全景拼接

- 总体步骤
  - 找到特征点
  - 建立特征点的描述符（比如向量）
  - 匹配对应的描述符
  - 求解变换矩阵
  - 图片乘几何变换矩阵，拼在一起
- 一些线性变换的表示
  - 旋转变换：$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ cos全正sin相反，一个自由度
  - 平移变换-欧氏变换：加一个$[x, y]^T$即可。为了避免相加这个问题，采用$[x, y, 1]^T$的方式来表示一个变换向量（这里叫齐次向量），这样就可以把变换矩阵写成 $\begin{bmatrix} \cos\theta & -\sin\theta & s \\ \sin\theta & \cos\theta & t \\ 0 & 0 & 1 \end{bmatrix}$。没什么道理，单纯是为了让变换过程全是矩阵乘法方便而建立的。三个自由度：s，t，θ.
  - 相似变换：基于欧氏变换，旋转完再对图形进行放大缩小$\begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$,将图片放大或缩小s倍。四个自由度：$s, \theta, t_x, t_y$
  - 仿射变换：左上角为什么非要是旋转和平移呢？我们放宽为$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$,要求矩阵为非奇异矩阵（如果奇异了那么这个行列式值为0，矩阵秩就小于3，那么二维平面就被降维为一维了。）这样就可以实现在z=1的平面上修改z的方向投影到平面的内容了。仿射变换的自由度为6
  - 射影变换：要求3×3矩阵非奇异。自由度最高的变换方式。由于一般来说齐次坐标可以归一化，意味着第三列整体扩大缩小是同一个向量，因此8个自由度
- 检测特征点
  - 哈里斯角点检测：对于一个小区域x，将这个小区域向上下左右平移一个小位移dx，如果区域内的像素有很明显变化的话，说明这是一个特征点。这样可以最大限度避免检测导致的误差。
    - 为了评估像素变化的明显程度，我们使用一个函数来记录变化

      $$s_w(\Delta x, \Delta y) = \sum_{(x_i, y_i \in W)} (f(x_i, y_i) - f(x_i + \Delta x, y_i + \Delta y))^2$$

      其中f代表坐标x，y处像素值，平方求和取差以比较区别。

      由泰勒展开，我们可以展开如下

      $$f(x_i + \Delta x, y_i + \Delta y) \approx f(x_i, y_i) + I_x(x, y)\Delta x + I_y(x, y)\Delta y$$

      ,其中I是f的偏导数。那么我们就可以得到s的表达式：

$$u=\Delta x, v=\Delta y$$

$$s=[u,v]\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}[u,v]^T$$

中间的矩阵是半正定的，一般来说是正定的（0的时候太难取到了）。证明过程如下：

$$\text{To prove this conclusion,We only need to prove the following definition}$$

$$for\ \forall\alpha=\begin{bmatrix} x \\ y \end{bmatrix}, letA=\sum_{x_i,y_i\in W}(I_x^2), B=\sum_{x_i,y_i\in W}(I_x\times I_y), C=\sum_{x_i,y_i\in W}(I_y^2)$$

$$\alpha^TM\alpha=[x \quad y]\times\begin{bmatrix} A & B \\ B & C \end{bmatrix}\times\begin{bmatrix} x \\ y \end{bmatrix}=[x \quad y]\times\begin{bmatrix} Ax+By \\ Bx+Cy \end{bmatrix}=Ax^2+2Bxy+Cy^2\geq 0$$

$$when\ x\times y\geq 0,\ \text{it is obvious that}\ \alpha^TM\alpha\geq 0.$$

$$\text{So what we need to prove is that when}\ x\times y<0, \text{the above conclusion is still correct.}$$

$$when\ x\times y<0, \text{Origin formula} = Ax^2-2B\left|xy\right|+Cy^2\geq 2(\sqrt{(AC)}-B)xy.$$

$$\text{So to prove the conclusion, we only need to prove}\sqrt{(AC)}\geq B.$$

$$\sqrt{(AC)}\geq B.\iff\sqrt{(\sum_{x_i,y_i\in W}(I_x^2)\times\sum_{x_i,y_i\in W}(I_y^2))}\geq\sum_{x_i,y_i\in W}(I_x\times I_y),$$

$$\text{And according to Cauchy's inequality, this conclusion has already been proved}$$

$$\text{So our conclusion is correct, when and only when every}\frac{I_{x_i}}{I_{y_i}}\ \text{is equal and}Ax^2=Cy^2.$$

简单来说，将y看成常量，然后证明Δ小于等于0即可。用一个柯西就行。

对于正定矩阵来说，可以将它化成一个椭圆：

$$s=[y_1,y_2]\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}[y_1,y_2]^T=\frac{y_1^2}{1/\lambda_1}+\frac{y_2^2}{1/\lambda_2},$$

$$\text{当变化幅度为单位1时，原式就是一个椭圆}$$

公式的意思是在整个椭圆一周，它们的变化幅度相同（都是1）。因此半轴越短变化越明显，越应该当作特征点。因此 λ越大越好

实际操作中，判断这个点是不是角点太难算了，一般用经验公式

$$r(x)=det(M(x))-k(trace(M(x)))^2$$

其中k通常是0.06.

为了避免在一个角点周围检测出多个角点，进行非极大抑制。确保一块区域只有一个角点。

- 这种方法不具备尺度不变性。由于像素变化多少是方格和方格之间比较的，意味着当图像放大十倍时原来的角点现在可能变成平坦区域了

- 建立描述符
    - 最简单的就是将刚刚哈里斯角点检测出来的图像拉成一个向量，这个向量作为描述符
    - 没有尺度不变性和旋转不变性
    - 比较方法：SSD（L2范数）：两个坐标差平方之和；SAD（L1范数）：两个坐标差绝对值之和，皮尔森系数相关：

      皮尔森系数如下：

$$\frac{1}{n}\times\frac{(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}$$

其中σ_x是方差开方。这个值域在[-1,1]之间，因此可以用1减去皮尔森系数变成[0,2]来比较相似性（越大越像）或者用arccos（皮尔森系数）来比较，值域在[0,π]

- 要匹配两者，要满足下面三个要求
  1. 对于你我来说，我们之间的相似度大于设定阈值
  2. 对于你来说，我是所有点之间最像的；对于我来说，你是所有点之间最像的（双向奔赴）
  3. 我和你之间的相似度要比我和另一个的相似度要高一些。这里高一些指的是相似度之比大于一个阈值（比如一点几）（不能模棱两可）

这部分是最基础的特征点比较。由于哈里斯角点检测没有尺度不变性，用块转成向量连旋转不变性都没有了，因此我们需要引入另一种特征点：SIFT（scale-invariant feature transform,尺度不变特征变换）特征点。

- 哈里斯角点最大的问题是窗口大小（特征尺度）是人为设定的，因此解决不了尺度不变性。我们希望新的算法可以做到自适应尺度，图被放大尺度大，图被缩小尺度小。因此我们对于每个点都想办法找到一个函数，自变量是在该点处尺度大小，输出值为量化该点适不适合做角点。这样我们就可以先找到这个点输出最大值（此时尺度应该是自适应的尺度），然后再将所有最大值放在一起比较来找到最适合做角点的几个点。最后将尺度归一化进行比较。
- 因此我们希望这个函数只有一个极大值点。为此，我们挑了LoG作为这个特殊的函数，输入一个区域，输出该处作为角点的合适值。LoG就是正态函数对x,y分别求两次偏导然后相加，最后再乘一个σ^2。

  $$LoG = \frac{x^2+y^2-2\sigma^2}{2\pi\sigma^4}e^{-\frac{x^2+y^2}{2\sigma^2}} //$$

  其中σ是一组设定的算子。这样我们就得到了一个三维尺度空间，x，y以及尺度向量σ。对于每个点，我们都可以在σ轴上找到一个极值点。对于每个极值点，如果它是3×3×3空间中最大的那个点，那么它就会被放入特征点候选点。后面的匹配方法同哈里斯角点检测。由于极值点中极大值代表该斑点亮，极小值代表该斑点暗，因此这种检测方法不受光照影响，不受尺度影响。
- 然后需要对相关的点处理。要过滤掉边缘点和低对比度响应点。现在这个函数导出的的点对边缘响应也强（边缘也是非常好的极值点）。边缘的特征是对于一个椭圆光斑，短轴很短长轴很长。我们要剔除这部分内容。因此我们可以计算长短轴比值。长短轴与两个特征值有关，记比值为r，因此可得：

  $$tr(H) = \lambda_1+\lambda_2, det(H) = \lambda_1\lambda_2,$$
  $$\frac{tr(H)^2}{det(H)} = \frac{(\lambda_1+\lambda)^2}{\lambda_1\lambda_2} = \frac{(r+1)^2}{r}$$

  这样我们就求出了r。如果r比一个指定的阈值要大就说明这个点很可能是边缘的响应点，应该剔除。
- 接下来我们要建立描述子。由于上面框范围的方法只框出了一个区域（光斑），没有处理旋转不变性。我们需要为每一个光斑设置出一个方向，然后将方向归一化再比对相似度。
- 最简单的方向就是求特征点的梯度向量。求法如下
  - 对于特征点附近的一块区域（在论文中，选择的是关键点附近3σ范围内的像素点）中的所有像素点，我们都可以求出该点的梯度方向以及幅度。获得幅度后，我们会对每个像素点的方向根据距离关键点的距离和幅度进行加权构建直方图（实际操作中会采用高斯函数进行加权）。论文中采取的方式是取36个方向做直方图，然后取出这块区域中方向幅度和最大的方向作为特征区域方向。（为了鲁棒性，论文还添加了要求：对于幅度和超过主值80%的方向也会被算作另一个主方向。这样同一个坐标的像素可以作为不同方向的关键点）。完成这些步骤后我们需要把关键点区域旋转成统一方向再进行匹配。
- 下一步求特征点描述子（即特征向量）。将特征点邻域划成4*4的一个区域，每个区域再建立一个8个方向的直方图来描述这个区域的方向权值。这样对于一个点来说就有$4\times4\times8=128$维的特征向量。然后同上面哈里斯角点匹配法来匹配特征点，就可以构建一一对应了
  - 这样我们就从一张图片提取出了如下内容
    - n个特征点的特征向量：$n\times128$矩阵

    - n个特征点的特征尺度空间大小：n×1矩阵
    - n个特征点的方向参数（方向的角度）:n×1矩阵
    - n个特征点的坐标：n×2矩阵
- 下一步我们要求解矩阵变换方程。
  - 首先要理解矩阵向量求导。对于f和x来说，
    - 如果f是vector，x是标量，那么是[f1(x),f2[x]...,fn(x)],反过来就是[f(x1),f(x2),...,f(xn)].如果f是矩阵或者x是矩阵就是把vector扩大成矩阵
    - 如果f是vector，x也是vector，那么f(x)可以理解为两个vector相乘成为的matrix（因为是要一一对应）。
    - 因此在求导的适合一个是横另一个必须是竖着的，比如x=[x1,x2],y=[y1,y2],yn=f(xn),那么：

$$\frac{dy}{dx^T} = \begin{bmatrix} \frac{dy_1}{dx_1} & \frac{dy_2}{dx_1} \\ \frac{dy_1}{dx_2} & \frac{dy_2}{dx_2} \end{bmatrix}, \frac{dy^T}{dx} = \begin{bmatrix} \frac{dy_1}{dx_1} & \frac{dy_1}{dx_2} \\ \frac{dy_2}{dx_1} & \frac{dy_2}{dx_2} \end{bmatrix}$$

      因此在求导的时候需要有个T。
  - 一些常用的矩阵求导公式：

$$\frac{da^Tx}{dx} = \frac{dx^Ta}{dx} = \begin{bmatrix} \frac{d(a_1x_1+a_2x_2+...+a_nx_n)}{dx_1} & \frac{d(a_1x_1+a_2x_2+...+a_nx_n)}{dx_2} \cdots & \frac{d(a_1x_1+a_2x_2+...+a_nx_n)}{dx_n} \end{bmatrix}$$

$$= [a_1, a_2, \dots a_n] = a$$

$$\frac{dAx}{dx^T} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = A$$

$$\frac{dx^TA^T}{dx} = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} = A^T$$

$$\frac{dx^TAx}{dx} : x = [x_1, x_2]^T, A = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}, Ax = \begin{bmatrix} a_{11}x_1 + a_{21}x_2 \\ a_{12}x_1 + a_{22}x_2 \end{bmatrix},$$

$$x^TAx = a_{11}x_1^2 + a_{21}x_1x_2 + a_{12}x_1x_2 + a_{22}x_2,$$

$$\frac{dx^TAx}{dx} = \begin{bmatrix} 2a_{11}x_1 + a_{21}x_2 + a_{12}x_2 \\ a_{21}x_1 + a_{12}x_1 + 2a_{22}x_2 \end{bmatrix} = (A + A^T)x$$

$$\text{由于都是二次型}, \frac{dx^Tx}{dx} = 2x$$

  - 我们要求的就是射影变换的矩阵（八个自由度）。要确保射影变换，必须保证图中所有点在同一平面上。（如果有深度值问题的话就没办法用摄影变换矩阵来将两张图拼在一起了）。
  - 如果原图上的点(x,y)和现在要拼接的图的点(u,v)相互匹配的话，他们满足

$$c\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h13 \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$x = \frac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + h_{33}}$$

$$y = \frac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + h_{33}}$$

    (其中分母是齐次坐标归一化的产物)

    将分母乘到等式左边然后化成Ah=0的形式：

$$\begin{bmatrix} u & v & 1 & 0 & 0 & 0 & -ux & -vx & -x \\ 0 & 0 & 0 & u & v & 1 & -uy & -vy & -y \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0$$

由于其实只有8个自由度（h33是需要归一的），可以化成Ah=b的形式：

$$\begin{bmatrix} u & v & 1 & 0 & 0 & 0 & -ux & -vx \\ 0 & 0 & 0 & u & v & 1 & -uy & -vy \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

对于一组点，我们就可以得到两行。那么对于n组点，我们就可以得到2n行。对于这组方程，有解要求左边矩阵秩小于8.然而一般来说一张图不会只有四组点。所以左边的矩阵通常秩大于8.因此要找出0解基本是不可能的。我们换一个方式，找到一组h使得Ah-b最接近0来近似找到矩阵解。

- 线性最小二乘法

  非齐次线性方程求近似解：

  我们可以用$||Ax-b||^2$二范数的方式来衡量最小值h在哪。极值点解法如下：

$$|Ax-b|^2 = (Ax-b)^T(Ax-b) \uparrow$$

$$\frac{d[(Ax-b)^T(Ax-b)]}{dx} = 0 \Rightarrow |Ax-b|^2 \ min.$$

$$\frac{d[(Ax-b)^T(Ax-b)]}{dx} = \frac{d[(x^TA^T-b^T)(Ax-b)]}{dx} = \frac{d(x^TA^TAx - x^TA^Tb - b^TAx + b^Tb)}{dx}$$

$$= \frac{d\,x^T(A^TA)x}{dx} - \frac{dx^TA^Tb}{dx} - \frac{d\,b^TAx}{dx} + \frac{d\,b^Tb}{dx}$$

$$= 2(A^TA)x - A^Tb - A^Tb + 0 = 2(A^TA)x - 2(A^Tb) = 0$$

$$A^TAx = A^Tb \Rightarrow x = (A^TA)^{-1}A^Tb$$

  由于$||Ax-b||^2$是一个凸函数，求出来的极值点就是最小值

  齐次线性方程求近似解：

  对于Ax=0方程来说，由于x=0时是平凡解，我们设置了一个限制条件$x^Tx=1$。我们采用拉格朗日乘子来计算。对于拉格朗日乘子法来说，我们要找到一组x和λ，使得方程导数为0.在这个问题中，即求$||Ax||_2^2 + \lambda(1-||x||_2^2)$这个函数的最小值点。对x和λ求导，令式子为0可得：

$$A^TAx = \lambda x$$
$$x^Tx = 1$$

第一个式子告诉我们，λ这个值就是A^TA这个矩阵的特征值。同时我们可以导出下面这个式子：

$$||Ax||_2^2 = x^TA^TAx = x^T\lambda x = (\lambda\text{是常数})\lambda x^Tx = \lambda$$

因此原式最小值就是A^TA这个矩阵的特征值最小值。接下来就是找到一组解使得Ax=λ就行了

- RANSAC：随机一致采样
  - 很简单，首先随机选几个点，然后用这个点拟合你的模型，最后用模型来检测有几个点在误差范围内是在你的模型上面的。一直随机选直到你的模型能拟合的点超过了阈值或者已经是所有模型中最好的就结束
  - 比如用直线估计离散点，先随机选两个点画一条直线，然后检查有几个点在直线上或者误差范围内在直线上（这些点被称为内点）。一直迭代直到90%的点都在直线上或者这是迭代过程中拟合点个数最多的直线就停止。
  - 在图像拼接中，我们可以通过选择4个以上（5个，6个，只要你的秩大于等于8）的点来计算射影矩阵，然后用矩阵来检查其他特征点在变换后有几个能到对应的特征点。一直选直到你的射影矩阵最优为止。
- 图像插值
  - 简单来说，当你用变换矩阵乘了要拼接的图片时，往往原来是整数坐标的像素点会变成不是整数。这时拼接的时候就采取双线性插值办法：对这个坐标的周围四个点的像素颜色取加权平均（以距离为加权）

单目测量

- 预备知识：射影几何
  - 对于齐次坐标，我们有另一种解释：齐次坐标（x，y，1）代表的是z轴为1的平面处的一点。这样我们可以将这个平面坐标看作是向量（x,y,1)和平面z=1的交点。
    - 因此当我们要将点挪远的时候，我们可以将坐标变成（x，y,0.5），这样就等于交点往外移了两倍。
    - 同时这也是为什么k(x,y,1)齐次坐标代表的是同一个点：因为你本身是相同的一个向量。
  - 那么问题来了：如果齐次坐标是(x,y,0)呢?
    - 在三维坐标中，它被认为是和平面平行。
    - 在射影几何中，它被认为是于平面相交于无穷远点。
    - 因此，对于这些无穷远点来说，坐标为k(x,y,0)
  - 无穷远点构成一条直线。（两个平面相交成一条直线，无穷远点就是两个平行平面交成的直线）
  - 一个点在线上可以看作这个点所在的向量在两个线上的点代表的向量构成的平面上->叉乘为0
    - 因此直线齐次坐标如图：

$$\left(\begin{vmatrix} y_1 & z_1 \\ y_2 & z_2 \end{vmatrix}, \begin{vmatrix} z_1 & x_1 \\ z_2 & x_2 \end{vmatrix}, \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix}\right)^T$$

      点x和直线坐标点乘为0
    - 两线求交点即求一组(x,y,1)，使得该点与两个直线齐次坐标点乘为0，也就是说，两条直线齐次坐标叉乘出来的方向就是x的方向
  - 所谓的射影变换，就是找到一个H，使得$x_1 = Hx_2$对x1上的点都成立。

- 预备知识2：非线性最小二乘法（函数：非线性，目的：找到最小，衡量指标：对误差二乘（平方））
    - 雅可比矩阵：对于F（x），一阶导就是$\frac{dF}{dx_i}$,二阶导就是$\frac{d^2F}{dx_i dx_j}$,这样就构成了一组矩阵，叫雅可比矩阵。
    - 矩阵函数的二阶泰勒展开：

    $$F(x+h) = F(x) + h^T F'(x) + 1/2h^T F'(x)h + O(||h||^2)$$

    - 一阶导为0是驻点，极值点是驻点，驻点不是极值点（看二阶导）。二阶导如果是正定矩阵就是极值点
    - 下降法找极值点
        - 一般分为两步：第一步线找到下降方向，第二步找可以下降多远
        - 两步法
            - 找方向：最速下降法，牛顿法
                - 对于起始点，搜索周围的方向。如果没有可以下降的方向说明到底了，否则就向指定方向移动指定步长。遍历到指定次数为止。
            - 找长度：线搜索法（求步长）
        - 一步法：置信区间法，阻尼法
    - 最速下降法（steepest descent）
        - 每次都对F（x）求导得到F ’（x），当下降方向为-F'(x)时是当前函数的梯度，下降最快。
        - f(x+h)=f(x)+J(x)h,F(x+h)=1/2||f(x+h)||^2.
    - 牛顿法
        - 找到哪个方向更快到达驻点：使x不动，求F'(x+h)=0的h值
        - 用泰勒一阶展开来逼近：F'(x+h)=F'(x)+hF''(x)+O(h^2)=0,h=-F'(x)F''$^{-1}$(x).要是下降而不是反方向上升，二阶导必须正定（否则就是去找局部最大值了）。因此这一方法往往是快收敛到底的时候用的（它没法翻山）。同时如果矩阵半正定，直接没法解。

    - SD+Newton hybrid：大人我全都要
        - 对于海森矩阵非正定时，用最速下降
        - 对于正定矩阵，用牛顿下降
    - 线搜索法：先随便猜一个a，如果步长太短导致下降太慢，a增大；如果步长太大导致越过最低点（比当前值还大），a减小
    - 信赖区间法：用一个近似的二次模型（最简单的有极小值的模型）来模拟一段区间。区间由人为设定（通常是一次下降到最低点所在的横坐标）。在这段近似区间内沿着二次模型下降。区间范围可以迭代：如果下降的快就增大区间，下降的慢就减小区间（防止过头了）
        - 近似的二次模型是用泰勒展开二次项拟合的
        - 用增益比来更新置信区间：

$$\rho = \frac{F(\mathbf{x}) - F(\mathbf{x}+\mathbf{h})}{L(\mathbf{0}) - L(\mathbf{h})}$$

the actual decrease

the predicted decrease

依据 Rho 更新 delta 的方式可以如下：

$$\begin{aligned} &\text{if } \rho < 0.25 \\ &\quad \Delta := \Delta / 2 \\ &\text{elseif } \rho > 0.75 \\ &\quad \Delta := \max\{\Delta, 3 \times \|\mathbf{h}\|\} \end{aligned}$$

    - 阻尼法（Damp）：给h添加一个惩罚项，然后用牛顿法来找到最小值

- 如图：

$$\phi_{\mu}'(\mathbf{h}) = \frac{d\left(L(\mathbf{h}) + \frac{1}{2}\mu\mathbf{h}^T\mathbf{h}\right)}{d\mathbf{h}} = \frac{d\left(F(\mathbf{x}) + \mathbf{h}^T\mathbf{c} + \frac{1}{2}\mathbf{h}^T\mathbf{B}\mathbf{h} + \frac{1}{2}\mu\mathbf{h}^T\mathbf{h}\right)}{d\mathbf{h}}$$

$$= \mathbf{c} + \frac{1}{2}\left(\mathbf{B} + \mathbf{B}^T\right)\mathbf{h} + \mu\mathbf{h} = \mathbf{c} + \mathbf{B}\mathbf{h} + \mu\mathbf{h} = \mathbf{0}$$

$$\Rightarrow \quad \mathbf{h}_{dm} = -(\mathbf{B} + \mu\mathbf{I})^{-1}\mathbf{c} \qquad \textbf{(Eq. 3)}$$

- 一般来说，$B + \mu I$就是正定的，也就是说这个项一定是全局最小值。
- 如果下降的ρ太小，就增大μ来惩罚避免走太快，否则就减小μ来加速

- 高斯牛顿法
  - 由于二阶导的海森矩阵太难求，我们直接处理$||f(x+h)||^2$使得这一部分尽可能的小（没有必要一定要找到等于0的极值点）.这部分算出来是

$$F(\mathbf{x}+\mathbf{h}) \approx L(\mathbf{h}) \equiv \frac{1}{2}\left(\mathbf{f}(\mathbf{x}+\mathbf{h})\right)^T \mathbf{f}(\mathbf{x}+\mathbf{h}) = \frac{1}{2}\mathbf{f}^T\mathbf{f} + \mathbf{h}^T\mathbf{J}^T\mathbf{f} + \frac{1}{2}\mathbf{h}^T\mathbf{J}^T\mathbf{J}\mathbf{h}$$

    对它求导令导数为0就可以求出极小值。

  - 问题在于J^TJ不一定是正定可逆的，算法可能会出错。

- LM法：F'(x)=c,F''(x)=B.当惩罚项很大的时候是最速下降法（乘了一个很小的步长），可以知道离最低点还很远，当惩罚项很小的时候就是牛顿法，说明快到了
  - $F'(x)=J(x)F(x), F''(x)=J^TJ$
- 如果下降慢到比指定值还小（导函数小于阈值），说明至少到了一个极值点。
- 或者，我们也可以评判如果x到一个地方不往前动了（动的距离小于阈值）,说明下降到极值点了

- 单目测量正式开始
  - 我们要求：待测量物体在一个物理平面上（变换至少是射影变换），然后生成图像的平面和该物理平面保持线性几何变换（说明变换是相似变换，不考虑镜头畸变，理想照相过程只有放大缩小以及图片旋转）。
  - 我们要标定的参数有两部分：内参和外参。内参是相机自身的参数，比如镜头畸变（鱼眼相机），以及一些相机制造过程中的误差。我们需要将这部分作为矩阵标定出来。外参标定是理想无畸变的相机生成的图像与世界系下的图像的矩阵变换。
    - 内参：畸变参数，焦距，成像像素大小，成像平面坐标系夹角
    - 外参：相机位置以及旋转角度，物品世界坐标

- 我们将根据下面这个针孔相机模型来进行理解与处理：

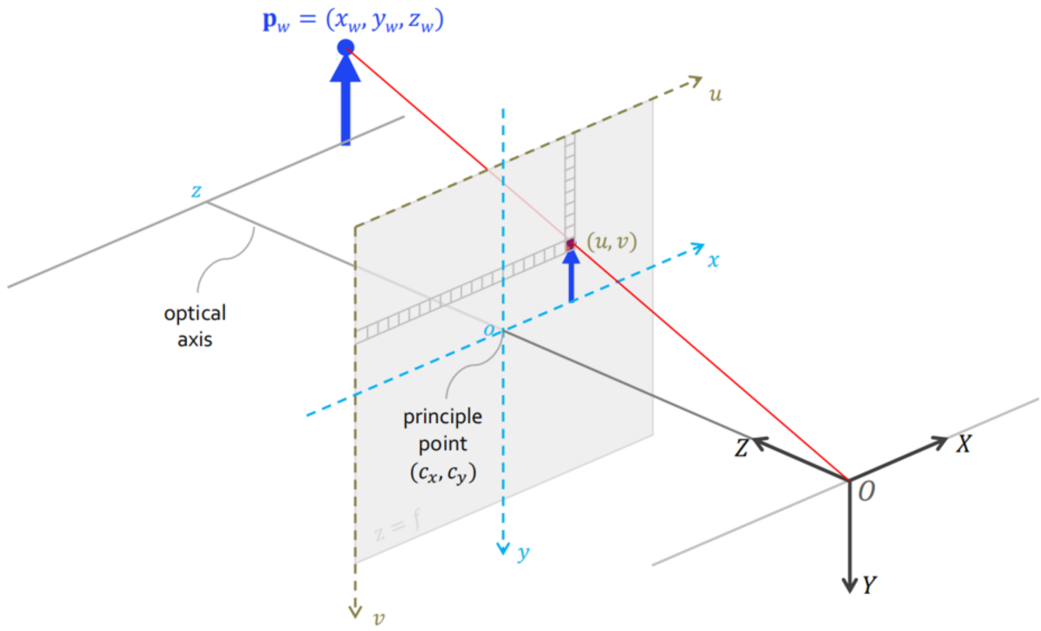
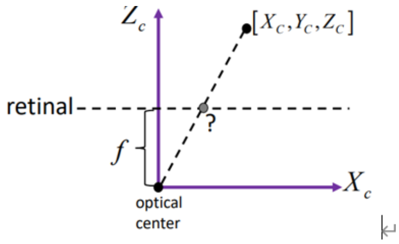


图 10-1：针孔相机模型。

  - pw代表的是世界坐标系下人的坐标（world CS）。中间的平面代表成像平面下的坐标。成像过程的第一步，是将世界坐标系下的人物坐标（pw代表的是蓝色xyz下的坐标）转换成相机坐标系（camera CS）下的坐标（黑色部分的xyz）。因此：

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = R_{3\times3} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + t_{3\times1}, \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = [R\ t]_{3\times4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

  - (其实这部分很像图形学中的视角变换，将世界系移动到相机系就是相机系从世界系0点处变换到指定坐标和方向的逆过程)
  - 下一步是将相机系中的人物投影到成像平面上。对于无畸变相机来说，这时成像平面距离相机为焦距f，因此一步相似可得：



$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f\dfrac{X_C}{Z_C} \\ f\dfrac{Y_C}{Z_C} \end{bmatrix} \xrightarrow{\text{homogeneous form}} \underset{\text{normalized homogeneous}}{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}} = \frac{1}{Z_C}\begin{bmatrix} fX_C \\ fY_C \\ Z_C \end{bmatrix} = \frac{1}{Z_C}\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \underset{\text{inhomogeneous}}{\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix}} \quad (2)$$

- 由于f这部分是内参标定要解决的事情，首先我们先不考虑具体成像到哪个平面，先统一成像到z=1的平面（normalize retinal CS）上，这样我们就得到了归一化成像平面下图像的坐标：

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} \to \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \frac{X_C}{Z_C} \\ \frac{Y_C}{Z_C} \end{bmatrix} \xrightarrow{\text{homogeneous form}} \underbrace{\begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}}_{\text{normalized homogeneous}} = \frac{1}{Z_C} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} \quad (3)$$

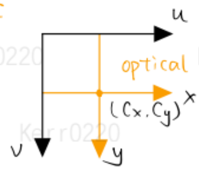
- 这里得到的xn和yn是在归一化成像平面上的坐标。下一步，我们要将这个坐标转换到成像坐标（pixel CS）（即图中的uv，注意，这个平面的z坐标不是1）。如果镜头没有畸变（即uv垂直），那很好，我们只需要首先将这部分坐标投射到z坐标为f的平面上，然后计算单位1的x长度代表几个像素点，（比如单位1长度代表了三个像素，那么(0,1)坐标实际代表的是（0，3））， 最后将0点从画面中心移动到画面左上角（图像坐标系要求）。后两步的矩阵变换如下：

④ Retinal CS → Pixel CS

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & C_x \\ 0 & \frac{1}{dy} & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

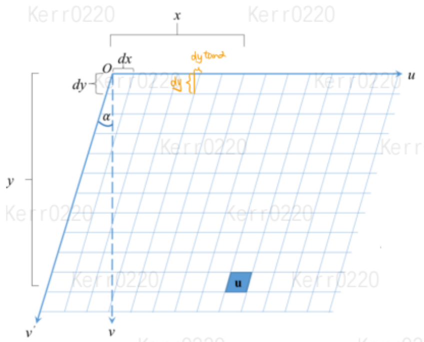
光心在 (Cx, Cy) 处

一个像素在y方向上的距离



- 如果很不幸，成像平面上的轴并不严格垂直，那么就需要加一个tan来判断像素在的地方。（当然，现在的相机都很棒了，通常都几乎是完全垂直的）推导过程如下：

⑤ 成像平面 2个轴之间不严格垂直.



$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \to \begin{bmatrix} x + y\tan\alpha \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \tan\alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & C_x \\ 0 & \frac{1}{dy} & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \tan\alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{dx} & \frac{\tan\alpha}{dx} & C_x \\ 0 & \frac{1}{dy} & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

最后结果的3×3矩阵乘上前面推出来的焦距矩阵就是内参矩阵。

- 我们得到每个参数的意思：

$$P_{4\times1}：\text{世界坐标系（三维）}$$

$$[R\ t]_{3\times4}P_{4\times1}：\text{相机坐标系（三维）}$$

$$\frac{1}{Z_c}[R\ t]_{3\times4}P_{4\times1}：\text{归一化平面坐标（二维）}$$

$$K_{3\times3}\frac{1}{Z_c}[R\ t]_{3\times4}P_{4\times1}：\text{成像平面坐标（二维）}$$

- 以上都是基于无畸变的理想相机进行成像的结果。事实上，当光线进入相机时，会产生径向和切向畸变。建模如下：

  考虑到径向畸变，有模型：
  $$\begin{cases} x_{dr} = x_n\left(1+k_1r^2+k_2r^4+k_3r^6\right) \\ y_{dr} = y_n\left(1+k_1r^2+k_2r^4+k_3r^6\right) \end{cases}$$

  考虑到切向畸变，有模型：
  $$\begin{cases} x_{dt} = x_n+\left(2\rho_1x_ny_n+\rho_2\left(r^2+2x_n^2\right)\right) \\ y_{dt} = y_n+\left(2\rho_2x_ny_n+\rho_1\left(r^2+2y_n^2\right)\right) \end{cases}$$

  结合起来，有模型：
  $$\begin{cases} x_d = x_n\left(1+k_1r^2+k_2r^4\right)+2\rho_1x_ny_n+\rho_2\left(r^2+2x_n^2\right)+x_nk_3r^6 \\ y_d = y_n\left(1+k_1r^2+k_2r^4\right)+2\rho_2x_ny_n+\rho_1\left(r^2+2y_n^2\right)+y_nk_3r^6 \end{cases}$$

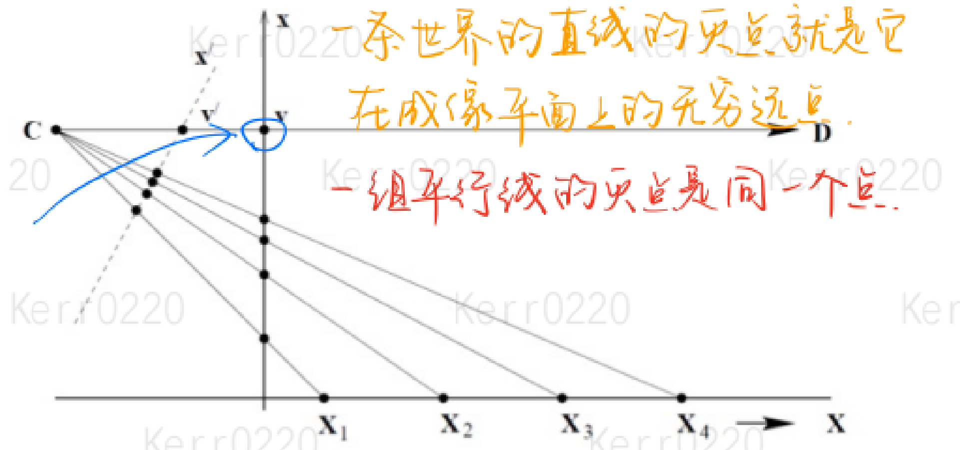
  k1、k2、ρ1、ρ2、k3 称为相机的畸变参数，它们当然也是相机的内参数。

  - 注意一点：这部分的畸变是在归一化平面下变换的，以及畸变参数也是属于内参数
  - 因此我们在算出归一化平面坐标后需要乘一个D进行镜头畸变

$$P_{4\times1}：\text{世界坐标系（三维）}$$
$$[R\ t]_{3\times4}P_{4\times1}：\text{相机坐标系（三维）}$$
$$\frac{1}{Z_c}[R\ t]_{3\times4}P_{4\times1}：\text{归一化平面坐标（二维）}$$
$$D\frac{1}{Z_c}[R\ t]_{3\times4}P_{4\times1}：\text{归一化平面坐标加畸变（二维）}$$
$$K_{3\times3}D\frac{1}{Z_c}[R\ t]_{3\times4}P_{4\times1}：\text{成像平面坐标加畸变（二维）}$$

- 接下来我们要做的就是标定参数。内参有9个要标定（2平移2缩放5畸变），而外参是好解决的：标定板（zhengyou zhang张正友提出）记录了世界坐标，同时我们拍照就有成像坐标。
- 首先解决外参R和t的问题。t好说，只是相机坐标位置。R旋转矩阵要用罗德里德斯公式来算。对于旋转矩阵，其本身只有3个自由度，只需要考虑旋转轴方向与旋转角度（$0-\pi$）
  - 对于R来说，特征值为1的特征向量就是n，$\theta=\arccos\left(\frac{\operatorname{tr}(R)-1}{2}\right)$，我们考虑几个问题
    - 一定有一个特征值为1吗？一定，不然没法正定（数学我也不会证，好像是R一定是正定还是什么的，不记得了）
    - 多个特征值为1怎么办？可以，但一定是3个1.这样arccos=0说明相机没转
    - 角度在0-180，180-360怎么办？将旋转向量反过来就行。用罗德里德斯公式来判断哪边方向才是旋转方向。
- 对于一张照片来说可以有n个点，因此有如下公式：
  $$\Theta^* = \arg\min_{\Theta} \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{1}{2}\left\|\mathbf{K}\cdot\mathcal{D}\left\{\frac{1}{Z_{Cij}}\left[\mathcal{R}\left(\mathbf{d}_i\right)\mathbf{t}_i\right]\mathbf{P}_j\right\}-\mathbf{u}_{ij}\right\|_2^2$$
  ，其中i代表第i张照片，j代表这张照片中的第j个点。每张照片要优化的参数有如下：
  $\Theta=\left\{f_x,f_y,c_x,c_y,k_1,k_2,\rho_1,\rho_2,k_3,\left\{\mathbf{d}_i\right\}_{i=1}^{M},\left\{\mathbf{t}_i\right\}_{i=1}^{M}\right\}$，其中t有3个自由度，d有三个自由度。因此对于N张照片我们可以得到2×M×N个参数，而要优化的参数个数为9+6M。
- 参数量很多，因此我们需要找到一个好的优化起点确保快速迭代到最低点。

- 畸变参数没有办法估计（这个是厂商做工问题），所以我们先假装相机没有畸变。
- cx和cy差不多处于图像的中心，直接初始化为长和宽的一半
- 对于两个和焦距、像素有关的参数，首先，先根据相机上的像素坐标求出归一化平面下的坐标：xn=K^{-1}u
- 接下来我们要找到归一化平面坐标和相机系下的坐标之间的关系。
    - 对于图像中的一组平行线，他们最终会在无穷远点处相交。他们的交点是灭点。灭点的几何意义如下：

      计算很简单，作过相机原点的一条平行于成像平面的平行线，灭点就在这条线与归一化成像平面的交点。
    - 由于相机原点与灭点相交的这条线代表了一组平行线，那么我们可以在标定板上找到一组相互垂直的线。对于每个方格，我们取（0，1，0），（1，0，0）垂直，（1，1，0），（1，-1，0）垂直（齐次向量来代表无穷远点）。这样一个方格我们能取出两组垂线。
    - 由于世界系上，$cos\theta = \frac{l_1 l_2}{|l_1||l_2|} = \frac{OV_1 OV_2}{|OV_1||OV_2|}$,而OV向量其实就是我们上面提到的$K^{-1}$u,我们就可以得到：对于一组垂线：$u^T K^{-T} K^{-1} v=0$.
    - 问题是u，v怎么找？很简单。像素平面和世界系系下的平面之间可以通过射影变换转换过来。因此我们首先用线性最小二乘法来估计这个射影变换矩阵，然后我们就可以使用这个射影变换矩阵将世界系下的齐次坐标（0,1,0)转换成像素平面坐标系下的像素（图中的v'点）这样u和v有了，我们就可以估计K中的$f_x, f_y(\frac{f}{dx}, \frac{f}{dy})$ 了。
        - （其实呢，K应该拆成两部分，一部分是平移，另一部分是缩放投影。opencv直接首先将K拆成了PQ两部分平移和缩放，先把确定的平移和已知点乘了，再去用方程算fxfy）
    - 接下来是外参初始化（这时候我们已经算出来内参初始值了）。首先是世界系坐标。我们直接设置标定板所在平面z为0（这样对于旋转矩阵R来说第三列就不用算了）.在刚刚的运算过程中，我们算了从世界系坐标到成像坐标的单映矩阵H。我们刚刚算出了图中的u，v点，同时我们知道

      $$K_{3\times3}\frac{1}{Z_c}[R\ t]_{3\times4}P_{4\times1}$$

      且知道

      $$c[x_n, y_n, 1]^T = H[X_j,\ \ Y_j, 1]^T$$
      $$Z_{cj}[x_n, y_n, 1]^T = [R, t][X_j, Y_j, Z_j, 1]^T = [r1, r2, t][X_j, Y_j, 1]^T$$

      这样我们就知道了：

      $$H = \lambda[r1, r2, t]$$

我们就直接去找差的这个λ就行。

旋转矩阵是标准正交阵，r1和r2的长度应该为1.因此差的lambda就是将H中的h1h2归一的那个值。（opencv中lambda是h1h2长度求平均）。理想状态下 $\lambda = |h_1| = |h_2|$.矩阵t就直接除lambda就行

对于r3，它与r1r2垂直且长度唯一，r1r2叉乘然后归一化就行。

- 我们完成了标定。注意：生成鸟瞰图的时候不要先用逆过程将图片转到世界坐标系下，再根据世界坐标系做相似变换转到鸟瞰图下。这样需要算前面算的所有矩阵的逆。正确做法是：对于鸟瞰图上的每一个坐标，首先根据相似变换算出在物理平面下的坐标，然后用射影变换投影到理想无畸变的图像系下，最后再投影到归一化平面-算畸变-投影回成像平面找到映射的坐标像素。最后将插值获得的像素值画在鸟瞰图下的坐标上。整个投影规则如下：