

CSci 4968 and 6270 Computational Vision Lecture 16 Structure from Motion

Charles Stewart

Department of Computer Science
Rensselaer Polytechnic Institute

2009/10/26

Structure From Motion

Problem overview

- Given:
 - An image sequence taken from a moving camera, or
 - A set of images of the same scene taken from different cameras and viewpoints
- Estimate:
 - Determine which images show overlapping views
 - Estimate camera positions and internal parameters
 - Estimate 3d locations of a set of keypoints

What Can We Already Do?

Based on our discussion of keypoints, matching and cameras:

- Extract keypoints and descriptors from each image
 - Could generalize the descriptors to affine invariant
- Match keypoints between images
- Estimate fundamental matrix, \mathbf{F} , relating points in two images.
Recall,
 - If $\bar{\mathbf{x}}_1$ and $\bar{\mathbf{x}}_2$ are corresponding points in I_1 and I_2 , then

$$\bar{\mathbf{x}}_2^T \mathbf{F} \bar{\mathbf{x}}_1 = 0.$$

- \mathbf{F} is 3×3 and its rank is 2.
- Estimating \mathbf{F} requires either 7 or 8 correspondences, depending on the choice of technique.

Keypoints and Features, Revisited

Choice depends on expected camera motion

- When there is a single camera, small incremental movements, we don't necessarily need SIFT:
 - Harris corners
 - KLT tracking
- When there are multiple cameras and arbitrary views, use SIFT matching. Here, speed issues are a concern.
- High-speed implementations of SIFT, mostly based on GPU hardware, have made it more attractive to use SIFT descriptors as the basis for feature matching in the incremental motion case.

What Images To Pair Up in Matching

Choice depends on choice of scenario

- For single, continuously-moving camera, match consecutive images
- For multiple camera scenario:
 - Match all $C(N, 2)$ pairs, or
 - Use recent vocabulary-tree techniques to quickly identify which image pairs may have enough features in common. (We will discuss these when we cover "instance recognition".)

After Image-to-Image Matching

Assume fundamental matrix estimated for each matching pair

- Extract estimates of the intrinsic and extrinsic parameters from the fundamental matrix
- Gather matched feature points into "tracks"
- Backproject tracks to 3d
- Add more images
- Refine camera parameters and 3d point locations using "bundle adjustment"

We'll begin by returning to the fundamental matrix, since that is what we need to "take apart".

Deriving the Fundamental Matrix

Unlike when we first discussed it a few weeks ago, we will now derive its form.

- Given are two cameras and their associated images I_1 and I_2 .
- We will start by deriving what's called the "essential matrix", \mathbf{E} , under the assumption that the intrinsic camera parameter matrices, \mathbf{K}_1 and \mathbf{K}_2 , are known.
- We will then write the fundamental matrix \mathbf{F} in terms of \mathbf{E} .

The Essential Matrix

Camera coordinates

- Let $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{x}}_2$ be the homogeneous coordinates of corresponding image points in I_1 and I_2 .
- We convert these to "normalized" or "camera coordinates" through multiplication with the the inverses of the intrinsic camera matrices:

$$\tilde{\mathbf{u}}_1 = \mathbf{K}_1^{-1} \tilde{\mathbf{x}}_1 \quad \text{and} \quad \tilde{\mathbf{u}}_2 = \mathbf{K}_2^{-1} \tilde{\mathbf{x}}_2$$

- An important but subtle point: we can think of $\tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{u}}_2$ as both
 - Homogeneous coordinate vectors in an image with focal length 1
 - Direction vectors for the line (in image coordinates) that projects onto the image points.

The Coplanarity Constraint

Exploiting the coplanarity of three vectors

- The two camera centers and the point in the world that corresponds to both $\tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{u}}_2$ are coplanar.
- Form world coordinates:
 - Camera 1 is at origin with line of sight on the z axis
 - Camera 2 is rotated and translated from camera 1 by \mathbf{R} and \mathbf{t} .
- In this coordinate system, the vectors joining our three coplanar points are parallel to
 - $-\mathbf{R}^T \mathbf{t}$
 - $\tilde{\mathbf{u}}_1$
 - $-\mathbf{R}^T \tilde{\mathbf{u}}_2$

Applying the Co-Planarity Constraint

Use the vector triple product:

- Rotate these vectors by \mathbf{R} to simplify things a bit without losing planarity. This yields

$$-\mathbf{t}, \quad \mathbf{R}\tilde{\mathbf{u}}_1, \quad \tilde{\mathbf{u}}_2.$$

- The normal to the plane is parallel to the cross product

$$\boldsymbol{\eta} = \mathbf{t} \times \mathbf{R}\tilde{\mathbf{u}}_1 = [\mathbf{t}]_{\times} \mathbf{R}\tilde{\mathbf{u}}_1$$

where

$$[\mathbf{t}]_{\times} = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}$$

is the "skew-symmetric" matrix that represents the cross-product operation.

Completing the coplanarity constraint

- Since $\boldsymbol{\eta}$ is normal to any vector in the plane, it is, in particular, normal to $\tilde{\mathbf{u}}_2$, which means

$$\tilde{\mathbf{u}}_2^T [\mathbf{t}]_{\times} \mathbf{R}\tilde{\mathbf{u}}_1 = 0.$$

The Essential Matrix

- Writing $\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$, we have what's called the *essential matrix*.
 - This is the analog to the fundamental matrix, \mathbf{F} , but for "calibrated" cameras.
- For calibrated cameras, with points written in "camera coordinates", we have the relation

$$\tilde{\mathbf{u}}_2^T \mathbf{E} \tilde{\mathbf{u}}_1 = 0.$$

- The essential matrix has 5 degrees of freedom:
 - 2 from the translation vector, since $[\mathbf{t}]_{\times}$ may be scaled arbitrarily.
 - 3 from rotation matrix \mathbf{R}
- Like \mathbf{F} , \mathbf{E} is rank 2.

Back to the Fundamental Matrix

The form of F is obtained from E :

- Recall from earlier today that

$$\tilde{\mathbf{u}}_1 = \mathbf{K}_1^{-1} \tilde{\mathbf{x}}_1 \quad \text{and} \quad \tilde{\mathbf{u}}_2 = \mathbf{K}_2^{-1} \tilde{\mathbf{x}}_2$$

- Therefore, we easily obtain

$$\mathbf{F} = \mathbf{K}_2^{-T} [\tilde{\mathbf{t}}]_{\times} \mathbf{R} \mathbf{K}_1^{-1}$$

Resetting the Stage

Matching two images

- Apply keypoint matching as before.
- When the cameras are calibrated, you may convert keypoint locations to "camera coordinates", then estimate E .
 - Random-sampling algorithms may be used with special techniques for the 5 degrees of freedom of E .
- When the cameras are uncalibrated, the image coordinates, with the normalization technique we discussed for estimating H , may be used to estimate F .
- The next step is to try to extract \mathbf{t} and \mathbf{R} from E , or extract \mathbf{t} , \mathbf{R} and matrices \mathbf{K}_1 and \mathbf{K}_2 from F .

Taking Apart the Fundamental Matrix

Intrinsic camera parameters

- Make assumptions:
 - No skew between image coordinate axes
 - Image center is the center of the pixel array
 - Aspect ratio is known
- This reduces each intrinsic camera matrices to just a single degree of freedom:

$$\mathbf{K}_1 = \begin{pmatrix} f_1 & 0 & 0 \\ 0 & f_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{K}_2 = \begin{pmatrix} f_2 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- These focal lengths may be estimated using what's called the *Kruppa equations*, or they may be read from the EXIF data associated with modern digital images.

The Extrinsic Parameters from the Essential Matrix

- It can be shown that the singular value decomposition of a true essential matrix is $\mathbf{U} \text{diag}(1, 1, 0) \mathbf{V}^T$, where the 3rd row of \mathbf{U} is the direction of the translation vector \mathbf{t} .
 - In practice, the singular values will not be 1, 1, and 0, but they should be close.
 - Remember: we can only recover \mathbf{t} up to an unknown scale factor!
- Let

$$\mathbf{D} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Then, ensuring that \mathbf{U} and \mathbf{V} have positive determinants, the rotation matrix is

$$\mathbf{R} = \mathbf{U} \mathbf{D} \mathbf{V}^T \quad \text{or} \quad \mathbf{R} = \mathbf{U} \mathbf{D}^T \mathbf{V}^T.$$

Reconstructing the Points

Once we have camera estimates we can reconstruct estimates of the point locations

- Form two lines with parametric equations:
 - $I_1(a) = a\mathbf{u}_1$
 - $I_2(b) = b\mathbf{R}\mathbf{u}_2 + \mathbf{t}$
- Find point \mathbf{p} in 3d world coordinates that minimizes the distance to these two lines
- Repeat for each "correct" keypoint match
- Points are in world coordinate system that coincides with camera 1 and image I_1 .
 - A 3d similarity transformation can put them into any other coordinate system.
- Choose between two different rotation matrices \mathbf{R} based on which produces the most points in front of the cameras!

A Word of Caution

Make sure a homography does not fit just as well

- We want to ensure that the translation is not too small and that we aren't dealing with a planar surface.
- For any pair of images I_1 and I_2 , fit a homography matrix \mathbf{H} to the correspondences used to estimate \mathbf{F} .
- Then compute the root mean squared error in the mapping
- If the resulting error is substantially higher than the uncertainty in the feature point positions, then \mathbf{F} is preferred over \mathbf{H} .

Reconstruction with More than Two Images

Concentrate on the multiple cameras, multiple viewpoints case

- Choose two images that yield a relatively large number of correspondences and a relatively large error in the estimated homography (see previous slide).
- Extract initial estimates of focal lengths, \mathbf{t} and \mathbf{R} , as described above.
- Estimate 3d point positions.
- At this point we have our initial "structure-from-motion" estimate.
- Add new images to the reconstruction based on overlapping "feature tracks"
- Re-estimate final positions using what are known as "bundle adjustment" techniques.

Feature Tracks

- A set of corresponding feature locations across two or more images, with one feature per image.
 - It is intended that each track corresponds to a single location in the scene.
- The name "feature tracks" arose from the original SFM scenario of a single, continuously-moving camera, but has been applied to the multiple-camera, multiple viewpoint scenario.
- For a continuously-moving camera, a track is found based on frame-to-frame matching
- For the multi-camera scenario, the feature track extraction algorithm is graph based:
 - Each feature forms a vertex in the graph
 - An edge is formed between vertices if the features are matched
 - Breadth-first search starting from a feature/vertex is used to establish tracks.
 - Additional consistency checks are applied to tracks
- Per usual, we can not assume all tracks are correct.

Bundle Adjustment

Obtain final cameras and 3d positions of corresponding points

- \mathbf{p}_k is the world coordinate point in \mathcal{R}^3 corresponding to track k
- \mathbf{c}_i is the vector of parameters for the i -th camera
 - Includes translation, rotation, focal length and, usually, one or two radial lens distortion terms.
- $\mathbf{x}_{i,k}$ is the image position of the feature belonging to track k in image i
- $\mathbf{T}(\mathbf{p}_k; \mathbf{c}_i)$ is the projection of world point \mathbf{p}_k in the images coordinates (not camera coordinates) of camera i .
- The bundle adjustment objective function is

$$\sum_i \sum_k \|\mathbf{T}(\mathbf{p}_k; \mathbf{c}_i) - \mathbf{x}_{i,k}\|^2$$

- This non-linear least-squares technique is generally solved using bundle-adjustment methods

Applications

We have already discussed these, but we will reconsider them in class

- Photo tourism
- Match-move