



人脸识别相关技术 及其嵌入式应用



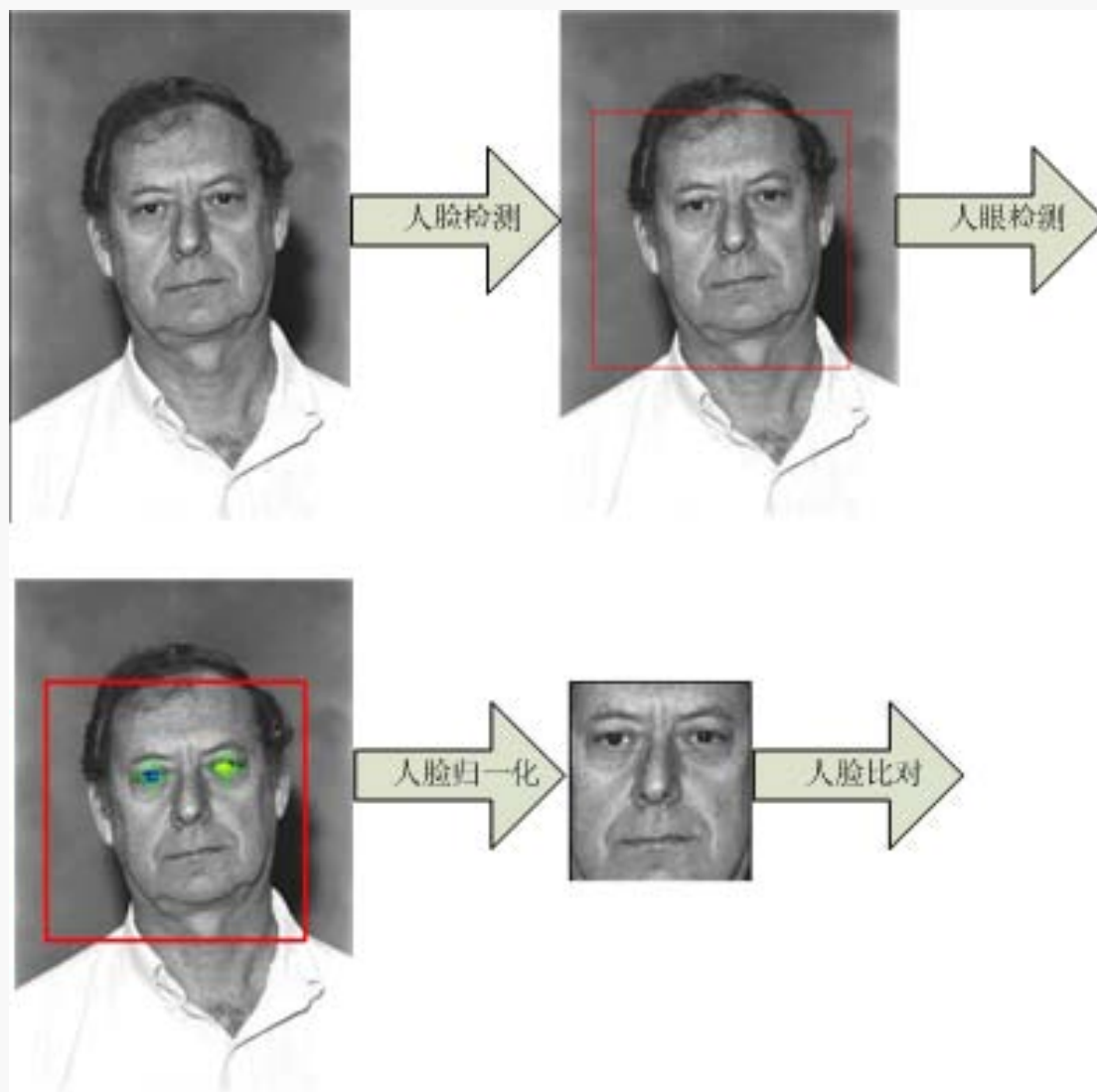
导师：赵衍运

学生：熊金水

研究内容

- 1 人脸检测
- 2 人眼检测及人脸归一化
- 3 人脸识别系统实现及优化

研究内容

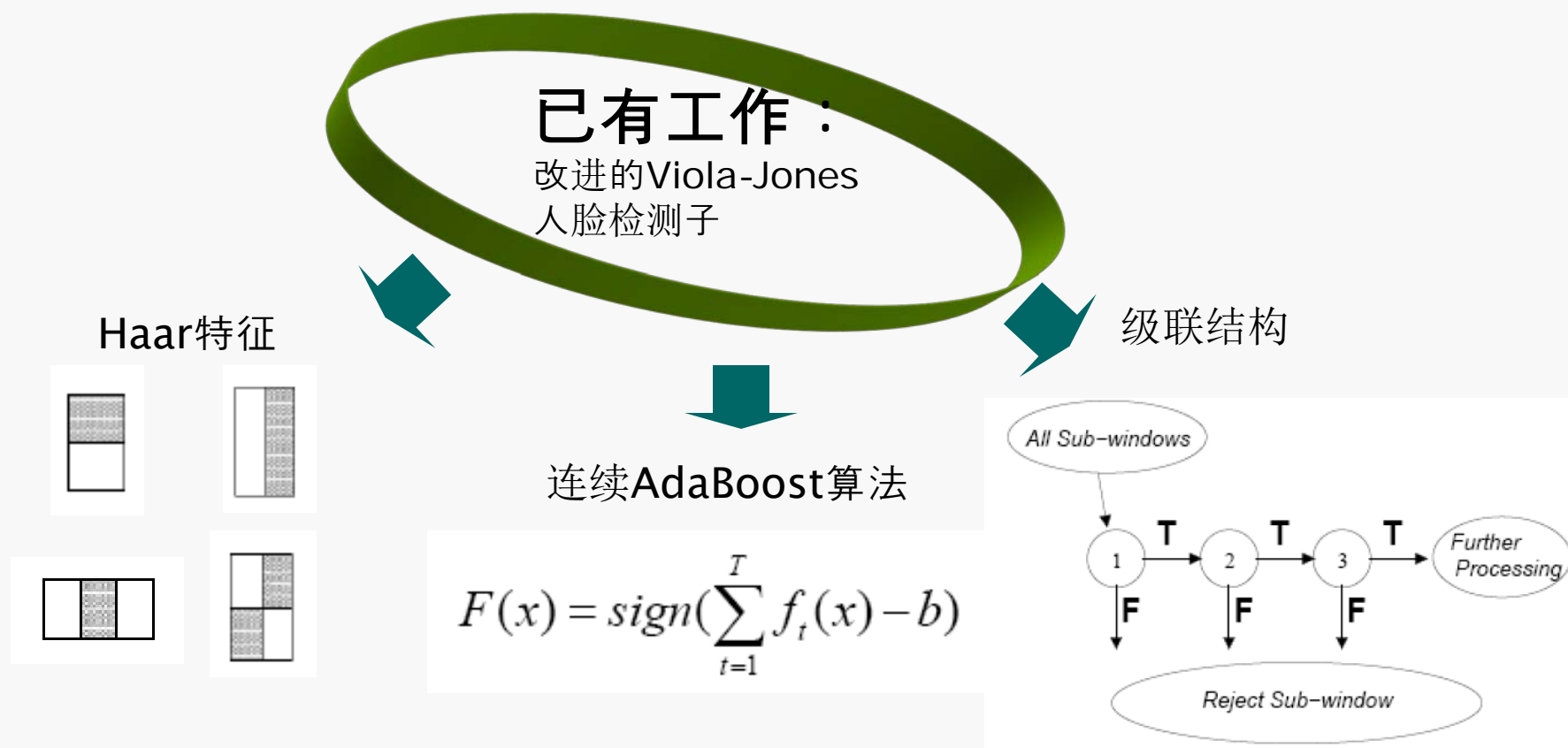


优化

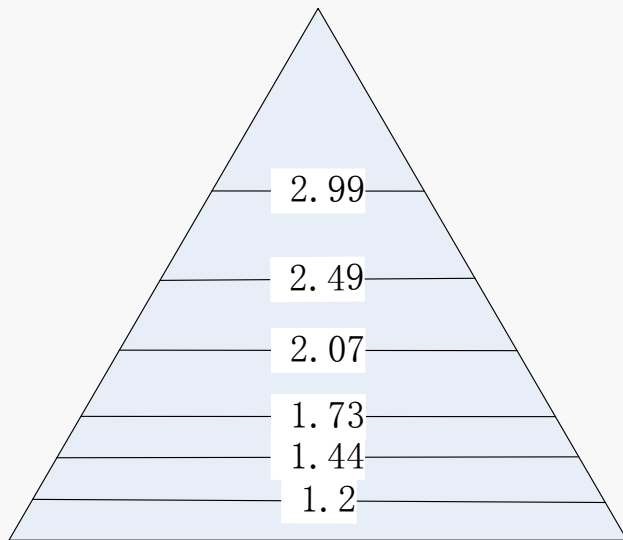
- **1.1 工作目标及已有工作**
- **1.2 我的工作**
- **1.3 结果**
- **1.4 进一步考虑**

1.1 工作目标及已有工作

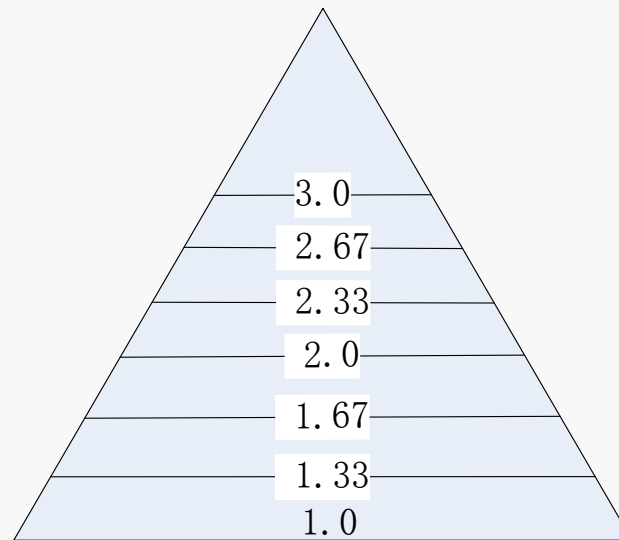
人脸检测目标：输入图像，输出人脸的位置，大小和姿态等。



1.2 我的工作——对图像金字塔的优化

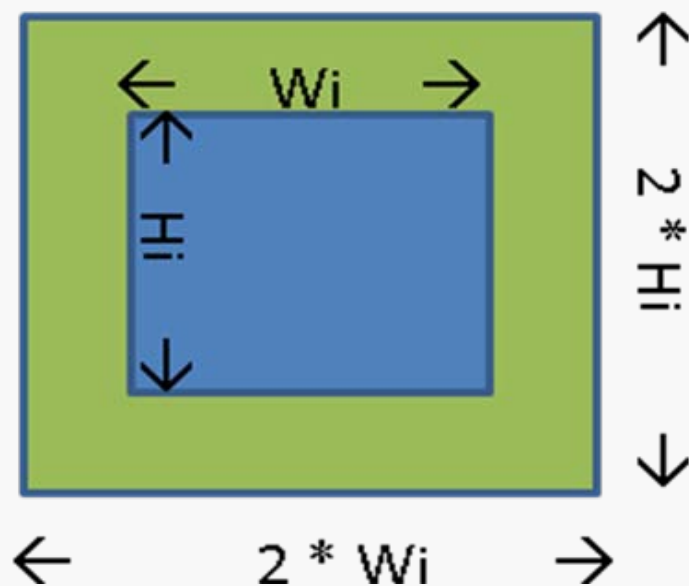


(a) 优化前：
指数变化的图像金字塔



(b) 优化后：
线性变化的图像金字塔

1.2 我的工作——对搜索策略的优化



- 中间蓝色方形代表人脸位置，绿色区域为屏蔽掉的区域。
- 从金字塔的顶层开始遍历子窗口

1.3 结果

- 测试集：FERET人脸库的Fa部分（含1196幅正面人脸图像）；
- 测试条件： Intel Xeon CPU 2.53GHz， 2.00GB的内存。

| / | 检测率TPR (%) | 错检 (个) | 平均耗时 (ms) |
|-----|------------|--------|-----------|
| 优化前 | 96.49 | 5 | 26.32 |
| 优化后 | 99.08 | 0 | 18.83 |



人脸检测部分结果

1.4 进一步考虑

- 有必要制定更加精确的人脸检测算法的指标



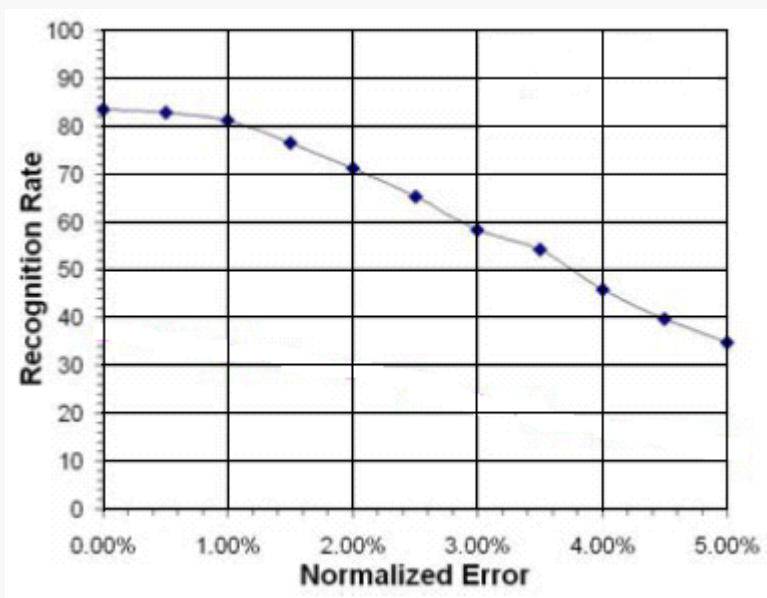
需要改进的人脸检测结果

人眼检测及人脸归一化

- **2.1 工作背景**
- **2.2 人眼粗定位**
- **2.3 人眼检测后处理**
- **2.4 人脸归一化**
- **2.5 结果**
- **2.6 进一步考虑**

2.1 工作背景

- 人眼检测：输入人脸图像，输出双眼的位置。
- 人脸归一化：根据人脸的标志点（通常为眼睛），使用旋转、平移和拉伸变换，使人脸图像有相同的尺寸。



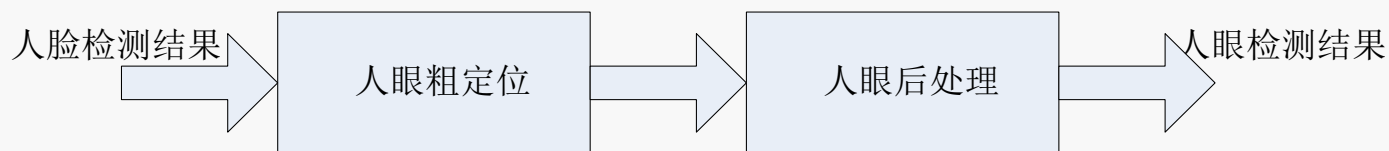
精确的人眼检测的重要性

横坐标：对人工标注的人眼位置，加入噪声的幅度。

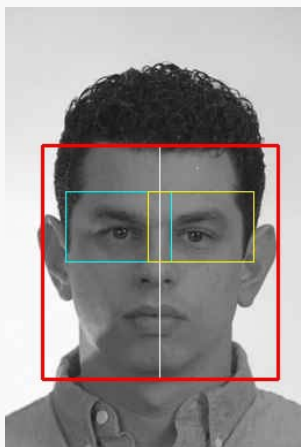
纵坐标：在FRGC人脸库中使用PCA方法得到的人脸识别率

2.2 人眼粗定位

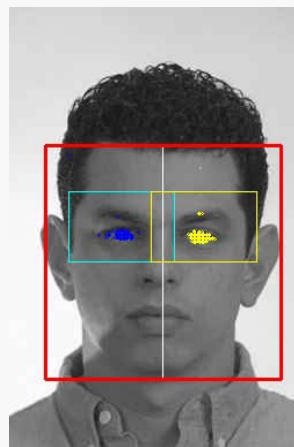
- 人眼检测的流程图



- 粗定位的方法：改进的粗定位人眼检测子。



人眼粗定位的搜索范围



人眼粗定位的结果

2.3 人眼检测后处理 ——AdaBoost方法

- 方法：用AdaBoost方法验证左眼和右眼能够组成一双眼睛的置信度，输出置信度最高的三双眼睛的平均值。
- 正负样本如下图所示：

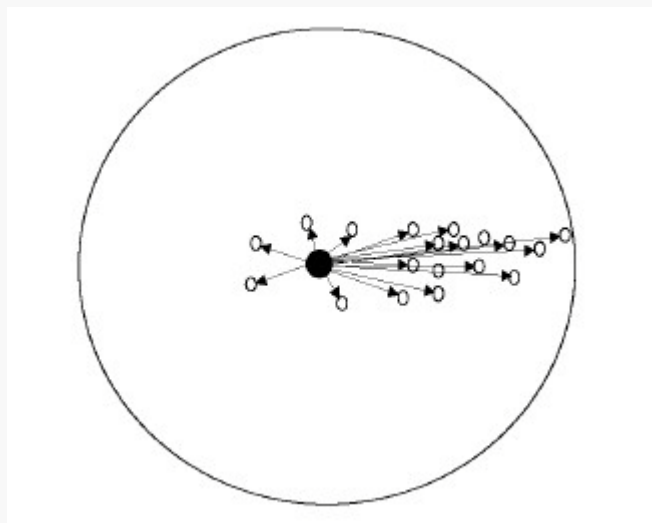


- 缺点：验证需经过级联结构的很多级；对丰富的候选点，需有效的下采样方法。

2.3 人眼检测后处理 ——mean shift方法（1）

- Mean shift: 使用迭代法寻找概率密度估计的最值。
- 最朴素的mean shift: 给定N个样本点，其中表示n维向量空间。在点处的均值偏移（mean shift）为：

$$m(x_0) = \frac{1}{k} \sum_{x_i \in S} (x_i - x_0)$$



2.3 人眼检测后处理 ——mean shift方法（2）

- 设 $\mathbf{y} = [x, y, s]$ ，其中 x, y 表示候选点的横纵坐标， s 表示候选点所在的尺度（指数变化），每个候选点的置信度（权值）为 ω ，则当迭代中心位于 \mathbf{y} 处时，使用高斯核函数的核密度估计为：

$$\hat{f}(\mathbf{y}) = \frac{1}{N(2\pi)^{3/2}} \sum_{i=1}^N \frac{\omega_i \exp(-\frac{D^2[\mathbf{y}, \mathbf{y}_i, \mathbf{H}_i]}{2})}{\sqrt{|\mathbf{H}_i|}}$$

其中

$$D^2[\mathbf{y}, \mathbf{y}_i, \mathbf{H}_i] = (\mathbf{y} - \mathbf{y}_i)^T \mathbf{H}_i^{-1} (\mathbf{y} - \mathbf{y}_i)$$

$$\mathbf{H}_i = \begin{bmatrix} (\exp(s_i)\delta_x)^2 & 0 & 0 \\ 0 & (\exp(s_i)\delta_y)^2 & 0 \\ 0 & 0 & \delta_s^2 \end{bmatrix}$$

2.3 人眼检测后处理 ——mean shift方法 (3)

- 推导得均值漂移为:

$$\mathbf{y}_{m+1} = \mathbf{H}_h(\mathbf{y}_m) \left(\sum_{i=1}^N \varpi_i(\mathbf{y}_m) \mathbf{H}_i^{-1} \mathbf{y}_i \right)$$

其中,

$$\mathbf{H}_h^{-1}(\mathbf{y}) = \sum_{i=1}^N \varpi_i \mathbf{H}_i^{-1}$$

$$\varpi_i(\mathbf{y}) = \frac{|\mathbf{H}_i|^{-1/2} \omega_i \exp(-\frac{D^2[\mathbf{y}, \mathbf{y}_i, \mathbf{H}_i]}{2})}{\sum_{i=1}^N |\mathbf{H}_i|^{-1/2} \omega_i \exp(-\frac{D^2[\mathbf{y}, \mathbf{y}_i, \mathbf{H}_i]}{2})}$$

2.3 人眼检测后处理 ——mean shift方法（4）

- 实现的考虑
 - 初始条件
 - 收敛条件
 - delt_x , delt_y , delt_s 调整
- AdaBoost方法和mean shift方法的比较
 - 相同点：~~精度相同。~~

AdaBoost

- 速度慢
- 利于后期人脸比对

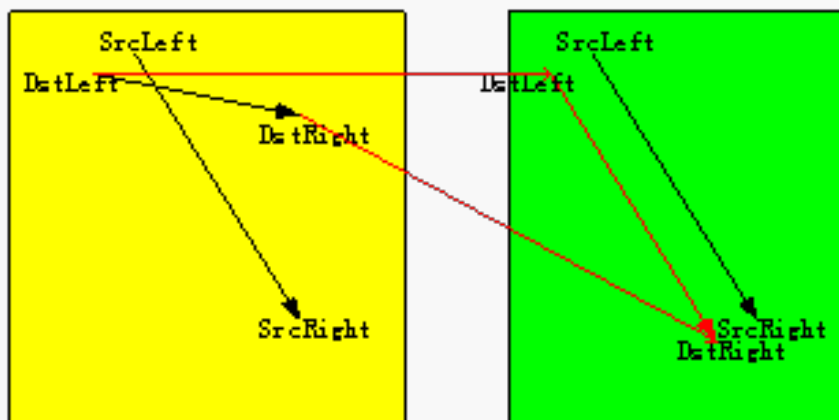
Mean shift

- 速度快
- 不利于后期人脸比对

2.4 人脸归一化算法

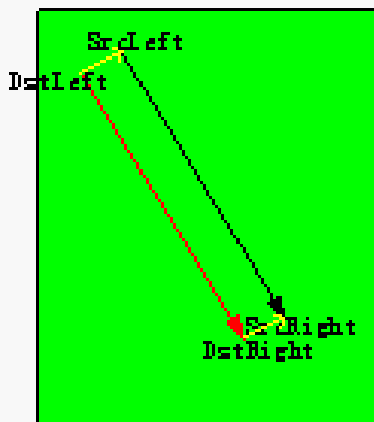
旋转
缩放

$$\begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} m \cos \alpha & -m \sin \alpha \\ m \sin \alpha & m \cos \alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

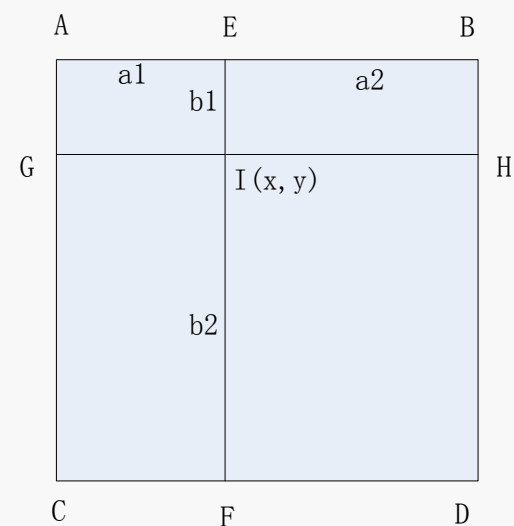


平移

$$TransVector = \begin{bmatrix} SrcLeft.x \\ SrcLeft.y \end{bmatrix}$$



双线性内插



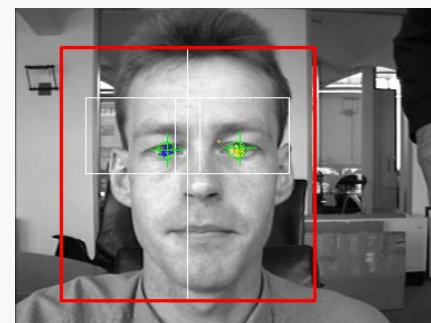
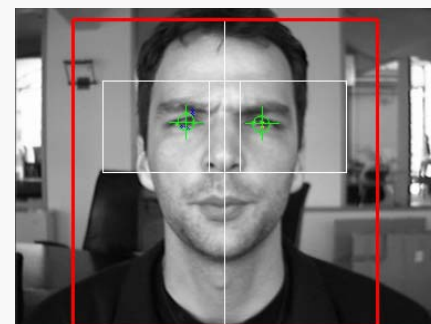
2.5 结果

普遍采用的人眼检测误差 err 定义如下，当 $\text{err} < 0.25$ 时，认为人眼检测正确。

$$\text{err} = \frac{\max(d_{\text{left_detect}}, d_{\text{right_detect}})}{d_{\text{standard}}}$$

测试集：BioID人脸库，共有1521幅包含人脸图片。

| 算法 | 检测率 (%) | 时间 (ms) | CPU主频 (GHz) |
|-----------------------|------------|---------|----------------|
| SVM + 边缘特征 | 99.46 | 80 | - |
| MIC + mean shift | 97.45 | 0.4 | 2.4 |
| AdaBoost + AdaBoost | 98.90 | 11.32 | 2.5 |
| AdaBoost + mean shift | 98.31 | 1.8 | 2.5 |



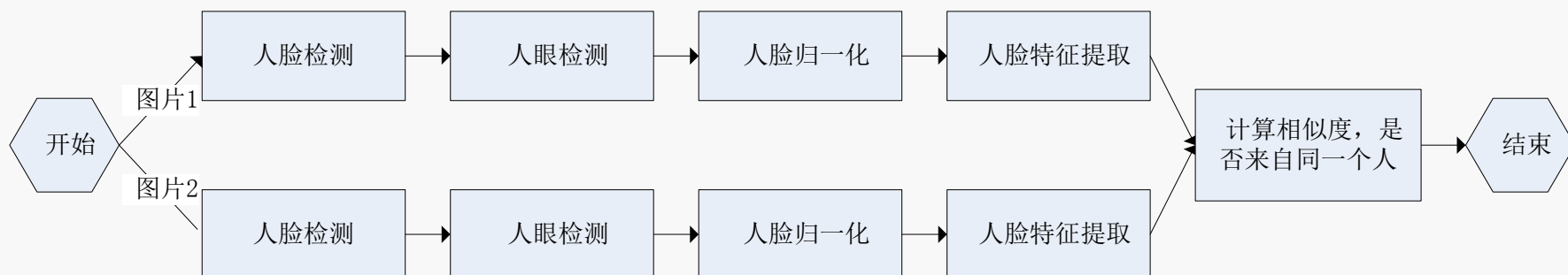
2.6 进一步考虑

- 有必要制定更加严格的指标；
- 能否利用人脸检测的结果，加速人眼检测；
- 是否有一种更有效的特征表征人眼模式；
- 是否可以合并人脸检测和人眼检测。

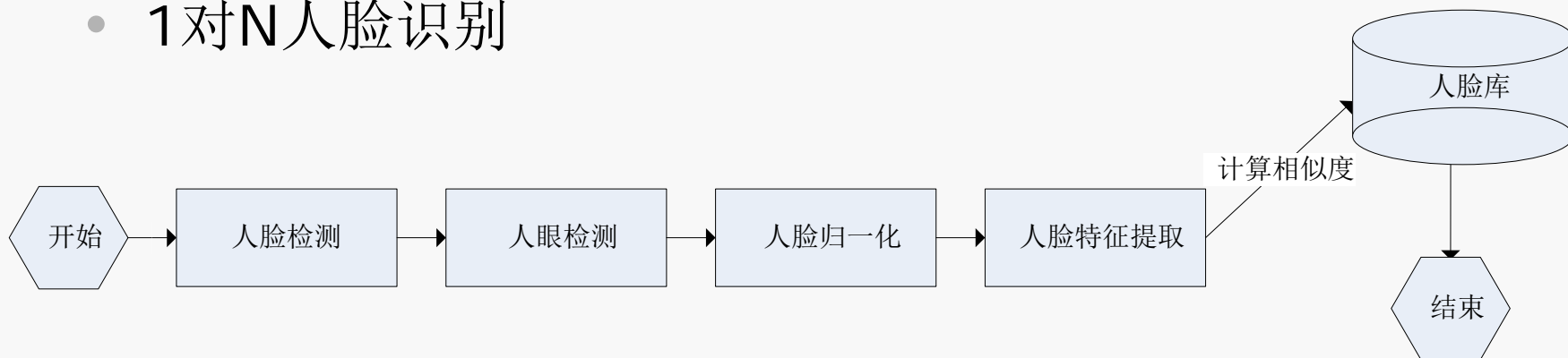
- **3.1 模块划分**
- **3.2 已有工作**
- **3.3 PC平台的优化**
- **3.4 嵌入式平台的优化**
- **3.5 进一步考虑**

3.1 模块划分及已有工作

- 1对1人脸识别

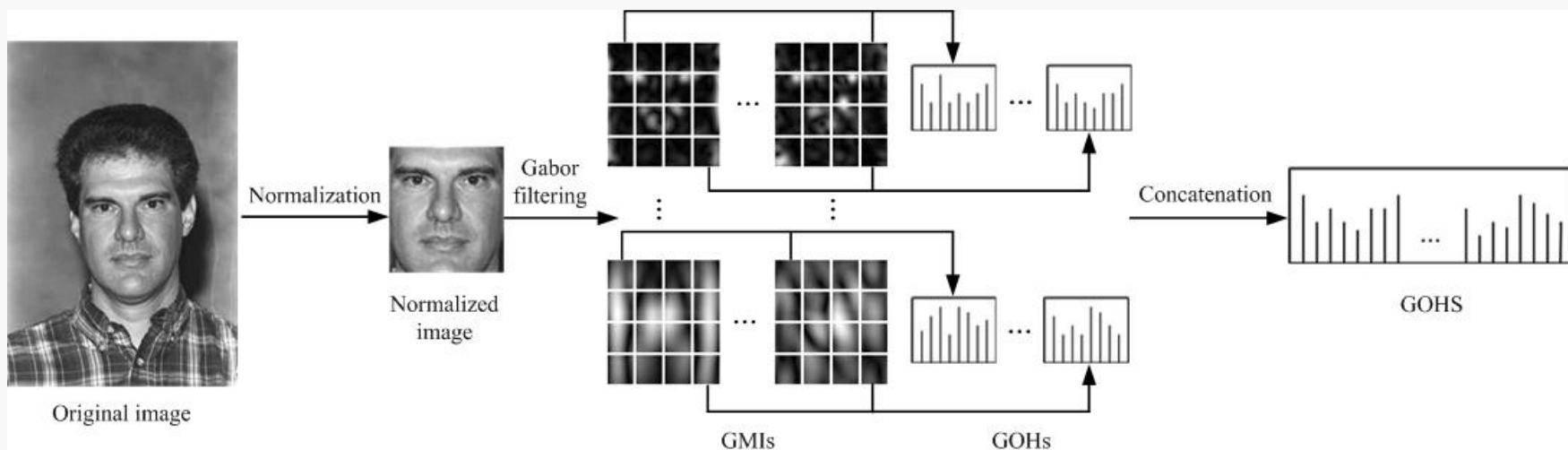


- 1对N人脸识别



3.2 已有工作

- GOH人脸描述子的提取



- LDA降维
- 余弦距离

3.3 PC平台的优化

- 使用FFTW开源库计算DFT和IDFT。

提取一幅图片特征的耗时

| 优化前 (ms) | 优化后 (ms) |
|-------------|-------------|
| 280 | 50 |

3.4 嵌入式平台的优化

- C++语言修改为C语言
 - 成员函数，成员变量
 - 构造函数，析构函数
 - 重载，如“=”
- 浮点运算优化为定点运算
 - 定标
 - 基本运算的优化，+，-，*，/等。
- 测试环境：FERET人脸库，DM642的DSP

| 优化前耗时 (ms/frame) | 优化后耗时 (ms/frame) |
|---------------------|---------------------|
| 2325 | 223 |

3.5 进一步考虑

- C++语言修改为C语言是否可以程序化



Thank you