# 引言

相机的视野 (FOV)

- 一般相机的 FOV = 50 x 35°
- 人眼的 FOV = 200 x 135°
- 全景 Mosaic = 360 x 180°

广角镜头、鱼眼镜头：产生（接近）半球的 FOV

获得全景图的方法：使用广角镜头

- 优点：全部基于光学，只需拍照即可
- 缺点：非常昂贵、体积大、失真大（可以后期处理）

# 图像拼接

图像拼接是将从不同视角拍摄的多幅有重叠的图像合并成一个视野更大的图像。这一过程包括多个关键步骤和算法，每个步骤都有特定的目标和实现思路。以下是详细的介绍：

## 1. 拍摄图像

**在同一位置拍摄一系列图像**

- **目标**：获取不同视角的图像，以覆盖更广的视野。
- **实现思路**：让相机围绕其光心旋转，确保每幅图像都有足够的重叠区域（一般重叠区域应占图像的30%至50%）。

## 2. 计算图像之间的运动 (Transformation)

**特征点提取**

- **目标**：识别每幅图像中的独特点或区域，以便后续匹配。
- **实现思路**：使用特征点检测算法如 Harris 角点检测、SIFT、SURF、ORB 等。

  **Harris 角点检测**：

  - **目标**：检测图像中的角点。
  - **实现思路**：计算每个像素点的自相关矩阵，根据响应值判断角点。

  **SIFT（Scale-Invariant Feature Transform）**：

  - **目标**：提取对尺度和旋转不变的特征点。
  - **实现思路**：通过高斯差分（DoG）检测特征点，计算特征点的梯度方向直方图描述子。

  **SURF（Speeded-Up Robust Features）**：

  - **目标**：提取快速且鲁棒的特征点。
  - **实现思路**：利用 Hessian 矩阵的行列式检测特征点，使用 Haar 小波响应计算描述子。

  **ORB（Oriented FAST and Rotated BRIEF）**：

  - **目标**：提取快速且对旋转不变的特征点。
  - **实现思路**：结合 FAST 角点检测和 BRIEF 描述子，进行方向校正。

**特征点描述**

- **目标**：为提取的特征点生成唯一描述，以便进行匹配

- **目标**：为提取的特征点生成唯一描述，以便进行匹配。
- **实现思路**：计算特征点描述子（例如 SIFT 描述子、SURF 描述子等）。

**特征点匹配**

- **目标**：找到两幅图像之间的对应特征点对。
- **实现思路**：使用最近邻搜索或其他匹配算法，比较特征点描述子并找到最佳匹配。

  **最近邻搜索（Nearest Neighbor Search）**：

  - **目标**：找到特征点的最近邻匹配。
  - **实现思路**：计算特征点描述子之间的距离，选择距离最近的匹配点。

  **K-D 树（k-d Tree）**：

  - **目标**：高效地进行多维空间中的最近邻搜索。
  - **实现思路**：构建 k-d 树结构，通过树的遍历查找最近邻。

**计算变换矩阵**

- **目标**：根据匹配的特征点对，计算从一幅图像到另一幅图像的变换关系。
- **实现思路**：
  - 使用 RANSAC（随机抽样一致性）算法来剔除错误匹配，保证变换矩阵的鲁棒性。
  - 根据特征点对，使用最小二乘法（DLT）求解变换矩阵。

  **RANSAC（随机抽样一致性）**：

  - **目标**：从含有异常值的匹配点中鲁棒地估计模型参数。
  - **实现思路**：随机选择一个子集计算模型参数，评估内点数目，迭代直到找到最佳模型。

  **DLT（直接线性变换）**：

  - **目标**：线性地求解变换矩阵。
  - **实现思路**：将变换问题表示为线性系统，通过最小二乘法求解。

## 3. 图像移动 (Warping)

**图像变形**

- **目标**：将第二幅图像变换到第一幅图像的视角下，使得两幅图像在重叠区域对齐。
- **实现思路**：
  - **坐标变换**：根据计算的变换矩阵，计算目标图像中每个像素的位置。
  - **像素值插值**：使用双线性插值、最近邻插值等方法对目标图像的像素进行插值计算。

  **双线性插值**：

  - **目标**：在目标位置的像素值通过邻近四个像素的加权平均获得。
  - **实现思路**：计算目标位置的上下左右四个像素值的加权平均。

  **最近邻插值**：

  - **目标**：在目标位置的像素值由最近的原始像素值决定。
  - **实现思路**：选择距离目标位置最近的原始像素值赋给目标位置。

# 4. 图像融合 (Blending)

图像融合是图像拼接中的一个重要步骤，目的是使拼接后的图像在重叠区域无缝衔接，避免出现明显的拼接痕迹或鬼影。常用的图像融合技术包括处理鬼影和羽化问题、金字塔融合以及梯度域融合。

## 1. 处理鬼影和羽化问题

**鬼影（Ghosting）**

- **问题**：由于不同图像对齐不准确或存在移动物体，重叠区域可能出现重影。
- **解决方案**：通过调整融合策略，如加权平均、选择性融合或使用高级融合技术来减少鬼影。

**羽化（Feathering）**

- **问题**：在图像边缘进行简单的加权平均可能导致明显的过渡区域。
- **解决方案**：使用线性渐变权重，使得图像在重叠区域平滑过渡。

## 2. 金字塔融合（Pyramid Blending）

**Laplacian 金字塔融合**

- **目标**：在不同分辨率下融合图像，减少拼接痕迹并处理不同尺度的图像细节。
- 实现思路：
    1. **构建高斯金字塔**：对图像进行多层次的高斯模糊，得到多层次的图像金字塔。
    2. **构建Laplacian金字塔**：通过将每一层高斯金字塔与下一层的扩展版本相减，得到Laplacian金字塔。
    3. **图像融合**：在每一层的Laplacian金字塔中进行加权融合。
    4. **图像重构**：从最底层开始逐层向上重构图像，最终得到融合后的图像。

**多分辨率样条（Multiresolution Spline）**

- **目标**：处理不同尺度下的图像细节，使得拼接结果更自然。
- **实现思路**：类似Laplacian金字塔，但更侧重于处理图像的边缘和细节部分。

## 3. 梯度域融合（Gradient Domain Blending）

**Poisson 图像编辑**

- **目标**：保持图像的梯度连续性，实现无缝融合。
- 实现思路：
    1. **计算梯度**：计算待融合图像的梯度场。
    2. **设定边界条件**：使用目标图像的边界条件。
    3. **求解Poisson方程**：通过求解带Dirichlet边界条件的Poisson方程，得到融合后的图像。

**去除鬼影**

- **目标**：消除由于图像对齐不精确或光照变化导致的重影现象。
- **实现思路**：使用金字塔融合、梯度域融合等高级融合技术。

  **梯度域融合（Poisson Image Editing）**：

    - **目标**：实现无缝融合，保持图像的梯度连续性。
    - **实现思路**：在梯度域进行融合，通过求解 Poisson 方程，使得融合后的图像在梯度上连续。

## 5. 重复以上步骤

**拼接多幅图像**

- **目标**：处理多幅图像的拼接问题，获得全景图。
- **实现思路**：将每一幅图像与前一幅图像进行拼接，直到所有图像拼接完成。

# 2D 射影几何

## 图像之间变换的分类

图像之间的变换可以根据其复杂程度和几何性质分为以下几类：

### 1. 平移变换（Translation）

- **描述**：仅包括水平和垂直方向的平移。
- **应用**：简单的图像移动，不涉及旋转或缩放。

### 2. 旋转变换（Rotation）

- **描述**：图像绕某一点旋转一个角度。
- **应用**：相机固定但旋转拍摄的情况。

### 3. 缩放变换（Scaling）

- **描述**：图像按比例缩放。
- **应用**：不同分辨率图像的对齐。

### 4. 仿射变换（Affine Transformation）

- **描述**：包括平移、旋转、缩放和剪切，保持平行线的平行性。
- 矩阵形式：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & tx \\ c & d & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

1. **平移（Translation）**： 平移变换使得点 $(x, y)$ 移动 $tx$ 和 $ty$，其矩阵表示为：

$$\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix}$$

2. **旋转（Rotation）**： 旋转变换使得点绕原点旋转一个角度 $\theta$，其矩阵表示为：

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3. **缩放（Scaling）**： 缩放变换改变点的大小，其矩阵表示为：

$$\begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4. **剪切（Shearing）**： 剪切变换使得图形在一个方向上拉伸，其矩阵表示为：

$$\begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- **应用**：多用于建筑物、平面图像的变换。

### 5. 投影变换（Projective Transformation/Homography）

- **描述**：包括所有仿射变换，并能处理透视投影，保持直线的直线性，但不保持平行性。
- 矩阵形式：

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- **应用**：广泛应用于图像拼接、全景图制作、立体视觉等。

射影几何（Projective Geometry）是研究几何图形之间关系的数学学科，它强调图形在投影变换下的性质。射影几何广泛应用于计算机视觉、图像处理和摄影测量中。在图像拼接中，射影几何用于处理从不同视角拍摄的图像之间的关系，将它们变换并对齐到同一个平面上。

## 齐次坐标与射影几何

## 齐次坐标、点和线段的关系

### 点和齐次坐标

在二维空间中，一个点 $(x, y)$ 可以通过其齐次坐标 $(x, y, 1)$ 表示。齐次坐标引入了一个额外的维度，使得可以用矩阵运算统一处理各种几何变换。

### 仿射变换中的平移

在普通的笛卡尔坐标系中，平移变换不能通过简单的矩阵乘法表示，但在齐次坐标中，平移变换可以通过矩阵乘法表示：

1. **点的表示**：在齐次坐标中，点 (x,y)(x, y)(x,y) 表示为 $(x, y, 1)$。
2. 平移变换矩阵：平移变换矩阵为：

$$T = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix}$$

3. 平移操作：将点 $(x, y, 1)$

   通过平移变换矩阵 $T$ 进行变换：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = T \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+tx \\ y+ty \\ 1 \end{bmatrix}$$

在齐次坐标中，平移变换可以通过矩阵乘法来表示和计算，这就是为什么引入第三个维度为1的原因。

### 线段的齐次表示

#### 线段的表示

- 在二维平面上，一条线段可以由两个端点 $(x_1, y_1)$ 和 (x_2, y_2) 定义。
- 在齐次坐标中，这两个端点表示为 $(x_1, y_1, 1)$ 和 $(x_2, y_2, 1)$。

#### 线的表示

- 在齐次坐标中，一条直线可以表示为 $ax + by + c = 0$。
- 其对应的齐次坐标表示为向量 $(a, b, c)$。

## 点和线的关系

在射影几何中，点和线有一种对偶关系：

- 一个点 $(x, y, 1)$ 在直线 $(a, b, c)$ 上，当且仅当 $ax + by + c = 0$。

## 齐次坐标系的引入

引入齐次坐标系的目的是为了更方便地进行几何变换：

1. **平移变换**：在普通笛卡尔坐标系中，平移变换无法通过简单的矩阵乘法表示；但在齐次坐标系中，可以通过矩阵乘法表示平移变换。
2. **统一处理**：通过引入第三个维度，所有的二维几何变换（平移、旋转、缩放、剪切和投影）都可以通过3×3矩阵乘法进行统一处理。
3. **变换组合**：多个变换可以通过矩阵乘法组合在一起，从而实现复杂的变换。

### 齐次坐标的射影变换

射影变换是一种从二维射影空间到自身的变换，它可以表示为 3×3 的非奇异矩阵 H：

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

对于齐次坐标 (x, y, 1)，射影变换可以表示为： $\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ 其中 $(x', y', w')$ 是变换后的齐次坐标。

# 2D 运动的射影几何表示

## 运动模型

在图像处理和计算机视觉中，常用的二维运动模型包括：

- **平移**：仅有平移参数。
- **相似变换**：包括平移、旋转和等比例缩放。
- **仿射变换**：包括平移、旋转、缩放和剪切。
- **射影变换**：包括所有仿射变换以及透视投影。

## 运动模型的参数求解

为了对齐两幅图像，需要计算从一幅图像到另一幅图像的变换矩阵 H。可以通过以下步骤实现：

1. **特征点检测和匹配**：使用 SIFT、SURF、ORB 等算法检测特征点，并匹配特征点对。
2. **计算变换矩阵**：使用 RANSAC 算法剔除错误匹配点，并用 DLT 算法求解变换矩阵 H。
3. **图像变形**：应用变换矩阵 H 将图像变形到同一平面。

# 射影几何在 3D 世界到 2D 图像之间的映射中的应用

## 射影几何在相机成像中的作用

射影几何在从3D世界到2D图像的映射中起着重要作用。相机成像可以看作是将3D场景通过投影变换映射到2D图像平面。下面详细介绍相机模型的概念，并通过具体例子阐释其原理。

# 针孔相机模型

**针孔相机模型**是一种简单而有效的相机模型，用于描述从3D点到2D点的映射过程。这个模型假设光线通过一个小孔投射到成像平面上，形成图像。

## 1. 3D点到2D点的映射

假设一个3D点 $(X, Y, Z)$ 通过针孔相机投影到2D图像平面上的点 $(x, y)$。

- **3D点**：$(X, Y, Z)$
- **投影点**：$(x, y)$

在针孔相机模型中，投影关系可以表示为：$x = \frac{fX}{Z}, \quad y = \frac{fY}{Z}$

其中 $f$ 是相机的焦距。

## 2. 齐次坐标表示

为了方便矩阵运算，我们使用齐次坐标表示：

- 3D点 $(X, Y, Z)$ 用齐次坐标表示为 $(X, Y, Z, 1)$
- 2D点 $(x, y)$ 用齐次坐标表示为 $(x, y, 1)$

# 相机投影矩阵

相机投影矩阵是一个3×4的矩阵，结合了相机的内参（Intrinsic Parameters）和外参（Extrinsic Parameters），用于将3D齐次坐标点 $(X, Y, Z, 1)$ 投影到2D齐次坐标点 $(x, y, w)$。

## 内参矩阵（Intrinsic Matrix）

内参矩阵描述了相机的内在特性，如焦距和主点位置。假设内参矩阵为 $K$：

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

其中：

- $f_x$和 $f_y$ 是图像平面上的焦距，通常$f_x = f_y = f$。
- $c_x$ 和 $c_y$ 是主点（光轴与图像平面的交点）在图像上的坐标。

## 外参矩阵（Extrinsic Matrix）

外参矩阵描述了相机在世界坐标系中的位置和方向。假设外参矩阵为 $[R|t]$，其中 $R$ 是旋转矩阵， $t$ 是平移向量。

## 3. 投影矩阵的构成

投影矩阵 ( P ) 是内参矩阵和外参矩阵的组合： $P = K[R|t]$

# 具体例子

假设一个3D点 ( (X, Y, Z) ) 通过相机投影到2D图像平面上。

1. **内参矩阵 ( K )**:
2. **外参矩阵 [R|t][R|t][R|t]**（假设没有旋转和平移，即 $R$ 为单位矩阵，$t$ 为零向量）：

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad t = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$[R|t] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

3. **投影矩阵 PPP**:

$$P = K[R|t] = \begin{bmatrix} 800 & 0 & 320 & 0 \\ 0 & 800 & 240 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

4. **投影关系**： 对于3D点 $(X, Y, Z, 1)$，投影到2D图像点 $(x, y, w)$：

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} 800 & 0 & 320 & 0 \\ 0 & 800 & 240 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} 800X + 320Z \\ 800Y + 240Z \\ Z \end{bmatrix}$$

5. **得到2D图像坐标**：

$$x = \frac{800X + 320Z}{Z}, \quad y = \frac{800Y + 240Z}{Z}$$

因此，3D点 $(X, Y, Z)$ 投影到2D图像上的坐标为 $\left(\frac{800X+320Z}{Z}, \frac{800Y+240Z}{Z}\right)$。

# 点、线、二次曲线的表示及其变换

## 点和线的表示及变换

**点的变换**：点 P 在射影变换 H 下的变换为 $P' = HP$。

**线的变换**：直线 L 在射影变换 H 下的变换为 $L' = (H^T)^{-1}L$。

## 二次曲线的表示及其对偶

### 二次曲线

- 二次曲线可以用二阶方程 $ax^2 + bxy + cy^2 + dx + ey + f = 0$ 表示。
- 在齐次坐标下，二次曲线可以用对称矩阵 Q 表示，方程为 $x^TQx = 0$。

### 对偶二次曲线

- 点和线在射影空间是对偶的，点对偶的二次曲线满足 $x^TQx = 0$。

# 图像的重投影

## 重投影概述

为了将不同视角的图像投影到同一平面进行拼接，需要对图像进行重投影。常见的投影表面有平面、圆柱和球面。

## 投影表面的选择

### 平面投影

- 适用于视角差异较小的图像拼接。
- 变换简单，但视角差异较大时会产生畸变。

### 圆柱面投影

- 适用于相机水平旋转拍摄的全景图像。
- 将图像投影到圆柱面，然后展开圆柱面，减少畸变。

### 球面投影

- 适用于视角差异较大的全景图像。
- 将图像投影到球面，然后展开球面，能更好地处理大视角差异。

### 圆柱面投影、球面投影及其反投影

#### 圆柱面投影

- **转换成圆柱坐标**：将 3D 点 (X, Y, Z) 转换为圆柱坐标 (θ, h)。
- **展开圆柱**：将圆柱坐标 (θ, h) 转换为 2D 图像坐标 (u, v)。

#### 球面投影

- **转换成球面坐标**：将 3D 点 (X, Y, Z) 转换为球面坐标 (θ, φ)。
- **展开球面**：将球面坐标 (θ, φ) 转换为 2D 图像坐标 (u, v)。

#### 反投影

- **圆柱面反投影**：将 2D 图像坐标 (u, v) 转换回圆柱坐标 (θ, h)，然后转换回 3D 坐标 (X, Y, Z)。
- **球面反投影**：将 2D 图像坐标 (u, v) 转换回球面坐标 (θ, φ)，然后转换回 3D 坐标 (X, Y, Z)。

# 图像拼接的实际应用：虚拟广告的实现

#### 概述

- **目标**：通过图像拼接和单应变换技术，将虚拟广告无缝地嵌入到真实场景中。
- **实现思路**：使用单应变换将虚拟广告图像投射到目标平面，实现与真实场景的无缝融合。

#### 步骤

1. **图像拍摄**：获取包含目标平面的图像。
2. **特征点匹配**：在目标平面图像和虚拟广告图像中提取和匹配特征点。
3. **计算单应矩阵**：使用RANSAC算法计算单应矩阵。
4. **图像变换**：应用单应矩阵将虚拟广告图像变换到目标平面。
5. **图像融合**：使用金字塔融合或梯度域融合技术，处理拼接缝隙和鬼影问题。