Computer Vision - Lecture 19

Uncalibrated Reconstruction

20.01.2015

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
  - Active Stereo
  - Structure-from-Motion
- Motion and Tracking

Recap: A General Point

- Equations of the form

$$Ax = 0$$

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

SVD

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

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.

Recap: 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 R
  - Translation t
    (both relative to world coordinate system)
- Camera projection matrix $P = K[R \mid t]$
  - ⇒ General pinhole camera: 9 DoF
  - ⇒ CCD Camera with square pixels: 10 DoF
  - ⇒ General camera: 11 DoF

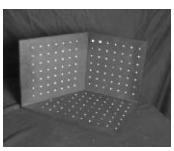
Recap: Calibrating a Camera

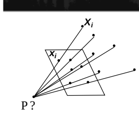
Goal

- 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 $P = P_{int}P_{ext}$





Slide credit: Kristen Grauman

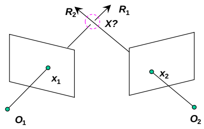
Recap: Camera Calibration (DLT Algorithm)

$$\begin{bmatrix} 0^T & X_1^T & -y_1X_1^T \\ X_1^T & 0^T & -x_1X_1^T \\ \cdots & \cdots & \cdots \\ 0^T & X_n^T & -y_nX_n^T \\ X_n^T & 0^T & -x_nX_n^T \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} = 0 \qquad Ap = 0$$

- P has 11 degrees of freedom.
- Two linearly independent equations per independent 2D/3D correspondence.
- Solve with SVD (similar to homography estimation)
  - Solution corresponds to smallest singular vector.
- 5 ½ correspondences needed for a minimal solution.

Slide adapted from Svetlana Lazebnik

Recap: Triangulation - Linear Algebraic Approach



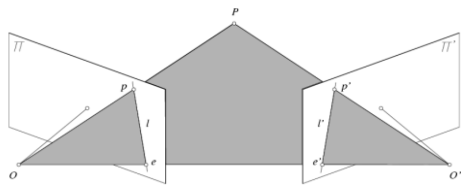
$$\lambda_1 x_1 = P_1 X \qquad x_1 \times P_1 X = 0 \qquad [x_{1\times}]P_1 X = 0$$
$$\lambda_2 x_2 = P_2 X \qquad x_2 \times P_2 X = 0 \qquad [x_{2\times}]P_2 X = 0$$

- Two independent equations each in terms of three unknown entries of X.
- Stack equations and solve with SVD.
- This approach nicely generalizes to multiple cameras.

Topics of This Lecture

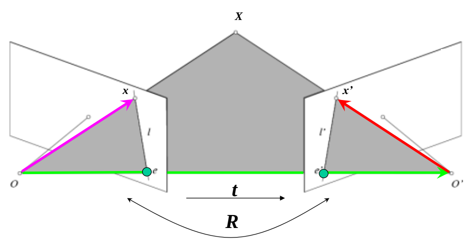
- Revisiting Epipolar Geometry
  - Calibrated case: Essential matrix
  - Uncalibrated case: Fundamental matrix
  - Weak calibration
  - Epipolar Transfer
- Active Stereo
  - Kinect sensor
  - Structured Light sensing
  - Laser scanning

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

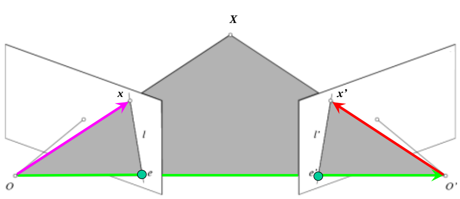


Camera matrix: $[I|0]$
$X = (u, v, w, 1)^T$
$x = (u, v, w)^T$

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

The vectors x, t, and Rx' are coplanar

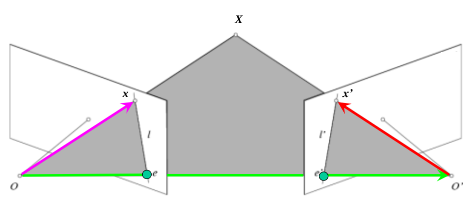
Epipolar Geometry: Calibrated Case



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

Essential Matrix
(Longuet-Higgins, 1981)

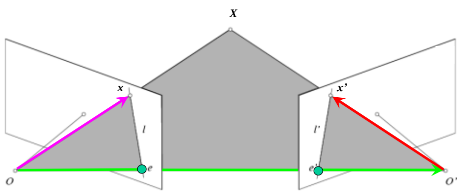
Epipolar Geometry: Calibrated Case



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

- $E x'$ 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)

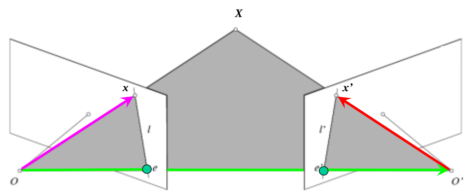
Epipolar Geometry: Uncalibrated Case



- 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



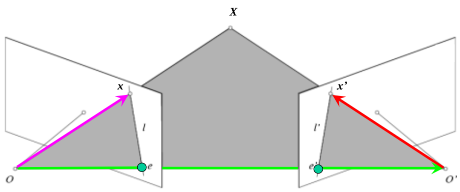
$$\hat{x}^T E \hat{x}' = 0 \;\Rightarrow\; x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

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

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

Fundamental Matrix
(Faugeras and Luong, 1992)

Epipolar Geometry: Uncalibrated Case

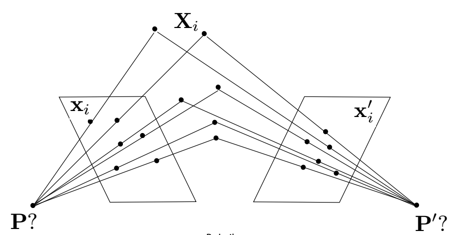


$$\hat{x}^T E \hat{x}' = 0 \;\Rightarrow\; x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

- *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

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

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

- Taking 8 correspondences:

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

$$\mathbf{Af} = 0$$

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

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

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:

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

Problem with the Eight-Point Algorithm

- In practice, this often looks as follows:

$$\begin{bmatrix} 250906.36 & 183269.57 & 921.81 & 200931.10 & 146766.13 & 738.21 & 272.19 & 198.81 & 1 \\ 2692.28 & 131633.03 & 176.27 & 6196.73 & 302975.59 & 405.71 & 15.27 & 746.79 & 1 \\ 416374.23 & 871684.30 & 935.47 & 408110.89 & 854384.92 & 916.90 & 445.10 & 931.81 & 1 \\ 191183.60 & 171759.40 & 410.27 & 416435.62 & 374125.90 & 893.65 & 465.99 & 418.65 & 1 \\ 48988.86 & 30401.76 & 57.89 & 298604.57 & 185309.58 & 352.87 & 846.22 & 525.15 & 1 \\ 164786.04 & 546559.67 & 813.17 & 1998.37 & 6628.15 & 9.86 & 202.65 & 672.14 & 1 \\ 116407.01 & 2727.75 & 138.89 & 169941.27 & 3982.21 & 202.77 & 838.12 & 19.64 & 1 \\ 135384.58 & 75411.13 & 198.72 & 411350.03 & 229127.78 & 603.79 & 681.28 & 379.48 & 1 \end{bmatrix}\begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

⇒ Poor numerical conditioning

⇒ Can be fixed by rescaling the data

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.

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

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 $T^T F T'$.

[Hartley, 1995]

The Eight-Point Algorithm

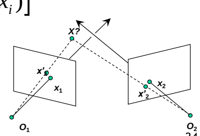
- Meaning of error $\sum_{i=1}^{N}(x_i^T F x'_i)^2$:

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

- Nonlinear approach: minimize

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

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



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.

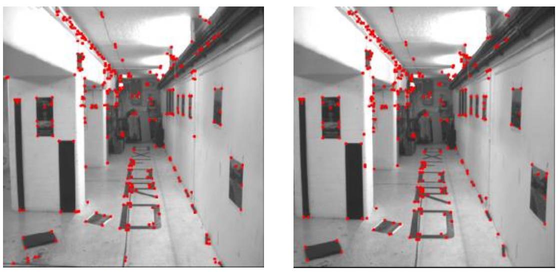
Stereo Pipeline with Weak Calibration

- 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

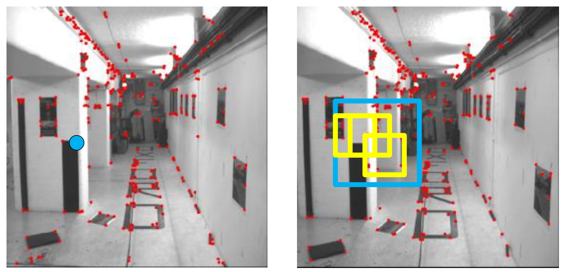
Stereo Pipeline with Weak Calibration

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

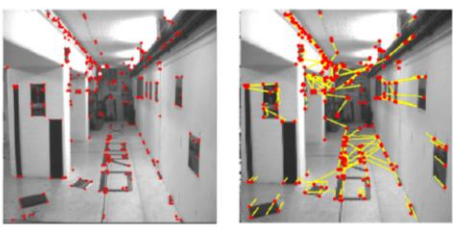


Stereo Pipeline with Weak Calibration

2. Match points using only proximity



Putative Matches based on Correlation Search

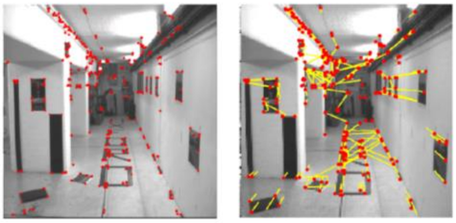


- Many wrong matches (10-50%), but enough to compute F

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)

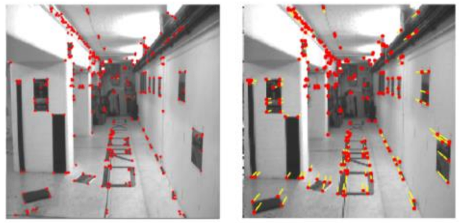
Putative Matches based on Correlation Search



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

Pruned Matches

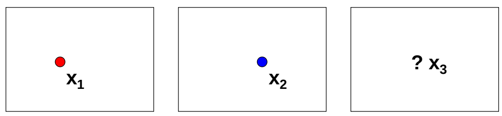
- Correspondences consistent with epipolar geometry



Resulting Epipolar Geometry

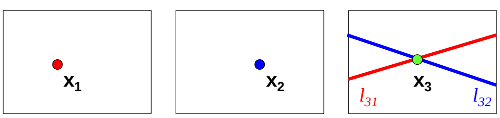
Epipolar Transfer

- 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?

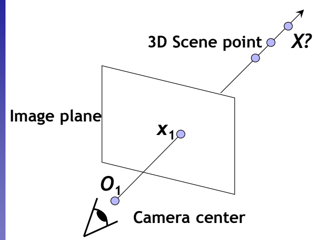
Topics of This Lecture

- Revisiting Epipolar Geometry
  - Calibrated case: Essential matrix
  - Uncalibrated case: Fundamental matrix
  - Weak calibration
  - Epipolar Transfer
- Active Stereo
  - Kinect sensor
  - Structured Light sensing
  - Laser scanning

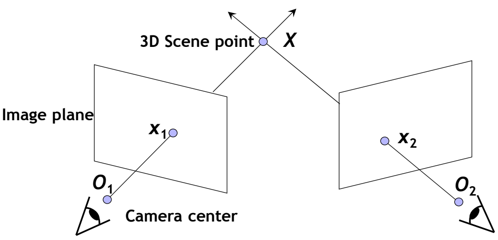
Microsoft Kinect - How Does It Work?

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

Recall: Optical Triangulation

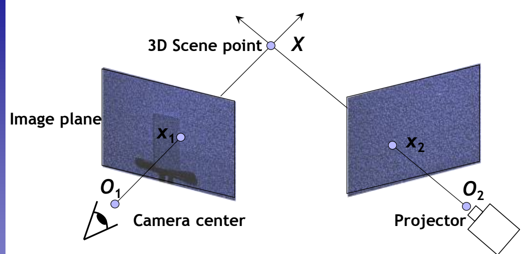


Recall: Optical Triangulation



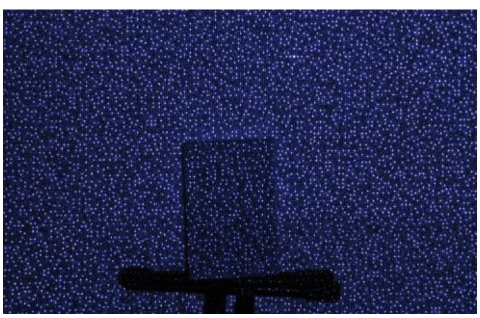
- Principle: 3D point given by intersection of two rays.
  - Crucial information: point correspondence
  - Most expensive and error-prone step in the pipeline…

Active Stereo with Structured Light



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

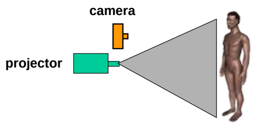
What the Kinect Sees…



3D Reconstruction with the Kinect

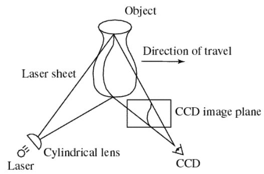
Active Stereo with Structured Light

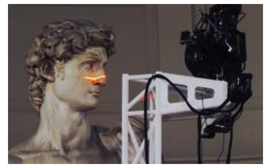
- Idea: Project “structured” light patterns onto the object
  - Simplifies the correspondence problem
  - Allows us to use only one camera



- The Kinect uses one such approach (“structured noise“)
  - What other approaches are possible?

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

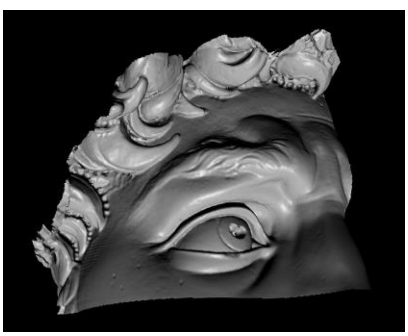
Laser Scanned Models

The Digital Michelangelo Project, Levoy et al.

Laser Scanned Models

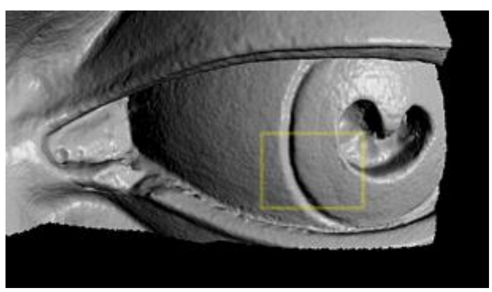
The Digital Michelangelo Project, Levoy et al.

Laser Scanned Models



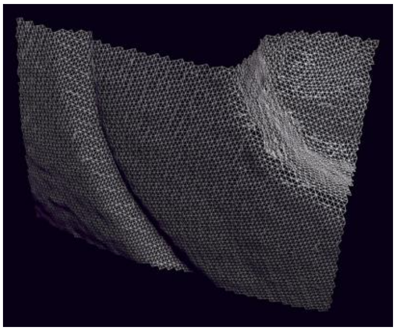
The Digital Michelangelo Project, Levoy et al.

Laser Scanned Models



The Digital Michelangelo Project, Levoy et al.

Laser Scanned Models



The Digital Michelangelo Project, Levoy et al.

Computer Vision WS 14/15

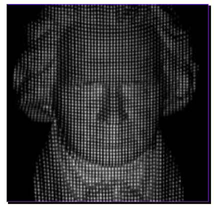
Slide credit: Steve Seitz

B. Leibe

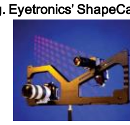
50

Multi-Stripe Triangulation

- To go faster, project multiple stripes
- But which stripe is which?
- Answer #1: assume surface continuity



e.g. Eyetronics' ShapeCam

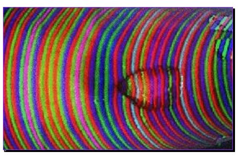


Computer Vision WS 14/15

Slide credit: Szymon Rusienkiewicz

Multi-Stripe Triangulation

- To go faster, project multiple stripes
- But which stripe is which?
- Answer #2: colored stripes (or dots)



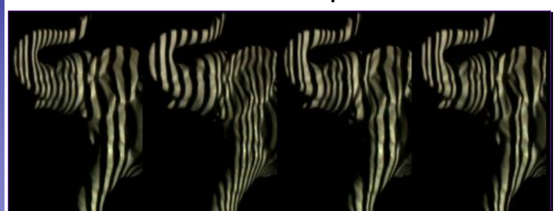
L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002

Computer Vision WS 14/15

Slide credit: Szymon Rusienkiewicz

Multi-Stripe Triangulation

- To go faster, project multiple stripes
- But which stripe is which?
- Answer #3: time-coded stripes



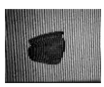
O. Hall-Holt, S. Rusienkiewicz, Stripe Boundary Codes for Real-Time Structured-Light Scanning of Moving Objects, ICCV 2001.

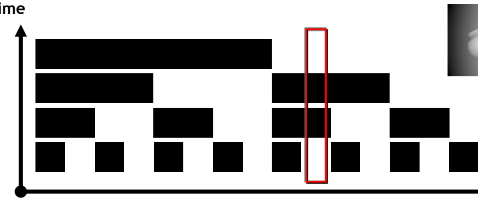
Computer Vision WS 14/15

Slide credit: Szymon Rusienkiewicz

Time-Coded Light Patterns

- Assign each stripe a unique illumination code over time [Posdamer 82]



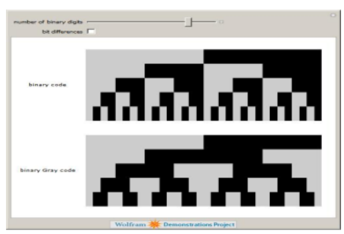


Computer Vision WS 14/15

Slide credit: Szymon Rusienkiewicz

Better codes…

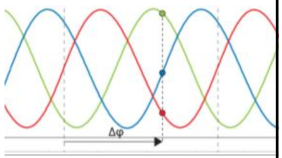
- Gray code
  Neighbors only differ one bit



Computer Vision WS 14/15

Slide credit: David Gallup

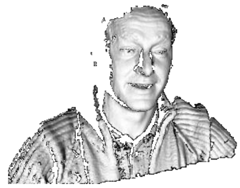
Phase-Shift Structured Light Scanning



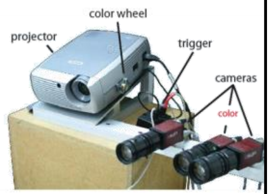
$$\Delta\varphi = \arctan\left(\frac{\sqrt{3}\,(I_r - I_b)}{(2\,I_g - I_r - I_b)}\right)$$

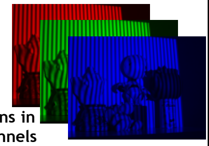
- Faster procedure by projecting continuous patterns
  - Project 3 sinusoid grating patterns shifted by 120° in phase.
  - For each pixel, compute relative phase from 3 intensities.
  - Recover absolute phase by adding a 2nd camera.

A High-Speed 3D Scanner



Dense 3D reconstructions at 30 fps (on the GPU)

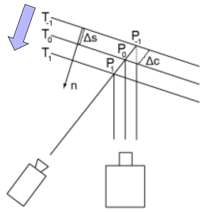


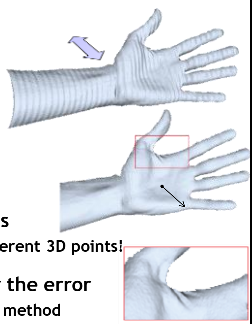


Project patterns in all 3 color channels (color wheel removed)

[Weise, Leibe, Van Gool, CVPR'07]

Problems with Dynamically Moving Objects





- Moving objects lead to artifacts
  - Measurements correspond to different 3D points!
- Derived a geometric model for the error
  - Designed a motion compensation method

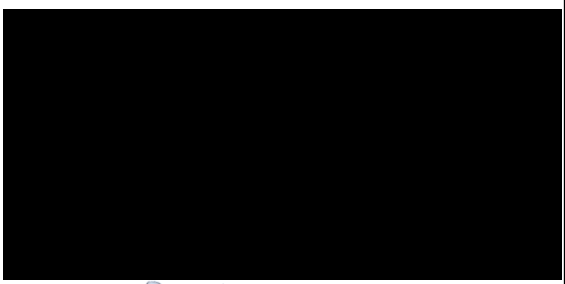
$\Rightarrow$ Result: Cleaned-up geometry + motion estimate!

[Weise, Leibe, Van Gool, CVPR'07]

Effect of Motion Compensation

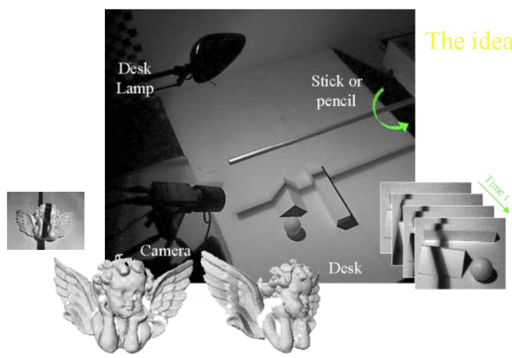
[Weise, Leibe, Van Gool, CVPR'07]

Application: Online Model Reconstruction



[Weise, Leibe, Van Gool, CVPR'08; 3DIM'09]

Poor Man's Scanner



Bouget and Perona, ICCV'98

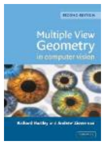
Slightly More Elaborate (But Still Cheap)

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.