Structure from Motion

Computer Vision
CSE P 576, Spring 2011
Richard Szeliski
Microsoft Research

Today's lecture

Geometric camera calibration
- camera matrix (Direct Linear Transform)
- non-linear least squares
- separating intrinsics and extrinsics
- focal length and optic center

Today's lecture

Structure from Motion
- triangulation and pose
- two-frame methods
- factorization
- bundle adjustment
- robust statistics

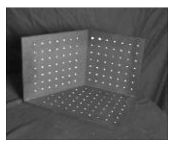
Photo Tourism

Camera Calibration

Camera calibration

Determine camera parameters from *known* 3D points or calibration object(s)

1. *internal* or *intrinsic* parameters such as focal length, optical center, aspect ratio: *what kind of camera?*
2. *external* or *extrinsic* (pose) parameters: *where is the camera?*



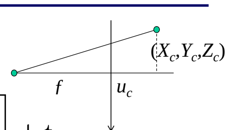
How can we do this?

Camera calibration – approaches

Possible approaches:

1. linear regression (least squares)
2. non-linear optimization
3. vanishing points
4. multiple planar patterns
5. panoramas (rotational motion)

Image formation equations



$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \mathbf{R} \end{bmatrix}_{3\times3} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{t}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} f & 0 & u_c \\ 0 & f & v_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Calibration matrix

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{K}\ \mathbf{X}_c$$

Is this form of K good enough?

- non-square pixels (digital video)
- skew
- radial distortion

$$\mathbf{K} = \begin{bmatrix} fa & s & u_c \\ 0 & f & v_c \\ 0 & 0 & 1 \end{bmatrix}$$

Camera matrix

Fold *intrinsic* calibration matrix **K** and *extrinsic* pose parameters (**R**,**t**) together into a *camera matrix*

M = **K** [**R** | **t**]

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

(put 1 in lower r.h. corner for 11 d.o.f.)

Camera matrix calibration

Directly estimate 11 unknowns in the **M** matrix using known 3D points (X_i,Y_i,Z_i) and measured feature positions (u_i,v_i)

$$\begin{aligned} u_i &= \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + 1} \\ v_i &= \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + 1} \end{aligned}$$

Camera matrix calibration

Linear regression:

- Bring denominator over, solve set of (over-determined) linear equations. How?

$$\begin{aligned} &u_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + 1) = \\ &\quad m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03} \\ &v_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + 1) = \\ &\quad m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13} \end{aligned}$$

- Least squares (pseudo-inverse)
- Is this good enough?

Levenberg-Marquardt

Iterative non-linear least squares [Press'92]

- Linearize measurement equations

$$\begin{aligned} \hat{u}_i &= f(\mathbf{m}, \mathbf{x}_i) + \frac{\partial f}{\partial \mathbf{m}} \Delta \mathbf{m} \\ \hat{v}_i &= g(\mathbf{m}, \mathbf{x}_i) + \frac{\partial g}{\partial \mathbf{m}} \Delta \mathbf{m} \end{aligned}$$

- Substitute into log-likelihood equation: quadratic cost function in **Δm**

$$\sum_i \sigma_i^{-2} (\hat{u}_i - u_i + \frac{\partial f}{\partial \mathbf{m}} \Delta \mathbf{m})^2 + \cdots$$

Levenberg-Marquardt

Iterative non-linear least squares [Press'92]

- Solve for minimum $\frac{\partial C}{\partial \mathbf{m}} = 0$

$$\mathbf{A}\Delta\mathbf{m} = \mathbf{b}$$

Hessian $$\mathbf{A} = \left[\sum_i \sigma_i^{-2} \frac{\partial f}{\partial \mathbf{m}} \left(\frac{\partial f}{\partial \mathbf{m}}\right)^T + \cdots\right]$$

error: $$\mathbf{b} = \left[\sum_i \sigma_i^{-2} \frac{\partial f}{\partial \mathbf{m}} (u_i - \hat{u}_i) + \cdots\right]$$

Camera matrix calibration

Advantages:

- very simple to formulate and solve
- can recover **K** [**R** | **t**] from **M** using QR decomposition [Golub & VanLoan 96]

Disadvantages:

- doesn't compute internal parameters
- more unknowns than true degrees of freedom
- need a separate camera matrix for each new view

Separate intrinsics / extrinsics

New feature measurement equations

$$\hat{u}_{ij} = f(\boxed{\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j}, \mathbf{x}_i)$$
$$\hat{v}_{ij} = g(\boxed{\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j}, \mathbf{x}_i)$$

Use non-linear minimization

Standard technique in photogrammetry, computer vision, computer graphics

- [Tsai 87] – also estimates κ_1 (freeware @ CMU) http://www.cs.cmu.edu/afs/cs/project/cil/ftp/html/v-source.html
- [Bogart 91] – *View Correlation*

Intrinsic/extrinsic calibration

Advantages:

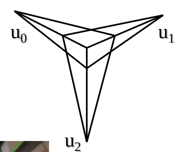
- can solve for more than one camera pose at a time
- potentially fewer degrees of freedom

Disadvantages:

- more complex update rules
- need a good initialization (recover **K** [**R** | **t**] from **M**)

Vanishing Points

Determine focal length *f* and optical center (u_c,v_c) from image of cube's (or building's) *vanishing points*
[Caprile '90][Antone & Teller '00]



Vanishing point calibration

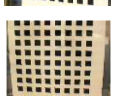
Advantages:
- only need to see vanishing points (e.g., architecture, table, …)

Disadvantages:
- not that accurate
- need rectihedral object(s) in scene

Multi-plane calibration

Use several images of planar target held at *unknown* orientations [Zhang 99]
- Compute plane homographies

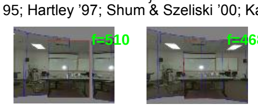
$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \sim \mathrm{K} \begin{bmatrix} \mathrm{r}_1 & \mathrm{r}_2 & \mathrm{t} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \sim \mathrm{HX}$$



- Solve for $\boldsymbol{K}^{-T}\boldsymbol{K}^{-1}$ from $\boldsymbol{H}_k$'s
  - 1plane if only *f* unknown
  - 2 planes if (f,u_c,v_c) unknown
  - 3+ planes for full $\boldsymbol{K}$
- Code available from Zhang and OpenCV

Rotational motion

Use pure rotation (large scene) to estimate *f*
1. estimate *f* from pairwise homographies
2. re-estimate *f* from 360º "gap"
3. optimize over all {$\boldsymbol{K}$,$\boldsymbol{R}_j$} parameters
   [Stein 95; Hartley '97; Shum & Szeliski '00; Kang & Weiss '99]



Most accurate way to get *f*, short of surveying distant points

Pose estimation and triangulation

Pose estimation

Once the internal camera parameters are known, can compute camera pose

$$\hat{u}_{ij} = f(\mathrm{K}, \boxed{\mathrm{R}_j, \mathrm{t}_j}, \mathrm{x}_i)$$
$$\hat{v}_{ij} = g(\mathrm{K}, \boxed{\mathrm{R}_j, \mathrm{t}_j}, \mathrm{x}_i)$$

[Tsai87] [Bogart91]

Application: superimpose 3D graphics onto video

How do we initialize (***R***,***t***)?

Pose estimation

Previous initialization techniques:

- vanishing points [Caprile 90]
- planar pattern [Zhang 99]

Other possibilities

- *Through-the-Lens Camera Control* [Gleicher92]: differential update
- 3+ point “linear methods”: [DeMenthon 95][Quan 99][Ameller 00]

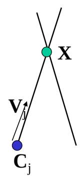
Triangulation

Problem: Given some points in *correspondence* across two or more images (taken from calibrated cameras), $\{(u_j,v_j)\}$, compute the 3D location **X**

Triangulation

Method I: intersect viewing rays in 3D, minimize:

$$\arg\min_{\mathbf{X}} \sum_j \|\mathbf{C}_j + s\mathbf{V}_j - \mathbf{X}\|$$



- **X** is the unknown 3D point
- $\mathbf{C}_j$ is the optical center of camera *j*
- $\mathbf{V}_j$ is the *viewing ray* for pixel (u_j, v_j)
- s_j is unknown distance along $\mathbf{V}_j$

Advantage: geometrically intuitive

Triangulation

Method II: solve linear equations in **X**

- advantage: very simple

$$u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + 1}$$

$$v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + 1}$$

Method III: non-linear minimization

- advantage: most accurate (image plane error)

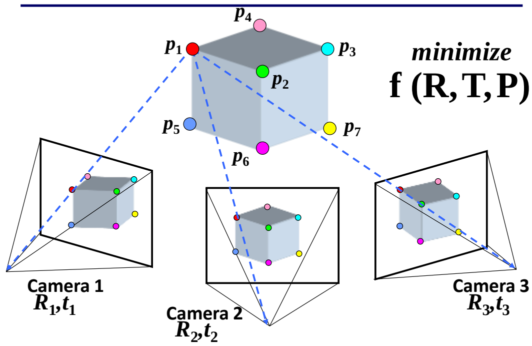
Structure from Motion

Today's lecture

Structure from Motion

- two-frame methods
- factorization
- bundle adjustment
- robust statistics

Structure from motion



Structure from motion

Given many points in *correspondence* across several images, $\{(u_{ij},v_{ij})\}$, simultaneously compute the 3D location $\mathbf{x}_i$ and camera (or *motion*) parameters ($\mathbf{K}$, $\mathbf{R}_j$, $\mathbf{t}_j$)

$$\begin{aligned}\hat{u}_{ij} &= f(\boxed{\mathrm{K}, \mathrm{R}_j, \mathrm{t}_j, \mathrm{x}_i}) \\ \hat{v}_{ij} &= g(\boxed{\mathrm{K}, \mathrm{R}_j, \mathrm{t}_j, \mathrm{x}_i})\end{aligned}$$

Two main variants: calibrated, and uncalibrated (sometimes associated with Euclidean and projective reconstructions)

Structure from motion

$$\begin{aligned}\hat{u}_{ij} &= f(\boxed{\mathrm{K}, \mathrm{R}_j, \mathrm{t}_j, \mathrm{x}_i}) \\ \hat{v}_{ij} &= g(\boxed{\mathrm{K}, \mathrm{R}_j, \mathrm{t}_j, \mathrm{x}_i})\end{aligned}$$

How many points do we need to match?

- 2 frames:
  ($\boldsymbol{R},\boldsymbol{t}$): 5 dof + $3n$ point locations $\leq$
  $4n$ point measurements $\Rightarrow$
  $n \geq 5$
- k frames:
  $6(k-1)-1 + 3n \leq 2kn$
- always want to use many more

Two-frame methods

Two main variants:

1. Calibrated: "Essential matrix" $\boldsymbol{E}$
   use ray directions ($\boldsymbol{x}_i$, $\boldsymbol{x}_i$')
2. Uncalibrated: "Fundamental matrix" $\boldsymbol{F}$

[Hartley & Zisserman 2000]

Essential matrix

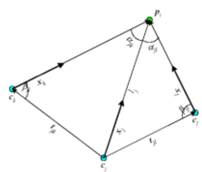
Co-planarity constraint:

$\boldsymbol{x}' \approx \boldsymbol{R}\,\boldsymbol{x} + \boldsymbol{t}$

$[\boldsymbol{t}]_\times \boldsymbol{x}' \approx [\boldsymbol{t}]_\times \boldsymbol{R}\,\boldsymbol{x}$

$\boldsymbol{x}'^T [\boldsymbol{t}]_\times \boldsymbol{x}' \approx \boldsymbol{x}'^{\,T} [\boldsymbol{t}]_\times \boldsymbol{R}\,\boldsymbol{x}$

$\boldsymbol{x}'^{\,T} \boldsymbol{E}\,\boldsymbol{x} = 0$ with $\boldsymbol{E} = [\boldsymbol{t}]_\times \boldsymbol{R}$



- Solve for $\boldsymbol{E}$ using least squares (SVD)
- $\boldsymbol{t}$ is the least singular vector of $\boldsymbol{E}$
- $\boldsymbol{R}$ obtained from the other two sing. vectors

CSE 576, Spring 2008 Structure from Motion 44

Fundamental matrix

Camera calibrations are unknown

$\boldsymbol{x}'\,\boldsymbol{F}\,\boldsymbol{x} = 0$ with $\boldsymbol{F} = [\boldsymbol{e}]_\times \boldsymbol{H} = \boldsymbol{K}'[\boldsymbol{t}]_\times \boldsymbol{R}\,\boldsymbol{K}^{-1}$

- Solve for $\boldsymbol{F}$ using least squares (SVD)
  - re-scale $(\boldsymbol{x}_i, \boldsymbol{x}_i')$ so that $|\boldsymbol{x}_i| \approx 1/2$ [Hartley]
- $\boldsymbol{e}$ (epipole) is *still* the least singular vector of $\boldsymbol{F}$
- $\boldsymbol{H}$ obtained from the other two s.v.s
- "plane + parallax" (projective) reconstruction
- use self-calibration to determine $\boldsymbol{K}$ [Pollefeys]

CSE 576, Spring 2008 Structure from Motion 45

Multi-frame Structure from Motion

Bundle Adjustment

$$\hat{u}_{ij} = f(\mathrm{K}, \mathrm{R}_j, \mathrm{t}_j, \mathrm{x}_i)$$
$$\hat{v}_{ij} = g(\mathrm{K}, \mathrm{R}_j, \mathrm{t}_j, \mathrm{x}_i)$$

What makes this non-linear minimization hard?

- many more parameters: potentially slow
- poorer conditioning (high correlation)
- potentially lots of outliers
- gauge (coordinate) freedom

CSE 576, Spring 2008 Structure from Motion 56

Levenberg-Marquardt

Iterative non-linear least squares [Press'92]

- Linearize measurement equations

$$\hat{u}_i = f(\mathbf{m}, \mathbf{x}_i) + \frac{\partial f}{\partial \mathbf{m}} \Delta \mathbf{m}$$
$$\hat{v}_i = g(\mathbf{m}, \mathbf{x}_i) + \frac{\partial g}{\partial \mathbf{m}} \Delta \mathbf{m}$$

- Substitute into log-likelihood equation: quadratic cost function in **Δm**

$$\sum_i \sigma_i^{-2} (\hat{u}_i - u_i + \frac{\partial f}{\partial \mathbf{m}} \Delta \mathbf{m})^2 + \cdots$$

Levenberg-Marquardt

Iterative non-linear least squares [Press'92]

- Solve for minimum $\frac{\partial C}{\partial \mathbf{m}} = 0$

$$\mathbf{A} \Delta \mathbf{m} = \mathbf{b}$$

Hessian $$\mathbf{A} = \left[\sum_i \sigma_i^{-2} \frac{\partial f}{\partial \mathbf{m}} \left(\frac{\partial f}{\partial \mathbf{m}} \right)^T + \cdots \right]$$

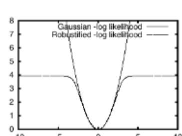
error: $$\mathbf{b} = \left[\sum_i \sigma_i^{-2} \frac{\partial f}{\partial \mathbf{m}} (u_i - \hat{u}_i) + \cdots \right]$$

Robust error models

Outlier rejection

- use robust penalty applied to each set of joint measurements

$$\sum_i \sigma_i^{-2} \rho \left(\sqrt{(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2} \right)$$



- for extremely bad data, use random sampling [RANSAC, Fischler & Bolles, CACM'81]

Structure from motion: limitations

Very difficult to reliably estimate *metric* structure and motion unless:

- large (x or y) rotation *or*
- large field of view and depth variation

Camera calibration important for Euclidean reconstructions

Need good feature tracker