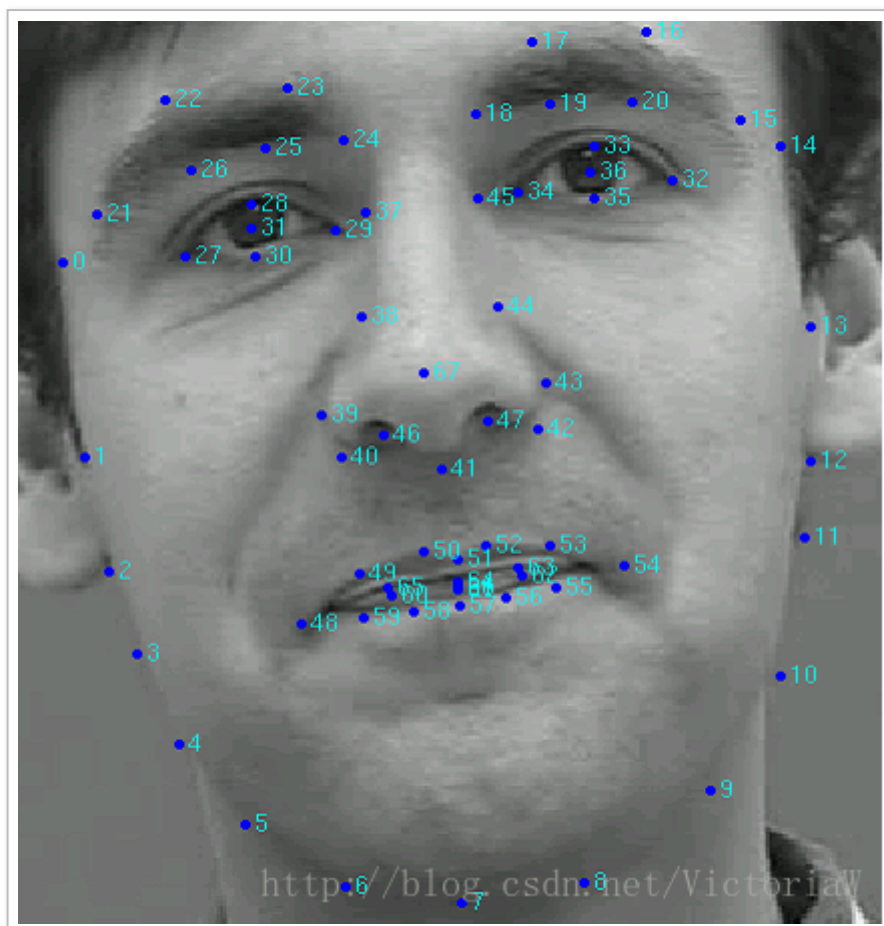


(21 条消息) 计算机视觉之统计形状模型——Active Shape Model_CodeTutor-CSDN 博客_统计形状模型

ASM (Active Shape Model) 由 Cootes 于 1995 年在论文 [1] 中提出。

ASM 是对图像中的 shape 进行建模的可变模型 (deformable model)，得到的形状模型既可以用来分析新的形状 (拟合模型到新形状，见第 4 节)，也可以用于生成形状 (在给定图像中搜索形状，见第 5 节)。

首先需要给出 shape 的定义。给定一幅图片，对感兴趣物体标注 landmark points，以人脸为例，就是选择人脸上的关键点，如下图所示：



得到点集合 $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ ，那么这些点在图片中对应的 shape 就可以表示成 $2n$ 维的向量 \vec{x} ：

$$\vec{x} = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)^T$$

可以看出每个 shape 都是 R^{2n} 空间中的一个点。并且我们规定对一个形状进行相似变换 (Similarity transformation), 即旋转 R、缩放 S、平移 T (RST), 并不会改变该形状。当然也可以定义形状在其他变换操作下是不变的。

假设现在有 N 个已经标注了 landmark 点的训练数据, 也就是说有了 N 个形状 $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$ 。

1、对齐

为了能比较不同 shapes, 需要将 shapes 对齐。这是因为不同的形状大小角度等都可能不同, 比如一个人脸特写与一张全身照中的人脸, 显然, 这两个人脸对应的形状是没有可比性的。

ASM 中的对齐算法是基于 [Procrustes Analysis](#) 的。Procrustes Analysis 在对齐形状时以最小化对齐后的形状和平均形状之间的距离之和为目的:

$$D = \sum_i |\vec{x}_i^{\rightarrow} - \vec{x}^{\rightarrow}|^2. \quad (1)$$

这个优化目标有意思, 首先式中的 \vec{x}_i^{\rightarrow} 是对齐后的第 i 个形状, \vec{x}^{\rightarrow} 是对齐后的形状的平均值。第一项是未知的, 而第二项又取决于第一项。假设 $\vec{x}_i^{\rightarrow} = T(\Theta_i; \vec{x}_i)$, 这个函数姑且称为对齐函数吧。那么每个形状都对应一个参数 Θ_i , 上述优化问题就是求使 D 最小的 $\Theta = \{\Theta_i, i = 1, \dots, N\}$:

$$\Theta^* = \arg \min_{\Theta} D = \arg \min_{\Theta} \sum_i |T(\Theta_i; \vec{x}_i) - \frac{1}{N} \sum_j T(\Theta_j; \vec{x}_j)|^2 \quad (2)$$

这个优化问题有解析解, 但是一般采用下面的迭代算法进行对齐:

1. 平移每个形状使其重心位于原点
2. 选择第一个形状 \vec{x}_1 , 对齐进行缩放 \vec{x}_1 以使 $|\vec{x}_1| = 1$, 以缩放后的形状作为初始平均形状 \vec{x}_0
3. 对齐所有形状到平均形状
4. 计算对齐后的所有形状的平均形状 \vec{x}
5. 对 \vec{x} 进行约束: 让 \vec{x} 对齐 \vec{x}_0 并且缩放到 $|\vec{x}| = 1$
6. 如果不收敛, 返回到 (3)

说明: 步骤 (5) 对平均形状进行约束, 是为了确保有唯一解 (如果不进行约束的话, 平均形状可能变得非常离谱)。另外判断是否收敛是看平均形状的变化是否超过某一幅度。

1.1、对齐操作

步骤 (3) 中的对齐操作可以有多种方式。比如可以移动每个形状使其以原点为中心，缩放使其模为 1，旋转使得 D 最小，这种方式下 D 的参数只有旋转角度。

文献 [1] 中介绍的方法是对每个形状进行相似变换 $T_{s,\theta,t_x,t_y}(\vec{x})$ 以对齐均值。假设两个形状 \vec{x}_1 和 \vec{x}_2 ，这两个形状以原点为中心，现在 \vec{x}_1 处理做相似变换使其对齐 \vec{x}_2 ，相似变换的参数通过最小化

$$E = |T_{s,\theta,t_x,t_y}(\vec{x}_1) - \vec{x}_2|^2 \quad (3)$$

来确定。相似变换的表达式为

$$T_{s,\theta,t_x,t_y}(\vec{x}) = s \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} x[\cdot] + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

变换得到：

$$T_{a,b,t_x,t_y}(\vec{x}) = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} x[\cdot] + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (4)$$

其中 $x[\cdot]$ 表示遍历 \vec{x} 中的每一个点。

现在来求解这个优化问题。令 E 对每个参数的偏导数为 0，得到下面的公式：

$$\begin{aligned} a \sum_{i=1}^{2n} [(x_1^i)^2 + (y_1^i)^2] + t_x \sum_{i=1}^{2n} x_1^i + t_y \sum_{i=1}^{2n} y_1^i &= \sum_{i=1}^{2n} x_2^i x_1^i + \sum_{i=1}^{2n} y_1^i y_2^i \\ b \sum_{i=1}^{2n} [(x_1^i)^2 + (y_1^i)^2] + t_y \sum_{i=1}^{2n} x_1^i - t_x \sum_{i=1}^{2n} y_1^i &= \sum_{i=1}^{2n} x_1^i y_2^i - \sum_{i=1}^{2n} y_1^i x_2^i \\ a \sum_{i=1}^{2n} x_1^i - b \sum_{i=1}^{2n} y_1^i + t_x &= \sum_{i=1}^{2n} x_2^i \\ a \sum_{i=1}^{2n} y_1^i + b \sum_{i=1}^{2n} x_1^i + t_y &= \sum_{i=1}^{2n} y_2^i \end{aligned} \quad (5)$$

假设 \vec{x}_1 已经平移到以原点为中心，那么 $\sum_{i=1}^{2n} x_1^i = \sum_{i=1}^{2n} y_1^i = 0$ ，于是有

$$\begin{aligned}
t_x &= \sum_{i=1}^{2n} x_2^i \\
t_y &= \sum_{i=1}^{2n} y_2^i \\
a &= \frac{\sum_{i=1}^{2n} x_2^i x_1^i + \sum_{i=1}^{2n} y_1^i y_2^i}{\sum_{i=1}^{2n} [(x_1^i)^2 + (y_1^i)^2]} = \frac{\vec{x}_1 \cdot \vec{x}_2}{\|\vec{x}_1\|^2} \\
b &= \frac{\sum_{i=1}^{2n} x_1^i y_2^i - \sum_{i=1}^{2n} y_1^i x_2^i}{\sum_{i=1}^{2n} [(x_1^i)^2 + (y_1^i)^2]} = \frac{\sum_{i=1}^{2n} x_1^i y_2^i - \sum_{i=1}^{2n} y_1^i x_2^i}{\|\vec{x}_1\|^2}
\end{aligned} \tag{6}$$

如果 \vec{x}_1 没有移动到以原点为中心，那么需要求解复杂的方程组 (5)。

最终得到：

$$\begin{aligned}
s &= \sqrt{a^2 + b^2} \\
\theta &= \tan^{-1} \frac{b}{a}
\end{aligned}$$

注意文献 [1] 在式 (3) 中加了一个权重矩阵，以区别不同的点对距离的影响程度，感兴趣的可以阅读 [1]。

2、对形状的变化进行建模

对齐后的 x 在 $2n$ 维空间中构成一个 cloud，这个 cloud 对应这一个分布。如果能得到这个分布，那么我们不仅可以生成和训练数据相似的新形状，还可以分析给定的一个形状是否和训练数据一致。

2.1、PCA

首先，我们采用 PCA 降维技术对数据进行降维以便处理。PCA 降维过程如下：

- 首先计算数据均值：

$$x = \frac{1}{N} \sum_{i=1}^N \vec{x}_i$$

- 然后计算数据的协方差矩阵：

$$S = \frac{1}{N-1} \sum_{i=1}^N (\vec{x}_i - x)(\vec{x}_i - x)^T$$

- 计算 S 的特征值 λ_i 和对应的特征向量 \vec{p}_i ，并且按特征值递减顺序进行排列，即 $\lambda_i \geq \lambda_{i+1}$

- 用 S 的前 t 个特征值对应的特征向量构成矩阵 Φ ($2n \times t$)，那么数据 \vec{x} 可以近似表示成

$$\vec{x} = x + \Phi \vec{b} \quad (7)$$

2.2、 t 的确定

我们知道，每个特征值 λ_i 给出了训练数据在相应特征向量方向上的方差，那么训练数据的总方差为 $V = \sum \lambda_i$ 。 t 可以通过下式来确定：

$$\sum_{i=1}^t \lambda_i \geq fV$$

其中 f 表示希望保留多少可变性。

2.3、分析 \vec{b}

由式 (7) 可知，不同的 \vec{b} 对应不同的形状 \vec{x} 。我们认为和训练数据一致的形状是合理的，那么自然 \vec{b} 也存在一个合理范围，就是说 \vec{b} 的取值不是任意的，需要约束到合理范围内。我们假设 \vec{b} 服从分布 $p(\vec{b})$ ， $p(\vec{b})$ 可以根据训练数据进行估计。我们认为当

$$p(\vec{b}) \geq p_t \quad (8)$$

时，形状是合理的。

假设 b 是独立的并且服从高斯分布，那么有

$$\log p(\vec{b}) = -0.5 \sum_{i=1}^t \frac{b_i^2}{\lambda_i} + \text{const.} \quad (9)$$

我们可以对

$$b$$

进行硬约束

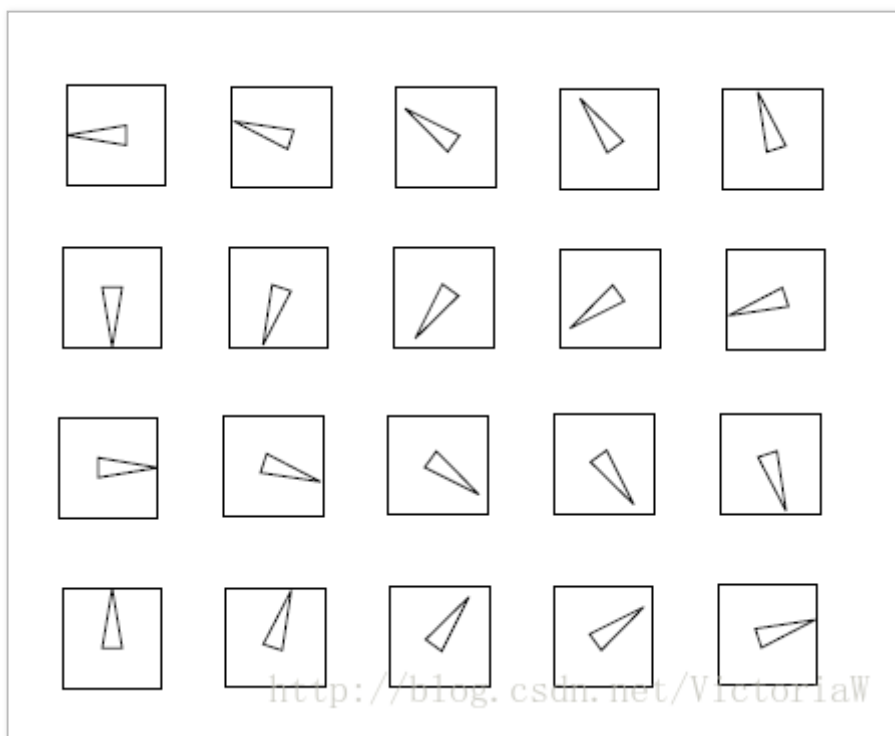
$$-3\sqrt{\lambda_i} \leq b \leq 3\sqrt{\lambda_i}. \quad (10)$$

也可以根据公式 (8) 并且结合公式 (9) 得到另一种约束：

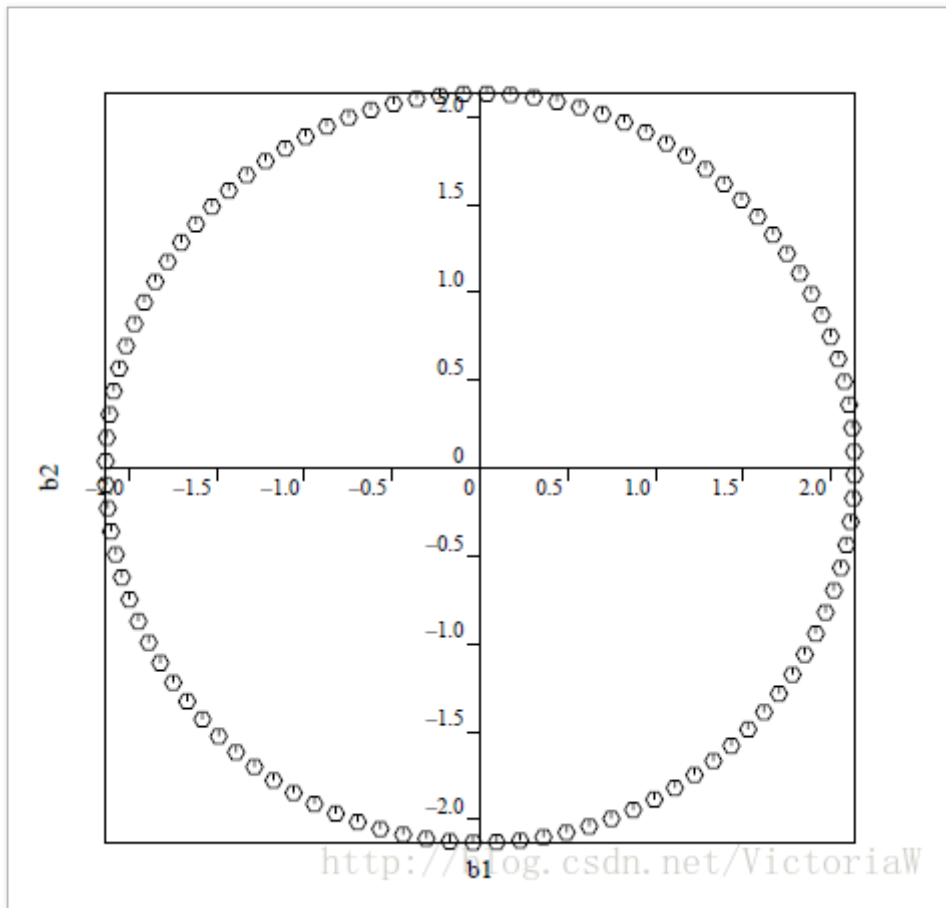
$$\left(\sum_{i=1}^t \frac{b_i^2}{\lambda_i} \right) \leq M_t \quad (11)$$

其中左边是 \vec{b} 和均值之间的 Mahalanobis 距离，右边的 M_t 可以通过卡方分布得到。

假设 \vec{b} 服从高斯分布可以解决大部分的问题，但是无法表示非线性的形状变换，比如只对形状的一部分进行旋转得到的训练数据。更具体地，我们通过旋转在一个正方形中的三角形得到一组训练数据，生成的样本如下图所示。



对训练数据应用 PCA 技术，发现有两个主要成分。我们把数据映射到二维空间中得到 \vec{b} 的分布如下图所示：



显然， \vec{b} 不是高斯分布。

这时可以用混合高斯模型 GMM 对 $p(\vec{b})$ 进行建模：

$$p(\vec{b}) = \sum_{j=1}^m w_j G(\vec{b} | \mu_j, S_j)$$

3、为当前模型形状 \vec{x} 找到最近的可能模型形状 \vec{x}

这个标题有点拗口，现在来解释一下为什么要给形状加定语“模型”。对于给定的一组训练形状数据 $\{X\}$ ，把每个形状看作 $2n$ 维空间中的一个点，可以想象得到这些点在 $2n$ 维空间中分布比较散乱，这就是因为形状还没有对齐。假设经过对齐（相似变换）后得到的形状集合为 $\{\vec{x}\}$ ，再在 $2n$ 维空间中找到对应的点，那么这些点就比较紧凑了。我们为了区分这两个形状，把 X 称为图像形状，把 \vec{x} 称为模型形状。这里处理的是模型形状。

而根据在 PCA 小节的介绍，每个模型形状可以由公式 (7) 表示，于是有

$$\vec{x} = x + \Phi \vec{b}$$

可以得到：

$$\vec{b} = \Phi^T (x^T - x)$$

我们在 2.3 节中提到过当

$$p(\vec{b}) \geq \rho_t$$

时，认为形状是合理的。而当

$$p(\vec{b}) < \rho_t$$

时，形状是不合理的，但是我们希望找到离形状最近的合理形状。这时无论

$$\vec{b}$$

是高斯分布还是混合高斯分布，我们对

$$p(\vec{b})$$

求梯度，然后沿着上坡方向移动直到超过阈值

$$\rho_t$$

。

4、用模型拟合新的形状 (Fitting a model to new shape)

现在我们讨论怎么求得最优 pose 参数 $\{s, \theta, X_t, Y_t\}$ 和 shape 参数 \vec{b} 以匹配给定的图像形状 Y 。

在前面的形状对齐过程中，我们是对图像形状 X 做相似变换 T ，映射到模型形状 \vec{x} 。那么也可以用一个相似变换 T' 把模型形状 \vec{x} 映射为图像形状 X ，并且 T' 可以由 T 得到

$$T' = T^{-1}$$

$$T^{-1} = \frac{1}{s} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} t_x \\ t_y \end{bmatrix} \right)$$

针对这一节的问题，我们通过优化问题

$$\min \|Y - T_{s,\theta,t_x,t_y}(x + \Phi\vec{b})\|^2$$

来求得最优参数。

下面给出优化问题的迭代法：

1. 初始化 $\vec{b} = 0$

2. 得到模型形状: $\vec{x} = x + \Phi \vec{b}$
3. 寻找最优参数 $\{s, \theta, t_x, t_y\}$ 使得最好地把 \vec{x} 匹配到 Y : 即 $\min \|Y - T(\vec{x})\|^2$, 这一步骤参考 1.1 节
4. 通过逆变换把 Y 映射为模型形状: $\vec{y} = T^{-1}(Y)$
5. 更新参数: $\vec{b} = \Phi^T (\vec{y} - x)$
6. 对 \vec{b} 进行约束
7. 判断是否收敛, 如果不收敛则返回步骤 2

判断是否收敛是通过检查参数是否变化。

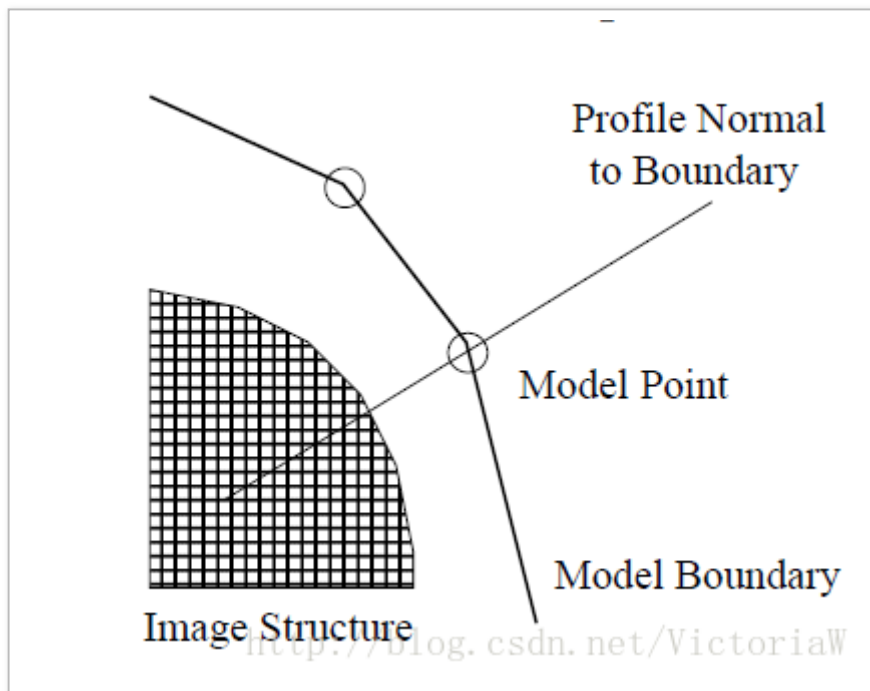
5、图像搜索——ASM

给定一张图片怎么找到其中的形状呢?

1. 初始化 $\vec{x} = x$, $\vec{b} = 0$, 并在图像中初始化一个形状 X
2. 检查 X 每个点附近的区域, 找到最好的新的形状 X'
3. 更新参数 $s, \theta, t_x, t_y, \vec{b}$ 以拟合新形状 X' : 第 4 节
4. 返回 2 直至收敛

这里需要着重介绍怎么根据当前形状 X 找到更好的形状 X' 。

对于以边缘作为形状的情况, 可以沿着模型边界的法线方向移动, 直至到达图像上梯度较高的点。如下图所示:



但是有些情况下形状不一定就是边缘，可能表示其他的结构。这时，可以通过对形状附近的图像结构进行建模，在搜索时，找最匹配所建模型的点。

常见的是对灰度结构进行建模。对训练形状 X_i 中的一个模型点沿法线方向在两侧各取 k 个点，于是每个模型点都得到 $2k + 1$ 个点，用灰度值构成向量 \vec{g}_i 。假设有 N 个训练数据，对一个模型点我们可以得到 N 个向量 $\{\vec{g}_i\}$ 。我们假设 \vec{g} 服从高斯分布，并且可以得到它的均值 \vec{g} 和协方差 S_g 。这样对一个模型点我们得到一个统计模型。

给出一个新的样本 \vec{g}_s ，Mahalanobis 距离：

$$f(\vec{g}_s) = (\vec{g}_s - \vec{g})S_g^{-1}(\vec{g}_s - \vec{g})^T$$

和

$$\vec{g}_s$$

来自于这个分布的 log 概率呈负相关性。

在搜索时，我们检查当前形状(model point)沿着模型边界的法线方向两侧各 $m(m > k)$ 个点，这样可以得到 $2(m - k) + 1$ 个 \vec{g} (自己在纸上画一画就知道为什么是这么多)，找到使得 Mahalanobis 距离最小的 \vec{g} 作为新的模型点。

6、实现

Stasm：作者 Stephen Milborrow，他研究生和博士生阶段一直在研究 ASM，于 2016 年博士毕业。在 [另一篇博客](#) 中简单介绍了 Stasm 源码。

参考：

- [1] Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: [Active shape models—their training and application](#) . CVIU 61, 38-59 (1995)
- [2] Cootes, T.F., Taylor, C.J.: Technical Report: Statistical Models of Appearance for Computer Vision. The University of Manchester School of Medicine (2004), www.isbe.man.ac.uk/~bim/refs.html
- [3] Youtube 视频: https://www.youtube.com/watch?v=53kx_czs7Es

全文完

本文由 简悦 SimpRead 转码，用以提升阅读体验，原文地址