深蓝学院
shenlanxueyuan.com

第四章 相机模型 非线性优化

主讲人 罗 瑞

纲要

深蓝学院
shenlanxueyuan.com

第一部分： 图像去畸变

第二部分： 双目视差的使用

第三部分： 矩阵运算微分

第四部分： 高斯牛顿法的曲线拟合实验

* 批量最大似然估计

图像去畸变

/* 由像素平面上一点（u,v), 求出未畸变情况下对应像平面坐标（x,y),
再由畸变方程得到该（x,y)在畸变情况下的投影的（u_distorted,v_distorted) */

```
double x = (u - cx) / fx,   y = (v - cy) / fy;
double r = sqrt(x * x + y * y);
double x_distorted = x * (1 + k1 * r * r + k2 * r * r * r * r) + 2 * p1 * x * y + p2 * (r * r + 2 * x * x);
double y_distorted = y * (1 + k1 * r * r + k2 * r * r * r * r) + p1 * (r * r + 2 * y * y) + 2 * p2 * x * y;
double u_distorted = fx * x_distorted + cx;
double v_distorted = fy * y_distorted + cy;
```

/* 将同一个（x,y) 的（u_distorted,v_distorted) 对应的像素值赋予（u,v)，就是在将该像素值由畸变位置还原到原位置*，遍历该图的所有坐标（u,v),就能得到去掉变形的图像*/

图像去畸变

深蓝学院 shenlanxueyuan.com

```
// 赋值 (最近邻插值)
if (u_distorted >= 0 && v_distorted >= 0 && u_distorted < cols && v_distorted < rows)
{
    image_undistort.at<uchar>(v, u) = image.at<uchar>((int) v_distorted, (int) u_distorted);
}
else
{
    image_undistort.at<uchar>(v, u) = 0;
}
```

/*对 v 行 u 列的这个像素赋值，注意取整，(v, u)对应的畸变坐标如果超出了图像范围找不到像素值时，取0.

双目视差的使用

假设双目计算的视差已经给定,根据双目模型,画出图像对应的点云,并显示到 Pangolin 中

```
Vector<Vector4d, Eigen::aligned_allocator<Vector4d>> pointcloud; // 生成点云
Vector4d point(0, 0, 0, left.at<uchar>(v, u) / 255.0); // 前三维为xyz,第四维为颜色
```

因此要得到点云，要求出对于每个像素点（v,u），其空间点的深度 z = fb/d.

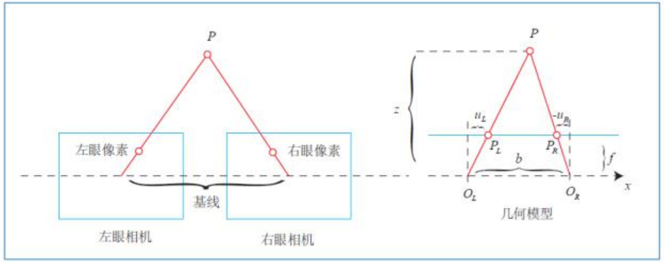
1. 针孔相机模型

- 双目模型
  - 左右相机中心距离称为基线
  - 左右像素的几何关系：

$$\frac{z-f}{z} = \frac{b-u_L+u_R}{b}.$$

- 整理得

$$z = \frac{fb}{d}, \quad d = u_L - u_R.$$



d称为视差（disparity），描述同一个点在左右目上成像的距离
d最小为1个像素，因此双目能测量的z有最大值：fb
虽然距离公式简单，但d不容易计算

双目视差的使用

```
//视差图disparity中(v,u)点，带有视差信息d，其深度d用像素值来表达0-255之间
 unsigned int d=disparity.at<uchar>(v,u);
//视差d=0时表示该点没有深度不用估计
 cout<<d<<endl;
          //基线长度b=0.573  Z=fb/d  f是内参--焦距，b是基线长度，d是视差
 double z=(fx*b)/d;
 point[2]=z;
 point[0]=(u-cx)*z/fx;
 point[1]=(v-cy)*z/fy;
 pointcloud.push_back(point);
```

矩阵运算微分

1. 矩阵 A ∈ R _(N × N) ,那么 d(Ax)/dx 是什么

$$A=\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ \vdots & & & & \vdots \\ a_{n1} & \cdots & & & a_{nn}\end{bmatrix} \qquad x=\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}$$

$$\frac{\partial Ax}{\partial x} = \partial \begin{bmatrix} a_{11}x_1+a_{12}x_2+\cdots a_{1n}x_n \\ a_{21}x_1+a_{22}x_2+\cdots a_{2n}x_n \\ \vdots \\ a_{n1}x_1+\cdots+\cdots a_{nn}x_n\end{bmatrix} \Big/ \partial x$$

Ax 为 n×1 维

对X的k分量求导:

$\frac{\partial AX}{\partial x_k}$ ⇒ 只有包含 x_k 的项，即 $a_{ik}x_k$ 对 x_k 求导有值

$$\frac{\partial AX}{\partial x_k} = \begin{bmatrix} 0+\cdots & a_{1k} & +0 \\ \vdots & a_{2k} & \vdots \\ & \vdots & \\ 0+\cdots & a_{nk} & +0\end{bmatrix} = \vec{a_k}$$

. A的第k列

若 $\vec{X}$ 为行向量. (X^T是行向量)

则 $\frac{\partial AX}{\partial X^T} = \left[\frac{\partial AX}{\partial x_1}, \frac{\partial AX}{\partial x_2}, \frac{\partial AX}{\partial x_3}, \cdots, \frac{\partial AX}{\partial x_n}\right] = [\vec{a_1},\vec{a_2},\cdots,\vec{a_n}] = A$

转置关系

但 $\vec{X}$ 为列向量:

所以 $\frac{\partial AX}{\partial X} = \begin{bmatrix}\frac{\partial AX}{\partial x_1}\\ \frac{\partial AX}{\partial x_2} \\ \vdots \\ \frac{\partial AX}{\partial x_n}\end{bmatrix} = \left(\frac{\partial AX}{\partial X^T}\right)^T = A^T$

$$\frac{\partial Ax}{\partial x} = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{1n} & a_{2n} & \cdots & a_{nn}\end{bmatrix} = A^T$$

矩阵运算微分

2. 矩阵 A ∈ R_(N ×N) ,那么 d(x ^T Ax)/dx 是什么

$$\frac{\partial x^T A x}{\partial x} = \left[\frac{\partial x^T A x}{\partial x_1} \quad \frac{\partial x^T A x}{\partial x_2} \quad \cdots \quad \frac{\partial x^T A x}{\partial x_n} \right]$$

先对x的第k个分量求导，结果如下：

$$\begin{aligned}\frac{\partial x^T A x}{\partial x_k} &= \frac{\partial \sum_{i=1}^{n} \sum_{j=1}^{n} x_i A_{ij} x_j}{\partial x_k} \\ &= \sum_{i=1}^{n} A_{ik} x_i + \sum_{j=1}^{n} A_{kj} x_j \\ &= a_k^T x + a_k' x\end{aligned}$$

可以看出第一部分是矩阵A的第k列转置后和x相乘得到，第二部分是矩阵A的第k行和x相乘得到，排列好就是:

$$\frac{\partial x^T A x}{\partial x} = A^T x + Ax$$

矩阵运算微分

3．证明：$x^TAx = tr(Axx^T)$

证明:

设a,b都是n维列向量，显然有

$$ab^T = \begin{bmatrix} a_1b_1 & a_1b_2 & \dots & a_1b_n \\ a_2b_1 & a_2b_2 & \dots & a_2b_n \\ \dots & \dots & \dots & \dots \\ a_nb_1 & a_nb_2 & \dots & a_nb_n \end{bmatrix}$$

$$b^Ta = \sum_{i=1}^{n} a_ib_i$$

显然，可以得到:

$$tr(ab^T) = b^Ta$$

令 $a = Ax, b = x$ 可得

$$tr(Axx^T) = tr((Ax)x^T) = x^TAx$$

表4　几种迹函数的微分矩阵与梯度矩阵的对应关系

迹函数 $f(\boldsymbol{X})$	微分矩阵 $\mathrm{d}f(\boldsymbol{X})$	梯度矩阵 $\partial f(\boldsymbol{X})/\partial \boldsymbol{X}$
$\mathrm{tr}(\boldsymbol{X})$	$\mathrm{tr}(\boldsymbol{I}\mathrm{d}\boldsymbol{X})$	$\boldsymbol{I}$
$\mathrm{tr}(\boldsymbol{X}^{-1})$	$-\mathrm{tr}(\boldsymbol{X}^{-2}\mathrm{d}\boldsymbol{X})$	$(\boldsymbol{X}^{-2})^{\mathrm{T}}$
$\mathrm{tr}(\boldsymbol{A}\boldsymbol{X})$	$\mathrm{tr}(\boldsymbol{A}\mathrm{d}\boldsymbol{X})$	$\boldsymbol{A}^{\mathrm{T}}$
$\mathrm{tr}(\boldsymbol{X}^2)$	$2\mathrm{tr}(\boldsymbol{X}\mathrm{d}\boldsymbol{X})$	$2\boldsymbol{X}^{\mathrm{T}}$
$\mathrm{tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})$	$2\mathrm{tr}(\boldsymbol{X}^{\mathrm{T}}\mathrm{d}\boldsymbol{X})$	$2\boldsymbol{X}$
$\mathrm{tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X})$	$\mathrm{tr}\left[\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})\mathrm{d}\boldsymbol{X}\right]$	$(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})\boldsymbol{X}$
$\mathrm{tr}(\boldsymbol{X}\boldsymbol{A}\boldsymbol{X}^{\mathrm{T}})$	$\mathrm{tr}\left[(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})\boldsymbol{X}^{\mathrm{T}}\mathrm{d}\boldsymbol{X}\right]$	$\boldsymbol{X}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})$
$\mathrm{tr}(\boldsymbol{X}\boldsymbol{A}\boldsymbol{X})$	$\mathrm{tr}\left[(\boldsymbol{A}\boldsymbol{X}+\boldsymbol{X}\boldsymbol{A})\mathrm{d}\boldsymbol{X}\right]$	$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}$
$\mathrm{tr}(\boldsymbol{A}\boldsymbol{X}^{-1})$	$-\mathrm{tr}\left(\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}^{-1}\mathrm{d}\boldsymbol{X}\right)$	$-(\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}^{-1})^{\mathrm{T}}$
$\mathrm{tr}(\boldsymbol{A}\boldsymbol{X}^{-1}\boldsymbol{B})$	$-\mathrm{tr}\left(\boldsymbol{X}^{-1}\boldsymbol{B}\boldsymbol{A}\boldsymbol{X}^{-1}\mathrm{d}\boldsymbol{X}\right)$	$-(\boldsymbol{X}^{-1}\boldsymbol{B}\boldsymbol{A}\boldsymbol{X}^{-1})^{\mathrm{T}}$
$\mathrm{tr}\left[(\boldsymbol{X}+\boldsymbol{A})^{-1}\right]$	$-\mathrm{tr}\left[(\boldsymbol{X}+\boldsymbol{A})^{-2}\mathrm{d}\boldsymbol{X}\right]$	$-\left[(\boldsymbol{X}+\boldsymbol{A})^{-2}\right]^{\mathrm{T}}$
$\mathrm{tr}(\boldsymbol{X}\boldsymbol{A}\boldsymbol{X}\boldsymbol{B})$	$\mathrm{tr}\left[(\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}+\boldsymbol{B}\boldsymbol{X}\boldsymbol{A})\mathrm{d}\boldsymbol{X}\right]$	$(\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}+\boldsymbol{B}\boldsymbol{X}\boldsymbol{A})^{\mathrm{T}}$

5 高斯牛顿法的曲线拟合实验

```
// 开始Gauss-Newton迭代
int iterations = 100;    // 迭代次数
double cost = 0, lastCost = 0;  // 本次迭代的cost和上一次迭代的cost

chrono::steady_clock::time_point t1 = chrono::steady_clock::now();
for (int iter = 0; iter < iterations; iter++) {

  Matrix3d H = Matrix3d::Zero();             // Hessian = J^T W^{-1} J in Gauss-Newton
  Vector3d b = Vector3d::Zero();             // bias
  cost = 0;

  for (int i = 0; i < N; i++) {
    double xi = x_data[i], yi = y_data[i];  // 第i个数据点
    double error = yi - exp(ae * xi * xi + be * xi + ce);
    Vector3d J; // 雅可比矩阵
    J[0] = -xi * xi * exp(ae * xi * xi + be * xi + ce);  // de/da
    J[1] = -xi * exp(ae * xi * xi + be * xi + ce);  // de/db
    J[2] = -exp(ae * xi * xi + be * xi + ce);  // de/dc

    H += inv_sigma * inv_sigma * J * J.transpose();
    b += -inv_sigma * inv_sigma * error * J;

    cost += error * error;
  }
```

```
// 求解线性方程 Hx=b
   Vector3d dx = H.ldlt().solve(b);
```

批量最大似然估计

深蓝学院 shenlanxueyuan.com

1. 根据所给方程构建误差方程

$$\begin{cases} e_{u,k} = x_k - x_{k-1} - u_k \\ e_{z,k} = y_k - x_k \end{cases}$$

对于不同的时刻写成矩阵形式

$$-e = -[z - Hx] = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} - \underbrace{\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{H} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$k \in [1, 2, 3]$$

$$V_k \begin{cases} e_{u_1} = x_1 - x_0 - u_1 = -u_1 & -x_0 + x_1 \\ e_{u_2} = x_2 - x_1 - u_2 = \quad -u_2 & -x_1 + x_2 \\ e_{u_3} = x_3 - x_2 - u_3 = \quad -u_3 & -x_2 + x_3 \end{cases}$$

$$= -\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad (1)$$

$$N_k \begin{cases} e_{z_1} = y_1 - x_1 \\ e_{z_2} = y_2 - x_2 \\ e_{z_3} = y_3 - x_3 \end{cases} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad (2)$$

$$e = Z - HX = \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}}_{Z} - \underbrace{\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{X}$$

6 * 批量最大似然估计

2.因为 $w_k \sim N(0,Q) \qquad n_k \sim N(0,R)$

$$
\begin{aligned}
(Z-HX)^T W^{-1}(Z-HX) = {}& [(Z-HX)^T Q^{-1}(Z-HX)]_{\substack{x=x_0\\ z=u_1}} + \\
& [(Z-HX)^T Q^{-1}(Z-HX)]_{\substack{x=x_1\\ z=u_2}} + [(Z-HX)^T Q^{-1}(Z-HX)]_{\substack{x=x_2\\ z=u_3}} + \\
& [(Z-HX)^T R^{-1}(Z-HX)]_{\substack{x=x_1\\ z=y_1}} + [(Z-HX)^T R^{-1}(Z-HX)]_{\substack{x=x_2\\ z=y_2}} + \\
& [(Z-HX)^T R^{-1}(Z-HX)]_{\substack{x=x_3\\ z=y_3}}
\end{aligned}
$$

$$
W = \begin{bmatrix} Q & & & & & \\ & Q & & & & \\ & & Q & & & \\ & & & R & & \\ & & & & R & \\ & & & & & R \end{bmatrix}
$$

$x^* \sim e^T w^{-1} e$ ， w^{-1} 信息矩阵，在高斯分布下 $w^{-1} = \Sigma^{-1}$

$$
w = \Sigma = \begin{bmatrix} Q_1 & & & & & \\ & Q_2 & & & & \\ & & Q_3 & & & \\ & & & R_1 & & \\ & & & & R_2 & \\ & & & & & R_3 \end{bmatrix}
$$

3 $$f(x)=\frac{1}{2}(Z-HX)^{\mathrm{T}}W^{-1}(Z-HX)$$

$$\frac{\partial f(X)}{\partial X}=-H^{T}W^{-1}(Z-HX)=0$$

$$H^{T}W^{-1}HX=H^{T}W^{-1}Z$$

其中W为6*6矩阵，上述方程是否有解关键在于$H^{T}W^{-1}H$是否可逆

$$rank(H)=4=rank(H^{T}) \longrightarrow rank(H^{T}W^{-1}H)=4$$

$$X=(H^{T}W^{-1}H)^{-1}H^{T}W^{-1}Z$$

$$X = (H^T W^{-1} H)^{-1} H^T W^{-1} Z$$

$$H = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Longrightarrow \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \Rightarrow rank(H) = 4$$

$W^{-1} = \Sigma^{-1}$，令 $Q_1 = \sigma_1^2$，$R_1 = \sigma_4^2$，则 $\Sigma^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2} & & & \\ & \frac{1}{\sigma_2^2} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma_6^2} \end{bmatrix}$

即 $W^{-1} = \Sigma^{-1} = A^T A$，$A = diag\left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_6}\right) = A^T$

故 $H^T W^{-1} H = H^T A^T A H = (AH)^T (AH) = \|AH\|^2 > 0$，正定，故可逆

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening