

Main Takeaway

计算机视觉CV——computer vision

50% hw + 50% 期末

作业提交

■ 提交内容 (以下三样, 缺一不可)

1. 源程序。
2. 对应的可执行二进制文件 (Win、MacOS、Linux都可以)
3. **实验报告** (包括开发软件说明、算法具体步骤、算法实现要点、实验结果展示及分析、编程体会等) , 提供模板。

references

- [7.1 相机校准 - OpenCV中文官方文档 \(woshicver.com\)](#)
- [jash-git/Learning-OpenCV-3: 《Learning OpenCV 3》學習OpenCV 3 範例/電子檔備份 \(github.com\)](#)

Contents

[Main Takeaway](#)

[Contents](#)

[Intro](#)

[Edge Detection](#)

概念

用模板实现卷积

基于一阶导数的边缘检测

基于二阶导数的边缘检测

Canny边缘检测

[Curves](#)

曲线表示

曲线拟合

Hough变换

直线检测

圆弧检测

评价

[Image Local Feature](#)

Feature detection

Feature descriptors

Image stitching

RANSAC

金字塔

[Eigenface](#)

PCA

Eigenface

Performance Evaluation

Motion Estimation

Visual Recognition

KNN

BoW

基于卷积的物体识别

Deep Learning

BP

CNN

Tips for training

Batch

Batch Normalization

Momentum

Self-attention Block

Camera Calibration

Camera model

Camera Calibration

Stereo Vision

Epipolar Geometry

Structured-lighting 3D Scan

ICP

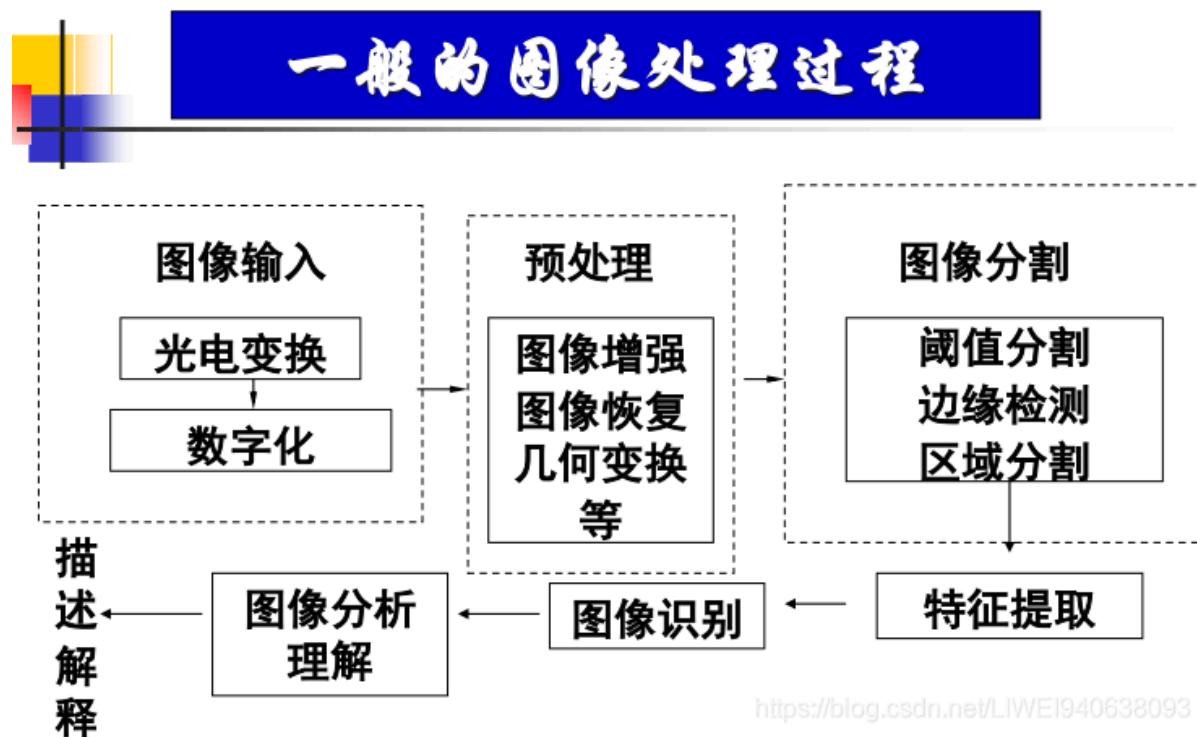
Object Categorization

期末

Intro

opencv 4.x 全面采用C++

[数字图像处理-Digital Image Processing\(DIP\)](#)



<https://blog.csdn.net/LIWEI940638093>

- <https://www.cc98.org/topic/5231792>)
- opencv tutorial
 - 基本的图像/视频操作

- 结构分析
- 摄像头定标
- 运动分析
- 目标识别
- 基本的GUI

what is an image

- binary: 0-black,1:white
- gray scale
- color: rgv,hsv——“vector-valued” function:

Gestalt laws



格式塔法则

1. 图像与背景关系原则 (figure-ground) : 物体/图形比背景更突出
2. 接近原则 (proximity) : 接近/邻近的物体会被认为是一个整体
3. 相似原则 (similarity) : 刺激物的形状/大小/颜色/强度等物理属性方面相似时, 刺激物被认为是一个整体
4. 连续性原则 (continuity) : 若图形的某些部分可以被看作连接在一起的, 则这些部分很可能被认为是一个整体
5. 封闭/闭合原则 (closure) : 对于有些没有闭合的图形, 主体能自行填补缺口使之被认为是一个整体
6. 蕴含律: 对复杂对象进行感知时, 人们倾向于把对象看作对称的, 简单的, 规则的图形

1. **Law of Proximity:** Elements that are close to each other are perceived as a group. For example, a cluster of dots is seen as a single group rather than individual dots.
2. **Law of Similarity:** Similar elements (in shape, size, color, etc.) are perceived as belonging together. For instance, a grid of squares and circles will be seen as rows or columns of similar shapes.
3. **Law of Closure:** People tend to perceive incomplete shapes as complete. Even if parts of a shape are missing, the mind fills in the gaps to perceive a whole object.
4. **Law of Continuity:** Elements that are arranged on a line or curve are perceived as more related than elements not on the line or curve. This principle helps in perceiving smooth, continuous lines rather than abrupt changes.
5. **Law of Common Fate:** Elements that move in the same direction are perceived as a group. This is particularly relevant in dynamic visual displays, such as animations.
6. **Law of Prägnanz (Good Figure):** People perceive objects in the simplest form possible. This principle suggests that we tend to see the most straightforward and stable arrangement of elements.
7. **Law of Figure-Ground:** People instinctively perceive objects as either being in the foreground or the background. The figure is the main object of focus, while the ground is the background.

Edge Detection

概念

Origin of Edges

- surface normal discontinuity 表面法线
- depth discontinuity
- surface color discontinuity
- illumination discontinuity

Edge detection : Identify sudden changes (discontinuities) in an image

求导：一阶局部极值/二阶过零点——但是有噪声的影响

用模板实现卷积

用模板(Template/Kernel，实质为系数矩阵)来对图像做卷积 (convolution)

$T(x,y)$ is a template ($n \times m$), $I(x,y)$ is an image ($M \times N$), then the convoluting of T with I is

$$T * I(X, Y) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} T(i, j) I(X + i, Y + j)$$

因为超出图像边界，所以resulting image一般会小一点

Sobel 算子 (水平方向) :

text

```
-1  0  1
-2  0  2
-1  0  1
```

Sobel 算子 (垂直方向) :

text

```
-1 -2 -1
 0  0  0
 1  2  1
```

通过将图像分别与水平和垂直方向的Sobel算子进行卷积，然后取两个结果的平方和的平方根，可以得到图像的边缘信息。

2. 模糊

模糊操作用于减少图像中的细节，使图像看起来更柔和。以下是使用均值滤波器进行模糊的示例：

均值滤波器 (3x3) :

text

```
1/9  1/9  1/9  
1/9  1/9  1/9  
1/9  1/9  1/9
```

通过将图像与均值滤波器进行卷积，可以实现简单的模糊效果。

3. 锐化

锐化操作用于增强图像中的细节，使图像看起来更清晰。以下是使用拉普拉斯算子进行锐化的示例：

拉普拉斯算子 (3x3) :

text

```
0  -1  0  
-1  4  -1  
0  -1  0
```

通过将图像与拉普拉斯算子进行卷积，然后将结果与原图像相加，可以实现锐化效果。

4. 浮雕效果

浮雕效果可以使图像看起来具有立体感，以下是使用浮雕滤波器实现浮雕效果的示例：

浮雕滤波器 (3x3) :

text

```
-1  -1  0  
-1  0  1  
0  1  1
```

通过将图像与浮雕滤波器进行卷积，然后将结果加上一个偏移量（如128），可以实现浮雕效果。

基于一阶导数的边缘检测

[图像处理——4个坐标系及相关转换](#) [图像像素坐标系](#) [图像物理坐标系](#) [相机坐标系](#) [世界坐标系](#) [图像坐标系](#)

梯度，连续的情况如下

- **梯度(Gradient):** 是图像对应二维函数的一阶导数

$$G(x, y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- **梯度的幅值:** isotropic operator anisotropic

$$|G(x, y)| = \sqrt{G_x^2 + G_y^2}$$



$$|G(x, y)| = |G_x| + |G_y|$$

$$|G(x, y)| \approx \max(|G_x|, |G_y|)$$

- **梯度方向:**

$$\alpha(x, y) = \arctan(G_y / G_x)$$

梯度方向为函数最大变化率方向

离散化: 使用差分近似偏导数

$$G_x = f[x+1, y] - f[x, y]$$

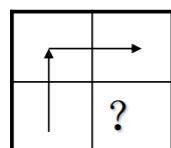
$$G_y = f[x, y] - f[x, y+1]$$

一般使用卷积模板进行计算

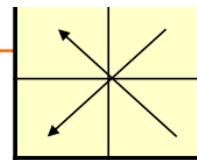
一般用卷积模板进行计算:

$$G_x = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

上述表示?



- Roberts交叉算子—— 2×2 梯度算子



$$G[i, j] = |G_x| + |G_y|$$



$$G[i, j] = |f[i, j] - f[i+1, j+1]| + |f[i+1, j] - f[i, j+1]|$$



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

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

- Sobel算子—— 3×3 梯度算子

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- Prewitt算子—— 3×3 梯度算子，运算较快

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

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

- 均值差分——一定邻域内灰度平均值之差

↓ **3×3 邻域加权**

$$G_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$G_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$

a_0	a_1	a_2
a_7	[i, j]	a_3
a_6	a_5	a_4

C=1: Prewitt算子

C=2: Sobel算子

C=3: Sethi算子

基于二阶导数的边缘检测

图像灰度二阶导数的过零点对应边缘点

- 拉普拉斯 (Laplacian) 算子

拉普拉斯算子是二阶导数的二维等效式：

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

用差分近似微分:

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \frac{\partial G_x}{\partial x} \\&= \frac{\partial(f[i, j+1] - f[i, j])}{\partial x} \\&= \frac{\partial f[i, j+1]}{\partial x} - \frac{\partial f[i, j]}{\partial x} \\&= (f[i, j+1] - f[i, j]) - (f[i, j] - f[i, j-1])\end{aligned}$$
$$\frac{\partial^2 f}{\partial x^2} = (f[i, j+1] - 2f[i, j] + f[i, j-1])$$
$$\frac{\partial^2 f}{\partial y^2} = (f[i+1, j] - 2f[i, j] + f[i-1, j])$$

表示为卷积模板

$$\nabla^2 \approx \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

邻域中心点具有更大权值的近似算子

$$\nabla^2 \approx \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

- LoG边缘检测算法 (LoG = Laplacian of Gaussian)

高斯滤波+拉普拉斯边缘检测

基本特征:

- 平滑滤波器是高斯滤波器.
- 采用拉普拉斯算子计算二阶导数.
- 边缘检测判据是二阶导数零交叉点并对应一阶导数的较大峰值.
- 使用线性内插方法在子像素分辨率水平上估计边缘的位置.

经过推导得到LoG算子

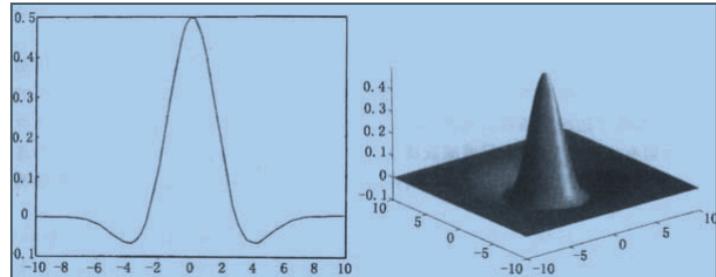
$$h(x, y) = \nabla^2 [g(x, y) * f(x, y)]$$

根据卷积求导法 ↓

$$h(x, y) = [\nabla^2 g(x, y)] * f(x, y)$$

$$\nabla^2 g(x, y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

墨西哥草帽算子:



两种等效计算方式

- 图像与高斯函数卷积，再求卷积的拉普拉斯微分
- 求高斯函数的拉普拉斯微分，再与图像卷积

Canny边缘检测

canny梯度较大的地方留下来，然后看周围和自己相似的像素也留下来

算法步骤：

1. 用高斯滤波器平滑图像
2. 用一阶偏导有限差分计算梯度幅值和方向
3. 对梯度幅值进行非极大值抑制 (NMS)
4. 用双阈值算法检测和连接边缘

以下对每步进行说明和解释

- 用高斯滤波器平滑图像——why 高斯滤波器

平滑去噪和边缘检测是一对矛盾，应用高斯函数的一阶导数，在二者之间获得最佳的平衡

步1. 图像与高斯平滑滤波器卷积：

$$S[i, j] = G[i, j; \sigma] * I[i, j]$$

- 一阶偏导差分计算梯度幅值和方向

步2a. 使用一阶有限差分计算偏导数阵列P与Q：

$$G_x[i, j] \approx (S[i, j+1] - S[i, j] + S[i+1, j+1] - S[i+1, j]) / 2$$

$$G_y[i, j] \approx (S[i, j] - S[i+1, j] + S[i, j+1] - S[i+1, j+1]) / 2$$

步2b. 计算梯度幅值与方向角：

$$M[i, j] = \sqrt{G_x[i, j]^2 + G_y[i, j]^2}$$

$$\theta[i, j] = \arctan(G_y[i, j] / G_x[i, j])$$

- 非极大值抑制 (NMS, Non-max Suppression)

NMS：找到局部极大值，并筛除（抑制）邻域内其余的值

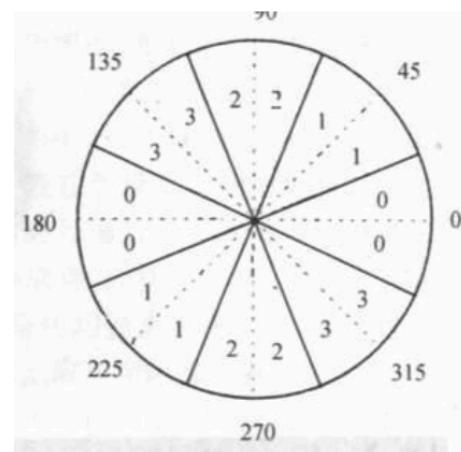
* 将梯度角离散为圆周的四个扇区之一，以便用 3×3 的窗口作抑制运算

* 方向角离散化：

$$\zeta[i, j] = \text{Sector}(\theta[i, j])$$

* 抑制，得到新幅值图：

$$N[i, j] = \text{NMS}(M[i, j], \zeta[i, j])$$



How抑制？ 若 $M[i, j]$ 不比沿梯度线方向上的两个相邻点幅值大，则 $N[i, j]=0$

- 将其梯度方向近似为以下值中的一个，包括0、45、90、135、180、225、270和315，即表示上下左右和45度方向。——简单，但效果不一定最好
 - 比较该像素点和其梯度正负方向的像素点的梯度强度，如果该像素点梯度强度最大则保留，否则抑制（删除，即置为0）
- 双阈值化并边缘链接
 - 取高低两个阈值(T_2, T_1)作用于新幅值图 $N[i, j]$ ，得到两个边缘图：高阈值和低阈值边缘图

高阈值图： $N[i, j] > T_2$

低阈值图： $N[i, j] > T_1$

- 连接高阈值边缘图，出现断点时，在低阈值边缘图中的8邻点域搜寻边缘点

阈值太高会出现部分轮廓丢失，阈值太低可能是假边缘

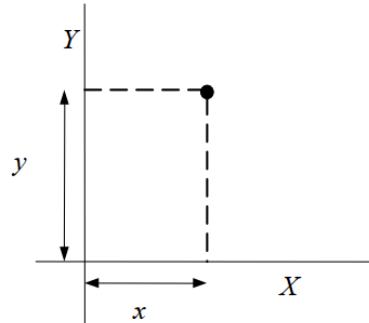
选用两个阈值：更有效的阈值方案

Curves

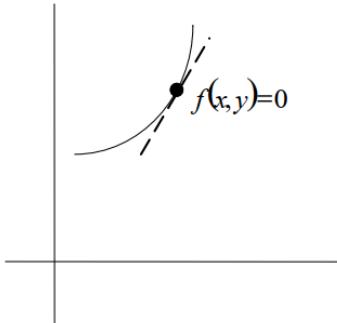
曲线表示

• 曲线表示

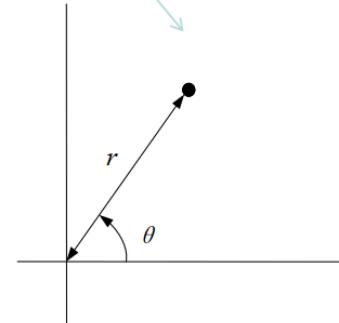
$$\vec{p}(t) = (x, y) = (x(t), y(t)) \quad t_0 \leq t \leq t_1$$



(a) 显式表达 **Explicit**



(b) 隐式表达 **Implicit**

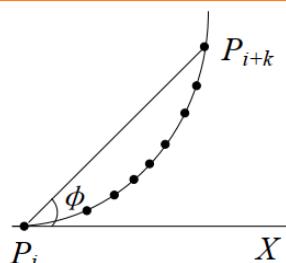


(c) 参数表达

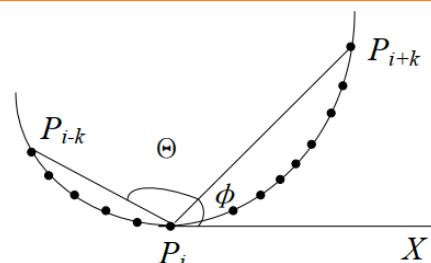
$$\begin{cases} y = f_1(x) \\ y = f_2(x) \end{cases}$$

$$\begin{cases} f_1(x, y) = 0 \\ f_2(x, y) = 0 \end{cases}$$

曲线的离散化



(a) 曲线斜率计算



(b) 曲线曲率计算

• 曲线长度

$$s = \sum_{i=2}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

• 曲线切向量(**k**斜率)

$$\phi = \arctan \frac{y_{i+k} - y_i}{x_{i+k} - x_i}$$

• 曲率(**k**曲率)

$$\Theta = \phi_l - \phi_r$$

曲线拟合

曲线拟合：给定一系列边缘点，设法找到一条曲线的函数表达式，通过调整参数尽量使该曲线接近所有的边缘点以描述对象的轮廓

- 直线段近似
 - 直线
 - 分线段
- 二次曲线
 - 圆弧
 - 圆锥
- 样条曲线（分段多项式）
 - 重要属性：连接处光滑（n阶导数连续）
 - **三次样条**：通过每个点
 - **B样条**：不必通过每个点
- 解析法：用p个边缘点。（缺点：不鲁棒）
- 回归法：用全部观测值来逼近（最小二乘法）

Hough变换

- Hough变换是基于投票(Voting)原理的参数估计方法——形状检测技术
- 解决问题：从图像中识别几何形状（如直线、圆、椭圆等）的图像处理方法。它**特别适用于处理有噪声或部分缺失的图像**，能够有效地检测出图像中的形状，即使这些形状存在断裂或旋转的情况
- 基本思想：图像中每一点对参数组合进行表决，组合为胜者（结果）赢得多数票的参数

直线检测

- 直线检测Hough变换
参数空间——常用极坐标（避免垂直直线带来的问题，范围有限制）
[霍夫变换\(Hough Transform\)详解](#)

Hough变换算法

1. 适当地**量化**参数空间（合适的精度即可）。
2. 假定参数空间的每一个单元都是一个累加器，把累加器**初始化**为零。
3. 对图像空间的每一点，在其所满足的参数方程对应的累加器上**加1**。
4. 累加器阵列的**最大值**对应模型的参数。

圆弧检测

圆弧拟合：参数组合(a, b): $\begin{cases} x = a + r \cos \theta \\ y = b + r \sin \theta \end{cases} \Rightarrow b = a \tan \theta - x \tan \theta + y$, 其中 θ 为边缘点处梯度
(可由a, b, x, y表示)

- (1) 量化关于**a, b**的参数空间到合适精度
- (2) 初始化所有累加器为**0**
- (3) 计算图像空间中边缘点的梯度幅度 $G_{mag}(x, y)$
和角度 $\theta(x, y)$
- (4) 若边缘点参数坐标满足 $b = a \tan \theta - x \tan \theta + y$
则该参数坐标对应的累加器**加1**
- (5) 拥有最大值的累加器所在的坐标即为图像空间中的圆心之所在
- (6) 得到圆心坐标之后，我们可以很容易反求**r**

评价

- 参数空间离散化：精度低/高
 - 低：集合形状不准确；丢失细节；噪声敏感
 - 高：计算复杂度增加；过拟合；难以全局最优解（可能多个局部峰值）
- 与RANSAC算法的比较
 - 相似

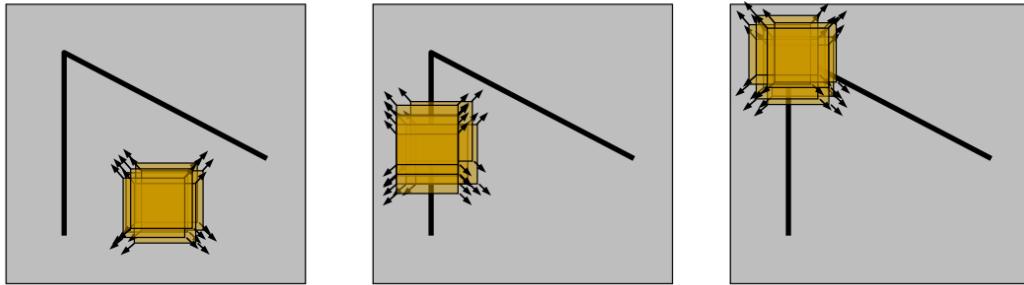
1. **目的**: Hough变换和RANSAC算法都是用于从包含噪声和异常点的数据集中提取几何形状（如直线、圆等）的算法¹¹。
 2. **鲁棒性**: 两者都对噪声和异常点具有一定的鲁棒性，能够从不完美的数据集中提取出有用的几何信息¹¹。
 3. **应用场景**: 它们在计算机视觉领域都有广泛的应用，特别是在图像处理、目标检测和模式识别等领域
 - 差别
1. **基本原理**:
 - **Hough变换**: 是一种数学上的变换，将原空间中的点映射到参数空间中。通过在参数空间中寻找峰值，来检测原空间中的几何形状⁷。
 - **RANSAC算法**: 是一种迭代的模型选择算法，通过随机采样数据点并拟合模型，然后评估模型与剩余数据点的一致性，来找到最佳的模型⁸。
 2. **处理方式**:
 - **Hough变换**: 对原空间中的每个点，在参数空间中对应的线上进行“投票”，通过寻找投票的峰值来检测几何形状⁷。
 - **RANSAC算法**: 通过随机采样数据点并拟合模型，然后评估模型与剩余数据点的一致性，来找到最佳的模型。它会丢弃与模型不一致的数据点（异常点）⁸。
 3. **适用性**:
 - **Hough变换**: 适用于检测特定的几何形状，如直线、圆等。它对形状的参数化有明确的要求⁷。
 - **RANSAC算法**: 适用于各种类型的模型拟合问题，不限于特定的几何形状。它对模型的形式没有特定的要求，只要能够通过数据点拟合出模型即可⁸。
 4. **计算复杂度**:
 - **Hough变换**: 计算复杂度较高，特别是对于高维的参数空间。它需要对原空间中的每个点进行参数空间的映射和投票⁷。
 - **RANSAC算法**: 计算复杂度相对较低，特别是对于简单的模型。它只需要进行有限次的随机采样和模型拟合⁸。
 5. **结果解释**:
 - **Hough变换**: 结果是参数空间中的峰值，需要进一步解释为原空间中的几何形状⁷。
 - **RANSAC算法**: 结果是最佳的模型参数，可以直接用于解释原空间中的数据

Image Local Feature

SIFT 尺寸不变特征变换 (Scale Invariant Feature Transform)

Feature detection

- Harris corner detector
 - 基本思想：用一个小窗来看图像，当移动时小窗内图像改变会非常大



“flat” region:
no change in
all directions

“edge”:
no change along
the edge direction

“corner”:
significant change
in all directions

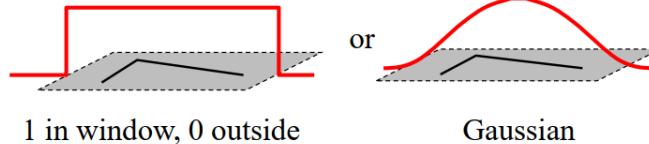
- Mathematics:

Change of intensity for the shift $[u, v]$:

$$E(u, v) = \sum_{x,y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$$

Window function
 Shifted intensity
 Intensity

Window function $w(x, y) =$



近似等价

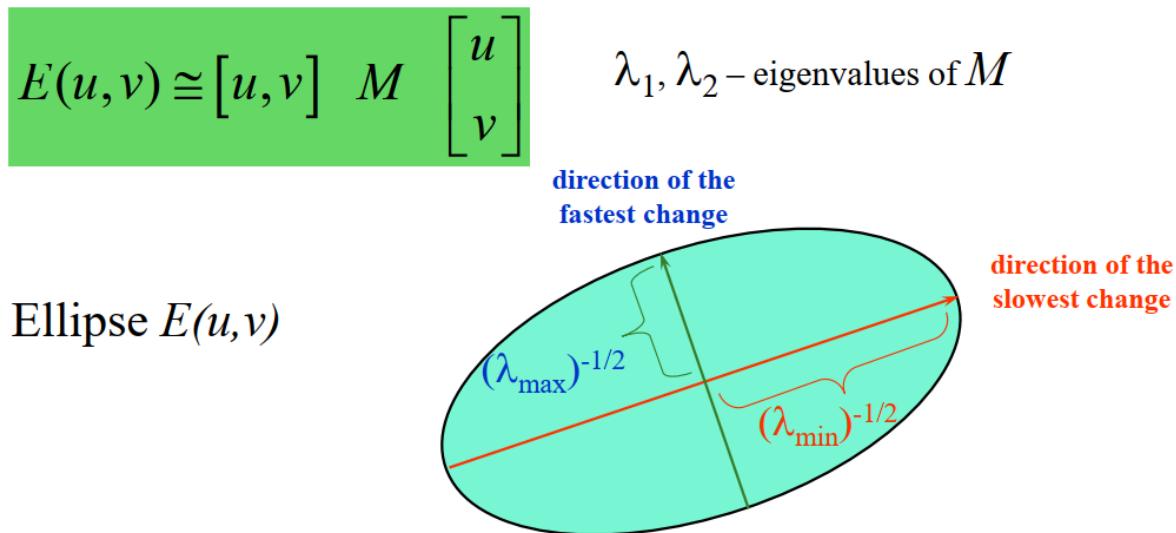
For small shifts $[u, v]$ we have a *bilinear* approximation:

$$E(u, v) \cong [u, v] \ M \ \begin{bmatrix} u \\ v \end{bmatrix}$$

where M is a 2×2 matrix computed from image derivatives:

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Intensity change in shifting window: eigenvalue analysis



二次项函数本质上就是一个椭圆函数

Measure of corner response:

$$R = \det M - k (\operatorname{trace} M)^2$$

$$\det M = \lambda_1 \lambda_2$$

$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

$(k - \text{empirical constant}, k = 0.04-0.06)$

- Corner response R is invariant to image rotation
 - Partial invariance to affine intensity change
 - But: non-invariant to image scale! [特征检测子 -- Harris & LoG & DoG 高斯金字塔 log金字塔](#)
- Scale Invariant Detection: 见下

一幅图像的尺度空间可被定义为原图像与可变尺度的高斯核卷积

那么如何在不同图像上找到的点是匹配的点

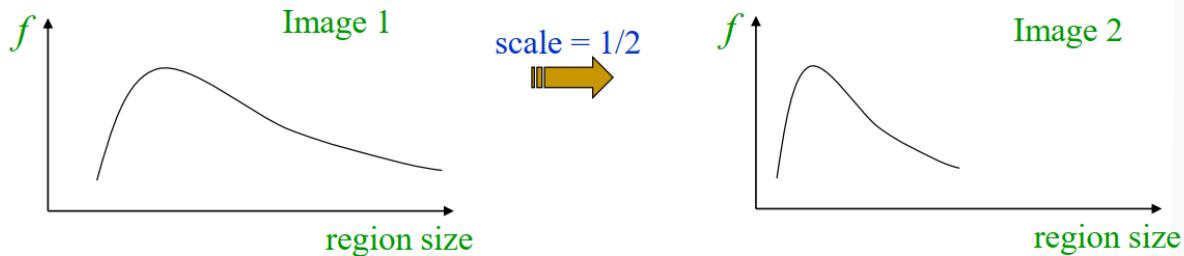
要解决scale的问题

■ Solution:

- Design a function on the region (circle), which is “**scale invariant**” (the same for corresponding regions, even if they are at different scales)

Example: average intensity. For corresponding regions (even of different sizes) it will be the same.

- For a point in one image, we can consider it as a **function of region size** (circle radius)



用一个函数来寻找每张图片合适的window size——LoG

Scale Invariant Detectors

- Harris-Laplacian

使用LoG算子，具有尺度不变性

对于二维图像，计算图像在不同尺度下的离散拉普拉斯响应值，然后，检查位置空间中每个点。如果该点的拉普拉斯响应值都大于或小于其他26个立方空间邻域的值，那么该点就是被检测到的图像斑点

LoG具有尺度不变性，但是要对高斯函数求二次导，计算量大。能不能简化LoG算子呢？——使用DoG算子 (Difference of Gaussians)

SIFT算法建议，在某一个尺度上对斑点的检测，通过对两个相邻高斯尺度空间的图像相减，得到一个DoG (Difference of Gaussians)的响应值图像

流程

- 确定尺度空间的一组取值 $\sigma_l = (\sigma_0, \sigma_1, \dots, \sigma_n) = (\sigma, k\sigma, k^2\sigma, \dots, k^n\sigma)$, $\sigma_D = s\sigma_l$, 经验值 $s = 0.7$
- 对于确定的尺度空间值 σ_D , 计算Harris角点响应值
- 尺度空间搜索：计算候选点的Laplace响应值，并对于给定阈值作比较 $F(x, y, \sigma_i) = \sigma_i^2 |L_{xx}(x, y, \sigma_i) + L_{yy}(x, y, \sigma_i)| \geq \text{threshold}$
- 将响应值 F 与相邻两个尺度空间对应的拉式响应值进行比较，找到 $F(x, y, \sigma_i) > F(x, y, \sigma_l), l = i - 1, i + 1$ 对应的 i

Discard points with DOG value below threshold (low contrast)

- However, points along edges may have high contrast in one direction but low in another
- Compute principal curvatures from eigenvalues of 2x2 Hessian matrix, and **limit ratio** (Harris approach):

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad \text{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta, \quad \text{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta. \quad \frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$

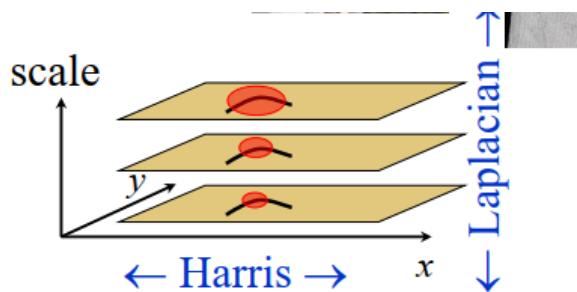
去除edge

- SIFT

■ Harris-Laplacian¹

Find local maximum of:

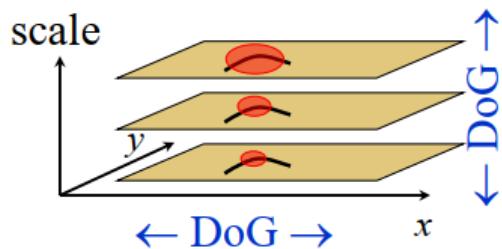
- Harris corner detector in space (image coordinates)
- Laplacian in scale



■ SIFT (Lowe)²

Find local maximum of:

- Difference of Gaussians in space and scale



¹ K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

² D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

Feature descriptors

SIFT算法是一种用于检测和描述图像中局部特征的算法。

[图像特征匹配方法——SIFT算法原理及实现](#)

Global histogram

SIFT:Scale Invariant Feature Transform descriptor在梯度上做

SIFT算法可以的解决问题：

1. 目标的旋转、缩放、平移 (RST)
2. 图像放射/投影变换 (视点viewpoint)
3. 光照影响 (illumination)
4. 部分目标遮挡 (occlusion)
5. 杂物场景 (clutter)
6. 噪声

1 Scale-space extrema detection

尺度空间极值检测

2 Keypoint localization

关键点定位

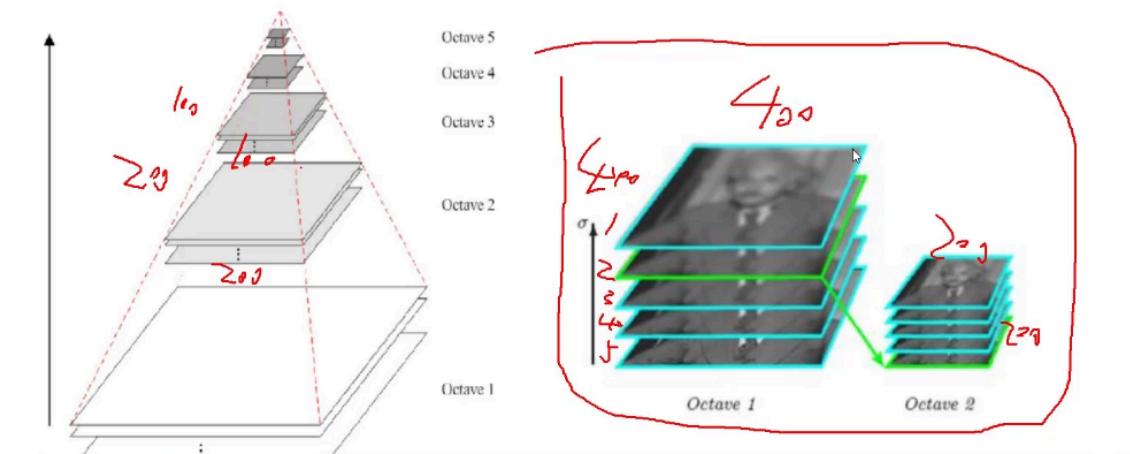
3 Orientation assignment

方向赋值

4 Keypoint descriptor

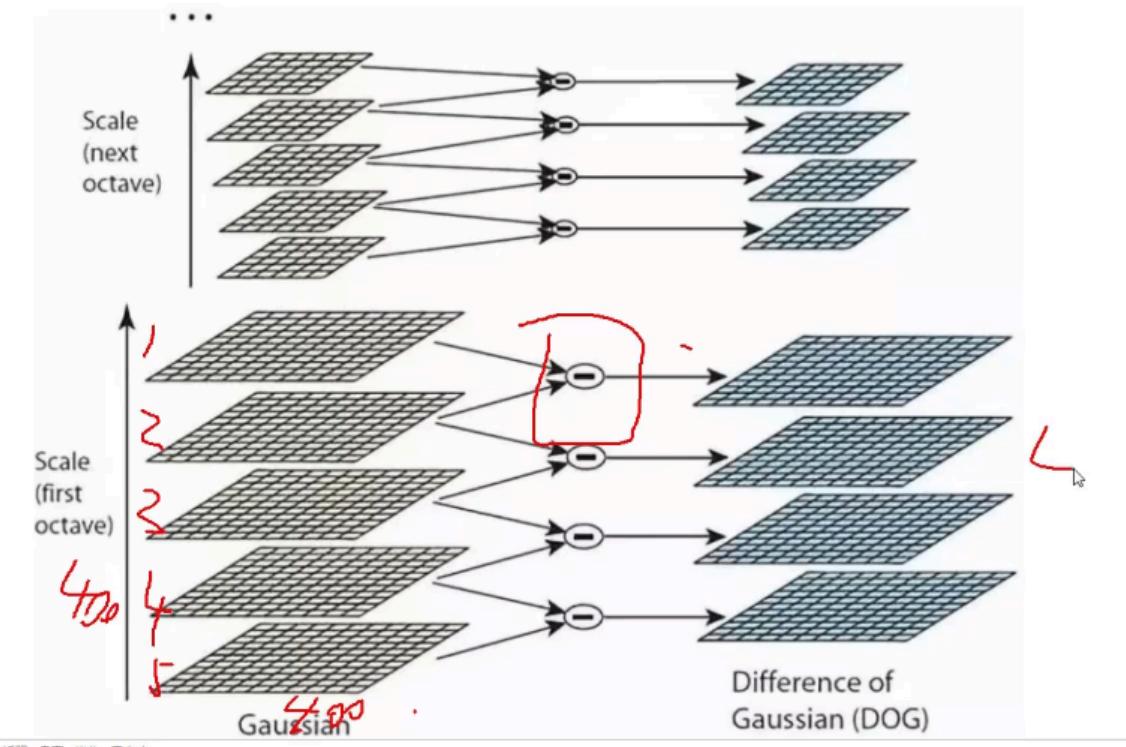
关键点描述

- 图像尺度空间：因为scale的变化，我们希望计算机对物体在不同尺度下有一个统一的认知，就要考虑图像在不同尺度下都存在的特点
- 多分辨率金字塔：不同尺度（塔的每层）下做不同分辨率的高斯滤波



有价值的东西：不同分辨率下不同的地方，因此我们有高斯差分金字塔（DOG）

- 高斯差分金字塔（DOG）：得到多层

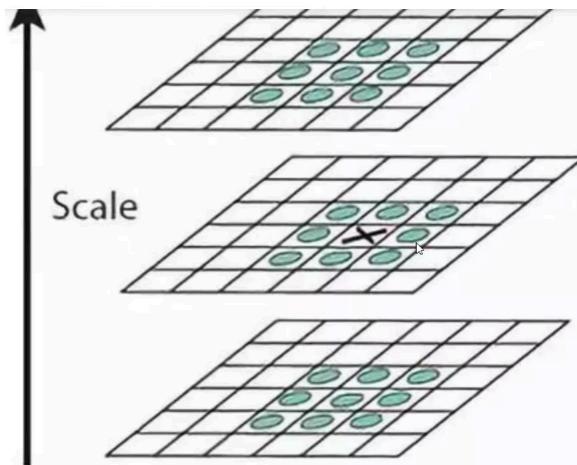


DOG定义公式:

$$D(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$$

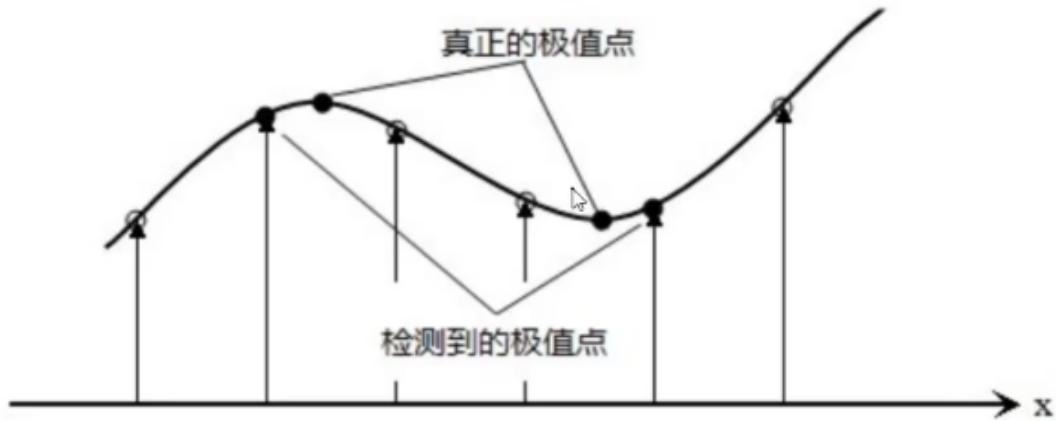
- DoG空间极值检测：找出极值点

特征点是由DOG空间的局部极值点组成的。为了寻找DOG函数的极值点，每一个像素点要和它所有的相邻点比较，看其是否比它的图像域和尺度域的相邻点大或者小



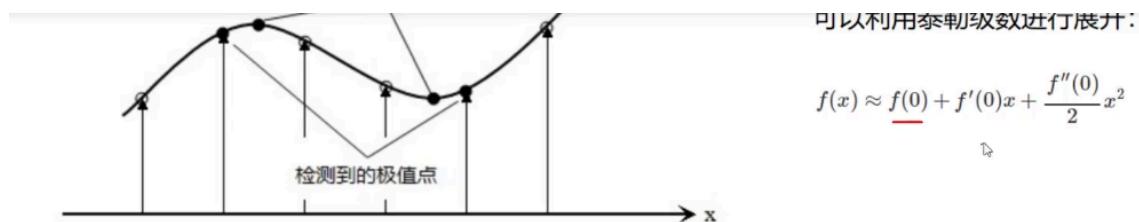
中间的检测点和它同尺度的8个相邻点和上下相邻尺度对应的9×2个点共26个点比较，以确保在尺度空间和二维图像空间都检测到极值点——keypoint

得到一堆离散的点——不一定全是真正的极值点



- 关键点的精确定位

对检测到的离散的点，对尺度空间DoG函数进行曲线拟合，计算其极值点，从而实现关键点的精确定位——利用泰勒级数进行展开



$$D(\Delta x, \Delta y, \Delta \sigma) = \underbrace{D(x, y, \sigma)}_{\text{检测到的极值点}} + \left[\frac{\partial D}{\partial x} \quad \frac{\partial D}{\partial y} \quad \frac{\partial D}{\partial \sigma} \right] \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \sigma \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta x & \Delta y & \Delta \sigma \end{bmatrix} \begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial x \partial \sigma} \\ \frac{\partial^2 D}{\partial y \partial x} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y \partial \sigma} \\ \frac{\partial^2 D}{\partial \sigma \partial x} & \frac{\partial^2 D}{\partial \sigma \partial y} & \frac{\partial^2 D}{\partial \sigma^2} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \sigma \end{bmatrix}$$

$$D(x) = D + \frac{\partial D^T}{\partial x} \Delta x + \frac{1}{2} \Delta x^T \frac{\partial^2 D^T}{\partial x^2} \Delta x \quad \Delta x = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D(x)}{\partial x}$$

求导令导数=0

关键点精确定位

1. DOG函数 $D(x)$ 的Taylor展开为 $D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X$
2. 令 $D(x)$ 的导数为0，得到极值点偏移量 $\hat{x} = -\left(\frac{\partial^2 D}{\partial x^2}\right)^{-1} \frac{\partial D}{\partial x}$
3. 若 $\hat{x} = (x, y, \sigma)^T$ 在任意维度大于0.5则说明极值点精确位置离另一个点更近，则改变当前关键点位置，定位到新点后重复上一步操作，迭代一定次数后仍不收敛则不认为该检测点为关键点
4. 精确关键点处函数值 $D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial X} \hat{x}$

- 消除边界响应

DOG算子有较强的边缘效应，边缘点的特征表现：某个防线有较大的主曲率，而其垂直方向主曲率较小

边缘效应是指在图像处理中，由于滤波器的应用，图像边缘区域的像素值受到不完整邻域的影响，导致这些区域的响应与图像内部区域的响应不同

Hessian矩阵：

$$H(x, y) = \begin{bmatrix} D_{xx}(x, y) & D_{xy}(x, y) \\ D_{xy}(x, y) & D_{yy}(x, y) \end{bmatrix} \quad \text{令} \alpha = \lambda_{\max} \text{为最大的特征值, } \beta = \lambda_{\min} \text{为最小的特征值}$$

$$\begin{aligned} Tr(H) &= D_{xx} + D_{yy} = \alpha + \beta \\ Det(H) &= D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \end{aligned} \quad \frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(\gamma + 1)^2}{\gamma}$$

Lowe在论文中给出的 $\gamma=10$, 也就是说对于主曲率比值大于10的特征点将被删除。

消除完后我们就得到了真正的最后的关键点, 下面要对得到的关键点进行描述

- 特征点的主方向

每个点 $L(x, y)$ 的梯度的模 $m(x, y)$ 以及方向 $\theta(x, y)$:

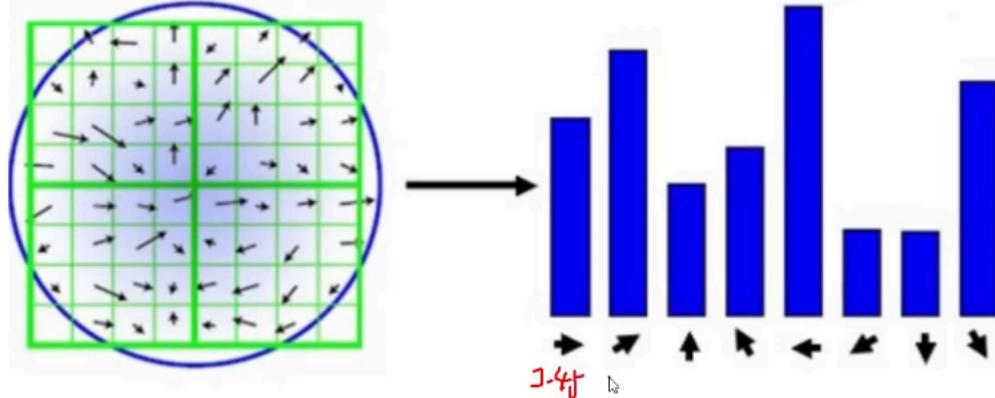
$$m(x, y) = \sqrt{[L(x+1, y) - L(x-1, y)]^2 + [L(x, y+1) - L(x, y-1)]^2}$$

$$\theta(x, y) = \arctan \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}$$

每个特征点可以得到三个信息(x, y, θ), 即位置、尺度和方向。具有多个方向的关键点可以被复制成多份, 然后将方向值分别赋给复制后的特征点, 一个特征点就产生了多个坐标、尺度相等, 但是方向不同的特征点。

- **关键点邻域:** 对于每个检测到的关键点, 选择一个以关键点为中心的局部区域 (通常是一个圆形区域, 半径为 $3 \times 1.5\sigma$, 其中 σ 是关键点的尺度)。
- 生成特征描述

在完成关键点的梯度计算后, 使用直方图统计邻域内像素的梯度和方向

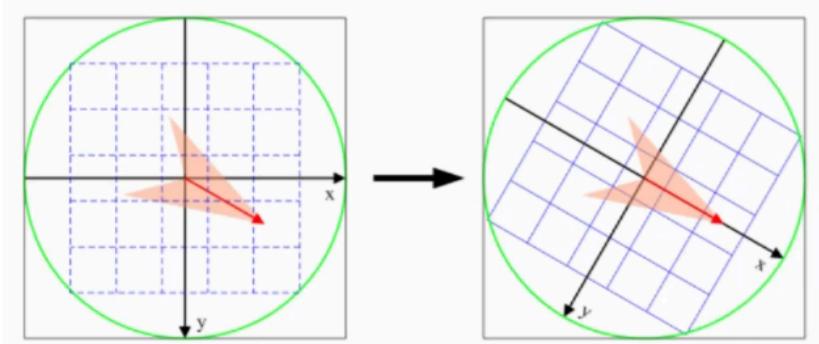


这样可以确定主方向

1. 在关键点为中心的8邻域窗口内计算关键点的梯度幅值和方向
2. 对每个关键点 (尺度为 σ) , 用直方图统计其一定领域内的像素梯度分布 (梯度方向可近似到 $\pm 45^\circ, \pm 90^\circ, \pm 135^\circ, 0^\circ, 180^\circ$ 上)
3. 找到直方图中对应个数最多的梯度方向, 即为当前关键点的方向

当有多个方向近似时, 我们可以把关键点复制成多份然后将方向分别赋给复制后的特征点——多峰值情况。每一份复制后的关键点具有相同的位置和尺度, 但分配了不同的主方向

为了保证特征矢量的旋转不变性，要以特征点为中心，在附近邻域内将坐标轴旋转 θ 角度，即将坐标轴旋转为特征点的主方向。

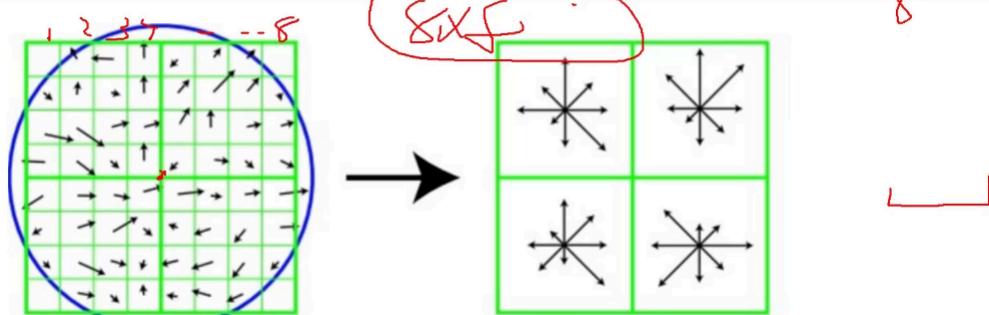


1. 将坐标轴旋转到关键点方向上：旋转后新坐标 $\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$
2. 对关键点的8*8领域内所有像素点计算其对应的梯度幅值和方向
3. 将8*8领域划分为4个4*4的小领域，在每个子领域上计算统计**每个梯度方向上对应所有梯度幅值的累加**⇒得到属于关键点的4*8共32个特征（每个子领域8个方向对应8个特征）
4. 将32个特征的值归一到(0,1)内，最后将关键点用上述得到的32个值描述，即L = $(l_1, l_2, \dots, l_{128}), l_i \in (0,1)$ ——有128维是因为原论文中作者推荐对16 × 16邻域进行切分

保证旋转不变性——预处理已经完成

- **旋转归一化**：在生成描述子时，将关键点邻域的梯度方向相对于主方向进行旋转归一化。这意味着描述子是基于关键点的局部坐标系生成的，而不是基于图像的全局坐标系。

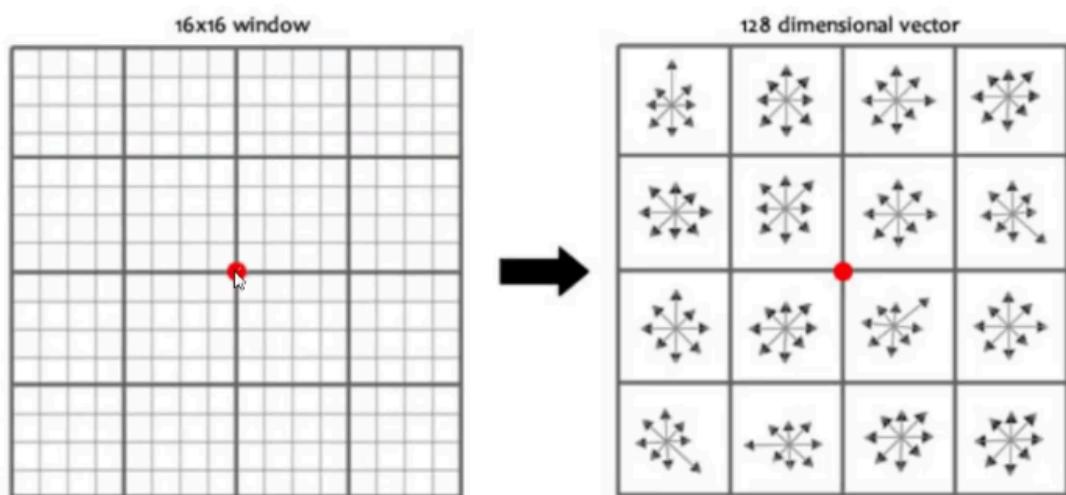
旋转之后的主方向为中心取8x8的窗口，求每个像素的梯度幅值和方向，箭头方向代表梯度方向，长度代表梯度幅值，然后利用高斯窗口对其进行加权运算，最后在每个4x4的小块上绘制8个方向的梯度直方图，计算每个梯度方向的累加值，即可形成一个种子点，即每个特征由4个种子点组成，每个种子点有8个方向的向量信息。



论文中建议对每个关键点使用4x4共16个种子点来描述，这样一个关键点就会产生128维的SIFT特征向量。

128=16*8维的SIFT特征向量

论文中建议对每个关键点使用4x4共16个种子点来描述，这样一个关键点就会产生128维的SIFT特征向量。



- SIFT特征的匹配——度量两幅图像中关键点的相似性

$$ratio = \frac{\text{最近邻距离}}{\text{次近邻距离}}$$

对图像A中某一关键点，找出图像B中与其距离最近的两个关键点，采用 $ratio = \frac{\text{最近邻距离}}{\text{次近邻距离}}$ 的评价方法：
ratio的阈值越小说明匹配准确度越高

- ratio = 0.4: 对准确度要求较高的匹配
- ratio = 0.5: 一般情况的取值
- ratio = 0.6: 对匹配点数目要求较多的匹配

一般地，ratio = 0.8则认为当前匹配为错误匹配

- 优点
 - 尺度/光照/旋转不变性
 - 在刚体的表征上尤其有效
 - 局部表征能力强
- 缺点
 - 耗时
 - 处理非刚性边缘时表现较差
 - 严重的仿射扭曲下效果较差
- 为什么使用梯度消息
 - 梯度信息反映了图像中像素值的变化方向和强度，能够捕捉图像中的边缘、角点和其他局部结构特征。使用梯度信息的好处包括：
 - **局部特征捕捉**: 梯度信息能够有效地描述图像中局部区域的形状和纹理特征。
 - **对光照变化鲁棒**: 梯度是像素值的相对变化，对光照的线性变化不敏感。
 - **计算简单高效**: 梯度计算是一种简单且高效的操作，适合实时应用。
 - 好处
 - **对旋转不变性的支持**: 通过梯度方向，可以为关键点分配主方向，使得描述子能够相对于关键点的方向进行归一化。
 - **对尺度变化的鲁棒性**: 梯度信息在高斯金字塔的不同尺度上计算，使得描述子对尺度变化具有鲁棒性。
 - **对噪声的鲁棒性**: 梯度信息在高斯平滑后的图像上计算，能够减少噪声的影响。

Image stitching

[机器视觉笔记：RANSAC算法以及思想 ransac 参数](#)

RANSAC算法就是一种剔除离群点的很好的一种方法

Procedure:

1. Detect feature points in both images

- 检测关键点
- 建立SIFT描述子

2. Find corresponding pairs

- 匹配SIFT描述子

Euclidean distance between descriptors ?

3. Use these pairs to align the images

- 计算转化矩阵

- 变换矩阵 $T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$ ——一组点对提供两个方程→需要至少三个点对

- RANSAC提高求解准确度

4. Image Blending

采用Pyramid Blending——还有更好的blending方法

RANSAC

- 解决问题：用于从包含噪声和异常值的数据集中估计数学模型的参数
- 核心思想：通过**随机采样**和**一致性检验**来估计模型参数，从而从包含噪声和异常值的数据集中恢复出正确的模型
 - 一致性的目的在于比较不同方法得到的结果是否具有一致性s
- 与最小二乘相比
 - 最小二乘法是一种通过最小化误差的平方和来拟合模型的方法。它**假设所有数据点都符合模型**，并试图找到使所有数据点误差平方和最小的模型参数
 - 它通过随机采样和一致性检验，能够有效地排除外点对模型估计的影响
 - RANSAC算法能够从多个模型中选择出最佳的模型，而最小二乘法通常只能拟合一个模型
 - RANSAC算法最多可以处理50%的外点情况，而最小二乘法在数据中存在大量异常值时，拟合结果会受到严重影响
- 缺点：
 - RANSAC算法需要进行大量的随机采样和模型估计，计算复杂度较高
 - 需要设置多个参数
 - RANSAC算法要求模型已知，且模型参数可以通过内点来估计，这限制了其应用范围
- 与Hough变换相比
 - 共同之处：都用于从包含噪声和异常值的数据集中提取几何形状或模型参数。它们都旨在提高模型拟合的鲁棒性，能够处理数据中的噪声和异常值
 - 两者在计算机视觉和图像处理领域都有广泛的应用
 - 具有一定的鲁棒性，能够从不完美的数据集中提取出有用的几何信息
 - 差异
 - Hough投票选择
 - 适用于检测特定的几何形状，如直线、圆等。它对形状的参数化有明确的要求，需要将形状表示为参数空间中的曲线
 - 结果是参数空间中的峰值，需要进一步解释为原空间中的几何形状
 - RANSAC：通过随机采样和一致性检验来拟合模型，它会丢弃与模型不一致的数据点（异常点）

适用于各种类型的模型拟合问题，不限于特定的几何形状。它对模型的形式没有特定的要求，只要能够通过数据点拟合出模型即可

- 基本假设
 - 正确数据：内点，可以被模型描述
 - 异常数据：偏离正常范围很远
- 基本步骤
 - 随机采样：在样本集中随机抽取n个样本，构成S，基于S中的样本对初始模型进行估计
 - 模型验证：计算样本集中其他样本到模型的误差——>误差小于阈值的样本+S中样本共同构成内点集S*
 - 重复上述步骤并保留到目前为止最好的内点集：数据点最多&其他样本到该集合的残差最小——称为最好
 - 输出最佳模型的参数
- 参数确定：每次的随机采样数n（用于确定模型的样本个数，例如对直线，n=2）和重复次数K
 - 定义内点比例 $\omega = \text{内点数}/\text{样本总数}$
 - 结论1： $K = \frac{\log(1-P)}{\log(1-\omega^n)}$ —— ω 不高时，n过大导致K急剧增大
 - 实际应用中，一般 ω 难以预先估计——采用自适应估计法
 - 初始化 $K = \infty$, count = 0, P = 0.99
 - 随机采样n个样本、计算模型并检查内点数
 - 计算 $\omega = \text{上一步中求得的内点数}/\text{样本总数}$
 - 计算 $K = \frac{\log(1-P)}{\log(1-\omega^n)}$, count++
 - 重复上述步骤，直到 $K < \text{count}$
- 优点：
 - 适用性强（能够解决很多模型拟合问题）
 - 易于实现
- 缺点：
 - 保证代价（迭代次数，计算耗时等）不过大的基础上，只能处理外点比例不高的数据——In contrast, Hough变换能处理外点比例很高的数据集合

分区 计算机视觉 的第 6 页

- 实际问题中很多数据集的外点比例很高——maybe可以通过随机选择子集提高性能
- outlier比例给定的情况下，k次采样后成功的概率是

$$1 - (1 - w^n)^k$$

RANSAC loop:

1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
2. Compute transformation from seed group
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, **re-compute** least-squares estimate of transformation on all of the inliers

Keep the transformation with the largest number of inliers

- How many samples are needed?
 - Suppose w is **fraction of inliers** (points from line).
 - n points needed to define **hypothesis** (2 for lines)
 - k **samples** chosen (each sample consists of **n points**).
- Prob. that a single sample of n points is correct: w^n
- Prob. that all k samples fail is: $(1-w^n)^k$

⇒ Choose k high enough to keep this below desired failure rate.

RANSAC: Computed k ($p=0.99$)

Sample size n	Proportion of outliers						
	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

金字塔

[你真正了解图像金字塔吗？详细介绍拉普拉斯金字塔和高斯金字塔（pyrDown\(\) and pyrUp\(\)）](#)

- 下采样 (Downsampling)

下采样是指将图像的分辨率降低，即减少图像的像素数量。通常通过隔行隔列采样或平滑后采样来实现。

- **直接下采样**：每隔一定间隔（如每隔2个像素）取一个像素值，直接降低图像分辨率。
- **平滑后下采样**：先对图像进行平滑（如高斯模糊），然后再进行下采样。这种方法可以减少下采样过程中引入的混叠效应（Aliasing）。

- 上采样 (Upsampling)

上采样是指将图像的分辨率提高，即增加图像的像素数量。通常通过插值来实现。

- **最近邻插值 (Nearest Neighbor Interpolation)**：将新像素的值设置为最接近的原始像素值。
- **双线性插值 (Bilinear Interpolation)**：根据周围4个原始像素的值进行线性插值。
- **双三次插值 (Bicubic Interpolation)**：根据周围16个原始像素的值进行三次插值，效果更好但计算量更大。

图像金字塔是由一幅图像的多个不同分辨率的子图所构成的图像集合

- 高斯金字塔 (Gaussian Pyramid)

一种多分辨率图像表示方法；通过一系列的高斯平滑和下采样操作，生成一组分辨率逐渐降低的图像层次结构。

高斯核的标准差 σ 决定了平滑的程度。

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

从尺度 (scale) 的角度理解高斯金字塔，可以将其视为对图像在不同尺度下的表示和处理

在图像处理中，**尺度**指的是图像的分辨率或细节的粗细程度。

- 高斯金字塔的每一层对应一个特定的尺度。
- **低层**（高分辨率）：捕捉图像的细节信息，如边缘、纹理等。
- **高层**（低分辨率）：捕捉图像的整体结构信息，如物体的轮廓、大范围的光照变化等。

- 拉普拉斯金字塔 (Laplacian Pyramid)

它通过捕捉图像在不同尺度下的高频信息，实现图像的多尺度表示和处理。从频率角度理解拉普拉斯金字塔，可以将其视为对图像**高频成分的分离和表示**

- 拉普拉斯金字塔的每一层是高斯金字塔相邻两层之间的差异，捕捉了图像在不同尺度下的高频信息
- 通过这种方式，拉普拉斯金字塔能够保留图像的细节信息，而高斯金字塔则更多地保留了图像的平滑信息。
- 拉普拉斯金字塔的作用在于，能够恢复高分辨率的图像

$$Li = Gi - pyrUp(Gi + 1)$$

怎么理解拉普拉斯金字塔的每一层是带通滤波？

- 拉普拉斯金字塔可以看作是对图像进行频带分解的工具。每一层捕捉了图像在不同频率范围内
的信息。
- 通过拉普拉斯金字塔，可以将图像分解为多个频带，从而实现对图像的多尺度分析。

Eigenface

PCA

用于降维

- PCA的核心思想是通过线性变换将原始数据集中的多个变量转换为少数几个不相关的主成分，从而
减少数据的维度，同时尽可能保留原始数据中最重要的信息。**最大化投影数据的方差**
- PCA有效的数据：

不是每一维协方差都大，不同特征之间有明显差别；线性空间，gap为数据，数据已经标准化，数
据中存在冗余特征

但是如果PCA样本点不太好，每一维协方差都大，即区分度不高，可能就G了

- 选择多少个特征向量：

方差解释率：每个主成分都有一个对应的特征值，特征值表示该主成分所解释的方差大小。通常，
我们按照特征值的大小对主成分进行排序，并计算每个主成分所解释的方差比例（即特征值与所有
特征值之和的比值）

保留前几个特征满足95%以上的方差即可

- PCA分析与DCT离散余弦变换的相同之处？不同之处？

什么是DCT离散余弦变换？[离散余弦变换\(DCT\)原理及应用 二维dct谱](#)

相同之处

1. 降维和压缩：

- **PCA**：通过线性变换将原始数据集中的多个变量转换为少数几个不相关的主
成分，从而减少数据的维度，同时尽可能保留原始数据中最重要的信息 [5](#)
-
- **DCT**：通过将时域或空域信号转换到频域，能够有效地捕捉信号的主要能
量，从而实现数据的降维和压缩 [3](#)。

2. 特征提取：

- **PCA**：通过计算数据的协方差矩阵，找到数据的主成分，这些主成分是数据
中的主要模式或特征 [5](#)。
- **DCT**：通过将信号转换到频域，能够提取出信号的频率特征，这些特征通常
与信号的主要模式或结构相关 [3](#)。

不同之处

1. 变换类型:

- **PCA**: 是一种线性变换，通过正交变换将数据从原始空间转换到主成分空间⁵。
- **DCT**: 是一种可分离的正交变换，通过将信号从时域或空域转换到频域，能够有效地捕捉信号的主要能量³。

2. 数学基础:

- **PCA**: 基于数据的协方差矩阵和特征值分解，通过最大化方差来找到主成分⁵。
- **DCT**: 基于余弦函数的正交性，通过将信号表示为余弦函数的线性组合来实现变换³。

3. 能量集中特性:

- **PCA**: 通过最大化方差来找到主成分，但并不特别强调能量的集中性。
- **DCT**: 具有能量集中特性，大多数自然信号（包括图像和音频）的能量都集中在DCT变换后的低频部分，这使其在信号压缩中特别有效⁶。
- PCA降维后，还能重构再升维：
 - 由于PCA的降维是通过线性变换实现的，因此可以通过逆变换将降维后的数据重构回原始空间。
 - 重构的过程实际上是通过降维后的数据和PCA过程中得到的变换矩阵，计算出原始数据的近似值

3. 为什么可以重构再升维?

- **主成分保留了主要信息**: PCA通过选择方差最大的主成分，保留了数据的主要信息。虽然降维过程中丢失了部分信息，但这些信息通常是噪声或冗余特征。
- **线性变换的可逆性**: PCA的降维和重构过程都是线性变换。由于主成分矩阵 \mathbf{U}_k 是正交矩阵 ($\mathbf{U}_k^T \mathbf{U}_k = \mathbf{I}$)，重构过程可以看作是对降维数据的逆变换。
 - **线性变换**: PCA的降维和重构都是通过线性变换实现的，因此重构后的数据是原始数据的线性组合。
 - **信息保留**: PCA在降维时会保留数据中的主要模式或特征，因此重构后的数据能够较好地还原原始数据的主要信息。
 - **近似恢复**: 由于PCA在降维时会丢失一些信息，因此重构后的数据通常是原始数据的近似值，而不是完全准确的值
- 以下是PCA的理论推导

- d-维空间 $\mathbf{x} = (x_1, x_2 \dots, x_d)$
- 投影方向 $a_1 = (a_1^1, a_1^2, \dots, a_1^d)^T$ where $a_1^T a_1 = 1$
- 投影值 $z_1 = a_1^T x = \sum_{i=1}^d a_1^i x_i$
- 问题
 - 最大化 $\text{var}(z_1)$
 - 求投影方向 $\arg \max_{a_1} \text{var}(z_1)$

$$\begin{aligned}
 \text{var}(z_1) &= E(z_1^2) - [E(z_1)]^2 = E\left[\left(\sum_{i=1}^d a_1^i x_i\right)^2\right] - \left[E\left(\sum_{i=1}^d a_1^i x_i\right)\right]^2 \\
 &= \sum_{i,j=1}^d a_1^i a_1^j E(x_i x_j) - \sum_{i,j=1}^d a_1^i a_1^j E(x_i) E(x_j) \\
 &= \sum_{i,j=1}^d a_1^i a_1^j [E(x_i x_j) - E(x_i) E(x_j)] \\
 &= \sum_{i,j=1}^d a_1^i a_1^j S_{ij} \quad \text{令 } S_{ij} = E(x_i x_j) - E(x_i) E(x_j) \\
 &= \mathbf{a}_1^T \mathbf{S} \mathbf{a}_1 \quad \text{S ?} \quad \text{cov}(x_i, x_j) = E(x_i x_j) - E(x_i) E(x_j)
 \end{aligned}$$

最大化: $\text{var}(z_k) = \mathbf{a}_k^T \mathbf{S} \mathbf{a}_k$

约束条件: $\mathbf{a}_k^T \mathbf{a}_k = 1$ and $\text{cov}(z_k, z_l) = 0$ for $k > l \geq 1$

\Downarrow

$$\text{cov}(z_k, z_l) = \mathbf{a}_k^T \mathbf{S} \mathbf{a}_l = \lambda_l \mathbf{a}_k^T \mathbf{a}_l = 0$$

类似, 得必要条件:

$$\mathbf{S} \mathbf{a}_k = \lambda \mathbf{a}_k \iff \mathbf{a}_k \text{ 为 k}^{\text{th}} \text{ 大的特征值对应的特征向量}$$

$$\mathbf{x} = (x_1, x_2 \dots, x_d) \quad \mathbf{z} = \mathbf{A}^T \mathbf{x} \quad \mathbf{A} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d] \quad z = (z_1, z_2 \dots, z_d)$$

且有，

$$\text{var}(z_i) = \lambda_i \quad \text{for } i \geq 1 \quad \text{and} \quad \text{cov}(z_i, z_j) = \lambda_i \delta_{ij}$$

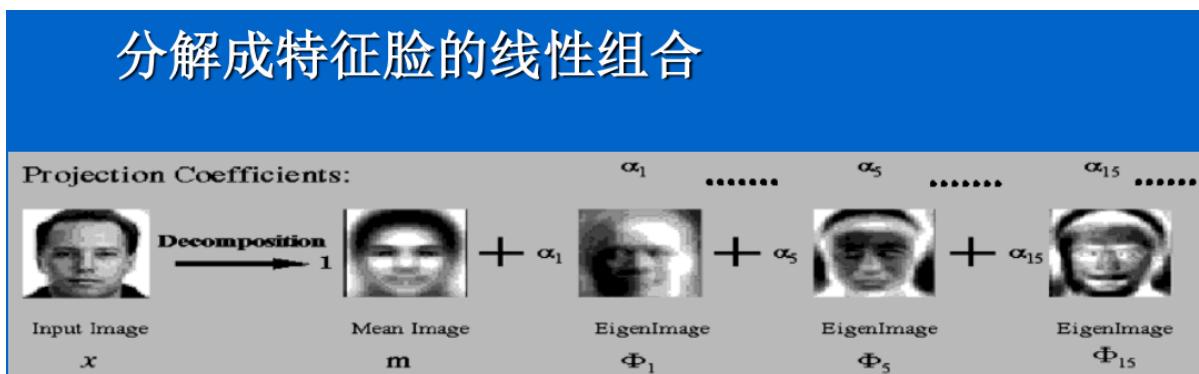
故，新坐标中的**协方差矩阵被对角化了**，即

$$C_z = [\text{cov}(z_i, z_j)] = \begin{pmatrix} \lambda_1 & & & 0 \\ & \ddots & & \\ 0 & & \ddots & \lambda_d \end{pmatrix}$$

Eigenface

Eigenface 算法的思想是希望能够将高维的图像数据**降维**，以此实现对不同人脸的特征刻画。

Eigenface 降维图像数据的方法是寻找一组特征脸，将特征脸作为一组基，**人脸信息便可以描述为特征脸的线性组合再加上一张平均脸**



[EigenFace的原理、实现及性能评估 eigenface算法](#)

Eigenface人脸识别算法

- 算法流程
 - 对数据库中人脸图像作归一化处理
 - 用PCA计算得到一组特征脸 (特征向量)
 - 计算数据库中每个人脸图像在该特征脸所张成的子空间上的坐标
 - 对每一输入图像，归一化后求解其在特征脸子空间中的坐标，并与库中人脸比较，验证相似性
- 预处理
 1. Mask: 根据人脸两只眼睛的中心位置 旋转/平移/缩放，使所有训练人脸图像与模板对齐——根据模板切出人脸区域
 2. 灰度值归一化：直方图均衡 & 直方图拉伸
- 训练过程
 - 向量一维化： x_i 为M * N的人脸图像对应的一维化矩阵， $1 \leq i \leq K$
 - 求协方差矩阵： $\Sigma = \frac{1}{K} \sum_{i=1}^K (x_i - u)(x_i - u)^T$, $u = mean(x_1, x_2, \dots, x_K)$
 - 求矩阵 Σ 的特征值和相应的单位特征向量 (MN维)
 - 构建转换矩阵，求出子空间中的坐标： $y_i = A^T x_i$, $A = [v_1, v_2, \dots, v_k]$, $k \ll K$
- 识别与重构 (待识别样本f)
 - 识别： $y_f = A^T f$, 比较 y_f 和 y_i
 - 重构： $\hat{f} = Ay_f$
- 特征脸个数k的选取： $\text{argmin}_k \left(\frac{\sum_1^K \lambda_i}{\sum_1^K \lambda_i} \geq \alpha \right)$, α 常取0.95~0.99 (即选取的特征值在总特征值中占比已经足够大)
- 理解利用人脸重构进行人脸检测的原理。如果一幅白噪声图像用Eigenface去重构，预计结果会是怎么样？原因是？

对于一个新的人脸图像，将其投影到特征脸空间，得到在该空间中的坐标。

使用这些坐标和特征脸，可以重构出该人脸图像。具体来说，重构公式为：

$$F = F_m + \sum_{i=1}^k \alpha_i F_i$$

如果使用Eigenface算法对一幅白噪声图像进行重构，预计结果会是：

1. 特征脸空间中的投影：

- 白噪声图像在特征脸空间中的投影坐标会非常接近于零，因为白噪声图像与训练集中的人脸图像差异很大，无法很好地表示为特征脸的线性组合。

2. 重构结果：

- 由于白噪声图像在特征脸空间中的投影坐标接近于零，使用这些坐标和特征脸进行重构时，得到的图像会非常接近于平均人脸图像 F_m 。
- 重构结果可能是一个模糊的人脸图像，缺乏具体的人脸特征，因为白噪声图像中没有包含有用的人脸信息。

- 思考：Eigen-X应用过程重点需要注意什么？

选择合适的降维参数k，以在保留足够信息的同时减少计算复杂度

确保所有训练图像和测试图像都经过相同的预处理步骤，如灰度化、尺寸归一化和面部对齐，以消除由于面部姿态、光照条件和表情变化引起的偏差

将所有图像归一化到相同的尺度，以确保在计算协方差矩阵时，每个像素的贡献是相同的

- 除上课提到的人脸、手型、人体形状之外，试举例，你觉得哪些数据可能比较适合用EigenX方法去建模？

文本分类、情感分析；金融数据

- **特征提取**：从金融数据中提取特征，如股票价格、交易量等，然后使用PCA降维。
- **应用**：风险评估、投资组合优化

基因表达水平，疾病诊断

Performance Evaluation

- **Verification/Authentication**
 - Claim ID at first!
 - 1 v.s. 1
 - ROC
 - EER
- **Identification**
 - 1 v.s. n
 - CMC
 - Rank-1 rate

FAR vs FRR

一个识别系统给出的一次识别结果的对错可分为四种情况：

- I. 合法的用户被正确接受,
- II. 合法的用户被错误拒绝;
- III. 非法的用户被正确拒绝,
- IV. 非法的用户被错误接受。

错误接受率
False Accept Rate, FAR
False Alarm Rate,

错误拒绝率
False Reject Rate, FRR

Motion Estimation

optical flow光流法

- 解决的是什么问题：
用于分析连续帧间像素运动；运动估计，目标跟踪，三维重建
- 三个基本假设
 - 亮度一致性brightness constancy：目标像素强度在相邻帧不发生变化—— $I(x + u, y + v, t + 1) = I(x, y, t)$
 - 空间一致性spatial coherence：相邻像素拥有相似运动。
 - 微小运动small motion：

一个点的约束等式

1. 假设一个目标像素在t时刻亮度为 $I(x, y, t)$, $t + \delta t$ 时刻亮度为 $I(x + u, y + v, t + \delta t)$, 则两者相等
2. Taylor展开得 $I_x u + I_y v + I_t = 0$, $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, $I_t = \frac{\partial I}{\partial t}$, 即 $-I_t = \nabla I \cdot \begin{pmatrix} u \\ v \end{pmatrix}$

■ Combining these two equations

$$\begin{aligned}
 0 &= I(x+u, y+v) - H(x, y) \\
 &\approx I(x, y) + I_x u + I_y v - H(x, y) \quad \text{shorthand: } I_x = \frac{\partial I}{\partial x} \\
 &\approx (I(x, y) - H(x, y)) + I_x u + I_y v \\
 &\approx I_t + I_x u + I_y v \\
 &\approx I_t + \nabla I \cdot [u \ v]
 \end{aligned}$$

■ In the limit as u and v go to zero, this becomes exact

$$\nabla I \cdot [u \ v]^T + I_t = 0$$

- 哪些位置的光流比较可靠？为什么？

使用技巧：尽量避免用边缘上的点计算光流——使用纹理复杂区域，梯度比较大且方向不同，求出来的特征值比较大（即角点，避免孔径问题(Aperture Problem)）

Lucas-Kanade flow

LK有一个window的概念，即我先划定一块区域比如(5x5)的像素区域，我们可以认为这块区域每个点的移动速度 u 、 v 是一致的

孔径问题(Aperture Problem)：所以我们在追光流的时候，选点通常会选目标的角点(cornor)

5x5 window: $\begin{matrix} A & d = b \\ 25 \times 2 & 2 \times 1 & 25 \times 1 \end{matrix} \longrightarrow \text{minimize } \|Ad - b\|^2$

- Problem: we have more equations than unknowns
- Solution: solve least squares problem
 - minimum least squares solution given by solution (in d)
of: $(A^T A) d = A^T b$
 - $A^T A$ $2 \times 2 \quad 2 \times 1 \quad 2 \times 1$
 - $A^T b$ $\left[\begin{array}{cc} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{array} \right] \left[\begin{array}{c} u \\ v \end{array} \right] = - \left[\begin{array}{c} \sum I_x I_t \\ \sum I_y I_t \end{array} \right]$
 - The summations are over all pixels in the $K \times K$ window
 - This technique was first proposed by Lucas & Kanade (1981)

直接最小二乘求解

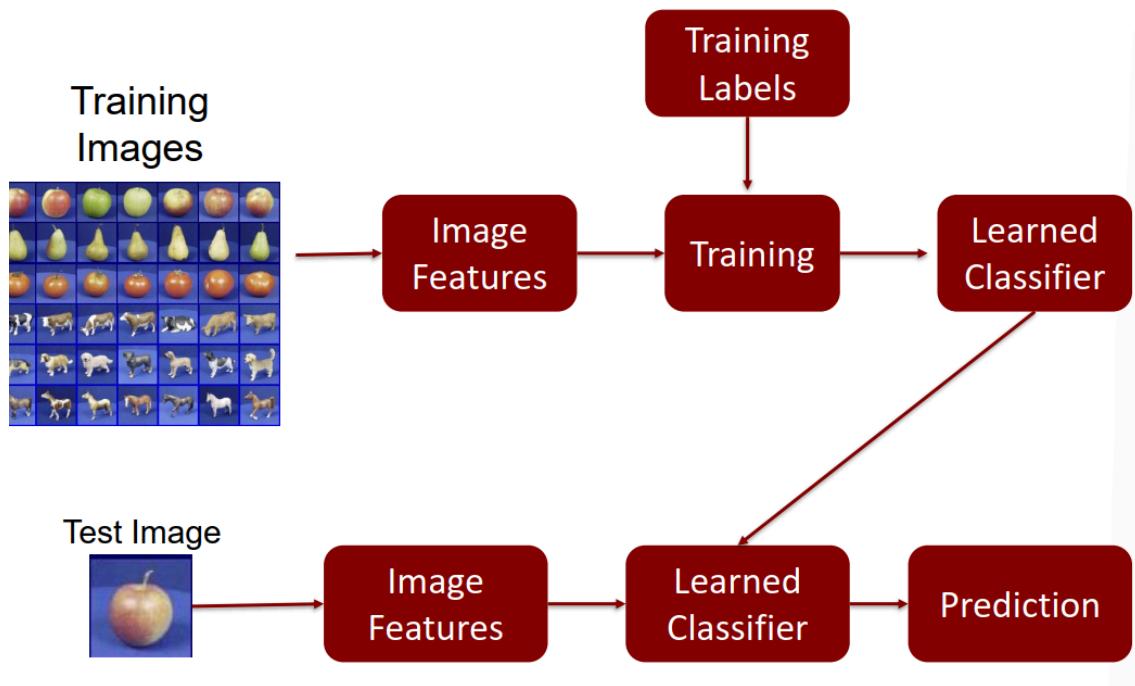
可解性

1. $A^T A = \sum \lambda I$ 需可逆
2. $A^T A$ 的特征值不能太小——防止被噪声干扰
3. $A^T A$ 的特征值 $\lambda_1, \lambda_2 (\lambda_1 > \lambda_2)$ 需满足 $\frac{\lambda_1}{\lambda_2}$ 不能太大

Visual Recognition

classification/detection

- 基本任务大概可以分为哪几大类
 - 图片分类
 - 检测和定位物体/图片分割
 - 估计语义和几何属性
 - 对人类活动和事件进行分类
- 都有哪些挑战因素
 - 视角变换
 - 光线变化
 - 尺度变化
 - 物体形变
 - 物体遮挡
 - 背景凌乱
 - 内部类别多样
- Bias-Variance Trade-off
 - Bias: how much the average model over all training sets differ from the true model? 所有训练集的平均模型与真实模型有多少差异?
 - | Error due to inaccurate assumptions/simplifications made by the model
 - Variance: how much models estimated from different training sets differ from each other 方差：从不同训练集估计的模型彼此之间的差异程度
- 模型复杂度和overfit underfit 的关系
 - Underfitting: model is too “simple” to represent all the relevant class characteristics
 - High bias and low variance
 - High training error and high test error
 - Overfitting: model is too “complex” and fits irrelevant characteristics (noise) in the data
 - Low bias and high variance
 - Low training error and high test error
- a simple pipeline



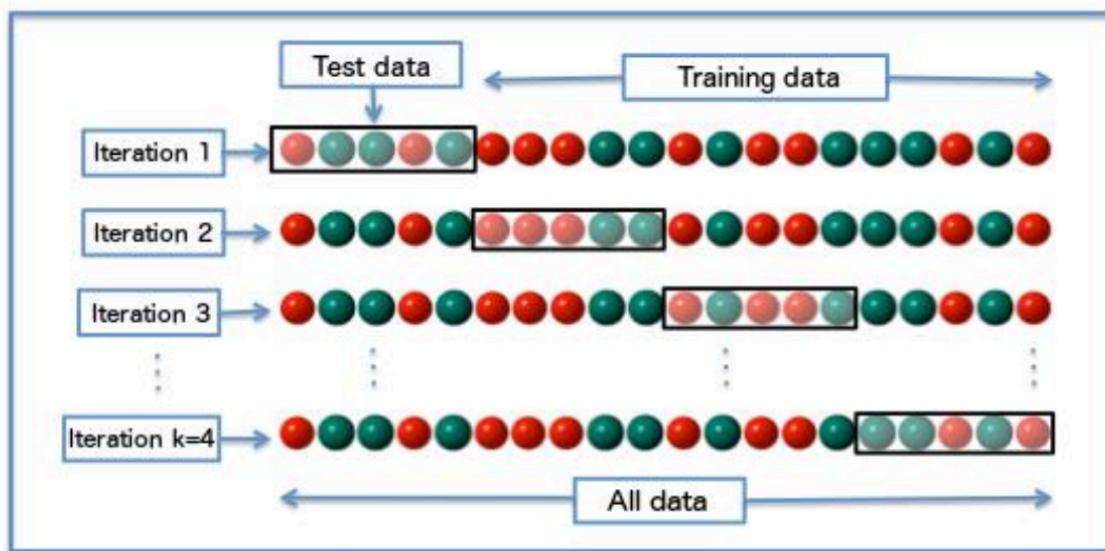
KNN

[KNN算法 \(k近邻算法\) 原理及总结](#)

特征空间匹配，然后选k个最近邻居，多数投票得到其类别

- K的选取

交叉验证选择cross validate



- 点距离的计算
- 归一化
- 维度爆炸

- Simple, a good one to try first
- Very flexible decision boundaries
- With infinite examples, 1-NN provably has error that is at most twice Bayes optimal error (out of scope for this class).

BoW

图像的 **BoW (Bag-of-Words, 词袋模型)** 是一种从图像中提取特征并表示为固定长度向量的方法。它最初源自自然语言处理 (NLP)，用于将文本表示为单词的频率向量。在计算机视觉中，BoW 模型被扩展用于图像处理，通过将图像中的局部特征（如关键点或局部描述子）类比为“视觉单词”，从而将图像表示为一个“视觉词袋”。

- 图像的BoW(bag-of-words)是指什么意思？
BoW 模型被扩展用于图像处理，通过将图像中的局部特征（如关键点或局部描述子）类比为“视觉单词”，从而将图像表示为一个“视觉词袋”。构建一个K维的直方图向量
- 如何构建visual words？

(1) 提取局部特征

- 使用特征检测算法（如 SIFT、SURF 或 ORB）从图像中提取局部特征。
- 每个特征点对应一个描述子（Descriptor），描述子是一个向量，表示该特征点的局部信息。

(2) 构建视觉词典 (Visual Vocabulary)

- 将所有图像的描述子集合起来，使用聚类算法（如 K-Means）将这些描述子聚类成 K 个簇。
- 每个簇的中心称为一个“视觉单词”，所有视觉单词构成一个“视觉词典”。

(3) 量化局部特征

- 对于每个图像的描述子，找到最近的视觉单词（即最近的簇中心），并将其映射到该视觉单词。
- 这个过程称为“量化”（Quantization），将局部特征映射到视觉词典中的某个单词。

(4) 构建词袋向量

- 统计每个视觉单词在图像中出现的频率，构建一个K维的直方图向量。
- 这个向量就是图像的 BoW 表示。

基本步骤

1. 特征提取与表示 (SIFT/SURF算法)
2. 通过训练样本聚类来建立字典(codewords dictionary) (常Kmeans)
3. 用字典的直方图来表达一张图像

Represent an image with histogram of codebook (i.e. Bag-of-words of an image)

4. 根据bag of words来分类未知图像：基于K个视觉词对未知图片建立直方图，并比较其与训练集的直方图的距离，取距离最短即为最佳匹配

Discriminative判别

基于卷积的物体识别

New Feature Extractor

$$Wx = \hat{y}$$

This simple computation is called a *fully-connected layer!*

W : the (10x1024) matrix of weight vectors

x : the (1024x1) image vector

\hat{y} : the (10x1) vector of class “probabilities”

- W : the (10x1024) matrix of weight vectors

Key Insight:

We want:

$$\arg \max(\hat{y}) = \arg \max(y)$$

Which we can accomplish by:

$$W^* = \arg \min_W \left(- \sum_{x,y} \log(p_c) \right)$$

Where p_c is the probability of the true class in \hat{y}

- Softmax 函数的主要作用是将网络的原始输出（通常称为 logits）转换为概率分布
- W 矩阵的组成权重矩阵，
含义权重矩阵 W 的每个元素 W_{ij} 表示输入特征 i 对输出特征 j 的贡献

Deep Learning

- 怎么理解被称为end-to-end的学习？

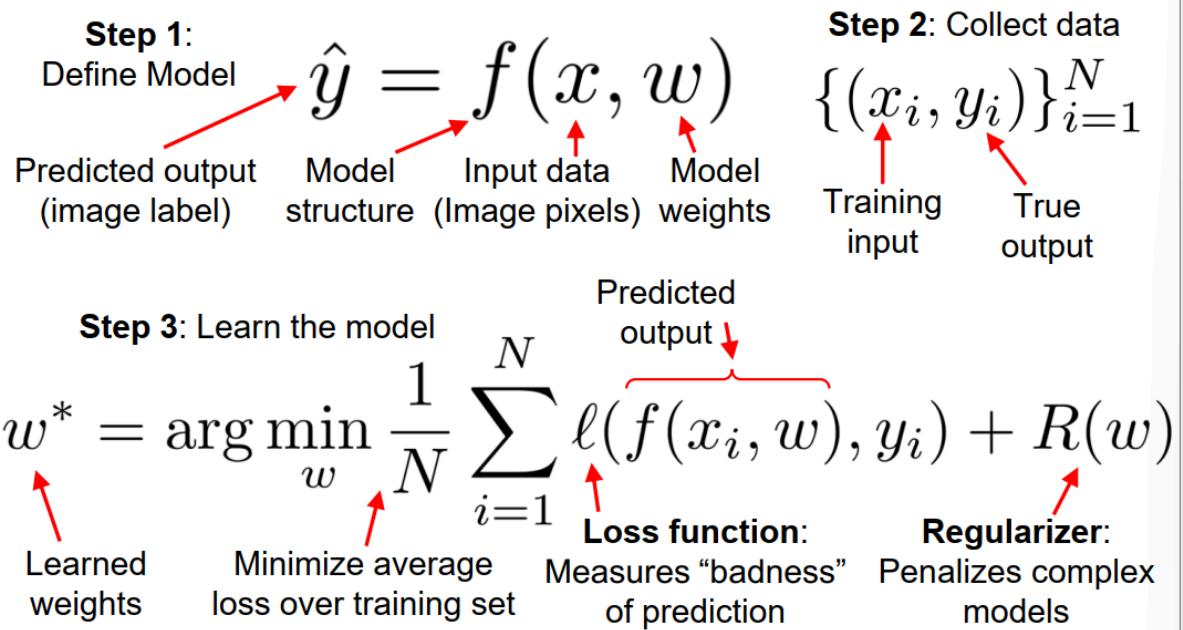
raw inputs to predictions；通过深度神经网络，直接从原始数据学习到最终结果，无需人工设计特征或中间步骤。

支持模型直接从输入数据学习到所需的输出结果，而不需要人为地将任务分割成多个独立的子任务或模块

- 神经网络的学习/训练，数学上本质是求解神经网络的什么？
求解神经网络的参数，使得神经网络能够逼近或拟合给定的目标函数。
- 会写出基于梯度下降法的学习框架
 - 定义模型，前向传播
 - 计算损失函数
 - 反向传播
 - 参数更新，使用优化算法（如梯度下降、Adam等）
 - 迭代训练

Supervised Learning in 3 easy steps

How to learn models from data



BP

反向传播Backpropagation: 反向传播用于计算损失函数对网络参数的梯度，即损失函数对权重矩阵 W 和偏置向量 b 的偏导数。这个过程基于链式法则，从输出层开始逐层向输入层传播梯度。

- 本质：复合求导
 - 关键：计算图的理解和使用
 - 节点：运算符
 - 连线上方：前向计算值
 - 连线下方：反向梯度值
 - 常用节点：加法/乘法/最大值节点
 - 作用：它的学习规则是使用梯度下降法，通过反向传播来不断调整网络的权值和阈值，使网络的误差平方和最小。
 - "梯度下降法"与BP算法的关系
 - 梯度下降法是一种优化算法，其基本原理是沿着函数梯度的反方向进行搜索，以寻找最小值
 - BP算法，即反向传播算法，是一种与最优化方法（如梯度下降法）结合使用的，用来训练人工神经网络的常见方法；
- BP算法中需要用到梯度下降法，用来配合反向传播，BP算法就是提供了给梯度下降法所需要的所有值。梯度下降法是求局部最好的w (权重)

Forward pass: Run graph “forward” to compute loss

Backward pass: Run graph “backward” to compute gradients with respect to loss

Easily compute gradients for big, complex models!

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

sigmoid function

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1 + e^{-x})^2} = \left(\frac{1 + e^{-x} - 1}{1 + e^{-x}} \right) \left(\frac{1}{1 + e^{-x}} \right) = (1 - \sigma(x))\sigma(x)$$

计算图如下：横线上写前向值，下面写后向值

Backpropagation: a simple example

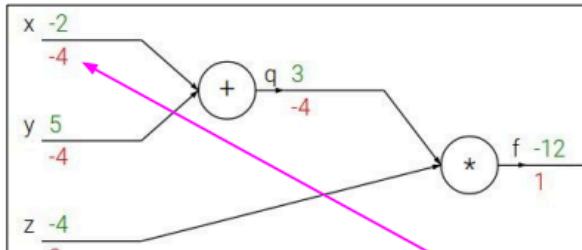
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2, y = 5, z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Chain rule:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}$$

$$\frac{\partial f}{\partial x}$$

CNN

CNN=卷积层+池化层+全连接层

与全连接网络相比，CNN在哪几个方面做了重要改变？为什么这么改？

- 局部连接：加了卷积层，参数减少
- 共享权重：同一个卷积核在输入图像的不同位置上共享相同的权重。这种权重共享机制进一步减少了参数的数量，并使得网络能够更好地捕捉到图像的平移不变性
- 有了池化层：通过下采样操作，减少特征图的尺寸，并保留最重要的特征。池化层有助于减少计算量和参数数量，并提高网络的平移不变性

卷积为什么有用？——Allow us to find interesting insights/features from images! 用于提取图像特征

Convolution = image-> Features

减小模型参数的技巧

参数计算

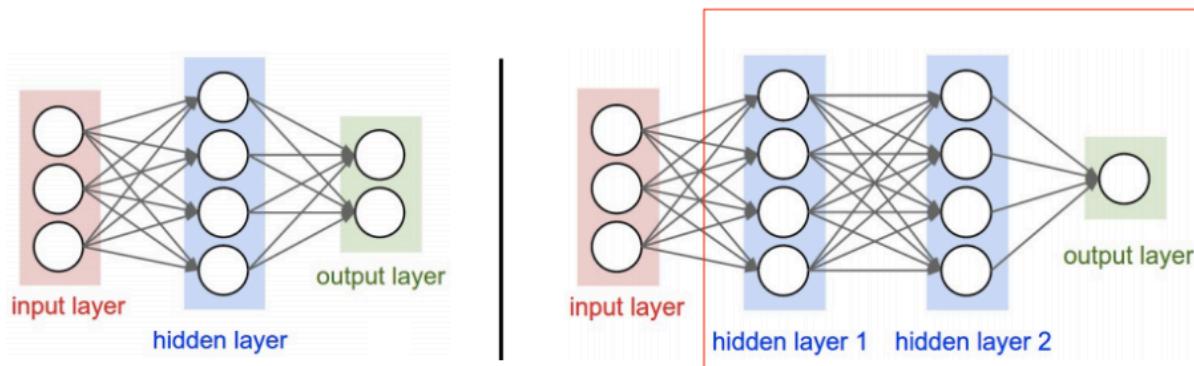
- 输出图像大小计算

记输出为 $M \times M$ 的 feature map, 输入为 $N \times N$, 卷积核为 $K \times K$, 步长为 S , padding 为 P , 则有
 $M = (N - K + 2P)/S + 1$

- 权重个数和神经元数目计算:

- 每一次 $K \times K$ 区域内的卷积都对应一个神经元, 因此每个神经元有 K^2 个权重, 1 个偏置——
多通道时权重要乘上通道数
 - 总神经元数为 M (输出图像大小), 连接数为 $(K^2 + 1) \times M$ (此处未考虑多通道) ——一般使用多个卷积核, 则神经元个数乘上卷积核数
- 更新的参数个数: CNN 中一个卷积核下对应的所有神经元 **共享参数**, 因此需要更新的参数个数为 $(K^2 + 1) \times \text{通道数} \times \text{卷积核数}$

上面算 feature_map 的公式不对



上述是如何计算NN的Neurons Weights Parameters

下面介绍CNN的计算

1. Local connectivity

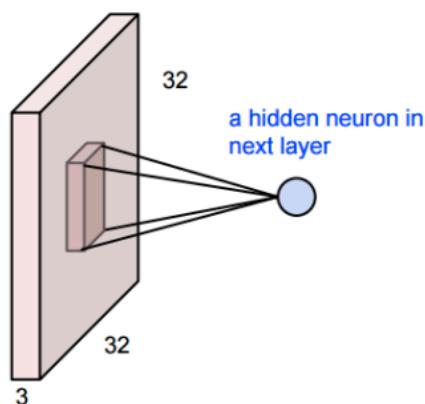


image: $32 \times 32 \times 3$ volume
before: full connectivity: $32 \times 32 \times 3$ weights

now: one neuron will connect to, e.g. $5 \times 5 \times 3$ chunk and only have $5 \times 5 \times 3$ weights.

note that connectivity is:

- local in space (5×5 inside 32×32)
- but full in depth (all 3 depth channels)

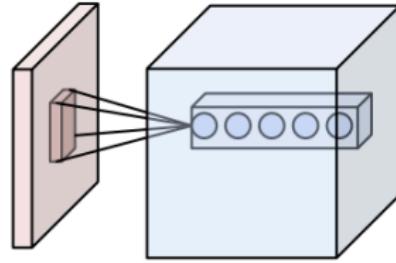


一个neuron的weights=卷积核的大小*输入图像的channel

note that connectivity is:

- local in space (5×5 inside 32×32)
- but full in depth (all 3 depth channels)

Examples time:



Input volume: **32x32x3**

Receptive fields: **5x5, stride 1**

Number of neurons: **5**

Output volume: $(32 - 5) / 1 + 1 = 28$, so: **28x28x5**

How many weights for each of the 28x28x5 neurons? **5x5x3 = 75**

output volume向下取整

池化层：在连续的卷积层之间会周期性地插入一个pooling层。它的作用是逐渐降低数据体的空间尺寸，这样的话就能减少网络中参数的数量，使得计算资源耗费变少，也能有效控制过拟合。

- 原理：同样是用一定大小的卷积核对输入图像进行卷积
- 与卷积层区别
 1. 参数不可更新
 2. stride一般较大
- 池化方式
 1. 单幅图像中池化——最大池化/平均池化
 2. 多幅特征图间池化——只使用最大池化

分区 计算机视觉 的第 11 页

-
- 作用
 1. 单图池化：减少参数数量 & 获得更大的感受野 & 使特征对微小变换更鲁棒 & 图像平滑作用
 2. 多图间池化：减少参数数量 & 找到多图间最显著的特征

全连接层，softmax回归

Softmax函数将前一层的输出（通常称为逻辑值或logits）转换为概率分布。每个逻辑值代表模型对应类别的原始预测数值，而Softmax函数的作用是将这些原始预测数值“压缩”成为一个真实的概率分布

Softmax层常常与交叉熵损失函数一起结合使用。交叉熵损失函数能够衡量预测的概率分布与真实标签之间的差异，从而指导模型的训练过程

System so far...

Feature extractor:

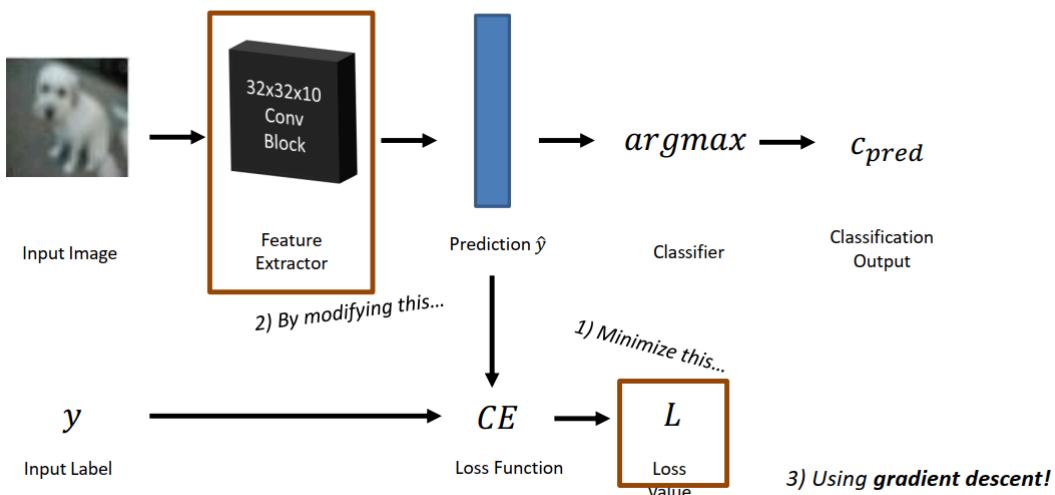
Classifier:

$$\hat{y} = SM(Wx)$$

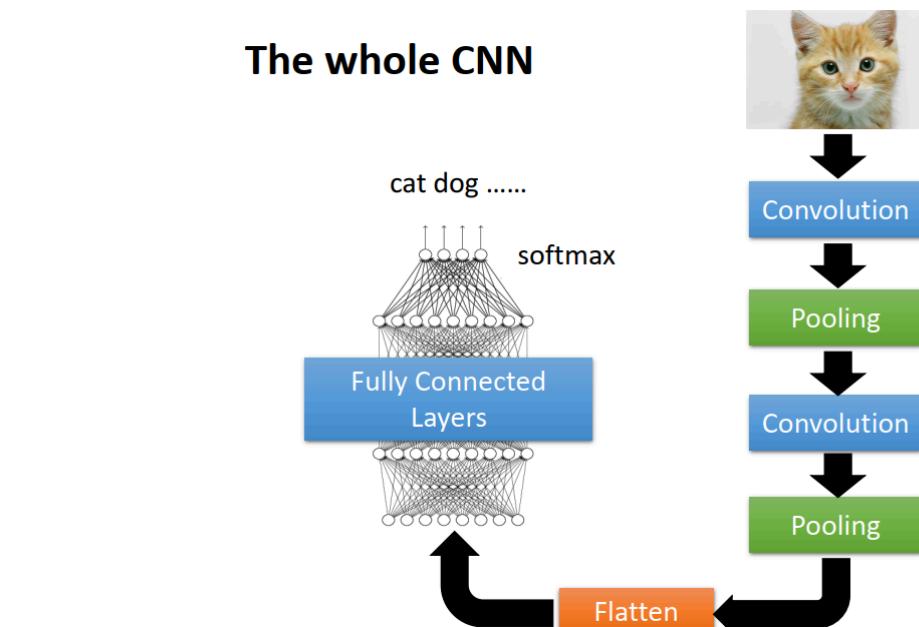
$$c_{pred} = \arg \max(\hat{y})$$

- 交叉熵 cross-entropy loss

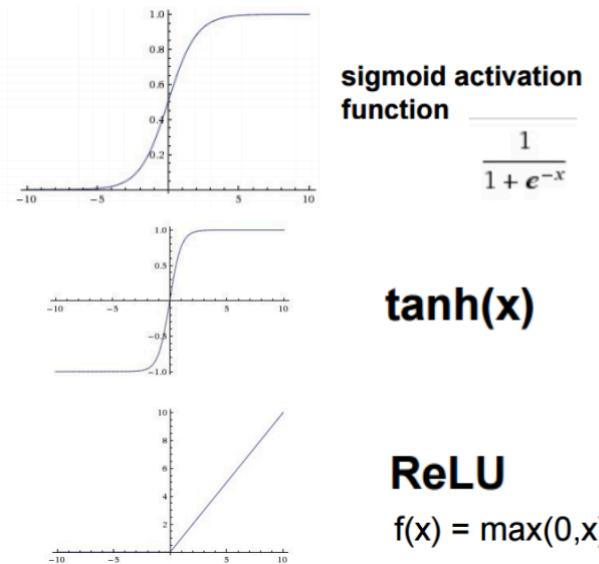
Our Classification System (modified)



The whole CNN



激活函数



Tips for training

Batch

- **小批量**: 更新频率高，梯度更新较为“嘈杂”，但有助于避免陷入局部最优，通常在小批量下模型的泛化能力更好。
- **大批量**: 更新频率低，梯度更新较为稳定，但可能导致模型陷入局部最优，泛化能力较差。

batch技巧是指什么？怎么理解该方法？

- - **Batch Size**是指在每次迭代中，模型同时处理的样本数量。
 - 它决定了模型在每次参数更新时所依据的数据量。
 - batch较大:
 - 可以加快训练速度，因为每次迭代处理更多的数据，减少了总的迭代次数3。
 - 可以减少模型训练过程中的随机性，使模型更稳定3。
 - 有助于Batch Normalization等技术更好地发挥作用
 - 需要更多的内存和计算资源
 - 可能陷入局部最优解
 - batch较小
 - 可以提高模型的泛化能力，因为每次迭代的梯度估计更准确3。
 - 有助于模型跳出局部最优解，找到更好的全局最优解
 - 训练速度较慢，因为需要更多的迭代次数
 - 可能会导致模型在训练过程中震荡较大，难以收敛

Batch Normalization

- **批量归一化的作用**: 通过对每个批量的数据进行归一化，使得每个特征的均值为0，方差为1，从而加速训练过程并提高模型的稳定性。

In general, feature normalization makes gradient descent converge faster

- **内部协变量偏移 (Internal Covariate Shift)**：批量归一化可以减少网络层之间的输入分布变化，使得训练过程更加稳定。
 - 在深度神经网络中，每一层的输入分布会随着前一层参数的变化而发生变化。这种输入分布的变化称为 **内部协变量偏移**。
 - 例如，假设网络的某一层在前向传播时，其输入分布发生了显著变化，这会导致后续层的训练变得困难。
 - 内部协变量偏移的影响
 - **训练不稳定**：输入分布的变化会导致梯度更新不稳定，训练过程可能变得非常缓慢。
 - **学习率限制**：为了避免训练不稳定，通常需要使用较小的学习率，但这会减慢收敛速度。
 - **梯度消失/爆炸**：输入分布的变化可能导致梯度消失或梯度爆炸问题，尤其是在深层网络中。
- **测试阶段的批量归一化**：在测试阶段，由于没有批量数据，通常使用训练阶段计算的移动平均值来进行归一化。
- **其他归一化方法**：文档还提到了其他归一化方法，如层归一化 (Layer Normalization)、实例归一化 (Instance Normalization) 等。

batch normalization的初衷是为了改变优化过程中的什么？

- batch normalization：初衷是为了解决深度神经网络训练过程中的 **内部协变量偏移 (Internal Covariate Shift)** 问题，从而加速训练并提高模型的稳定性和性能。
- 它通过调整神经网络中间层的输入分布，使得输入数据保持相对稳定的均值和方差，从而加速模型的训练收敛并提高模型的泛化能力

解决内部协变量偏移 (Internal Covariate Shift) 问题

Momentum

- **动量的作用**：动量是一种优化技术，通过在梯度下降中加入前几次更新的加权和，使得参数更新不仅依赖于当前的梯度，还依赖于之前的更新方向。这有助于**加速收敛并减少震荡并且跳出局部最优**。通过计算梯度的**指数加权平均**来更新参数
- **更新公式**

$$m_t = \gamma m_{t-1} + \eta \nabla L(\theta_t)$$

$$\theta_{t+1} = \theta_t - m_t$$

其中， m_t 是动量， γ 是动量系数， η 是学习率， $\nabla L(\theta_t)$ 是当前梯度。

- 有可能避免陷入局部最小值或鞍点
- 通过平滑梯度更新方向，减少震荡现象，使得参数更新更加稳定
- 在一定程度上缓解了对学习率的敏感性，使得在较大的学习率下也能实现稳定的收敛
- **优化失败的原因**
 - **局部最小值 (Local Minima)**：梯度下降可能会陷入局部最小值，导致无法继续优化。
 - **鞍点 (Saddle Point)**：在高维空间中，鞍点比局部最小值更常见，梯度在这些点附近接近于零，导致优化停滞。
 - **梯度消失 (Vanishing Gradient)**：当梯度接近于零时，参数更新会变得非常缓慢，导致训练停滞。

- 其他优化技巧

- 并行计算：在大批量训练中，可以通过并行计算来加速梯度计算。
- 大批量训练的挑战：尽管大批量训练可以加速训练过程，但可能会导致模型泛化能力下降。文档提到了一些研究，探讨如何在大批量训练中保持良好的泛化性能。

Self-attention Block

考虑上下文语义关系

- Self-attention机制主要是对什么样信息进行建模？

序列内部元素之间的依赖关系

- 理解self-attention机制中的q/k/v想代表的含义/意思？

查询 (Query) 键 (Key) 和值 (Value)

◦ q

▪ 含义：代表我们想要理解或关注的元素的向量表示。

▪ 作用：用于查询序列中其他元素与当前元素的相似度。

◦ k

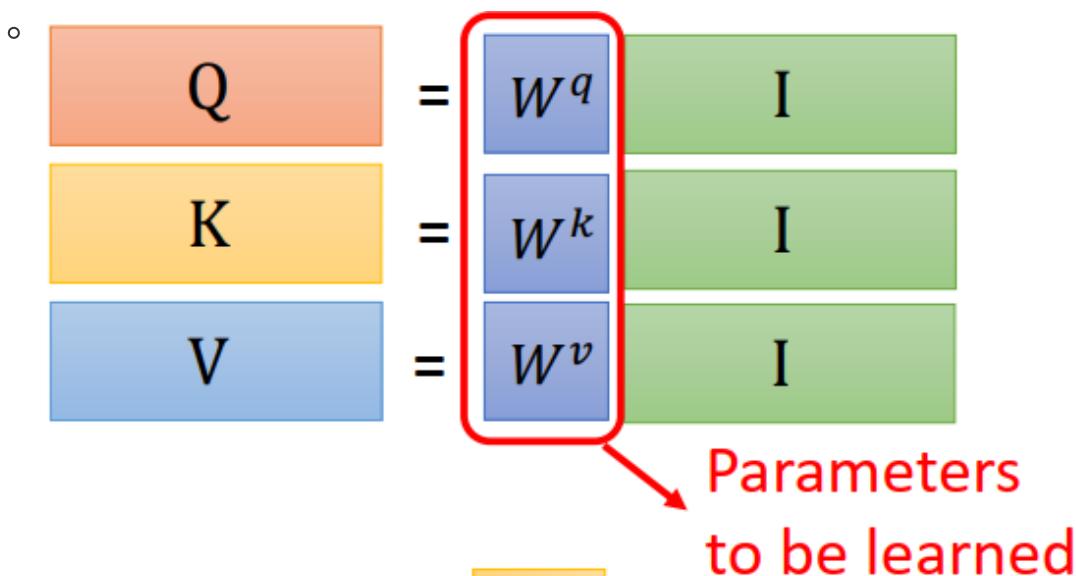
▪ 含义：代表序列中每个元素的向量表示，用于与Query进行相似度匹配。

▪ 作用：通过计算Query与所有Key之间的点积，得到一个权重分布，表示每个元素与当前关注元素的相关性。

◦ v

▪ 含义：代表序列中每个元素携带的实际信息的向量表示。

▪ 作用：根据Query与Key计算得到的权重分布，对Value进行加权求和，得到一个包含丰富上下文信息的新元素表示。



- 为什么要加位置编码 (positional encoding)

处理词元序列时，循环神经网络是逐个的重复地处理词元的，而自注意力则因为并行计算而放弃了顺序操作。为了使用序列的顺序信息，通过在输入表示中添加 位置编码 (positional encoding) 来注入绝对的或相对的位置信息——固定位置/学习位置

- Self-attention机制与CNN卷积机制的关系? 【定性理解】

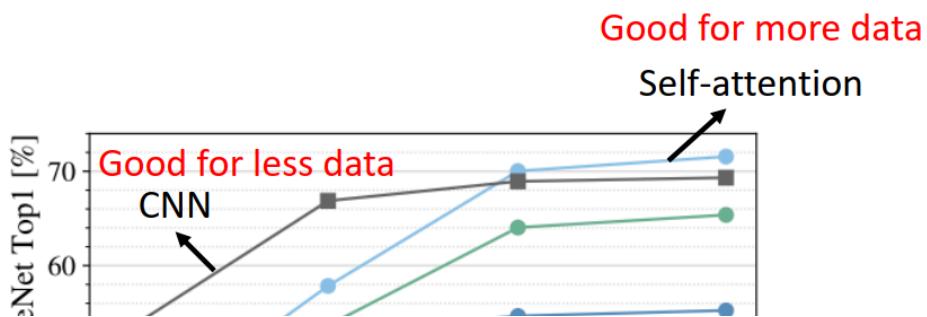
CNN: self-attention that can only attends in a receptive field

➤ CNN is simplified self-attention.

Self-attention: CNN with learnable receptive field

➤ Self-attention is the complex version of CNN.

Self-attention v.s. CNN

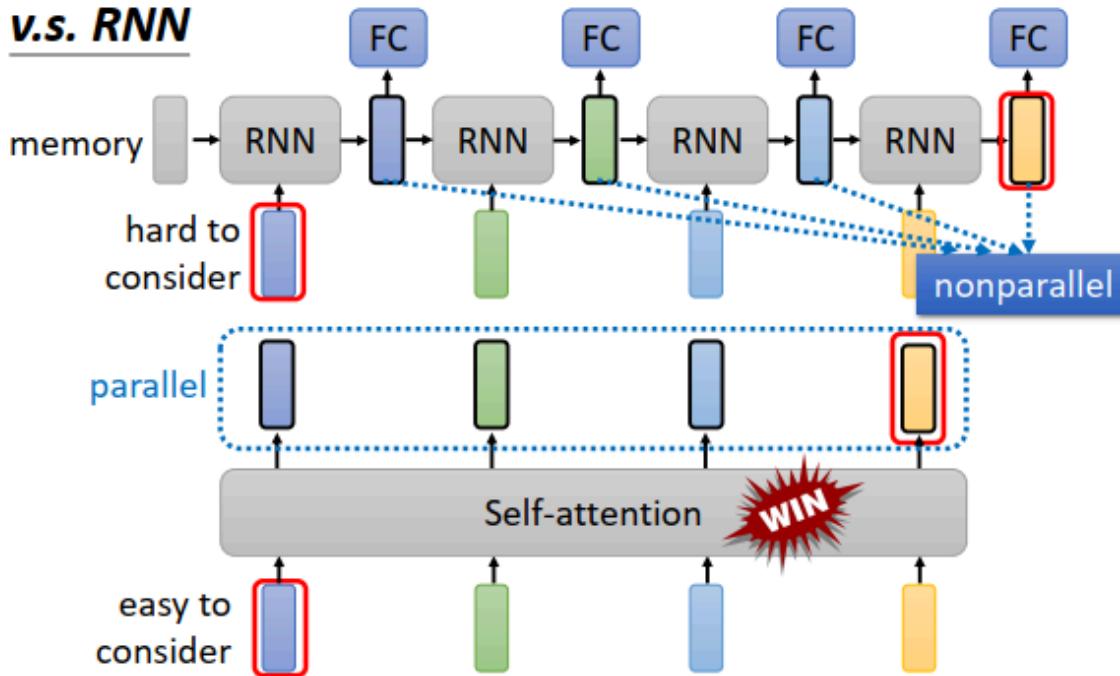


- Self-attention机制与循环神经网络模型（RNN）的关系? 【定性理解】

RNN: [史上最详细循环神经网络讲解 \(RNN/LSTM/GRU\)](#)

Self-attention

v.s. RNN



Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention

RNN没有办法并行化产生输出，只能一个接一个输出；

Self-attention可以并行化输出

RNN当结果要考虑比相对较远输入的位置时，比较难以考虑到；

Self-attention可以很好的对输入位置比较远的向量进行考虑

RNN每个输出只考虑了其左边的输入，没有考虑右边的输入

Self-attention则考虑了整个Sequence

- Self-attention机制与图神经网络模型（GNN）的关系？【定性理解】

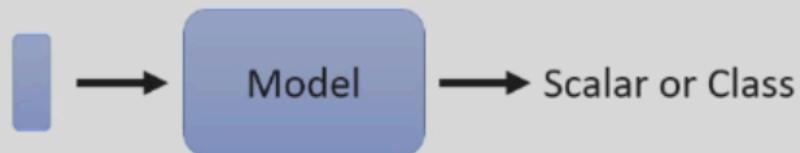
Self-attention for Graph(Consider edge: only attention to connected nodes) ——one type of Graph Neural Network (GNN)

图注意力网络（GAT）将 注意力机制 引入图神经网络领域，使图中的每个节点能够根据其邻居节点的重要性，动态地聚合邻居节点的信息

- input

Sophisticated Input

- Input is a **vector**



- Input is a **set of vectors**



输入视频/语言

Vector Set as Input

this is a cat

```
graph LR; A[this] --- B[is] --- C[a] --- D[cat]
```

One-hot Encoding

```
apple = [ 1  0  0  0  0 ..... ]  
bag   = [ 0  1  0  0  0 ..... ]  
cat   = [ 0  0  1  0  0 ..... ]  
dog   = [ 0  0  0  1  0 ..... ]  
elephant = [ 0  0  0  0  1 ..... ]
```

Word Embedding



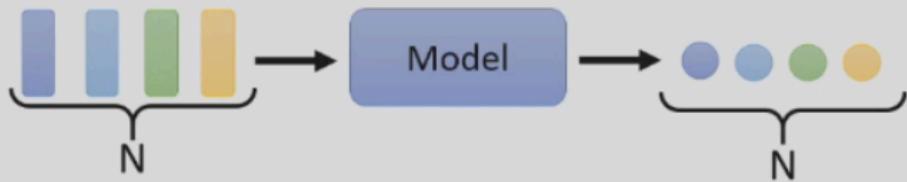
左边这种没有包含什么语义信息

Graph

- output

What is the output?

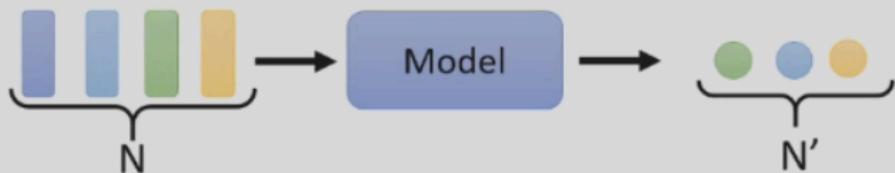
- Each vector has a label.



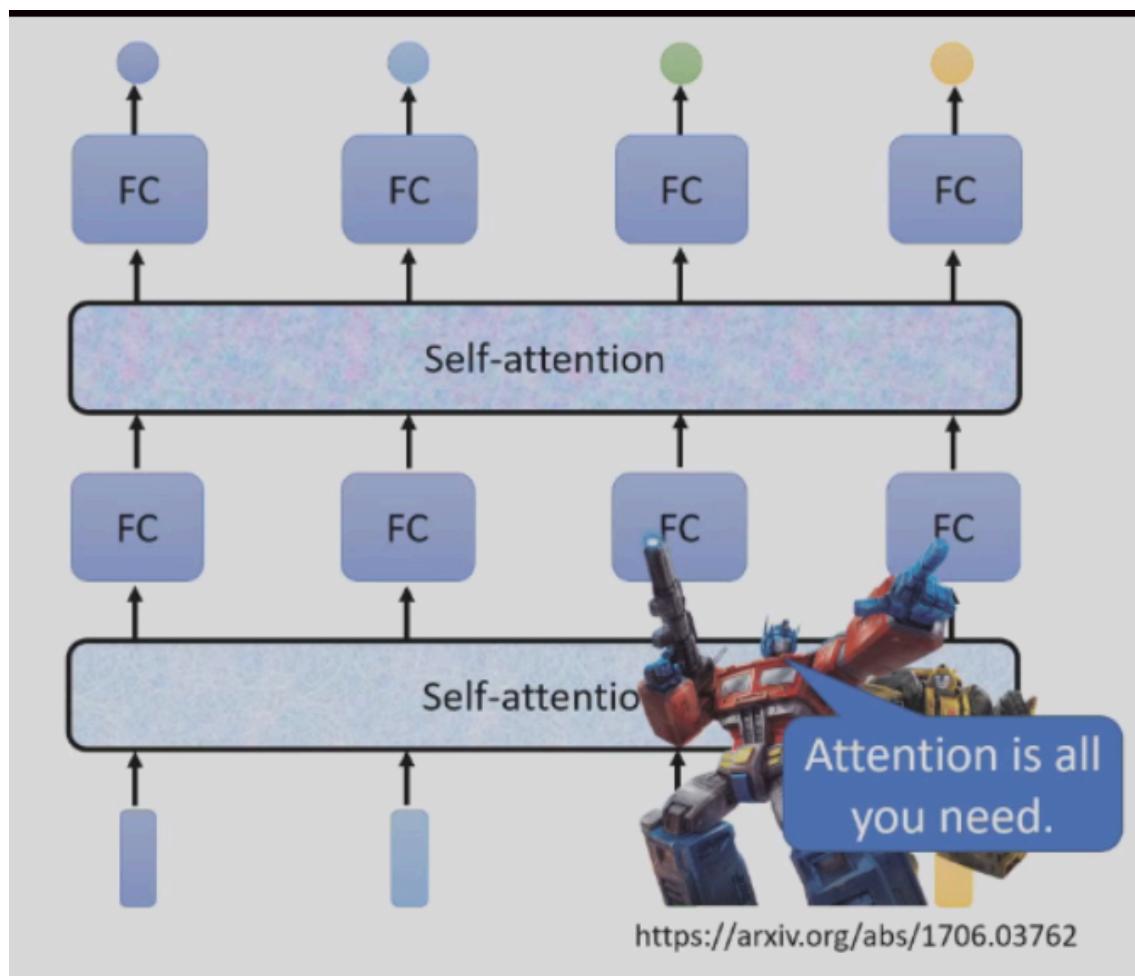
- The whole sequence has a label.



- Model decides the number of labels itself. seq2seq



sequence labeling

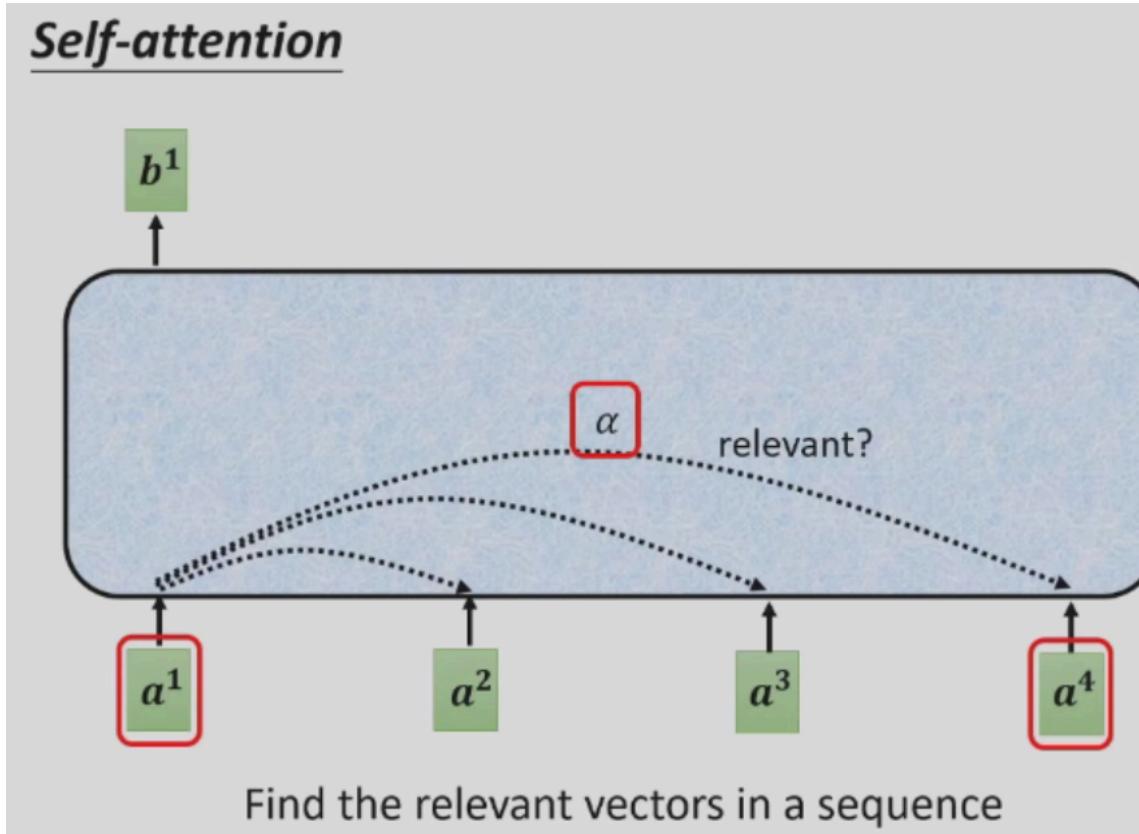


加入self-attention得到新的feature，其中包含自己和自己与前后的关系

- self-attention

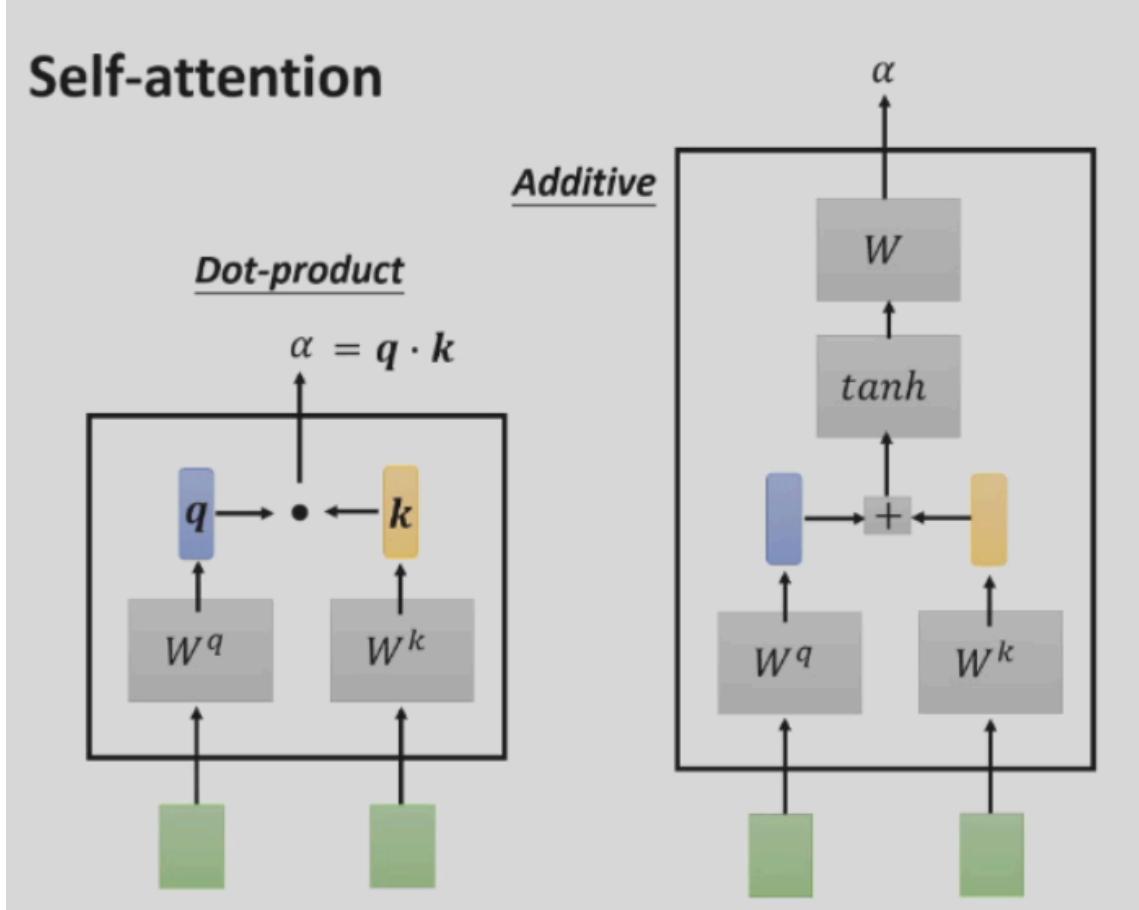
大的架构如下：

Self-attention



中间的算法可以随便替换

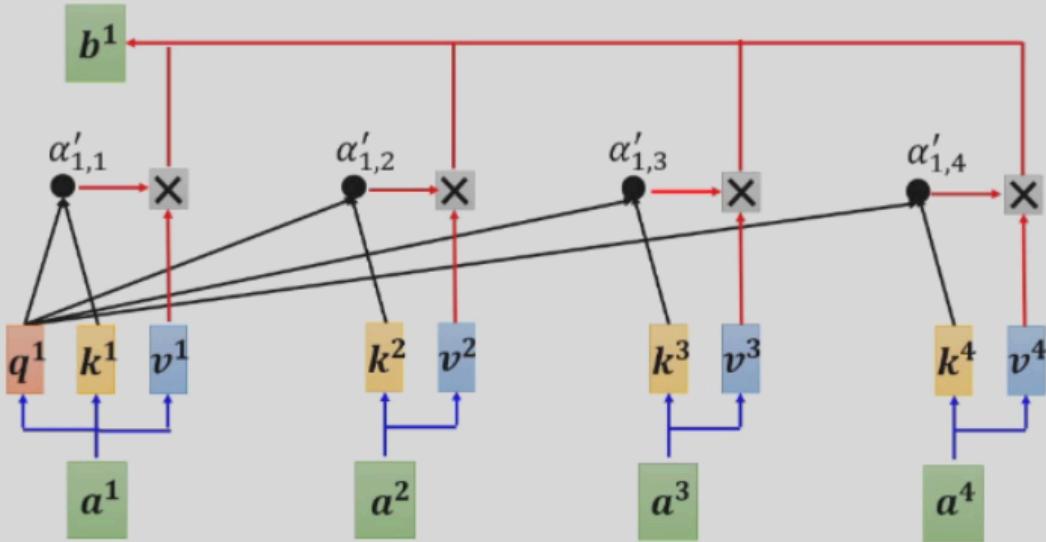
Self-attention



像这样提取不同的信息

Self-attention

Extract information based on
attention scores

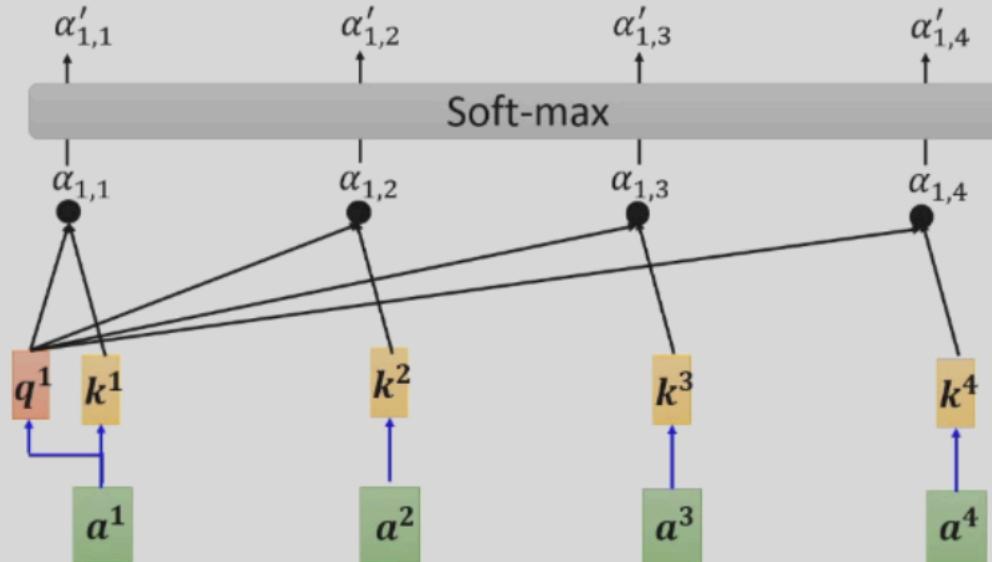


$$v^1 = W^v a^1 \quad v^2 = W^v a^2 \quad v^3 = W^v a^3 \quad v^4 = W^v a^4$$

如何考虑关联性呢？：每个a中都包含三个空间：自己与别人的关联性query/被别人比较的量key/自己的value

其中 $\alpha'_{i,j}$ 由如下获得：

Self-attention $\alpha'_{1,i} = \exp(\alpha_{1,i}) / \sum_j \exp(\alpha_{1,j})$



$$q^1 = W^q a^1 \quad k^2 = W^k a^2 \quad k^3 = W^k a^3 \quad k^4 = W^k a^4$$

$$k^1 = W^k a^1$$

Camera Calibration

Camera Calibration (single-view calibration)

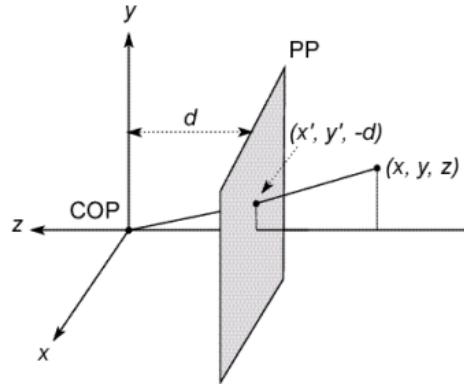
[7.1 相机校准 - OpenCV中文官方文档 \(woshicver.com\)](#)

[相机畸变产生原因与公式表示 \(基本原理\)](#)

Camera model

- 相机模型：小孔成像

Modeling projection



Projection equations

- Compute intersection with PP of ray from (x, y, z) to COP
- Derived using similar triangles (on board)
 $(x, y, z) \rightarrow (-d\frac{x}{z}, -d\frac{y}{z}, -d)$
- We get the projection by throwing out the last coordinate:
 $(x, y, z) \rightarrow (-d\frac{x}{z}, -d\frac{y}{z})$ (Fundamental Equations)

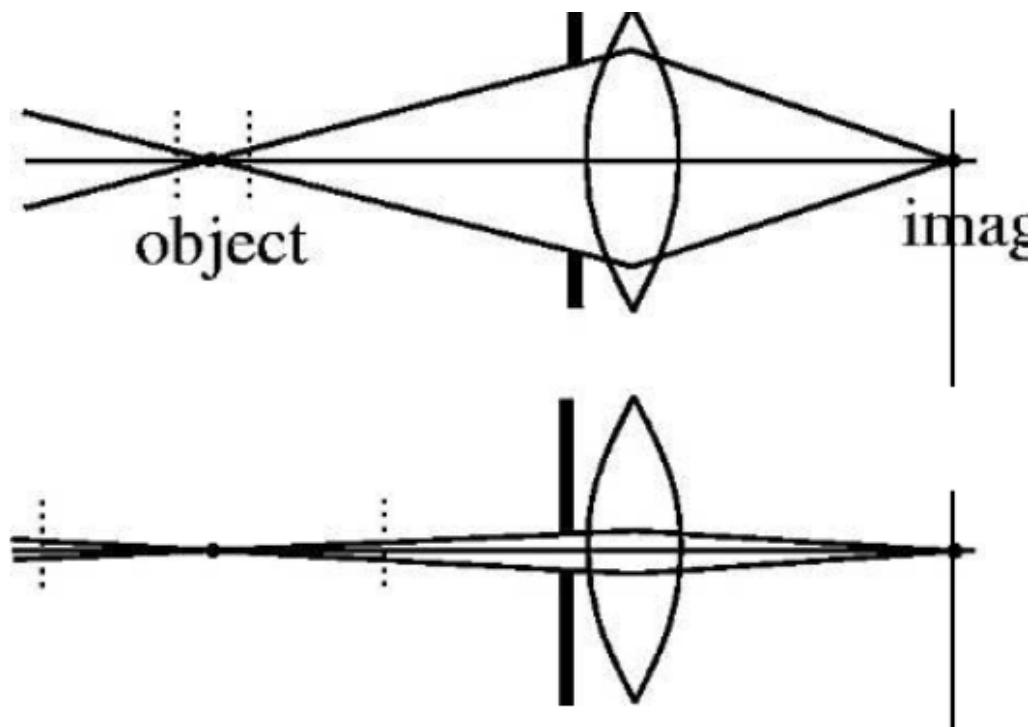
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ -z/d \end{bmatrix} \Rightarrow \left(-d\frac{x}{z}, -d\frac{y}{z}\right)$$

$$\begin{bmatrix} -d & 0 & 0 & 0 \\ 0 & -d & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} -dx \\ -dy \\ z \end{bmatrix} \Rightarrow \left(-d\frac{x}{z}, -d\frac{y}{z}\right)$$

not a linear transformation——所以我们改用齐次坐标

- 基本概念

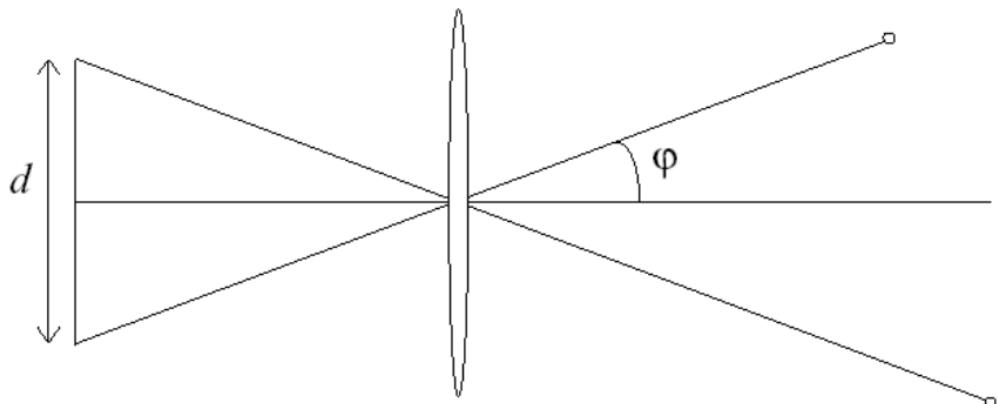
- 景深Depth of Field: 相机镜头能够取得清晰图像的成像所测定的被摄物体前后范围距离
- 光圈(aperture孔径): 镜头中用于控制光线透过镜头并进入机身内感光面光量的装置



大光圈景深小，小光圈景深大光路图里把上下两条线放近一点

small aperture reduces amount of light – need to increase exposure

- 焦距：从镜片中心到底片等成像平面的距离
- 视场(Field of View FOV)：镜头能够观察到的最大范围



$$\varphi = \arctan \frac{d}{2f}$$

(是视角的二分之一)

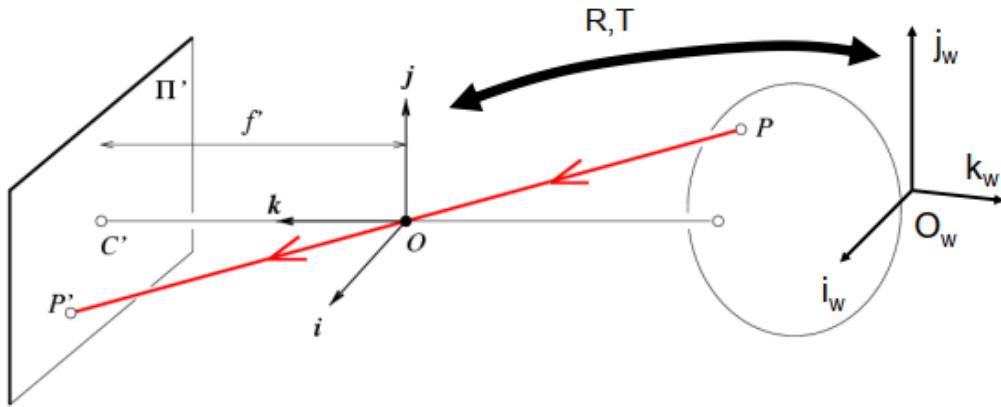
大焦距离得近：整个场景被缩短，远处的东西被拉到近处而且很大，但是虚化了，焦距内的物体也能看到

小焦距离得远：整个场景被拉长，远处的东西很小，但是都很清楚，焦距内的东西会在视野外

总结：焦距越大，视场越小

- Lens Flaws : Dispersion 色散
- 投影变换：

Projection



$$\mathbf{x} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}] \mathbf{X}$$

x: Image coordinates: $(u, v, 1)$

K: Intrinsic matrix (3x3)

R: Rotation (3x3)

t: Translation (3x1)

X: World coordinates: $(X, Y, Z, 1)$

- 不保角: 投影变换会改变角度。
- 不保距: 投影变换会改变距离。
- 不保平行: 投影变换会改变平行关系。
- 保共线: 投影变换保持共线性。
- 齐次坐标Homogeneous coordinates

欧式几何是投影几何的一个子集。齐次坐标是用N+1个数来表示N维坐标的一种方式。

- 齐次坐标与笛卡尔坐标之间的转换

$$(x, y, w) \Leftrightarrow \left(\frac{x}{w}, \frac{y}{w} \right)$$

Homogeneous \Leftrightarrow Cartesian(x, y, w)

刚体变换 $Rx + t$ 可以表示为齐次坐标下的矩阵乘法形式：

$$\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

- 好处
 - 统一表示
 - 点: 在二维空间中, 点 (x, y) 的齐次坐标为 $(x, y, 1)$ 。
 - 向量: 向量 (x, y) 的齐次坐标为 $(x, y, 0)$ 。
 - 矩阵计算
 - 无穷远点的处理
- 齐次坐标 $(x, y, 0)$ 表示无穷远点, 这在透视投影中用于表示平行线的交点。

内参矩阵

1. 相机坐标系→成像坐标系(3D TO 2D)

$$\begin{cases} x_{screen} = f \frac{X}{Z} \\ y_{screen} = f \frac{Y}{Z} \end{cases} \Rightarrow Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

其中f为焦距

2. 成像坐标系→像素坐标系(2D TO 2D): 考虑度量单位 (mm→pixel) 的不同, 以及两坐标系原点不同

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s_u & 0 & c_u \\ 0 & s_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \text{ 其中 } u, v \text{ 为像素坐标系坐标, } x, y \text{ 为成像坐标系坐标, } s_u, s_v \text{ 为表示度量单位转换的参数, } c_u, c_v \text{ 为成像坐标系相对于像素坐标系的偏移量}$$

$$\begin{bmatrix} U^{(new)} \\ V^{(new)} \\ S \end{bmatrix} = \begin{bmatrix} -fk_u & 0 & u_0 & 0 \\ 0 & -fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X^{(c)} \\ Y^{(c)} \\ Z^{(c)} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} U^{(new)} \\ V^{(new)} \\ S \end{bmatrix} = \begin{bmatrix} -f_u & 0 & u_0 & 0 \\ 0 & -f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X^{(c)} \\ Y^{(c)} \\ Z^{(c)} \\ 1 \end{bmatrix} \quad \text{Equation 3}$$

$$f_u = fk_u$$

$$f_v = fk_v$$

Silven [Heikkila97]). The projection of the points in the physical world into the camera is now summarized by the following simple form:

$$q = MQ, \text{ where } q = \begin{bmatrix} x \\ y \\ w \end{bmatrix}, M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

- M contains the **intrinsic parameters**

- **Intrinsic parameters**

$$(f_x, f_y, c_x, c_y)$$

1. 径向畸变

- 原因：由于透镜的几何形状不完美或安装位置引起的畸变
- 分类：枕形畸变/桶形畸变
- 校正模型：
$$\begin{cases} x_{corrected} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \\ y_{corrected} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \end{cases}$$

2. 切向畸变

- 原因：透镜平面和成像平面不平行引起的畸变
- 校正模型：
$$\begin{cases} x_{corrected} = x + [2p_1y + p_2(r^2 + 2x^2)] \\ y_{corrected} = y[2p_2x + p_1(r^2 + 2y^2)] \end{cases}$$

3. 畸变参数：(k_1, k_2, k_3, p_1, p_2)

$$x_{distorted} = x + [2p_1xy + p_2(r^2 + 2x^2)]$$

$$y_{distorted} = y + [p_1(r^2 + 2y^2) + 2p_2xy]$$

外参模型

外参模型：世界坐标系→相机坐标系(3D TO 3D)

分区 计算机视觉 的第 8 页

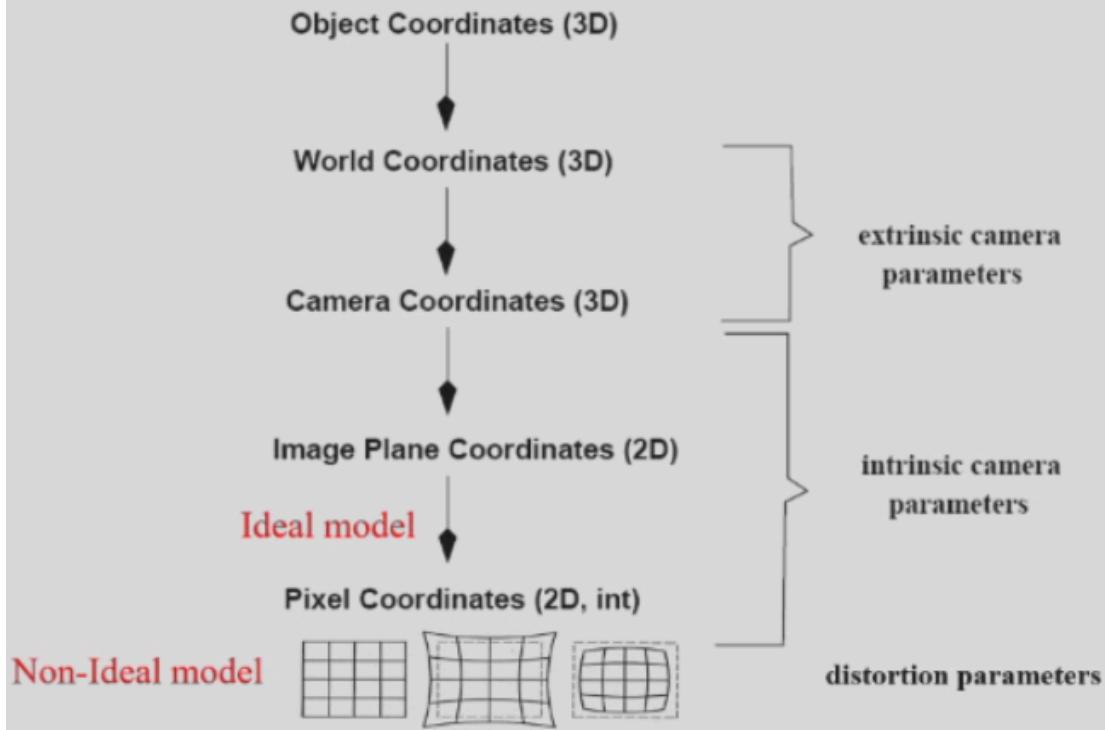
外参矩阵：
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} X_{world} \\ Y_{world} \\ Z_{world} \end{bmatrix}$$
 ⇒ 外参：($\theta, \varphi, \psi, t_x, t_y, t_z$) 分别表示世界坐标系相对相机坐标系的旋转、平移量

View Transformation

- Transformation between camera and object
- Called **extrinsic parameters**
 $(\theta, \varphi, \psi, t_x, t_y, t_z)$
- Called **intrinsic parameters**
 (f_x, f_y, c_x, c_y)
- Called **distortion parameters**
 $(k_1, k_2, p_1, p_2, k_3)$

distortion失真

Camera parameters

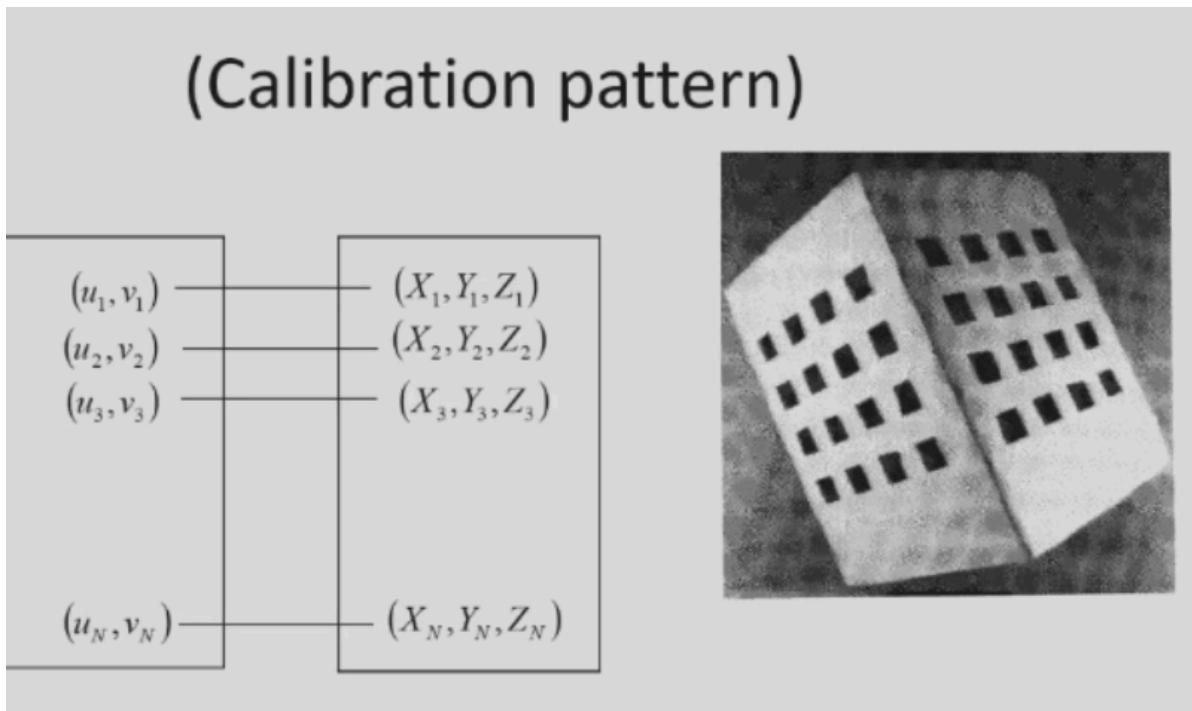


Camera Calibration

- what : Compute relation between pixels and rays in space
- why
- how

基本过程简述

1. 获取标定物体网格的角点在坐标系的位置
2. 找到图片的角点
3. 根据图像空间坐标系到世界坐标系列出等式
4. 求解相机参数



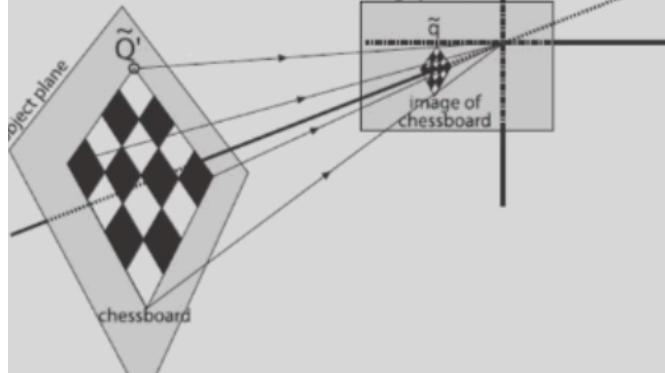
homography; chess定标

- Homogeneous matrix: $H_{33}=1$
 - Degree of freedom for H is **8**
- At least **4** points needed.
 - More is helpful.

Calibration by Homography

Plane → Plane

[Zhang99, Zhang00]



$$\tilde{Q} = [X \ Y \ Z \ 1]^t$$

$$\tilde{q} = [x \ y \ 1]^t$$

拍多个照片；测角点；建立等式

How Many Chess Corners (*pattern_size*) for Calibration?

- How many parameters we have?
 - 6 extrinsic parameters $(\theta, \varphi, \psi, t_x, t_y, t_z)$
 - 4 intrinsic parameters (f_x, f_y, c_x, c_y)
 - **5 distortion parameters** $(k_1, k_2, p_1, p_2, k_3)$
- 2D geometry -- 5
 - **3 points yield 6 constraints** (in principle)
 - Enough for 5 parameters!
 - More for robustness

外参会变

- **Each view:**
 - Gives **8** equations, because a square can be described by **4** points.
 - **Solve:** **6** individual parameters (extrinsic) : R, t
 - **Solve:** Common parameters (intrinsic):
 - intrinsic matrix (**4** parameters)
 - distortion coefficients
 - ...
 - How about giving **K** views?? **6K+4**

每个view=6个外参参数+4个共同的内参参数

N个点K个视角

$$2NK > 6k + M \quad (M = 4 \text{ or } 9)$$

一个点产生两个方程

每个匹配点对可以产生两个方程，分别对应于点的x坐标和y坐标的比例关系。

Stereo Vision

获得深度图

triangulation三角测量；Rectification整流，整改

How to Do Stereo?

Four major steps:

– Undistortion

- remove distortions -> undistorted images

– Rectification

- adjust cameras -> the two images row-aligned

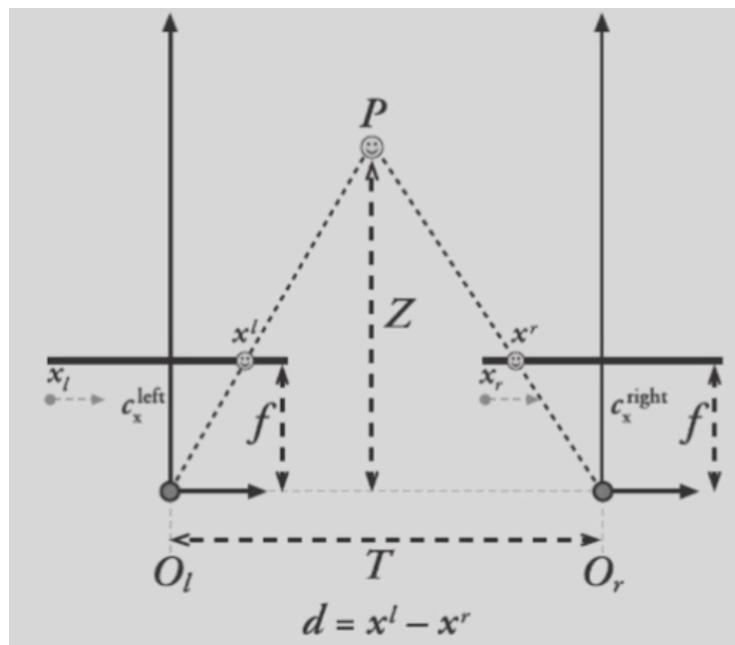
– Correspondence

- find the same features in the two images -> disparity

– Reprojection/triangulation

- triangulation -> disparity -> a depth map

disparity



$$\frac{T - (x^l - x^r)}{Z - f} = \frac{T}{Z} \quad \Rightarrow \quad Z = \frac{fT}{x^l - x^r}$$

- 基本步骤

Four major steps:

– Undistortion

- remove distortions -> undistorted images

– Rectification

- adjust cameras -> the two images row-aligned

– Correspondence

- find the same features in the two images -> disparity

– Reprojection/triangulation

- triangulation -> disparity -> a depth map

1. 去畸变：根据畸变模型消除畸变

2. 矫正

• 原因：根据对极几何，左右观测对于同一物体的投影处在同一水平线上（2D搜索 TO 1D）

• 目的：使左右观测所得图像行对齐

3. 角点匹配：在1D的直线上搜索匹配值函数的极值位置

4. 计算深度——Triangulation: $\frac{T-(x^l-x^r)}{z-f} = \frac{T}{z}$

将双目相机拍摄的左右两幅图像进行几何变换，使得两幅图像的对极线水平对齐

◦ 原始的2D匹配问题：在未校正的图像中，寻找一个图像上的点在另一个图像中的对应点，需要在二维平面上进行搜索，计算量大且容易出错。

◦ 转化为1D匹配问题：经过立体校正后，对应点的搜索范围被限制在水平方向上，即只需要在水平线上搜索对应点

如果不进行rectification，则会使得匹配复杂度高，匹配精度低，极线几何

Rectification

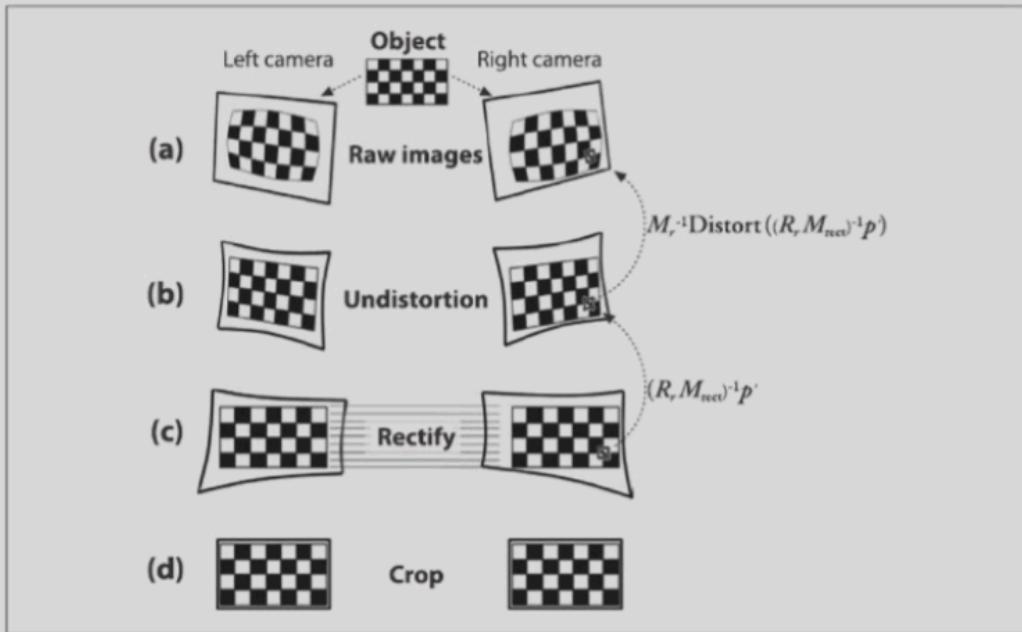


Figure 12-11. Stereo rectification: for the left and right camera, the raw image (a) is undistorted (b) and rectified (c) and finally cropped (d) to focus on overlapping areas between the two cameras; the rectification computation actually works backward from (c) to (a)

Epipolar Geometry

The basic geometry of a stereo imaging system

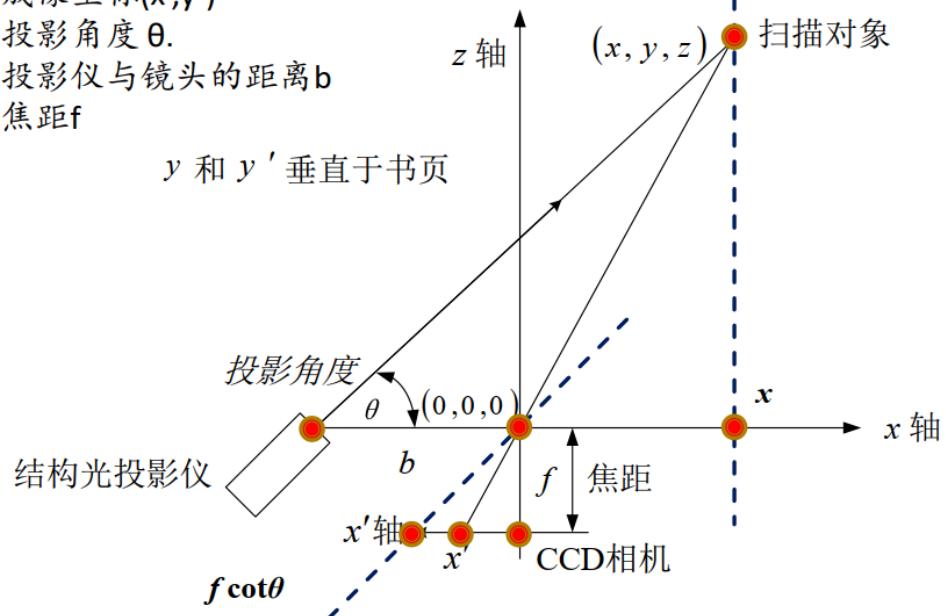
极几何（极线几何、核面几何、对极几何）——表述两个相机成像关系的几何

Structured-lighting 3D Scan

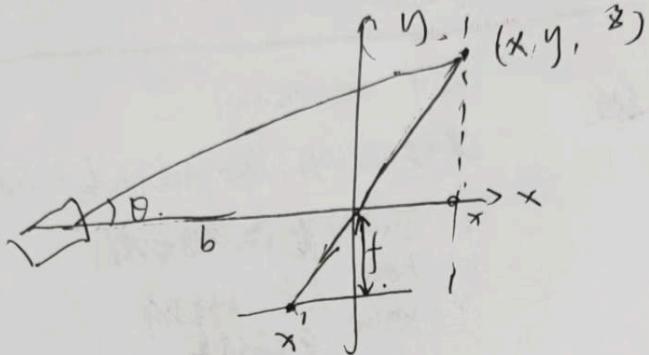
三维获取：结构光

- 结构光成像系统的构成
 结构光投影仪+CCD相机+深度信息重建系统
- 基本原理

- 观测对象坐标 (x, y, z) ?
- 成像坐标 (x', y')
- 投影角度 θ .
- 投影仪与镜头的距离 b
- 焦距 f



单流直成：仿射光束对齐 (Affine束对齐)



$$\frac{x'}{x} = \frac{f}{y} \quad y = (x+b) \cdot \tan \theta.$$

$$x = \frac{y}{f} x' = \frac{(x+b)\tan\theta}{f} x'$$

$$\Rightarrow \left(1 - \frac{\tan\theta}{f} x'\right)x = \frac{b\tan\theta}{f} x'$$

$$\therefore x = \frac{b\tan\theta}{f - \tan\theta x'} x'$$

$$= \frac{b}{f \cot\theta - x'} x'$$

2D \rightarrow 1D
只用一个面

点云，深度图，网格 (mesh)

ICP

Iterative Closest Point

- 目标：计算两组数据（两帧图像）间的旋转平移量，使之形成最佳匹配
- 常用的求解方法有奇异值分解 (SVD) 和非线性优化

INPUT: two point sets P_j and P'_j for surfaces S 、 S'

OUTPUT: $F : M_p = F(p)$

$$Cost = \sum_i d^2(Mp_i - P'_j)$$

or

$$Cost = \sum_i d^2(F(P_i) - P'_j)$$

where $d^2(Mp_i - P'_j)$

is the distance between Mp_i and P'_j

Mp_i is the point in the surface M_s

P'_j is the closest point in surface S' to Mp_i .

寻找F就变成了找到使
Cost最小的点的搜索过程，
这就是本算法称为ICP的
原因。

- 步骤：
 - 根据最近领域规则建立P和P'中点的关联——即初始化一个R和t，一般根据传感器和机器人移动参数得到
 - 两帧图像之间不能相差过大——会导致通过机器人运动学得到的R₀和t₀不准确
 - 利用线性代数/非线性优化的方式估计旋转平移量
 - 使用估计得到的旋转平移量对点集合P'的点进行旋转平移
 - 若旋转平移后的均方差小于阈值，则结束；否则迭代重复
- 线性代数求解法

分区 计算机视觉 的第 9 页

- 构建最小二乘问题 $\min_{R,t} J = \frac{1}{2} \sum_{i=1}^n \|p_i - (Rp'_i + t)\|_2^2$, 其中 $e_i = p_i - (Rp'_i + t)$ 为第i个匹配点的误差
- 旋转平移分解（去质心化）
 - 根据 $\arg\min_R \frac{1}{2} \sum_{i=1}^n \|q_i - Rq'_i\|_2^2$, 其中 $q_i = p_i - \bar{p}$, $q'_i = p'_i - \bar{p}'$, $\bar{p} = \frac{1}{n} \sum_{i=1}^n p_i$, $\bar{p}' = \frac{1}{n} \sum_{i=1}^n p'_i$
 - 根据R计算t: $t^* = \bar{p} - R^* \bar{p}'$
- 求解R：优化目标函数可变为 $\max_R \sum_{i=1}^n q_i^T R q'_i$
 - 奇异值分解：令矩阵 $W = \sum_{i=1}^n q_i q_i^T$, 对W进行SVD分解，可得 $W = U \Sigma V^T$ (Σ 为奇异值从大到小排列组成的对角阵, U,V为正交矩阵), 则 $R = VU^T$

laser scanning

3D capture system

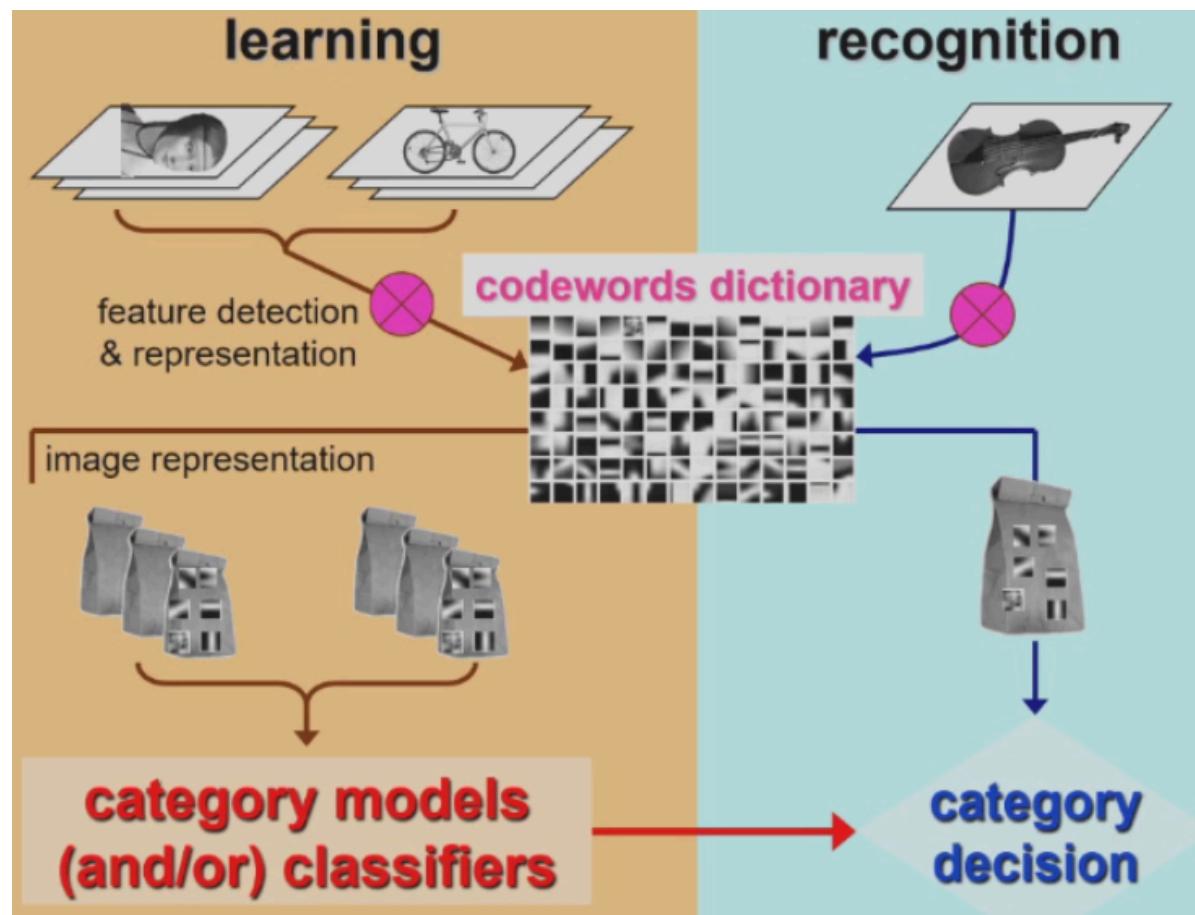
Object Categorization

BoW Bag of words

[计算机视觉中的词袋模型\(Bow,Bag-of-words\)](#)

其大概过程首先提取图像集特征的集合，然后通过聚类的方法聚出若干类，将这些类作为dictionary，即相当于words，最后每个图像统计字典中words出现的频数作为输出向量，就可以用于后续的分类、检索等操作。

以sift特征为例，假设图像集中包含人脸、自行车、吉他等，我们首先对每幅图像提取sift特征，然后使用如kmeans等聚类方法，进行聚类得到码本(dictionary)



1. Feature detection and representation

- **Regular grid**
 - Vogel & Schiele, 2003
 - Fei-Fei & Perona, 2005
- **Interest point detector**
 - Csurka, Bray, Dance & Fan, 2004
 - Fei-Fei & Perona, 2005
 - Sivic, Russell, Efros, Freeman & Zisserman, 2005
- **Other methods**
 - Random sampling (Vidal-Naquet & Ullman, 2002)
 - Segmentation based patches (Barnard, Duygulu, Forsyth, de Freitas, Blei, Jordan, 2003)

期末

期末考占50分，闭卷考试，题型为简答+公式推导+计算，可能因为CV东西太多太难，索性就当文科考了。考点都在复习提纲里，不需要看提纲之外的内容。复习的时候查到过以前同学的笔记，内容也都变化不大，所以就把提纲放上来了

整理了一份复习笔记（参考了网上一些同学的博客），只要背会，期末考试绝对没问题

- [CC98论坛](#)
- [2021-2022 冬 计算机视觉 回忆 - CC98论坛](