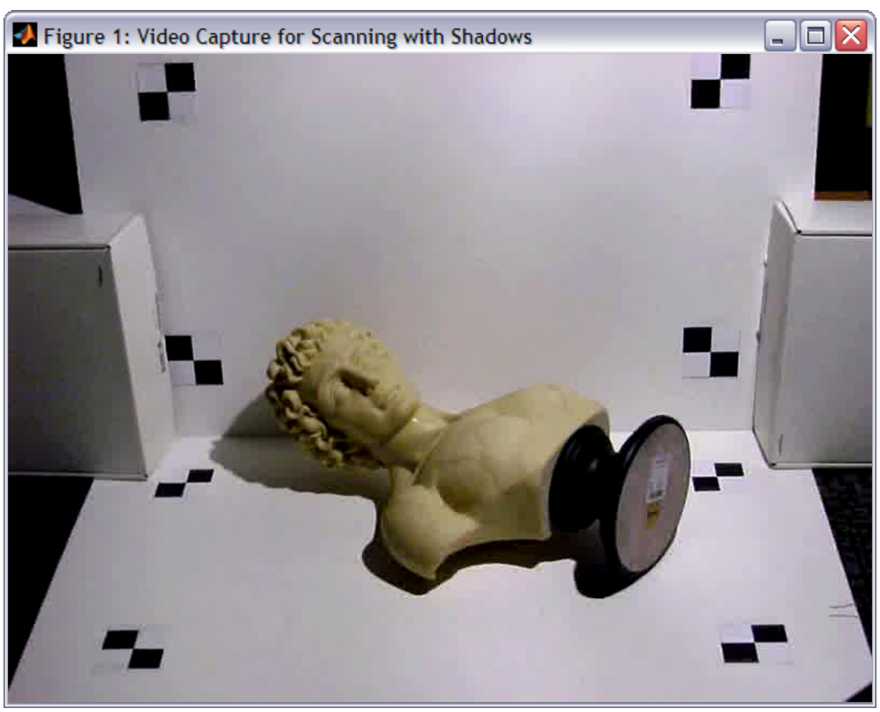
Swept-Plane Reconstruction Geometry



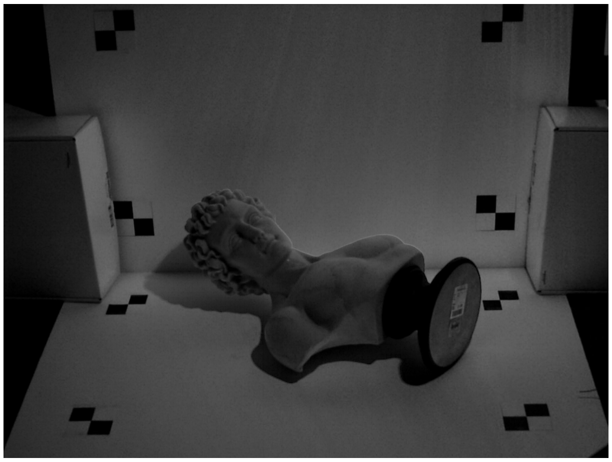


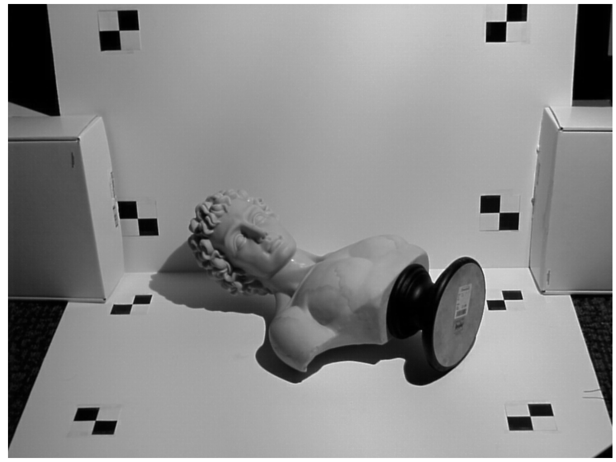
$$P = \Lambda_C(x,y) \cap \Pi_l(t)$$

Demo: Data Capture



Video Processing:
Assigning Per-Pixel Shadow Thresholds

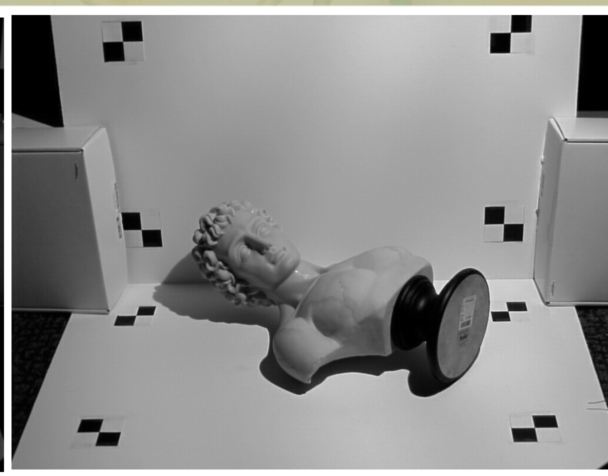




$$I_{\text{min}}(x, y) = \min_{t} I(x, y, t) \qquad I_{\text{max}}(x, y) = \max_{t} I(x, y, t)$$

- Convert from RGB to grayscale (for luminance-domain processing)
- Determine per-pixel minimum and maximum value over sequence

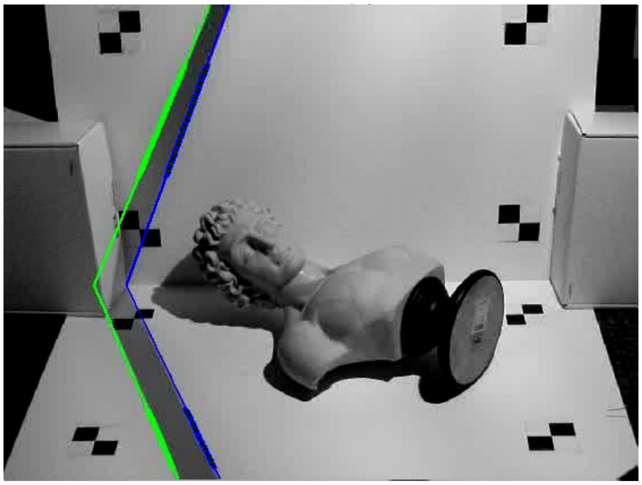
Video Processing:
Assigning Per-Pixel Shadow Thresholds

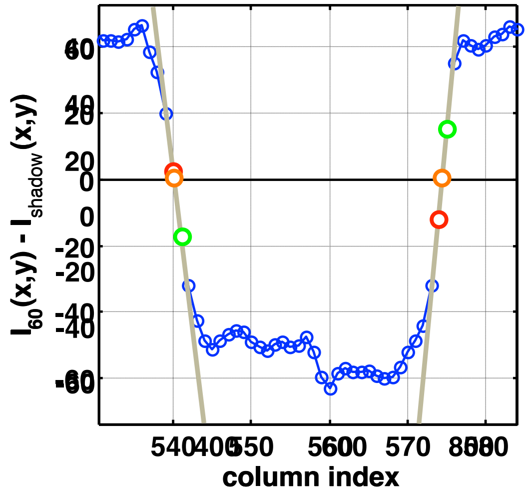


$$I_{\text{shadow}}(x, y) = \frac{I_{max}(x;y) + I_{min}(x;y)}{2}$$

- Convert from RGB to grayscale (for luminance-domain processing)
- Determine per-pixel minimum and maximum value over sequence
- Evaluate per-pixel “shadow threshold” as average of min. and max.

Video Processing:
Spatial Shadow Edge Localization

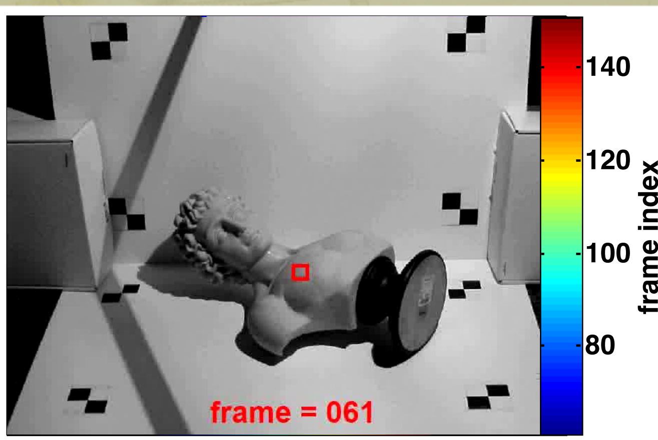


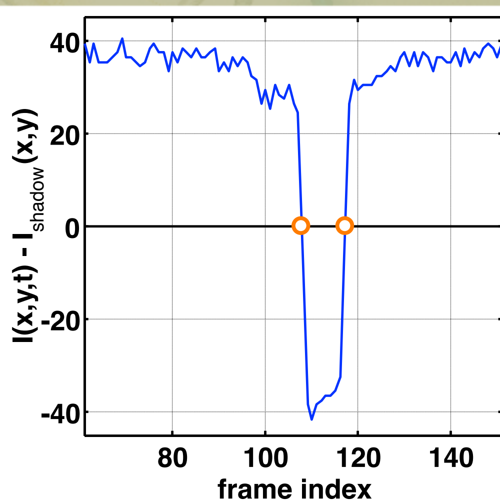


- Select region of interest on each calibration plane (occlusion-free)
- Estimate zero-crossings to find leading and trailing shadow boundaries
- Fit a line to the set of points along each shadow boundary

➔ **Result: Best-fit 2D lines for each shadow edge (in image coordinates)**

Video Processing:
Temporal Shadow Edge Localization



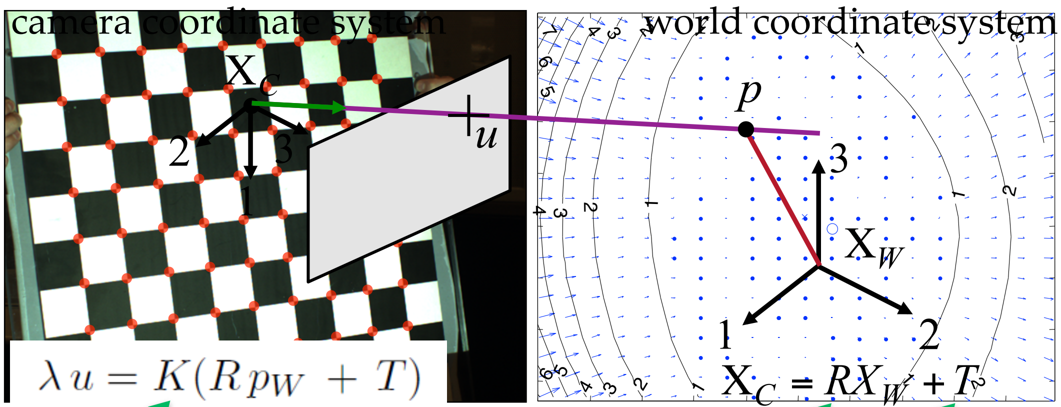


crossing frame index for leading shadow

- Tabulate per-pixel temporal sequence (minus shadow threshold)
- Estimate zero-crossings to find shadow-crossing times

➔ **Result: Use shadow-crossing time to lookup corresponding 3D plane**

Intrinsic Camera Calibration

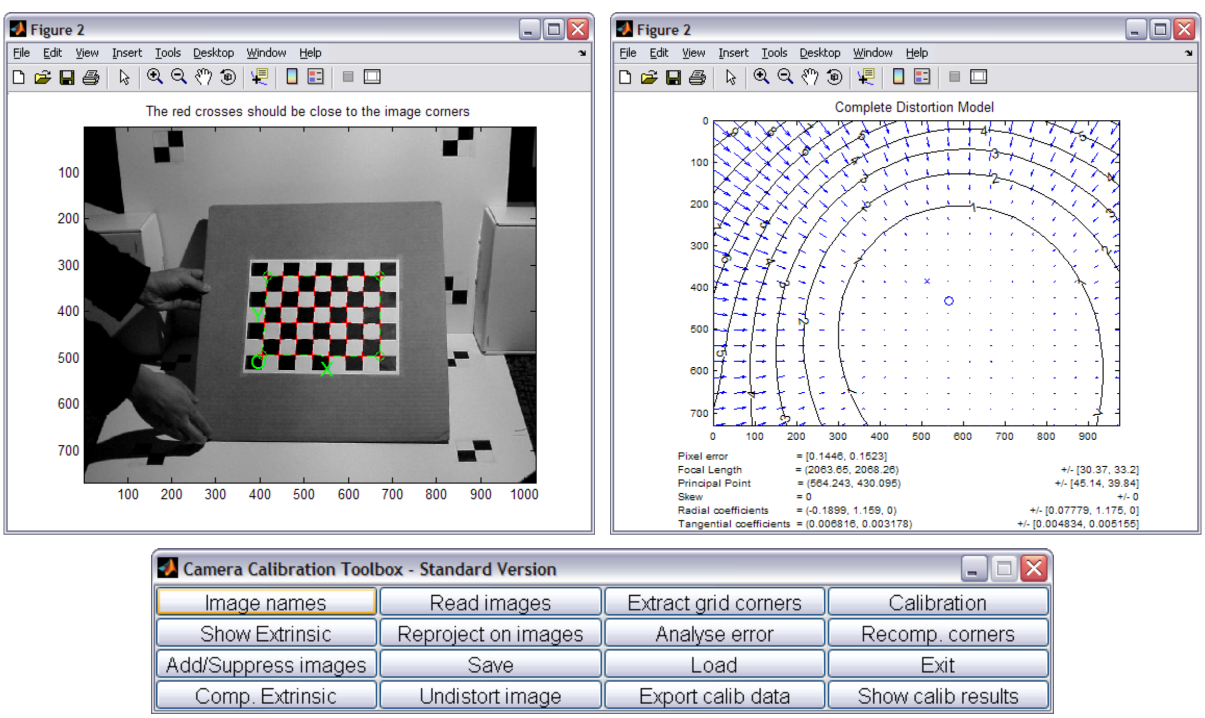


Camera Calibration Input
intrinsic parameters

Estimated Camera Lens Distortion Map
extrinsic parameters

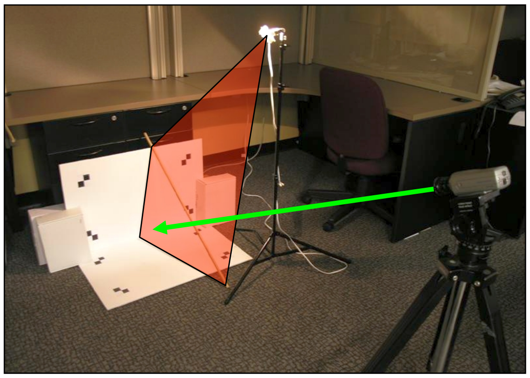
- How to estimate intrinsic parameters and distortion model? (unknowns: focal length, skew, scale, principal point, and distortion coeffs.)
- Popular solution: Observe a known calibration object (Zhang [2000])
- Each 2D chessboard corner yields two constraints on the 6-11 unknowns
- But, must also find 6 extrinsic parameters per image (rotation/translation)
- **➔ Result: Two or more images of a chessboard are sufficient**

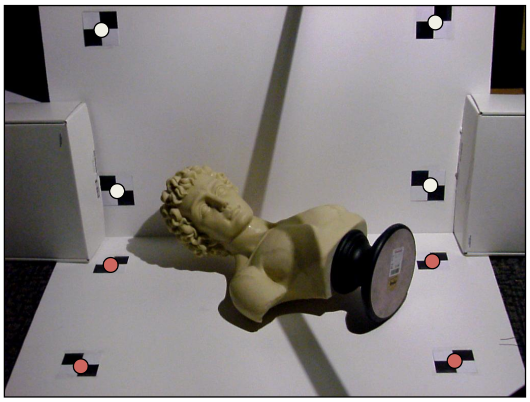
Demo: Camera Calibration in Matlab

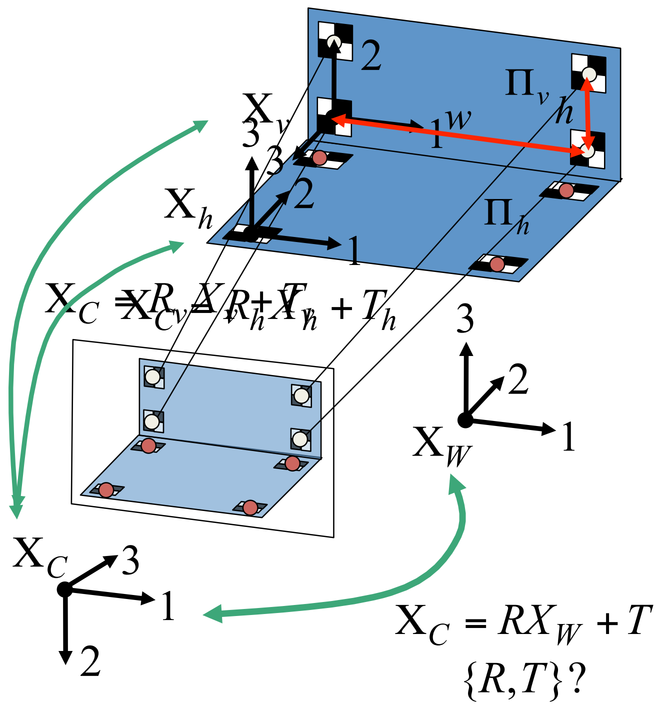


J.-Y. Bouguet. Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/

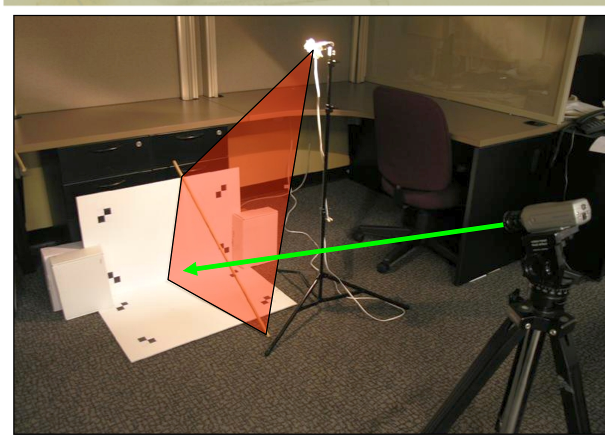
Extrinsic Camera Calibration

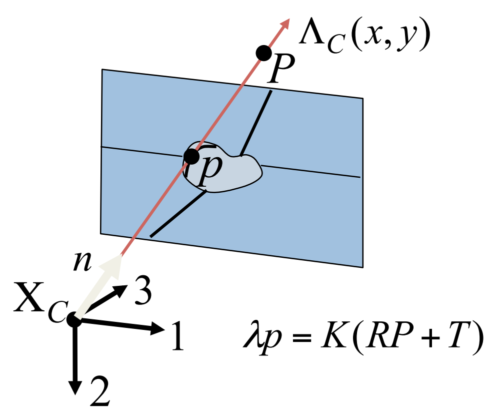






Demo: Mapping Pixels to Optical Rays

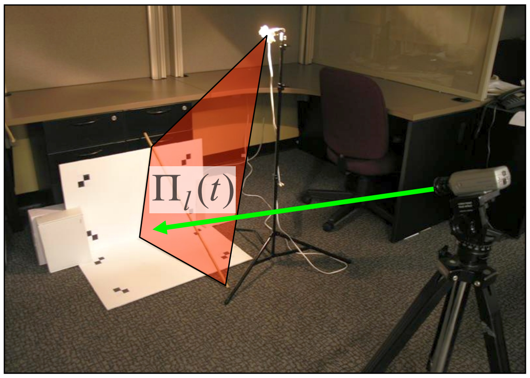


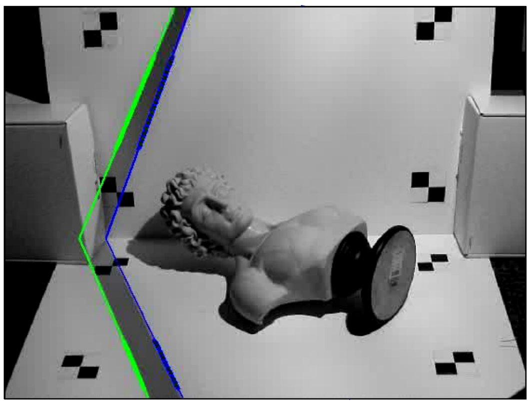


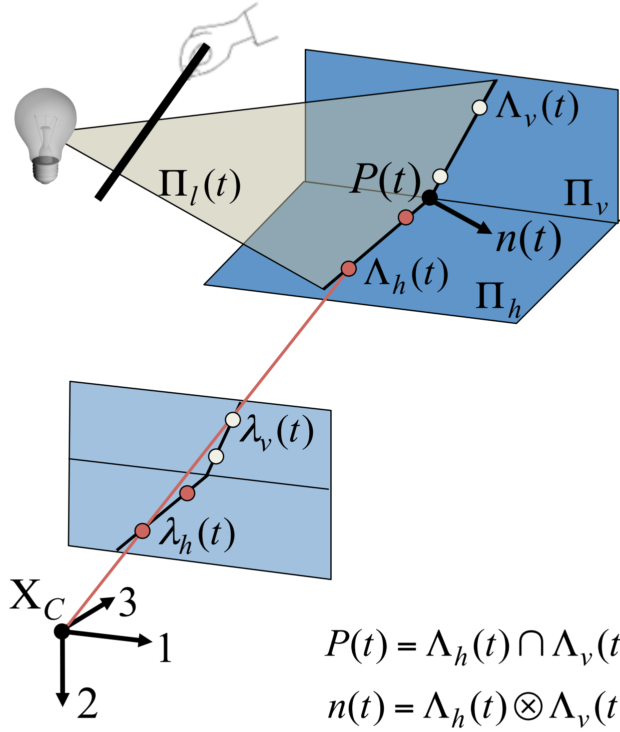
- How to map an image pixel to an optical ray?
- Solution: Invert the *calibrated* camera projection model
- But, also requires inversion of distortion model (which is non-linear)
- Mapping implemented in Camera Calibration Toolbox with **normalize.m**

➔ **Result: After calibration, pixels can be converted to optical rays**

Shadow Plane Calibration



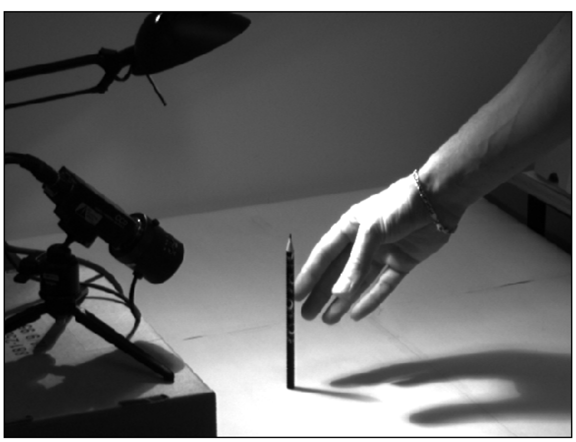


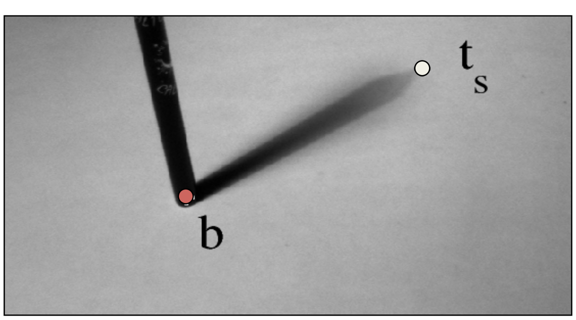


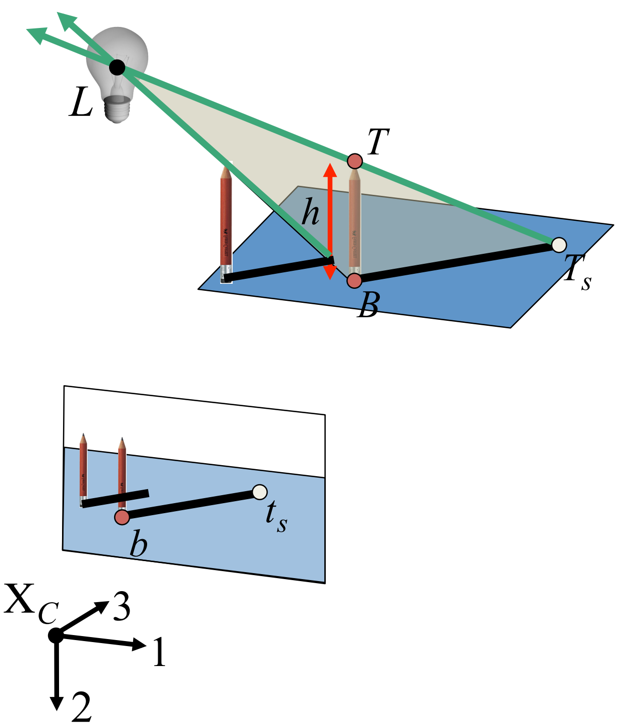
$$P(t) = \Lambda_h(t) \cap \Lambda_v(t)$$

$$n(t) = \Lambda_h(t) \otimes \Lambda_v(t)$$

Alternatives for Shadow Plane Calibration







J.-Y. Bouguet and P. Perona. 3D photography on your desk. *Intl. Conf. Comp. Vision*, 1998