CIS 580 Spring 2012 - Lecture 19

March 30, 2012

Notes and figures by Matthieu Lecce.

Review: single-view geometry

- The image center is the orthocenter of the projections of three orthogonal vanishing points.
- The cross-ratio $CR(A,B,C,D) = \frac{AC}{AD} : \frac{BC}{BD}$ is invariant w.r.t. to projective transformations.
- If D is at infinity, $CR(A,B,C,D) = \frac{AC}{BC}$. The knowledge of a point at infinity and *one* distance enables us to recover many other distances by computing corresponding cross-ratios in the image. *Remark*: no calibration is needed.

Cross-ratios and distance transfer (continued)

See figure.

$$\frac{AC}{BC} = \lambda_1, \quad \frac{AE}{CE} = \lambda_2, \quad AE = AC + CE$$

($(A,B,C,D)_{px}$ and $(A,C,E,D)_{px}$) x.

From the above we can retrieve CE.

This setting enables us to reconstruct all possible rectangular boxes in the view. See figure: P is on the same plane as the bottom face of the cube "ground plane"

Pose from a single view

We are interested in answering the question "Where is the camera?"

Given n points in the world coordinate system (measured as metric distances, not just ratios as previously) and their projections (in pixels), we want to compute the **camera extrinsics**.

Extrinsics: translation and rotation (T,R) w.r.t. the world coordinate system.

This problem is at the core of many applications: augmented reality, special effects, etc. The challenge is actually not to solve the linear system to recover the camera extrinsics (which we will do in this section), but to track and match beacon points, i.e. establish a correspondence betwee 2D points in the camera image and 3D points in the world coordinates

In precise applications like movie post-production, the tracking is done in a semi-automatic way, and sometimes hundreds of CG artists just have to click on thousands of images to manually track the reference points.

Remember the pixel position (u,v) of the projection of a beacon point verifies the camera model:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} \sim \underset{3\times3}{K} \underbrace{\begin{bmatrix} R & T \end{bmatrix}}_{3\times4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

In most applications, we assume K does not vary and is obtained by calibration, and R, T are the unknowns

When intrinsics are unknown

Assume we have a set of 2D-3D matchings $(u, v, w) \leftrightarrow (X_w, Y_w, Z_w)$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} M & m \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = M \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + m$$

We define $M = KR$ and $m = KT$ for clarity.

The following results simply states that a 3D point is on the ray going through the camera center and the point on the screen:

$$\boxed{\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \lambda M^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - M^{-1} m}$$

Notice that $-M^{-1}m$ is the projection (camera) center in world coordinates. This equation defines a ray (line) that goes through the camera center and the point on the screen.

Given n 2D-3D correspondences (associations between 3D beacon points and their 2D projections), we can recover $P = \begin{bmatrix} M & m \end{bmatrix}$ using the following equations:

$$u = \frac{p_1^T X_w}{p_3^T X_w}, \quad v = \frac{p_2^T X_w}{p_3^T X_w}$$

We note $P = \begin{bmatrix} p_1^\top \\ p_2^\top \\ p_3^\top \end{bmatrix}$.

We obtain two such equations per point. The projection matrix P can be computed up to a scale factor, so it has $3 * 4 - 1 = 11$ independent unknowns. Therefore, 6 points determine a unique (up to a scale) P-matrix *if and only if 4 of them are not coplanar*.

When the intrinsics are known (photogrammetry augmented reality)

Now, let's assume K is known, therefore we only have to estimate R and T. We know the **back-projection** of the point on the screen:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Note that (x, y) are metric coordinates, whereas (u, v) are pixel coordinates.

The following simple equation is similar to the one we derived when K is unknown:

$$\boxed{\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T}$$

See fig. 3. For three 3D points A, B, C, we know AB,BC,AC,δ_{AC},δ_{AB},δ_{BC}. Unknowns are $\lambda_A, \lambda_B, \lambda_C$.

We use the law of cosines:

$$AC^2 = \lambda_A^2 + \lambda_C^2 - 2\lambda_A \lambda_C \cos \delta_{AC}.$$

(to be continued)