

文章编号:1002-2082(2018)04-0491-09

# 一种抗姿态与表情变化的三维人脸识别方法

蔡川丽<sup>1</sup>, 张建平<sup>1</sup>, 张彦博<sup>2</sup>

(1. 延安大学 数学与计算机科学学院, 陕西 延安 716000; 2. 西安建筑科技大学 理学院, 陕西 西安 710055)

**摘要:**为了提高人脸在姿态和表情变化下的识别率,结合局部平面距离(DLP)对曲面局部凹凸性优良的判断能力,提出了一种采用人脸的等距不变表示形式来匹配的人脸识别方法。首先,对深度摄像头采集到的深度图像进行距离约束、位置约束、转换等操作,得到干净完整的三维人脸,利用三维脸上每一点DLP值确定鼻尖点,利用聚类的思想确定鼻根点;其次,采用改进的快速推进算法计算人脸的测地距矩阵,设置阈值并切割出有效的人脸区域;最后,计算有效的人脸区域的高阶矩特征,作为人脸的特征向量进行匹配。实验结果表明,对于不同的数据库,本文算法的识别率接近97%;将本文算法与基于轮廓线特征的人脸识别算法以及基于Gabor特征的人脸识别算法进行比较,其识别率分别提高了14.1%和8.3%,同时有着较高的运算效率。

**关键词:**人脸识别; 局部平面距离; 测地距离; 高阶矩; 等距不变

中图分类号:TN29; TP391

文献标志码:A

DOI:10.5768/JAO201839.0402002

## 3D face recognition method against gesture and expression change

Cai Chuanli<sup>1</sup>, Zhang Jianping<sup>1</sup>, Zhang Yanbo<sup>2</sup>

(1. College of Mathematics and Computers Science, Yan'an University, Yan'an 716000, China;  
2. School of Science, Xi'an University of Architecture and Technology, Xi'an 710055, China)

**Abstract:** In order to improve the facial recognition rate under the change of posture and expression, combined with the ability of local plane distance (DLP) to judge the convexity of local curved surface, a face recognition method based on face equidistant invariant representation was proposed. Firstly, several operations such as distance constraint, location constraint and transformation were conducted on the depth image captured by the deep camera to get the clean and complete 3D face; then the nose tip was determined by the DLP value of every point on the 3D face, and the nasal root was determined by the clustering idea; secondly, the improved fast propulsion algorithm was used to calculate the geodesic distance matrix of face, then the threshold value was set and the effective face area was cut out; finally, the high-order moment feature of the effective face area was calculated as the feature vector of face for matching. The experimental results show that the recognition rate of this algorithm is close to 97% for different databases. Compared with the face recognition algorithms based on contour features and Gabor features, the recognition rate of this algorithm is increased by 14.1% and 8.3%, respectively, while having a high computing efficiency.

**Key words:** face recognition; distance to local plane; geodesic distance; higher-order moment; isometry invariance

收稿日期:2017-12-30; 修回日期:2018-05-26

基金项目:国家自然科学基金(61763045);延安大学博士科研启动项目(YDBK2017-21);延安大学2016年度校级科研计划项目“伴随伸缩矩阵的二元周期小波框架的构造”(YDQ2016-23)

作者简介:蔡川丽(1988—),女,陕西韩城人,硕士,助教,主要研究方向为小波框架与图像处理。

E-mail:2855772727@qq.com

通信作者:张建平(1982—),男,陕西榆林人,博士,讲师,主要研究方向为小波分析。E-mail:2855772727@qq.com

## 引言

人脸识别技术<sup>[1]</sup>给人们的工作生活带来了很多方便,对社会的安稳发展具有一定贡献,人脸识别已经广泛应用于身份认证、打卡考勤、刑侦安检、信息安全等领域。目前关于二维人脸识别<sup>[2]</sup>的应用比较成熟,能够解决一些实际问题,但是二维人脸识别通常会受到一些外界条件(光照、姿态和表情)的影响,比如煤矿底下工作的工作人员,要记录他们的打卡考勤情况比较困难,因为脸庞往往被碳粉笼罩,往往会导致识别失败。为了解决这些问题,提高人脸识别的精确度,就出现了三维人脸识别<sup>[3]</sup>,三维人脸比二维人脸具有更丰富的信息,增加了一些刚性特征<sup>[4]</sup>,如人脸上两点之间的测地距<sup>[5]</sup>。

目前出现的人脸识别算法<sup>[6-7]</sup>主要采用3个技术,即主成分分析<sup>[10]</sup>(principal component analysis, PCA)、线性判别分析<sup>[11]</sup>(linear discriminant analysis, LDA)、独立成分分析<sup>[12]</sup>(independent component analysis, ICA)。比较熟知的特征脸算法,采用的就是PCA获取人脸的最优成分;ICA和PCA的区别在于,ICA假设人脸成分的分布不是服从高斯分布的;LDA算法主要是将相似人脸统计分布集聚到一类,差异最小化,不是一类的差异最大化的过程。这些方法在光照、姿态和表情变化下都会受到影响。

为了避免这些缺陷,文章提出了一种抗光照、表情、姿态变化的三维人脸识别方法,当人脸发生这些变化的时候,测地距并没有发生变化,采用多维尺度(multidimensional scaling, MDS)算法得到人脸的等距不变的表示形式,计算其高阶矩<sup>[13]</sup>作为人脸的特征向量,如果是同一个人在不同条件下的两个人脸,它们的等距不变的表示形式接近一样,因此有着相似的人脸特征向量,匹配误差非常小,如果是不同的人的两个人脸,因为它们本身脸上两点之间的测地距离就有差异,因此等距不变的表示形式是有明显差异的,计算两个特征向量会有比较大的匹配误差,这样可以达到识别的目的。当然,由深度摄像头采集的人脸数据是要经过一些预处理的,首先要经过距离约束、位置约束、去噪、补洞等得到干净完整的三维人脸,其次计算人脸上每一点的DLP值,利用3次筛选确定鼻尖点,利用聚类的方法确定鼻根点,以鼻尖点和鼻根点为中心,有效的测地距离为半径,切割出有

效的人脸区域,最后再计算测地矩阵并完成匹配过程。实验部分先展示了本文算法对不同数据库的识别效果,后面比较了不同文献算法的识别率,实验结果证明,文章算法的识别率和运算效率都有一定的提高。

## 1 三维人脸的等距不变表示形式

关键点检测和特征提取的精确度往往受到姿态表情变化的影响,从而影响人脸识别的准确度。这里提出了一种三维人脸的等距不变的表示形式,即同一个人脸的姿态的变化和微表情的变化(不撕裂的情况下)被看作是等距变换,等距变换的意思是三维人脸上任意两点之间的测地距离没有发生变化,将两个人脸用等距不变的形式表示,如果是同一个人,则等距不变的表示形式几乎一致,有着较高的匹配相似度,不同人的脸上两点之间的测地距本身就有差异,因此等距不变的表示形似也差别较大,最终得到较低的匹配相似度,达到了抗姿态,抗表情的目的。

### 1.1 三维人脸标准型以及MDS算法

人脸的皮肤在姿态和微表情变化时只发生稍微的拉伸变化,测地距离几乎没有发生变化,因此可以通过测地距离构建三维人脸的标准型。将人脸曲面映射到另一个空间,在映射的过程中,保持映射前两点之间测地距离等于映射后两点之间的欧氏距离,这种变换称为等距嵌入。用下面的表达式定义一个等距嵌入映射 $\varphi$ :

$$\varphi: (\{x_1, x_2, \dots, x_N\} \subset S, D) \rightarrow (\{x'_1, x'_2, \dots, x'_N\} \subset M^m, D') \quad (1)$$

其中: $S$ 表示原始人脸黎曼曲面; $M^m$ 表示映射后的 $m$ 维流形; $D$ 是三维人脸上人脸两点之间的测地距离构成的测地矩阵,即 $D = (d_{ij}) = (d(x_i, x_j))$ ; $D' = (d'_{ij}) = (d'(x'_i, x'_j))$ 且 $d_{ij} = d'_{ij}$ 。

MDS算法的目的是根据已知三维人脸点云,得到一些点的坐标,这些点尽可能地与对应原始曲面顶点之间的测地距离相近,连接这些点形成的曲面称为原始曲面的形变不变特征,实际上是一个等距嵌入的过程。(1)式中 $D$ 和 $D'$ 之间不可能完全相等,必然存在差异,称为嵌入误差,那么等距嵌入算法的主要工作就是寻找对应点集,确保嵌入误差最小。嵌入误差的表达式为

$$E(X') = \frac{\sum_{i < j} \omega_{ij} (d_{ij}(X) - d'_{ij}(X'))^2}{\sum_{i < j} d_{ij}^2(X)} \quad (2)$$

其中: $X$  为原始三维曲面; $X'$  为映射后的流形; $d_{ij}(X)$  为原始三维曲面上两点之间的测地距离; $d'_{ij}(X')$  为流形上对应两点之间的欧氏距离; $\omega_{ij}$  是对应两点距离的权重值,取决于对距离的贡献度,如果某些距离丢失,那么对应的权值设置为 0。这样,MDS 算法最终转化为一个最优化问题,即求解最小  $E(X)$  的问题。

## 1.2 MDS 算法实现

a) 设原始人脸上有  $n$  个顶点,计算测地距离  $\delta_{rs}$ ,其中  $r$  和  $s$  是人脸上任意两点;

b) 计算矩阵  $A = [-\frac{1}{2}\delta_{rs}^2]$ ;

c) 计算矩阵  $B = [a_{rs} - a_r - a_s + a_{..}]$ , $a_{rs} = A(r,s)$ , $a_r = \frac{1}{n} \sum_{s=1}^n a_{rs}$ , $a_s = \frac{1}{n} \sum_{r=1}^n a_{rs}$ , $a_{..} = \frac{1}{n^2} \sum_{r=1}^n \sum_{s=1}^n a_{rs}$ ;

d) 计算矩阵  $B$  的特征值  $\lambda_1, \lambda_2, \dots, \lambda_n$  以及对应的特征向量  $v_1, v_2, \dots, v_n$ ,并对特征向量进行归一化,确保  $v_i^T v_i = 1$ ;如果  $B$  不是半正定,可采取下面两种方式进行处理:1) 忽略负的特征值,并继续后续的步骤;2) 在测地距离上增加一个合适的常数  $c$ ,即  $\delta'_{rs} = \delta_{rs} + c(1 - \delta_{rs})$ ,然后重新回到第 2 步。其中  $\delta^s$  是克罗内克符号(Kronecker delta),表示如果  $r=s$ ,则  $\delta^s=1$ ;否则  $\delta^s=0$ 。

e) 选择一个适当的维数  $p$ ,使得  $\sum_{i=1}^p \lambda_i / \sum_{i=1}^n \lambda_i^+$  的值接近于 1( $\lambda_i^+$  指正的特征值),或者  $p$  直接选择 3 或者 4;

f) 按照下面的式子计算  $p$  维空间的坐标, $x_{ri} = v_{ir}$  ( $r=1, 2, \dots, n$ ,  $i=1, 2, \dots, p$ ),至此,得到了原始三维人脸的一种等距不变的表示形式。

## 1.3 对齐与匹配

原始三维人脸在转换成等距不变的表示形式时,在空间上的位置、大小和方向上并没有一个统一的标准,不利于后面的匹配识别,因此在匹配和识别之前需要对其进行对齐。采用下面的方法对点集进行对齐,定义三维点集  $\{x_i, y_i, z_i\}_{i=1}^N$  的  $k$  阶矩如下:

$$m_{pqr} = \frac{1}{N} \sum_{i=1}^N x_i^p y_i^q z_i^r \quad (3)$$

其中  $p, q, r$  为非负整数且  $p+q+r=1$ ,对齐的过程大致分为 4 步。

第 1 步,使用一阶矩寻找中心,把三维点平移到以质心为原点的位置,  $(m_{100}, m_{010}, m_{001})$  为三维

点集的质心坐标,平移公式为

$$[x'_i, y'_i, z'_i]^T = [x_i - m_{100}, y_i - m_{010}, z_i - m_{001}]^T \quad (4)$$

第 2 步,计算二阶矩阵  $M$ :

$$M = \begin{bmatrix} m_{200} & m_{110} & m_{101} \\ m_{110} & m_{020} & m_{011} \\ m_{101} & m_{011} & m_{022} \end{bmatrix} \quad (5)$$

第 3 步,对矩阵  $M$  采用奇异值分解(SVD 分解),得到矩阵  $U$  和  $\Lambda$ :

$$U\Lambda' U^T = \text{SVD}(M) \quad (6)$$

第 4 步,采用旋转矩阵  $U$  和伸缩矩阵  $\Lambda$  对点集进行如下变换:

$$[x''_i, y''_i, z''_i]^T = \frac{1}{\Lambda(1,1)} \cdot U \cdot [x'_i, y'_i, z'_i]^T \quad (7)$$

2 个模型进行对齐后,下面需要做的就是匹配了,本文使用高阶矩的曲面匹配方法,其主要思想是使用曲面的  $k$  阶矩来表示它的特征,然后把  $k$  阶矩作为欧氏空间上的向量,进行距离计算。(8)式定义基于矩的距离计算:

$$d_{mon}(S_1, S_2) = \sum_{p+q+r=k} (m_{pqr}(X_1) - m_{pqr}(X_2))^2 \quad (8)$$

式中: $S_1, S_2$  是 2 个匹配曲面; $X_1, X_2$  是对应曲面上的点集,得到的距离越小,匹配度越高。在实际操作中,数据库中的被匹配人脸曲面模型,其高阶矩数据已经预先计算好,作为特征值保存在数据库中,以减少运算量。

## 2 计算点云上两点之间的测地距

计算三维点云上点与点之间的测地距的经典算法有 Dijkstra 算法<sup>[14]</sup>,但经典的 Dijkstra 算法要求事先对三维点云进行网格化,生成面片信息,加大了计算的复杂度,影响了测地距的实现效率。这里提出了一种改进的快速行进算法<sup>[15]</sup>来计算三维点云上的测地线,首先,找到一个映射,将二维参数域上的点映射到三维点云(如  $x(u_1, u_2) = (\cos(u_1), \sin(u_2), \cos(u_1) + \sin(u_2))$ ),这里的  $x$  是一个二维参数域  $[-2\pi, 2\pi] \times [-2\pi, 2\pi]$  到三维点的一个映射),一个曲面上的局部坐标系可以通过(9)式的偏导计算得到:

$$\xi_i = \frac{\partial x}{\partial u_i} \quad (9)$$

设  $z=f(x, y)$  是一个曲面方程,若在  $(x, y)$  处有一个变化量  $(\Delta x, \Delta y)$ ,则  $z$  的变化量可以表示为

$\Delta z = A\Delta x + B\Delta y + \rho(\sqrt{(\Delta x)^2 + (\Delta y)^2})$ , 当  $(\Delta x, \Delta y) \rightarrow 0$  的时,  $dz = f_x dx + f_y dy$ , 则:

$$dz^2 = (f_x dx + f_y dy)^2 = \\ (dx, dy) \begin{pmatrix} f_x \cdot f_x & f_x \cdot f_y \\ f_y \cdot f_x & f_y \cdot f_y \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} \quad (10)$$

通过(11)式的弧微分计算近似表示曲面上两点之间的表面距离:

$$ds^2 = \left( \frac{\partial x}{\partial u_1} du_1 + \frac{\partial x}{\partial u_2} du_2 \right) \cdot \left( \frac{\partial x}{\partial u_1} du_1 + \frac{\partial x}{\partial u_2} du_2 \right) = \\ \frac{\partial x}{\partial u_1} \cdot \frac{\partial x}{\partial u_1} du_1^2 + \frac{\partial x}{\partial u_1} \cdot \frac{\partial x}{\partial u_2} du_1 du_2 + \\ \frac{\partial x}{\partial u_2} \cdot \frac{\partial x}{\partial u_1} du_1 du_2 + \frac{\partial x}{\partial u_2} \cdot \frac{\partial x}{\partial u_2} du_2^2 = \\ (du_1, du_2) \begin{pmatrix} \frac{\partial x}{\partial u_1} \cdot \frac{\partial x}{\partial u_1} & \frac{\partial x}{\partial u_1} \cdot \frac{\partial x}{\partial u_2} \\ \frac{\partial x}{\partial u_2} \cdot \frac{\partial x}{\partial u_1} & \frac{\partial x}{\partial u_2} \cdot \frac{\partial x}{\partial u_2} \end{pmatrix} \begin{pmatrix} du_1 \\ du_2 \end{pmatrix} = \\ du^T G du$$

(11)

式中:  $du = (du_1, du_2)$ ;  $G$  是一个度量矩阵, 从(11)式可以看出, 局部坐标系统是正交的当且仅当  $G$  是一个对角矩阵。结合上面的知识, 构造如下的一个程函方程, 通过求解这个方程得到曲面上任意两点之间的测地距:

$$\| \nabla_G t \| ^2 = \nabla_u^T t G(u)^{-1} \nabla_u^T = 1 \quad (12)$$

利用常规的笛卡尔网格采样参数域  $U$ , 得到离散的网格, 对于网格上一点  $u_0$ , 根据网格的连通性可以得到  $u_0$  的邻域点  $u_0 + m$ , 比如四邻域  $m = (\pm 1, 0)^T$ 、 $(0 \pm 1)^T$ , 或者 8 邻域  $m = (\pm 1, 0)^T, (0, \pm 1)^T, (\pm 1, \pm 1)^T$ 。这两种邻域可以形成以  $u_0$  为支撑点的 4 个或者 8 个三角形, 网格上一个三角形的三个顶点  $u_0, u_0 + m_1, u_0 + m_2$  投影到曲面上为  $x_0 = x(u_0), x_1 = x(u_0 + m_1), x_2 = x(u_0 + m_2)$ , 实际上, 在  $x_0, \xi_1, \xi_2$  组成的局部坐标系中有:

$$x_i = x_0 + m_i \xi + m_i^2 \xi^2 \quad (13)$$

或者写成  $X = TM$ , 其中:  $X = (x_1, x_2)$ ;  $T = (\xi_1, \xi_2)$ ;  $M = (m_1, m_2)$ 。令  $E = M^T GM$ , 则有:

$$E = M^T GM = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \begin{pmatrix} \xi_1 \cdot \xi_1 & \xi_1 \cdot \xi_2 \\ \xi_2 \cdot \xi_1 & \xi_2 \cdot \xi_2 \end{pmatrix} \begin{pmatrix} m_1 & m_2 \end{pmatrix} = \\ \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} (\xi_1 \quad \xi_2) (m_1 \quad m_2) = M^T T^T TM = \\ (TM)^T TM = X^T X = \begin{pmatrix} x_1 \cdot x_m & x_1 \cdot x_2 \\ x_2 \cdot x_1 & x_2 \cdot x_2 \end{pmatrix} \quad (14)$$

于是有  $E_{12} = x_1 \cdot x_2 = |x_1| |x_2| \cos(x_1, x_2)$ , 当  $E_{12} > 0$  时, 有  $\cos(x_1, x_2) > 0$ , 则角  $x_1 x_0 x_2$  为第

一象限或者第四象限, 又因为角  $x_1 x_0 x_2$  是三角形中的角, 所以角  $x_1 x_0 x_2$  为锐角, 因此可以从  $E_{12}$  的正负来判断角  $x_1 x_0 x_2$  是否为锐角。由深度摄像头采集到的一幅深度图像, 结合相机参数, 可以通过(15)~(17)式将深度图像转化为三维点云:

$$z = \text{double}(d) / \text{camera. scale} \quad (15)$$

$$x = (j - \text{camera. cx}) \times z / \text{camera. fx} \quad (16)$$

$$y = (i - \text{camera. cy}) \times z / \text{camera. fy} \quad (17)$$

其中:  $x, y, z$  为转化后三维点云上的点;  $d$  为深度图像在  $i, j$  处的深度值;  $\text{camera. fx}, \text{camera. fy}$  为摄像头焦距;  $\text{camera. cx}, \text{camera. cy}$  为摄像头主点;  $\text{camera. scale}$  为摄像头平移参数。本文将这种转换方式作为二维点到三维点的一个映射, 测地距算法的实现步骤如 1)~5):

步骤 1) 设深度摄像头获取的一幅深度图像为  $\text{depth1}$ , 采用一些约束分割出人脸深度图像  $\text{depth2}$ , 对  $\text{depth2}$  进行中值滤波得到  $\text{depth3}$ , 记  $\text{depth3}$  的长为  $a$ , 宽为  $b$ ;

步骤 2) 根据步骤 1) 的映射关系将  $\text{depth3}$  映射为三维点云, 采用 DLP 算法计算三维点云人脸的鼻尖点( $\text{nosetip\_x}, \text{nosetip\_y}, \text{nosetip\_z}$ ), 根据点云与深度图像的关系确定鼻尖点对应深度图像的位置( $i_0, j_0$ )以及对应的深度值  $d_0$ ;

步骤 3) 初始化一个  $a \times b$  的网格矩阵  $D_0$ ,  $D_0(i_0, j_0) = 0$  其余位置的值取  $\infty$ , 通过不断扫描更新的方式, 最终得到一个测地距矩阵。已有的快速行进算法虽然也能够达到扫描更新的目的, 但是有着固有的扫描顺序, 并且访问网格点的顺序依赖于波阵面的形状, 会导致缓存系统的利用不够充分。这里提出一种光栅扫描的顺序, 能够克服以上缺点, 一次完整的扫描更新顺序为: 从上往下, 从左往右; 从上往下, 从右往左; 从下往上, 从左往右; 从下往上, 从右往左。通过迭代扫描数次, 得到一个较为精确的测地距矩阵;

步骤 4) 对于即将更新的一点  $t_0$ , 分两种情况进行更新:

a) 设网格矩阵为  $D_0$ , 以第 1 次扫描为例(从上到下, 从左往右), 从第 1 个点开始扫描, 第 1 个点为已知点, 当扫到第 1 个点的时候, 跳过, 更新后面的点, 第 1 行和第 1 列的点按照(18)式更新:

$$t_0 = \min(D_0(i, j), t_1 + \|x_1\|) \quad (18)$$

其中  $\|x_1\|$  为这两个点对应的三维点构成向量的模的长度;

b) 剩下可以构成三角形的点按照如下方式进

行更新:

◇ 若和  $t_0$  构成三角形的邻域点上的值都是有界的,设即将更新的点和 2 个邻域点构成的第一个三角形邻域点上的值为  $t_1, t_2$ (按照扫描的先后顺序确定  $t_1, t_2, t_3$ ),求解一元二次方程:

$$t_0^2 \cdot I^T QI - 2t_0 \cdot I^T Qt + t^T Qt - 1 = 0 \quad (19)$$

取其最大的解为  $t_0$  暂时的值,其中:  $I = (1, 1)^T$ ;  $t = (t_1, t_2)^T$ ;  $Q = (X^T X)^{-1}$ ;  $X = (x_1, x_2)$ ,  $x_1, x_2$  是  $t_1, t_2$  位置对应的三维点。若  $Q(t - t_0 I) > 0$  或者  $t_0 < \max\{t_1, t_2\}$ ,按照此式确定  $t_0$  的值,  $t_0 = \max\{t_1 + \|x_1\|, t_2 + \|x_2\|\}$ ,否则,保留该方程的最大解为  $t_0$  的值。最后得到的  $t_0$  的值和网格上这点原有的值作比较,取最小值为  $t_0$  的更新值。对于第 2 个三角形,按照此方式再更新一次,得到  $t_0$  最终的值;

◇ 若  $t_0$  和 3 个邻域点构成的三角形只有一个三角形上的点的值是有界的,则按照上文方式更新  $t_0$ ;

◇ 若  $t_0$  和 3 个邻域点构成的三角形有 2 个三角形上对应的点的值都存在  $\infty$ ,则按(20)式更新  $t_0$ :

$$t_0 = \min\{D(i, j), t_1 + \|x_1\|, t_2 + \|x_2\|, t_3 + \|x_3\|\} \quad (20)$$

式中: $D(i, j)$  为更新点原有的值; $\|x_1\|, \|x_2\|, \|x_3\|$  为三维点云上更新点与邻域点构成的向量的模。

步骤 5) 另外 3 个方向的扫描按照步骤 1)~4)的方式进行,完成了一次迭代,设第  $n$  次迭代完之后得到的测地距矩阵为  $t^n$ ,如果对所有的元素有:

$$|t^n(i, j) - t^{n-1}(i, j)| < \epsilon \quad (21)$$

则停止迭代。图 1 是迭代了 10 次的测试结果。

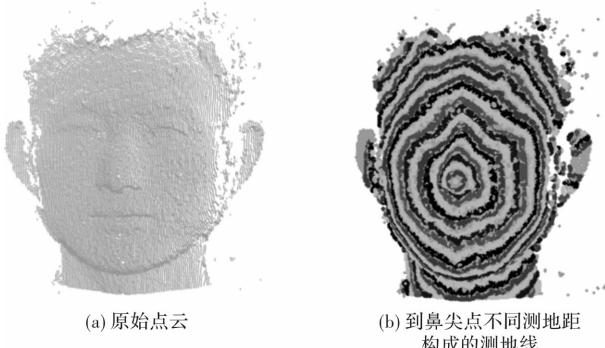


图 1 计算测地线测试图

Fig. 1 Test image of computing geodesic

### 3 局部平面距离确定曲面凹凸性

一般情况下,对于一个完整的不含噪点的三维人脸,鼻尖区域属于脸上最凸的区域,基于该思想,采用一种计算一点所在的邻域曲面的凹凸性的方法,即 DLP 算法,对于人脸点云  $\Omega$  上的任意一点  $v_i$ ,在  $v_i$  点的邻域找个点  $\{x_1, x_2, \dots, x_k\}$  作为  $v_i$  点的邻域点,计算  $v_i$  点的邻域的质心  $m_i$ :

$$m_i = \frac{1}{k} \sum x_i \quad (22)$$

然后计算协方差矩阵  $\text{cov}_i$ :

$$\text{cov}_i = \frac{1}{k} \sum (x_i - m_i)(x_i - m_i)^T \quad (23)$$

对  $\text{cov}_i$  进行奇异值分解得到 3 个特征值  $\lambda_1 \geq \lambda_2 \geq \lambda_3$  以及对应的特征向量  $e_1, e_2, e_3$ ,取  $e_3$  作为邻域面的  $z$  方向,计算  $v_i$  点的 DLP 值  $d_i$ :

$$d_i = e_3^T (v_i - m_i) \quad (24)$$

记  $d = [d_1, d_2, \dots, d_k, \dots]$  为所有点的 DLP 集合,为了避免特殊点对鼻尖点检测的影响,需要筛选 3 次确定最终的鼻尖点。第 1 次筛选,取  $d$  中大于 0 的点作为鼻尖点候选点  $c^1 = \{c_1^1, c_2^1, \dots, c_s^1\}$ ; 第 2 次筛选,计算三维人脸的质心  $m$ ,计算  $c^1$  中所有点与  $m$  的欧氏距离  $dc^1 = [dc_1^1, dc_2^1, \dots, dc_s^1]$ ,对  $dc^1$  进行从小到大排序,取前  $h$ ( $h$  为  $c^1$  中元素总个数的 50%,不是整数的往上进 1)个对应的点作为第 2 次鼻尖点筛选集合  $c^2 = \{c_1^2, c_2^2, \dots, c_h^2\}$ ; 第 3 次筛选,取  $c^2$  中 DLP 最大的点作为最终的鼻尖点  $nose\_tip$ 。图 2 为采用 DLP 算法进行鼻尖点检测的测试图。



图 2 鼻尖点检测测试图

Fig. 2 Test image of nose tip detection

### 4 算法流程

本文识别算法的总体思路为:对于输入的两个三维人脸,首先确定其鼻尖点和鼻根点,其次采用改进的快速行进算法计算三维脸上任意两点之间的测地距离,构造测地矩阵,并以鼻尖和鼻根为中心,有效的测地距离为半径,截取有效的人脸轮廓,最后将有效的人脸轮廓采用等距不变的方式表示,采用高阶矩完成最终的匹配。具体的算

法流程如下:

步骤1)由深度摄像头采集到的两幅深度图像,根据约束条件得到脖子以上的人脸深度图像,利用相机参数以及转换关系,将人脸深度图像转化为人脸点云数据,对人脸点云进行去噪、补洞等处理,得到干净完整的三维人脸;

步骤2)计算上述三维人脸上每一点的DLP值,记 $d=[d_1, d_2, \dots, d_k, \dots]$ 为所有点的DLP值得集合,经过3次筛选确定鼻尖点。第1次筛选:取 $d$ 中大于0的点作为鼻尖点候选点 $c'=\{c'_1, c'_2, \dots, c'_s\}$ ;第2次筛选:利用公式 $m=\frac{1}{N} \sum x_i$ 计算三维人脸的质心 $m$ ,计算 $c'$ 中所有点与 $m$ 的欧式距离 $d_c=[d_{c_1}, d_{c_2}, \dots, d_{c_s}]$ ,对 $d_c$ 进行从小到大排序,取前 $h$ 个( $h$ 为 $c'$ 中元素总个数的50%,不是整数的往上进1)对应的点作为第2次鼻尖点筛选集合 $c''=\{c''_1, c''_2, \dots, c''_h\}$ ;第3次筛选:取 $c''$ 中DLP值最大的点作为最终的鼻尖点。如图3所示。

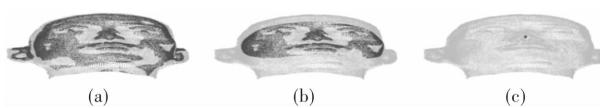


图3 鼻尖点筛选过程

Fig. 3 Filtering process of nose tip

步骤3)在三维人脸上每个点的DLP值已知的情况下,经过3次筛选确定眼内角点。第1次筛选:取 $d$ 小于0的点作为眼内角点候选点 $e'=\{e'_1, e'_2, \dots, e'_t\}$ ;第2次筛选:计算 $e'$ 中所有点与质心的欧式距离 $d_e=[d_{e_1}, d_{e_2}, \dots, d_{e_t}]$ ,取 $d_e$ 中前 $h_1$ 个( $h_1$ 为 $e'$ 中元素个数的5%,不是整数的往上进1)对应的点作为第2次眼内眼角点筛选集合 $e''=\{e''_1, e''_2, \dots, e''_{h_1}\}$ ;第3次筛选:对 $e''$ 中的点采用k-means聚类算法,将 $e''$ 中的点聚成两类 $e_a=\{e_{a_1}, e_{a_2}, \dots, e_{a_m}\}$ 和 $e_b=\{e_{b_1}, e_{b_2}, \dots, e_{b_n}\}$ ,分别取 $e_a, e_b$ 中DLP值最小的点作为眼内眼角点。如图4所示。

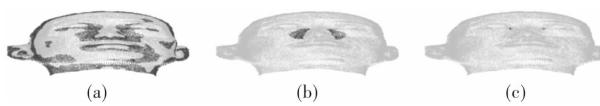


图4 鼻根点筛选过程

Fig. 4 Filtering process of nasal root point

步骤4)计算两个内眼角的中点作为鼻根点,采用改进的快速行进算法计算三维人脸上任意两点之间的测地距离,构成测地矩阵。以鼻尖为中

心,截取测地距离在 $r_1$ 以内的点构成区域 $D_1$ ,以鼻根为中心,截取测地距离在 $r_2$ 内的点构成区域 $D_2$ ,将 $D_1$ 和 $D_2$ 并起来得到有效的人脸轮廓;

步骤5)采用提出的MDS算法将有效的人脸轮廓表示成等距不变的形式并进行对齐,然后利用高阶矩得到表征人脸特征的特征向量,通过计算两个特征向量的欧式距离完成两个三维人脸的匹配。距离值越小,表示两个人脸越相似,距离值越大,说明两个人脸相差较大。具体流程图如图5所示。

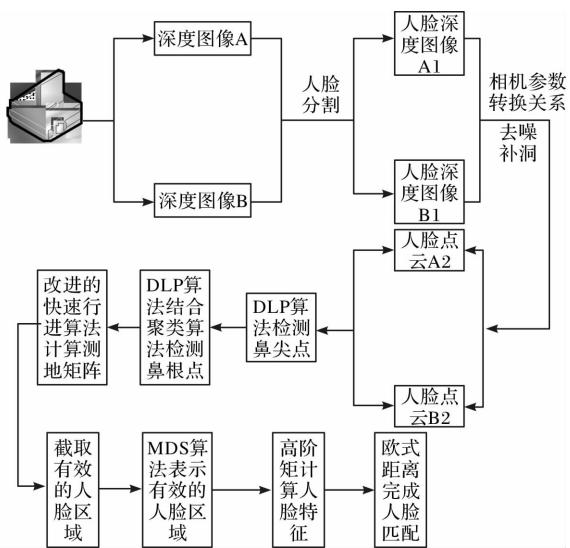


图5 人脸识别算法流程

Fig. 5 Algorithmic flow chart of face recognition

## 5 实验结果与分析

本文的实验机器为联想拯救者R720,处理器为i7-7700HQ,内存8G,显卡为GTX1050Ti,win10系统,实验软件为Matlab 2014a。实验通过两部分来完成,第1部分主要是为了说明本文算法的实用性较强,展示了算法在3个三维人脸数据库下的识别结果。选用的数据库分别是FRGC V2.0、Ear Database和CASIA-3D FaceV1,FRGC V2.0包含了466个人的4 007个人脸数据,这些人脸都是接近正面姿态,但带有丰富的表情变化,Ear Database主要包含了人脸的侧面姿态,有118个人在±45°的三维人脸和88个人在±60°的三维人脸,CASIA-3D FaceV1由中科院自动化研究所采集了123个人的三维人脸,其中每个人有38个不同光照、表情和姿态的三维数据,总共4 764个。第2部分通过比较不同文献的识别率来说明本文算法的优越性。

### 5.1 本文算法在不同数据库下的识别结果

这部分分为两个阶段,第1阶段,对每个数据库的人脸进行编号,同一个人有着相同的编号,并且事先计算好每个人的高阶矩特征,放入人脸特征库(这些人脸都是带编号的),作为每个人的表征。第2阶段为测试阶段,当测试某一个数据库时,随机提取数据库中一个人脸,计算其高阶矩特征和库中每个人脸特征的相似度,如果最终识别出的人脸和输入的人脸编号一致,表示识别成功,显示结果为1,否则,识别失败,显示结果为0,统计1的个数即得到当前数据库的识别率。图6分别为文章算法对FRGC V2.0、Ear Database和CASIA-3D FaceV1识别结果,表1为识别率统计结果。

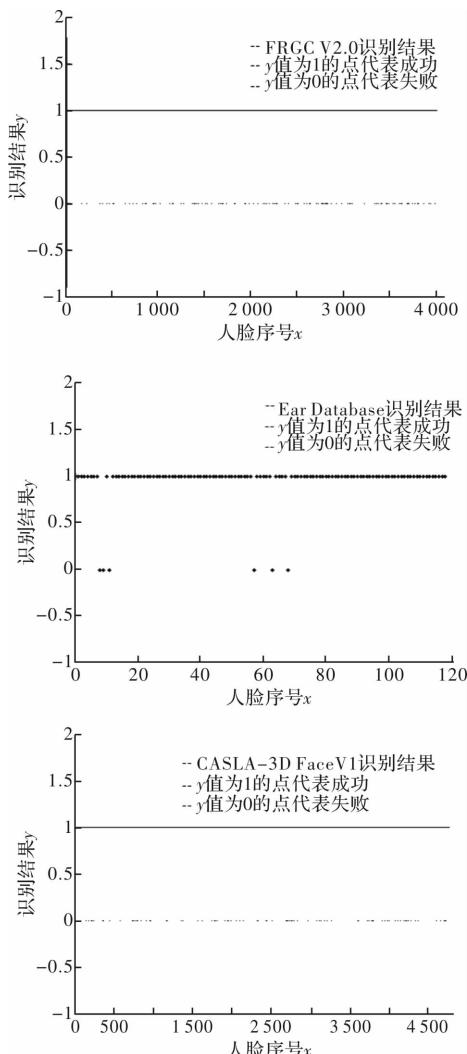


图6 FRGC V2.0、Ear Database 和 CASIA-3D FaceV1 的测试结果

Fig. 6 Test result of FRGC V2.0, Ear Database and CASIA-3D FaceV1

表1 识别率统计结果

Table 1 Statistical results of recognition rate

数据库	识别率
FRGC V2.0	96.26%
Ear Database	96.46%
CASIA-3D FaceV1	97.41%

分析可知,本文算法对FRGC V2.0和CASIA-3D FaceV1都有着较高的识别率,对Ear Database略微有些降低,总的来说,本文算法对不同的人脸姿态有着较强的鲁棒性和较高的识别率。

### 5.2 不同文献算法的识别结果比较

为了说明本文算法的优越性,实验数据从FRGC V2.0中随机提取400个人脸,从Ear Database中随机提取200个人脸,从CASIA-3D FaceV1随机提取400个人脸,总共1000个不同表情、姿态和光照的三维人脸,比较文献[16],文献[17]和本文算法的识别率,结果如表2所示。

表2 不同文献的识别结果比较

Table 2 Comparison of identification results from different literatures

比较文献	识别率
文献[16]	82.4%
文献[17]	88.2%
本文算法	96.5%

可以看出,尽管人脸数据丰富变化,但本文算法的识别率比文献[16]提高了14.1%,比文献[17]提高了8.3%,在识别率上都有了明显的提高。

### 5.3 运算效率

由于本文的识别系统事先计算了人脸库中每个人脸的高阶矩特征,这在识别的过程减少了很多的计算量,在测试的时候,对于一个输入模型,每个人脸点云大概有5万左右个点,各个分段需要的平均时间如下:DLP算法检测鼻尖点和鼻根点需要1.26 s,采用改进的快速推进算法计算地矩跟迭代的次数有关系,文章的迭代次数为4次,需要的时间是2.55 s,切割有效的人脸区域以及采用MDS算法表示有效的人脸区域需要0.92 s,计算人脸的高阶矩特征需要0.11 s,欧式距离完成最终匹配需要0.01 s。这些是采用开发工具Matlab 2014a进行试验所消耗的时间,由于Matlab编程效率高但运算效率略低,相信采用其他开发工具(如C++),各个阶段的运算时间会有大幅度

提升。

## 6 结束语

为了克服二维人脸识别受姿态和表情变化的影响,本文提出一种三维人脸的等距不变的表示形式,将人脸表情和姿态的变化看作是等距变化,这里的距离指的是三维人脸上两点之间的测地距离,即两点之间最短的表面距离。对于一个干净完整的三维人脸,计算人脸上每一点的DLP值,经过3次筛选,得到鼻尖点,利用聚类方法得到眼内角点,取两个内眼角的中点为近似鼻根点。采用改进的快速推进算法计算三维人脸的测地矩阵,以鼻尖点和鼻根点各为中心,有效测地距为半径,获取有效的人脸区域,将人脸表示成等距不变的表示形似,并计算高阶矩特征,作为最终的人脸特征向量。通过实验,验证了本文算法在不同数据库下的识别率,比较了该算法和其他文献算法的识别结果,实验结果证明,针对不同的人脸姿态和表情,本文算法有着较高的识别率和鲁棒性,识别率比基于轮廓线特征的人脸识别算法<sup>[16]</sup>提高了14.1%,比基于Gabor特征的人脸识别算法<sup>[17]</sup>提高8.3%,同时具有较高的运算效率。

## 参考文献:

- [1] ZOU Guofeng, FU Guixia, LI Haitao, et al. A survey of multi-pose face recognition[J]. Pattern Recognition & Artificial Intelligence, 2015, 28 (7): 613-625.
- [2] HU Min, CHENG Tianmei, WANG Xiaohua. Face recognition based on fusion of global and local feature [J]. Modern Computer, 2013, 27(9):817-822.
- [3] HU Binglian, WANG Dawei, LI Mingjun, et al. Study on 3D face recognition system against expression change[J]. Chinese Journal of Scientific Instrument, 2014, 35(2):299-304.
- [4] ZHOU Jun, YUAN Xian, YUAN Yanyan, et al. A novel 3D face recognition method based on profile and rigid regions[J]. Journal of Image & Graphics, 2010, 15(2):266-273.
- [5] ZHOU Jun, YUAN Xian, YUAN Yanyan, et al. Application of semantic features in face recognition[J]. Pattern Recognition Letters, 2008, 29(10):3251-3256.
- [6] LEE K C, HO J, KRIEGMAN D J. Acquiring linear subspaces for face recognition under variable lighting[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2005, 27(5):684.
- [7] LU J, YUAN X, YAHAGI T. A method of face recognition based on fuzzy c-means clustering and associated sub-NNs[J]. IEEE Transactions on Neural Networks, 2007, 18(1):150-60.
- [8] ZHU Lei, ZHU Shan'an. Face recognition based on two-dimensional image principal component analysis [J]. Journal of Zhejiang University, 2007, 41(2): 264-267.
- [9] 祝磊, 朱善安. 基于二维广义主成分分析的人脸识别[J]. 浙江大学学报, 2007, 41(2):264-267.
- [10] DONG Xiaoqing, CHEN Hongcai. Face recognition based on sub-pattern two-directional two-dimensional linear discriminant analysis[J]. Chinese Journal of Liquid Crystals and Displays, 2015, 30 (6): 1016-1023.
- [11] 董晓庆, 陈洪财. 基于子模式双向二维线性判别分析的人脸识别[J]. 液晶与显示, 2015, 30 (6): 1016-1023.
- [12] CHAI Zhi, LIU Zhengguang. Face recognition using complex wavelet and independent component analysis [J]. Journal of Computer Applications, 2010, 30 (7):1863-1866.
- [13] PERAKIS P, THEOHARIS T, PASSALIS G, et al. Automatic 3D facial region retrieval from multi-pose facial datasets[C]// Eurographics Conference

- on 3d Object Retrieval. [S. l]:[s. n.], 2009:37-44.
- [14] LU Huiming. 3D face recognition based on differential geometric method[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2009.  
陆慧铭. 基于微分几何方法的3D人脸识别[D]. 南京:南京邮电大学, 2009.
- [15] WEBER O, DEVIR Y S, BRONSTEIN A M, et al. Parallel algorithms for approximation of distance maps on parametric surfaces[J]. Acm Transactions on Graphics, 2008, 27(4):1-16..
- [16] LI Xiaojuan, YANG Tangwen, RUAN Qiuqi, et al. Modified 3D face recognition algorithm based on contour curve features[J]. Journal of the China Railway Society, 2012, 34(1):34-38.  
李晓娟, 杨唐文, 阮秋琦, 等. 基于轮廓线特征的三维人脸识别改进算法[J]. 铁道学报, 2012, 34(1):34-38.
- [17] XU C, LI S, TAN T, et al. Automatic 3D face recognition from depth and intensity Gabor features [J]. Pattern Recognition, 2009, 42(9):1895-1905.