

从本质矩阵恢复相机矩阵

本质矩阵

本质矩阵 (essential matrix) 是基本矩阵在归一化图像坐标下的一种特殊形式。

考虑相机矩阵 $P = K[R|t]$, 点 $x = P^{-1}X$ 是图像平面上的一个点, 若已知相机内参 K , 那么点 $\hat{x} = K^{-1}x$, 有 $\hat{x} = [R|t]X$, 则称点 \hat{x} 是在归一化坐标 (normalized coordinates) 下的表示。点 \hat{x} 可以被认为是在空间点 X 在内参矩阵为单位阵 I 的情况下的像, 而 $K^{-1}P = [R|t]$ 称为归一化相机矩阵 (normalized camera matrix)。

对于一对对应点 $x \leftrightarrow x'$ 基本矩阵定义为

$$x'^T F x = 0 \quad (1)$$

而对本质矩阵, 给定一对对应点 $x \leftrightarrow x'$, 归一化图像点对为 $\hat{x} \leftrightarrow \hat{x}'$, 定义为

$$\hat{x}'^T E \hat{x} = 0 \quad (2)$$

把点对关系 $\hat{x} = K^{-1}x$, 和 $\hat{x}' = K'^{-1}x'$ 代入(2)可以得到 $\hat{x}'^T K'^{-T} E K^{-1} \hat{x} = 0$, 则有如下关系

$$E = K'^T F K \quad (3)$$

考虑相机矩阵 $P = K[I|0]$ 和 $P' = [R|t]$, 根据 F 矩阵的性质有

$$F = K'^{-T} [t]_{\times} R K^{-1} = K'^{-T} R [R^T t]_{\times} K^{-1} \quad (4)$$

从而有

$$E = [t]_{\times} R = R [R^T t]_{\times} \quad (5)$$

本质矩阵的性质

其中 t 和 R 分别有3个自由度, 除去一个齐次因子, E 只有5个自由度, 因此需要满足比 F 矩阵更多的约束。

反对称矩阵的性质

如果一个矩阵 M 是 $d \times d$ 的反对称矩阵 (skew-symmetric/antisymmetric matrices), 那么其满足 $M^T = -M$, 有

$$\det M = \det(-M^T) = \det(-M) = (-1)^d \det M \quad (6)$$

可以观察到, 当 d 为奇数时, $\det M = 0$ 。所以, 反对称矩阵 M 的秩一定为偶数。

如果 M 的维度为偶数的非奇异 $2n \times 2n$ 反对称矩阵, 那么, 存在正交矩阵 U , 有

$$U^T M U = N \equiv \text{diag} \left\{ \begin{pmatrix} 0 & m_1 \\ -m_1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & m_2 \\ -m_2 & 0 \end{pmatrix}, \dots, \begin{pmatrix} 0 & m_n \\ -m_n & 0 \end{pmatrix} \right\} \quad (7)$$

这里的 m_j 都是正实数。矩阵 N 称为非奇异反对称矩阵的实标准型 (real normal form)。如果矩阵 M 是奇异的且秩为 $2n$ 的 $d \times d$ 矩阵 (d 可以是奇数可以是偶数), 则存在 $d \times d$ 矩阵 U 使得

$$U^T M U = N \equiv \text{diag} \left\{ \begin{pmatrix} 0 & m_1 \\ -m_1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & m_2 \\ -m_2 & 0 \end{pmatrix}, \dots, \begin{pmatrix} 0 & m_n \\ -m_n & 0 \end{pmatrix}, O_{d-2n} \right\} \quad (8)$$

这里的 O_{d-2n} 是一个 $(d-2n) \times (d-2n)$ 的零矩阵, 当 $d = 2n$ 则(8)式退化为(7)式。

综上所述, 实反对称矩阵 M 可以分解为 $M = U N U^T$ 的形式, U 为正交矩阵。其中 N 是形如 $\text{diag}(m_1 D, m_2 D, \dots, m_n D, 0, \dots, 0)^T$ 的分块矩阵, 其中 $D = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ 。

本质矩阵的性质

本质矩阵 $E = [t]_{\times} R = S R$, 我们考虑 3×3 反对称矩阵 S , 其可表示为 $S = k U Z U^T$, 根据上述的反对称矩阵的性质, 3×3 的反对称矩阵 $\det S = 0$, 对应上述 $d = 3$, $n = 1$ 的情况, 则矩阵 Z 可以表示为

$$Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

在相差一个尺度因子的情况下, $E = U Z U^T R$ 。为把矩阵 E 写成奇异值分解 $E = U \Sigma V^T$ 的形式, 则需要把 Z 构造为一个对角矩阵和正交矩阵相乘的形式。根据初等行变换

$$[Z|I] = \left[\begin{array}{ccc|ccc} 0 & 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \Rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] = [\text{diag}(1, 1, 0)|W]$$

观察到这里的 W 是一个正交矩阵且 $W^{-1} = W^T$, 有

$$\begin{aligned} ZW &= \text{diag}(1, 1, 0) \\ ZW^T &= -\text{diag}(1, 1, 0) \end{aligned} \quad (9)$$

因此本质矩阵的奇异值分解可以表示为

$$\begin{aligned} E &= UZU^T R = U \text{diag}(1, 1, 0) (W^T U^T R) \equiv U \text{diag}(1, 1, 0) V_1^T \\ E &= UZU^T R = U \text{diag}(1, 1, 0) (-W U^T R) \equiv U \text{diag}(1, 1, 0) V_2^T \end{aligned} \quad (10)$$

所以本质矩阵分解有两种情况, 但都有如下形式

$$E = U \text{diag}(1, 1, 0) V^T \quad (11)$$

并且我们有如下结论 (性质)

一个矩阵是本质矩阵的充要条件是其奇异值中有两个相等且第三个是0

本质矩阵的分解

我们希望通过本质矩阵的SVD分解得到 R 和 \mathbf{t} 。考虑本质矩阵两个SVD分解的情况, 如果我们通过SVD分解得到

$$E = U \Sigma V^T = U \text{diag}(1, 1, 0) V^T$$

设 $E = SR$, S 的形式和上述相同 $S = UZU^T$, 则分解得到的旋转矩阵可以记为 $R = UXV^T$, 这里的 X 是某个旋转矩阵。则有

$$U \text{diag}(1, 1, 0) V^T = E = SR = (UZU^T)(UXV^T) = U(ZX)V^T$$

因此有 $ZX = \text{diag}(1, 1, 0)$, 从而有 $X = W$ 或者 $X = W^T$, 因此旋转矩阵 R 有如下两种情况

$$\rightarrow R_1 = UW^T V^T \quad R_2 = U W V^T \quad (12)$$

这里回顾(10)式, 由于 R 为旋转矩阵, 则有 $\det R = 1$, 因此(10)式中 $\det V_1 = \det(R^T U W) = \det(R^T) \det(U) \det(W) = \det(U)$, 则有 $\det(UV) = 1$, 而对于 $\det V_2 = \det(-R^T U W^T) = -\det(U)$, 则有 $\det(UV) = -1$, 所以对应于 $\det(UV) = -1$ 的情况, (12)式中求得的旋转矩阵的行列式就为 -1 , 所以在结果中需要取反。

接下来我们考虑 \mathbf{t} , 根据 $E = [\mathbf{t}]_{\times} R = SR$, 即我们从 $S = [\mathbf{t}]_{\times}$ 中得到 \mathbf{t} , 考虑

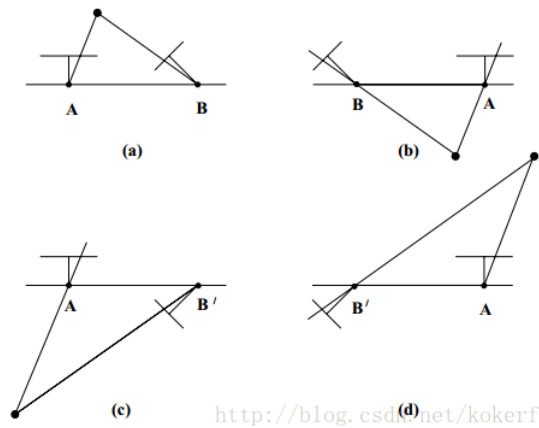
$$S \mathbf{t} = [\mathbf{t}]_{\times} \mathbf{t} = 0$$

则 \mathbf{t} 属于 S 的零空间, 通过对前两行的线性变换, 可以把式子化为奇异值分解的形式, 而考虑到 Z 的最后一行为零, 也就是对应了最小奇异值(0), 因而解就是 $\mathbf{t} = U(0, 0, 1)^T = \mathbf{u}_3$, 即 U 最后一列。考虑到给 \mathbf{t} 乘以一个非零尺度因子得 $\lambda \mathbf{t}$, 有 $[\lambda \mathbf{t}]_{\times} R = \lambda [\mathbf{t}]_{\times} R = \lambda E$, 而对于 E 而言(相差一个尺度因子)这种情况也是等效的, 而对于 \mathbf{t} 而言, 当 $\lambda = \pm 1$ 时, 其物理上的意义(方向)却是不同的。所以, 不考虑尺度因子, 即取 $\|\mathbf{t}\| = 1$, \mathbf{t} 的方向依然无法确定, 所以有两种可能的解。

综上, 本质矩阵的分解一共有4种可能的解, 即已知本质矩阵 $E = U \text{diag}(1, 1, 0) V^T$ 和第一个相机矩阵 $P = [I|0]$, 则第二个相机矩阵 P' 有如下4种可能的解

$$\begin{aligned} P' &= [UWV^T | \mathbf{u}_3] & \text{or} & & [UW^T V^T | \mathbf{u}_3] & (\lambda = 1) \\ &[UWV^T | -\mathbf{u}_3] & \text{or} & & [UW^T V^T | -\mathbf{u}_3] & (\lambda = -1) \end{aligned} \quad (13)$$

下图的四种情况就是上述四种解对应的两个相机之间的关系。其实这四种情况中只有一种是符合实际的解, 只需要根据上述的解根据三角法去计算3D点的坐标, 只有当两个相机观测到3D点都在前方, 也就是深度都为正, 才是最终的解。



<http://blog.csdn.net/kokerf>

下面是ORB-SLAM2中本质矩阵分解的代码

```
1 void Initializer::DecomposeE(const cv::Mat &E, cv::Mat &R1, cv::Mat &R2, cv::Mat &t)
2 {
3     cv::Mat u,w,v;
4     cv::SVD::compute(E,w,u,v);
5
6     u.col(2).copyTo(t);
7     t=t/cv::norm(t);
8
9     cv::Mat W(3,3,CV_32F,cv::Scalar(0));
10    W.at<float>(0,1)=-1;
11    W.at<float>(1,0)=1;
12    W.at<float>(2,2)=1;
13
14    R1 = u*W*v;
15    if(cv::determinant(R1)<0)
16        R1=-R1;
17
18    R2 = u*W.t()*v;
19    if(cv::determinant(R2)<0)
20        R2=-R2;
21 }
```

以及OpenCV中的代码

```
1 void cv::decomposeEssentialMat( InputArray _E, OutputArray _R1, OutputArray _R2, OutputArray _t )
2 {
3     Mat E = _E.getMat().reshape(1, 3);
4     CV_Assert(E.cols == 3 && E.rows == 3);
5
6     Mat D, U, Vt;
7     SVD::compute(E, D, U, Vt);
8
9     if (determinant(U) < 0) U *= -1;
10    if (determinant(Vt) < 0) Vt *= -1;
11
12    Mat W = (Mat_<double>(3, 3) << 0, 1, 0, -1, 0, 0, 0, 0, 1);
13    W.convertTo(W, E.type());
14
15    Mat R1, R2, t;
16    R1 = U * W * Vt;
17    R2 = U * W.t() * Vt;
18    t = U.col(2) * 1.0;
19
20    R1.copyTo(_R1);
21    R2.copyTo(_R2);
22    t.copyTo(_t);
23 }
```

之后对四个解做判断的代码在这里，篇幅过长则不贴出

- ORB-SLAM2,主要在CheckRT这个函数
- OpenCV

参考

- [Multiple View Geometry in Computer Vision II, 9.6](#)
 - [Properties of antisymmetric matrices](#)
 - [Camera Computation and the Essential Matrix](#)
-