

射影变换

陈烁龙

2022 年 7 月 12 日

目录

1	射影变换的描述	1
2	通过高斯牛顿法求解	1
3	通过 SVD 分解求解	1
3.1	利用原始射影矩阵进行求解	1
3.2	求解 $AX=0$	2
4	案例	2

插图

1	射影变换	2
2	射影变换进行文档扫描处理	3

表格

摘要

是的，没错，本次要分享的是射影变换，其是最一般的几何变换。

关键词：射影变换

1 射影变换的描述

射影变换是最一般的几何变换。对于二维平面的变换而言，其有 8 个自由度：

$$P' = \begin{pmatrix} a' & b' & c' \\ d' & e' & f' \\ g' & h' & i' \end{pmatrix}$$

注意到：虽然矩阵有 9 个元素，但是如果将矩阵最后一个元素归化为 1，则一共有 8 个元素。我们将矩阵进行归化，同时除以 i' ，则有：

$$P = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix}$$

假设我们有从点 $p_1(x_1, y_1)$ 到点 $p_2(x_2, y_2)$ 的射影变换，则有：

$$\alpha p_2 = P p_1 \rightarrow \alpha \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

注意： α 为一个比例因子，因为最后一个乘积出来的元素不会为 1：

$$g x_1 + h y_1 + 1 \neq 1$$

2 通过高斯牛顿法求解

基于上文的公式，我们有：

$$\rightarrow x_2 = \frac{a x_1 + b y_1 + c}{g x_1 + h y_1 + 1}$$

$$\rightarrow y_2 = \frac{d x_1 + e y_1 + f}{g x_1 + h y_1 + 1}$$

这里我们使用比值的方式，消除了比例因子 α 。

进一步，我们有：

$$\rightarrow 0 = a x_1 + b y_1 + c - x_1 x_2 g - x_2 y_1 h - x_2$$

$$\rightarrow 0 = d x_1 + e y_1 + f - x_1 y_2 g - y_1 y_2 h - y_2$$

我们定义我们的误差函数为：

$$e(X) = \begin{pmatrix} a x_1 + b y_1 + c - x_1 x_2 g - x_2 y_1 h - x_2 \\ d x_1 + e y_1 + f - x_1 y_2 g - y_1 y_2 h - y_2 \end{pmatrix}$$

其中：

$$X = (a \ b \ c \ d \ e \ f \ g \ h)^T$$

那么，误差函数对待求参数的雅可比矩阵为：

$$\frac{\partial e(X)}{\partial X} = \begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 x_2 & -x_2 y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1 y_2 & -y_1 y_2 \end{pmatrix}^T$$

采用高斯牛顿法，即可求解得到射影变换矩阵 P ：

$$J(X) J^T(X) \Delta X = -J(X) e(X)$$

3 通过 SVD 分解求解

3.1 利用原始射影矩阵进行求解

当我们不归化射影矩阵时，有：

$$\beta p_2 = P p_1 \rightarrow \beta \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} a' & b' & c' \\ d' & e' & f' \\ g' & h' & i' \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

得到：

$$\rightarrow x_2 = \frac{a' x_1 + b' y_1 + c'}{g' x_1 + h' y_1 + i'}$$

$$\rightarrow y_2 = \frac{d' x_1 + e' y_1 + f'}{g' x_1 + h' y_1 + i'}$$

最后化简可以得到：

$$\begin{pmatrix} x_1 & 0 \\ y_1 & 0 \\ 1 & 0 \\ 0 & x_1 \\ 0 & y_1 \\ 0 & 1 \\ -x_1x_2 & -x_1y_2 \\ -x_2y_1 & -y_1y_2 \\ -x_2 & -y_2 \end{pmatrix}^T \begin{pmatrix} a' \\ b' \\ c' \\ d' \\ e' \\ f' \\ g' \\ h' \\ i' \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

这是对于一个点对构造的方程，多个点对时，用相同的方法构建。利用 SVD 法求解的时候，会有约束：

$$\|X\|^2 = 1$$

3.2 求解 $AX=0$

对于一个 m 行、 n 列的矩阵 $A_{m \times n}$ ，可以通过 SVD 分解，得到：

$$A_{m \times n} = U_{m \times m} \Sigma_{m \times n} V_{n \times n}^T$$

且满足：

$$\begin{cases} U^T U = I \\ V^T V = I \end{cases}$$

易知，当 $m < n$ 时，该方程有无数多解，这是我们不感兴趣的。通常使用 SVD 方法求解该种问题时，我们都会有一个额外的条件：

$$\|X\|^2 = 1$$

即：待求参数向量的二范数为 1。在该种情况下，我们的最小二乘解为：

$$X = Col(V, n)$$

即：解为 V 矩阵的最后一列。

举例，对于方程：

$$x + y = 0 \rightarrow \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \end{pmatrix}$$

在使用 SVD 分解法求解的时候，我们会附加条件：

$$x^2 + y^2 = 1$$

最后的结果为：

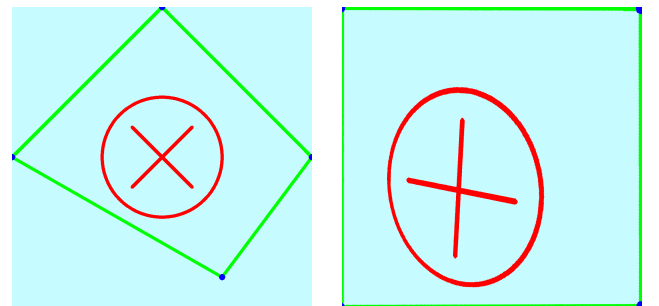
$$\begin{aligned} U_{1 \times 1} &= (-1) \\ \Sigma_{1 \times 2} &= (\sqrt{2} \ 0) \\ V_{2 \times 2} &= \begin{pmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} \end{aligned}$$

显然有：

$$\begin{cases} A = U \Sigma V^T \\ U^T U = 1 \\ V^T V = 1 \\ Col(V) = \begin{pmatrix} x & y \end{pmatrix}^T \\ x^2 + y^2 = 1 \end{cases}$$

4 案例

我们首先构造一个图像，然后在该图像上取 4 个点构造一个感兴趣区域（绿色框所示范围）。然后把在这个感兴趣区域通过射影变换，让其充满整个空间。具体的结果如下图所示。



(a) 原始图像

(b) 射影变换后的图像

图 1: 射影变换

具体的射影变换细节如下表所示。

Listing 1: 高斯牛顿法求解

```
1 // 求解得到的射影矩阵
2 -- trans:
```

```
3 0.600816 0.600816 -300.408
4 -0.946573 0.94468 473.286
5 -0.000346103 -0.000106854 1
6
7 // 利用求解得到的射影矩阵将pc2转为pc1的结果
8 -- p1: [500, 0], p2: [0, 0], trans(p2): [500, -5.00253e-14]
9 -- p1: [999, 500], p2: [999, 0], trans(p2): [999, 500]
10 -- p1: [700, 900], p2: [999, 999], trans(p2): [700, 900]
11 -- p1: [0, 500], p2: [0, 999], trans(p2): [1.07613e-13, 500]
```

可以看到，我们基于射影变换，将原始图片中的纸张进行反射影变换，得到了纸张的正射投影图像，对于人眼而言，更加易于识别。

Listing 2: SVD 法求解

```
1 // 求解得到的射影矩阵
2 -- trans:
3 0.00107178 0.00107178 -0.535889
4 -0.00168856 0.00168519 0.844282
5 -6.17404e-07 -1.90615e-07 0.00178387
6
7 // 利用求解得到的射影矩阵将pc2转为pc1的结果
8 -- p1: [500, 0], p2: [0, 0], trans(p2): [500, 2.02693e-11]
9 -- p1: [999, 500], p2: [999, 0], trans(p2): [999, 500]
10 -- p1: [700, 900], p2: [999, 999], trans(p2): [700, 900]
11 -- p1: [0, 500], p2: [0, 999], trans(p2): [-4.26567e-11,
    500]
```

另外，为了比较明显的看到该射影变换的用处，本处取了一个比较实际的例子来进行展示。我们知道，相机拍摄场景实际上就是一种射影变换。为此，我们可以对像片中感兴趣区域进行射影变换，如下图所示：



(a) 原始图像



(b) 射影变换后的图像

图 2: 射影变换进行文档扫描处理