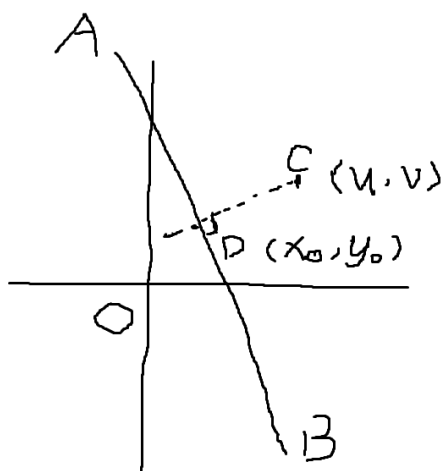


问题包-3

1. 证明: 点(u, v)到一条线(a, b, c)的距离为: $|au + bv + c|$, 这里 $a^2 + b^2 = 1$

这条直线AB的参数方程为: $ax + by + c = 0$, 点C的坐标为(u, v), 过点C做对直线AB的垂线, 垂足为D, 其坐标为(x_0, y_0), 如下图:



则可以得出直线CD的参数方程为: $y - v = \frac{b}{a}(x - u)$, AB和CD的交点为D(x_0, y_0), 联立得到:

$$\begin{cases} ay_0 - bx_0 + bu - av = 0 \\ ax_0 + by_0 + c = 0 \end{cases}$$

可以解出 $x_0 = \frac{b^2u - abv - ac}{a^2 + b^2}$, 由于 $a^2 + b^2 = 1$,

所以可以解得目标距离 $d = \sqrt{1 + \frac{b^2}{a^2}} |x_0 - u| = \frac{|au + bv + c|}{\sqrt{a^2 + b^2}} = |au + bv + c|$

2. 简述EM算法的基本原理和流程 (以高斯混合模型求解为例)

EM算法可以用于解决数据缺失的参数估计问题, 以高斯混合模型(μ, Σ, π)为例, 这里的数据缺失就是隐变量 $\gamma_{t,k}$ (表示 $y_{t,k}$ 这个样本来源于第k个模型), 即只知道混合模型中各个类的分布模型 (譬如都是高斯分布) 和对应的采样数据, 而不知道这些采样数据分别来源于哪一类 (隐变量)

- 基本原理:

主要分为E-step(expectation-step)和M-step(Maximization-step), 前者就是基于上次迭代的混合高斯分布参数来对采样数据做分类划分, 即对隐变量 $\gamma_{t,k}$ 计算期望, 对各个类所划分到的样本数据求样本产生的概率; 后者则是基于前者的基础上, 求采样数据产生的概率Q函数, 然后通过最大化Q函数来优化混合高斯分布的参数, 如此不断迭代循环。至于为什么需要多次迭代循环进行, 是因为EM算法中对于 γ 的估计利用是初始化或者某一步迭代的混合高斯分布参数(μ_i, Σ_i, π_i)。

- 流程:

- E-step:

我们现在有样本集 $Y = (y_1, y_2, \dots, y_T)$, 再引入隐变量 $\gamma_{t,k}$ (表示 y_t 这个样本来源于第k个模型), 由此可以占城完全数据:

$$(y_t, \gamma_{t,1}, \gamma_{t,2}, \dots, \gamma_{t,K}), t = 1, 2, \dots, T$$

上面完全数据的似然函数为:

$$\begin{aligned} p(y, \gamma | \mu, \Sigma, \pi) &= \prod_{t=1}^T p(y_t, \gamma_{t,1}, \gamma_{t,2}, \dots, \gamma_{t,K} | \mu, \Sigma, \pi) \\ &= \prod_{t=1}^T \prod_{k=1}^K (\pi_k N(y_t; \mu_k, \Sigma_k))^{\gamma_{t,k}} \\ &= \prod_{k=1}^K \pi_k^{\sum_{t=1}^T \gamma_{t,k}} \prod_{t=1}^T (N(y_t; \mu_k, \Sigma_k))^{\gamma_{t,k}} \end{aligned}$$

完全数据的对数似然函数为:

$$\ln p(y, \gamma | \mu, \Sigma, \pi) = \sum_{k=1}^K \left(\sum_{t=1}^T \gamma_{t,k} \right) \ln \pi_k + \sum_{t=1}^T \gamma_{t,k} \left(-\ln(2\pi) - \frac{1}{2} \ln |\Sigma_k| - \frac{1}{2} (y_t - \mu_k)^T (\Sigma_k)^{-1} (y_t - \mu_k) \right) \quad (2)$$

但是我们不知道 γ 的分布无法对其最大化，所以这里对以上对数似然函数取期望作为Q函数，最大化Q函数 $Q(\mu, \Sigma, \pi, \mu^i, \Sigma^i, \pi^i) = E_{\gamma}[\ln p(y, \gamma | \mu, \Sigma, \pi) | Y, \mu^i, \Sigma^i, \pi^i]$ ，即把完全数据的对数似然函数中的 $\gamma_{t,k}$ 替换为 $E(\gamma_{t,k} | y_t, \mu^i, \Sigma^i, \pi^i)$:

$$\begin{aligned} E(\gamma_{t,k} | y_t, \mu^i, \Sigma^i, \pi^i) &= p(\gamma_{t,k} = 1 | y_t, \mu^i, \Sigma^i, \pi^i) \\ &= \frac{p(\gamma_{t,k}=1, y_t | \mu^i, \Sigma^i, \pi^i)}{p(y_t)} \\ &= \frac{p(\gamma_{t,k}=1, y_t | \mu^i, \Sigma^i, \pi^i)}{\sum_{k=1}^K p(\gamma_{t,k}=1, y_t | \mu^i, \Sigma^i, \pi^i)} \\ &= \frac{p(y_t | \gamma_{t,k}=1, \mu^i, \Sigma^i, \pi^i) p(\gamma_{t,k}=1 | \mu^i, \Sigma^i, \pi^i)}{\sum_{k=1}^K p(y_t | \gamma_{t,k}=1, \mu^i, \Sigma^i, \pi^i) p(\gamma_{t,k}=1 | \mu^i, \Sigma^i, \pi^i)} \\ &= \frac{\pi_k^i N(y_t; \mu_k^i, \Sigma_k^i)}{\sum_{k=1}^K \pi_k^i N(y_t; \mu_k^i, \Sigma_k^i)} \end{aligned}$$

■ M-step:

有个Q函数，就可以对Q函数进行最大化，得到下一次迭代的模型参数了，即：

$$\mu^{i+1}, \Sigma^{i+1}, \pi^{i+1} = \arg \max Q(\mu, \Sigma, \pi, \mu^i, \Sigma^i, \pi^i)$$

对Q函数进行求导，并另其导数为0，可得：

$$\mu_k^{i+1} = \frac{\sum_{t=1}^T \frac{\pi_k^i N(y_t; \mu_k^i, \Sigma_k^i)}{\sum_{k=1}^K \pi_k^i N(y_t; \mu_k^i, \Sigma_k^i)} y_t}{E(\gamma_{t,k} | y_t, \mu^i, \Sigma^i, \pi^i)}, k = 1, 2 \dots K$$

$$\Sigma_k^{i+1} = \frac{\sum_{t=1}^T \frac{\pi_k^i N(y_t; \mu_k^i, \Sigma_k^i)}{\sum_{k=1}^K \pi_k^i N(y_t; \mu_k^i, \Sigma_k^i)} (y_t - \mu_k^i)^2}{E(\gamma_{t,k} | y_t, \mu^i, \Sigma^i, \pi^i)}, k = 1, 2 \dots K$$

$$\pi_k^{i+1} = \frac{E(\gamma_{t,k} | y_t, \mu^i, \Sigma^i, \pi^i)}{T}, k = 1, 2 \dots K$$

其中 $\mu_k^{i+1}, \Sigma_k^{i+1}, \pi_k^{i+1}$ 分别表示第(i+1)次迭代，第k个类的均值，协方差矩阵和所占的权重。

3. 用伪代码写出Mean-shift的算法流程（以图像分割为例），并分析影响算法性能的主要因素
Mean-shift的伪代码如下：

Algorithm 1 Mean-shift

Input: Γ : Picture features(X, Y, L, U, V) Φ : Kernel B : Bandwidth D : Min_{distance}**Output:** Λ : Picture Segmentation(X, Y, k)1: $\Gamma_{shifted} = \Gamma$ 2: **while** not all shifted points' shifting distance $< D$ **do**3: **for** P_i in Λ **do**4: $P_{i,shifted} = f_{shift}(P_i, \Phi, B)$ 5: $d_{i,shifted} = distance(P_i, P_{i,shifted})$ 6: $\Gamma_{shifted}.append(P_{i,shifted})$ 7: **end for**8: **end while**9: $clusters = cluster_points(\Gamma_{shifted})$

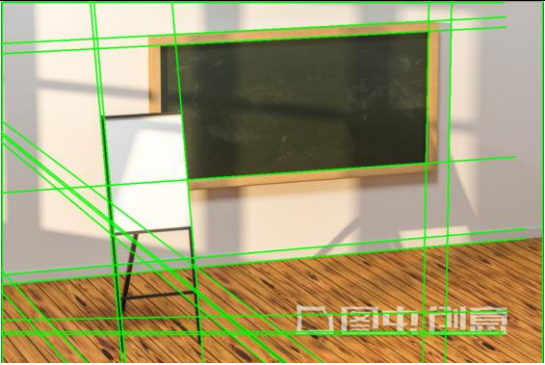
{function cluster_points is used to cluster the shifted points due to their distance, create and return clusters}

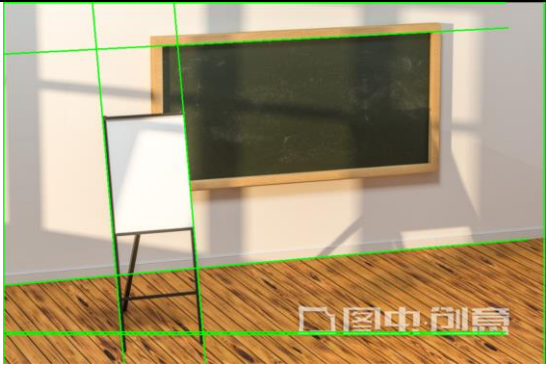
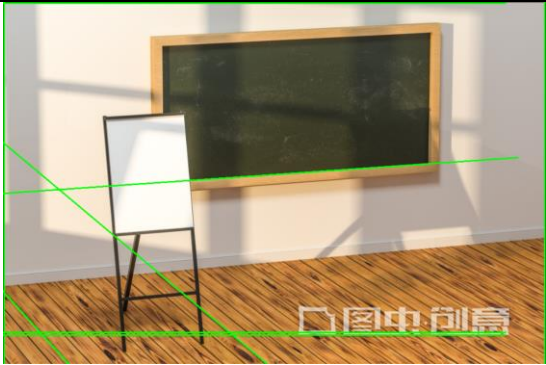
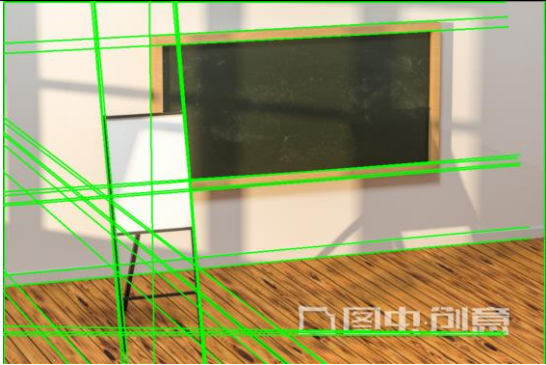
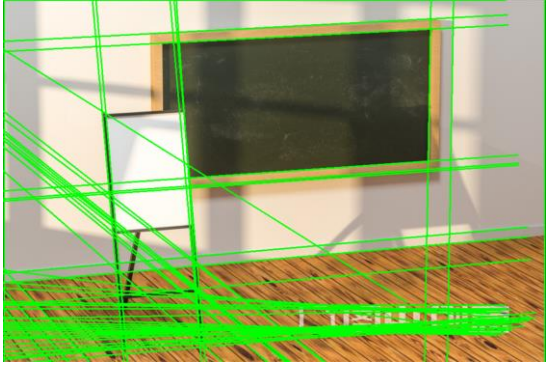
10: $\Lambda = segment(\Gamma, clusters)$

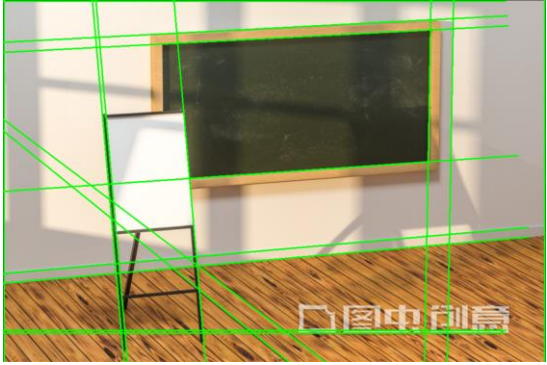
{function segment is used to remap the pixel value from the origin to the shifted points due to the clusters and return the result image}

对于Mean-shift算法，其输入参数为图像特征、计算偏移向量所在高维球的bandwidth，计算偏移向量时的核函数、最小偏移向量的模，所以很显然影响算法性能的主要因素有

1. 特征提取算法(SIFT显然要比ORB、LUV要慢，但是效果也许更好高)
 2. Bandwidth大小
 3. 计算漂移向量时所使用的核函数
 4. 判断是否收敛的阈值，即最小偏移向量的模
4. 找一张包含线条的图像，用霍夫变换进行线检测，并统计线条的数目。尝试不同的参数设置，并给出结果比较

霍夫直线检测结果	rho	theta	threshold	线条数
	1	Pi/180	150	22

	0.5	$\text{Pi}/180$	150	9
	1	$\text{Pi}/90$	150	7
	1	$\text{Pi}/240$	150	29
	1	$\text{Pi}/180$	120	57

	1	$\pi/180$	180	17
---	---	-----------	-----	----

由此可见，半径的步长 ρ 越大，所检测到的直线数量越小；每次偏转的角度 θ 越大，所检测到的直线越多；置信阈值 lines 越大，所检测到的直线越少，此参数过大可能导致检测不到任何直线。

5. 用线拟合的方式，对下图中的各文字行，插入删除线：

处理结果如下图所示，具体代码思路与实现详见代码中的注释

