Geometry of image formation

Tomáš Svoboda, svoboda@cmp.felk.cvut.cz
Czech Technical University in Prague, Center for Machine Perception
http://cmp.felk.cvut.cz
Last update: November 3, 2008
Motivation

- parallel lines
- window sizes
- image units
- distance from the camera
What will we learn

- how does the 3D world project to 2D image plane?
- how is a camera modeled?
- how can we estimate the camera model?
Pinhole camera

1

http://en.wikipedia.org/wiki/Pinhole_camera
Camera Obscura

http://en.wikipedia.org/wiki/Camera_obscura
Camera Obscura — room-sized

Used by the art department at the UNC at Chapel Hill

3http://en.wikipedia.org/wiki/Camera_obscura
1D Pinhole camera
1D Pinhole camera projects 2D to 1D

\[ \frac{x_1}{-f} = \frac{X_1}{Z_1} \]

\[ x_1 = -f \frac{X_1}{Z_1} \]
Problems with perspective I

\[ X_1 = X_2 \]

\[ x_1 \neq x_2 \]
Problems with perspective II

\[ Z_2 \]

\[ Z_3 \]

\[ x \]

\[ z \]

image plane

optical axis

\[ x_2 \]

\[ f \]

\[ X_2 \neq X_3 \]

\[ x_2 = x_3 \]
Get rid of the \((-\)\) sign
How does the 3D world project to the 2D image plane?
A 3D point $\mathbf{X}$ in a world coordinate system
A pinhole camera observes a scene

\[ [0, 0, 0] \]

\( x \, y \, z \)
Point $\mathbf{X}$ projects to the image plane, point $\mathbf{x}$.
Scene projection

\[ [0, 0, 0] \]
Scene projection

\[ [0, 0, 0] \]

\[ C \]
3D Scene projection – observations

- 3D lines project to 2D lines
- but the angles change, parallel lines are no more parallel.
- area ratios change, note the front and backside of the house
Put the sketches into equations
3D → 2D Projection

We remember that: $\mathbf{x} = \left[ \frac{fX}{Z}, \frac{fY}{Z} \right]^T$

$$
\begin{bmatrix}
\mathbf{x} \\
1
\end{bmatrix}
\simeq
\begin{bmatrix}
fX \\
fY \\
Z
\end{bmatrix}
$$

$$
\begin{bmatrix}
\mathbf{x} \\
1
\end{bmatrix}
\simeq
\begin{bmatrix}
f & 0 \\
f & 0 \\
1 & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{X} \\
1
\end{bmatrix}
$$

Use the homogeneous coordinates\(^4\)

$$
\lambda_{[1 \times 1]} \mathbf{x}_{[3 \times 1]} = K_{[3 \times 3]} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{X}_{[4 \times 1]}
$$

but . . .

\(^4\)for the notation conventions, see the talk notes
we need the $X$ in camera coordinate system

Rotate the vector:

$$X = R(X_w - C_w)$$

$R$ is a $3 \times 3$ rotation matrix. The point coordinates $X$ are now in the camera frame.

Use homogeneous coordinates to get a matrix equation

$$\begin{bmatrix} X \\ 1 \end{bmatrix} = \begin{bmatrix} R & -RC_w \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ 1 \end{bmatrix}$$

The camera center $C_w$ is often replaced by the translation vector

$$t = -RC_w$$
External (extrinsic parameters)

The translation vector $t$ and the rotation matrix $R$ are called External parameters of the camera.

$$x \simeq K \begin{bmatrix} I | 0 \end{bmatrix} \begin{bmatrix} X \\ 1 \end{bmatrix}$$

$$\lambda x = K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X_w \\ 1 \end{bmatrix}$$

Camera parameters (so far): $f, R, t$

Is it all? What can we model?
What is the geometry good for?

How would you characterize the difference?

Would you guess the motion type?
What is the geometry good for?

video: Zoom out vs. motion away from scene
Enough geometry$^5$, look at real images

$^5$just for a moment
From geometry to pixels and back again
Problems with pixels
Is this a straight line?
Problems with pixels
What are we looking at?
Did you recognize it?
Pixel images revisited

- There are no negative coordinates. Where is the principal point?
- Lines are not lines any more.
- Pixels, considered independently, do not carry much information.
Pixel coordinate system

Assume normalized geometrical coordinates \( \mathbf{x} = [x, y, 1]^\top \)

\[
\begin{align*}
    u &= m_u(-x) + u_0 \\
    v &= m_v y + v_0
\end{align*}
\]

where \( m_u, m_v \) are sizes of the pixels and \( [u_0, v_0]^\top \) are coordinates of the principal point.
Put pixels and geometry together

From 3D to image coordinates:
\[
\begin{bmatrix}
\lambda x \\
\lambda y \\
\lambda 
\end{bmatrix} = \begin{bmatrix}
f & 0 & 0 \\
0 & f & 0 \\
0 & 0 & 1 
\end{bmatrix} \begin{bmatrix}
R \\
t
\end{bmatrix} X_{[4 \times 1]}
\]

From normalized coordinates to pixels:
\[
\begin{bmatrix}
u \\
v \\
1
\end{bmatrix} = \begin{bmatrix}
-m_u & 0 & u_0 \\
0 & m_v & v_0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

Put them together:
\[
\frac{1}{\lambda} \begin{bmatrix}
u \\
v \\
1
\end{bmatrix} = \begin{bmatrix}
-fm_u & 0 & u_0 \\
0 & fm_v & v_0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
R \\
t
\end{bmatrix} X
\]

Finally:
\[u \simeq K \begin{bmatrix}
R \\
t
\end{bmatrix} X\]

Introducing a $3 \times 4$ camera projection matrix $P$:
\[u \simeq PX\]
Non-linear distortion

Several models exist. Less standardized than the linear model. We will consider a simple on-parameter radial distortion. $x_n$ denote the linear image coordinates, $x_d$ the distorted ones.

$$x_d = (1 + \kappa r^2) x_n$$

where $\kappa$ is the distortion parameter, and $r^2 = x_n^2 + y_n^2$ is the distance from the principal point.

Observable are the distorted pixel coordinates

$$u_d = K x_d$$

Assume that we know $\kappa$. How to get the lines back?
Undoing Radial Distortion

From pixels to distorted image coordinates: \( x_d = K^{-1}u_d \)

From distorted to linear image coordinates: \( x_n = \frac{x_d}{1 + \kappa r^2} \)

Where is the problem? \( r^2 = x_n^2 + y_n^2 \). We have unknowns on both sides of the equation.

Iterative solution:

1. initialize \( x_n = x_d \)
2. \( r^2 = x_n^2 + y_n^2 \)
3. compute \( x_n = \frac{x_d}{1 + \kappa r^2} \)
4. go to 2. (and repeat few times)

And back to pixels \( u_n = Kx_n \)
Undoing Radial Distortion
Estimation of camera parameters—camera calibration

The goal: estimate the $3 \times 4$ camera projection matrix $P$ and possibly the parameters of the non-linear distortion $\kappa$ from images.

Assume a known projection $[u, v]^\top$ of a 3D point $X$ with known coordinates

$$
\begin{bmatrix}
\lambda u \\
\lambda v \\
\lambda
\end{bmatrix}
= 
\begin{bmatrix}
P_1^\top \\
P_2^\top \\
P_3^\top
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
$$

$$
\frac{\lambda u}{\lambda} = \frac{P_1^\top X}{P_3^\top X} \text{ and } \frac{\lambda v}{\lambda} = \frac{P_2^\top X}{P_3^\top X}
$$

Re-arrange and assume$^6$ $\lambda \neq 0$ to get set of homogeneous equations

$$
u X^\top P_3 - X^\top P_1 = 0$$
$$v X^\top P_3 - X^\top P_2 = 0$$

$^6$see some notes about $\lambda = 0$ in the talk notes
Estimation of the $P$ matrix

\[ uX^TP_3 - X^TP_1 = 0 \]
\[ vX^TP_3 - X^TP_2 = 0 \]

Re-shuffle into a matrix form:

\[
\begin{bmatrix}
-X^T & 0^T & uX^T \\
0^T & -X^T & vX^T
\end{bmatrix}
\begin{bmatrix}
P_1 \\
P_2 \\
P_3
\end{bmatrix} = 0_{[2 \times 1]}
\]

A correspondence $u_i \leftrightarrow X_i$ forms two homogeneous equations. $P$ has 12 parameters but scale does not matter. We need at least 6 2D $\leftrightarrow$ 3D pairs to get a solution. We constitute $A_{[\geq 12 \times 12]}$ data matrix and solve

\[ p^* = \text{argmin} \|Ap\| \text{ subject to } \|p\| = 1 \]

which is a constrained LSQ problem. $p^*$ minimizes algebraic error.
Decomposition of $P$ into the calibration parameters

\[ P = \begin{bmatrix} KR & Kt \end{bmatrix} \quad \text{and} \quad C = -R^{-1}t \]

We know that $R$ should be $3 \times 3$ orthonormal, and $K$ upper triangular.

\( P = P./\text{norm}(P(3,1:3)); \)

\( [K,R] = \text{rq}(P(:,1:3)); \)
\( t = \text{inv}(K)*P(:,4); \)
\( C = -R'*t; \)

See the slide notes for more details.
An example of a calibration object
2D projections localized

![Image of 2D projections localized]
Reprojection for linear model
Reprojection for full model
Reprojection errors—comparison between full and linear model

sorted 2D reprojection errors
The book [2] is the ultimate reference. It is a must read for anyone wanting to use cameras for 3D computing.

Details about matrix decompositions used throughout the lecture can be found at [1]


End
\[
\frac{x_1}{-f} = \frac{X_1}{Z_1}
\]

\[
x_1 = -f \frac{X_1}{Z_1}
\]
$x_1 \neq x_2$
\[ x_2 \neq x_3 \quad \text{or} \quad x_2 = x_3 \]
$X^°$
sorted 2D reprojection errors

pixels

full model
linear model