# NLML-HPE: 通过流形学习在有限数据下进行头部姿态估计

Mahdi Ghafourian, Federico M. Sukno
Department of Engineering, Universitat Pompeu Fabra, Spain
(mahdi.ghafourian,federico.sukno)@upf.edu

## Abstract

头部姿态估计（HPE）在各种计算机视觉应用中扮演着关键角色，例如人机交互与人脸识别。本文提出了一种新颖的深度学习方法，利用称为 NLML-HPE 的非线性流形学习，在有限的训练数据下进行头部姿态估计。该方法基于张量分解（即 Tucker 分解）和前馈神经网络的结合。与传统基于分类的方法不同，我们的方法将头部姿态估计形式化为一个回归问题，将输入的地标映射到姿态角度的连续表示。为此，我们的方法使用张量分解将每个欧拉角（偏航、俯仰、滚动）分割为独立的子空间，并将底层流形的每个维度建模为余弦曲线。我们解决了两个关键挑战：1. 几乎所有的 HPE 数据集都存在不正确和不准确的姿态标注。因此，我们通过旋转 3D 头部模型到一组固定姿态并渲染相应的 2D 图像，为我们的训练集生成了一个精确且一致的 2D 头部姿态数据集。2. 我们在有限的训练数据下实现了实时性能，因为我们的方法能够准确捕捉面部地标的物体旋转特性。一旦学习了围绕每个轴的旋转的底层流形，模型在预测未见数据时非常快。我们的训练和测试代码以及训练好的模型均可在线获取：https://github.com/MahdiGhafoorian/NLML_HPE 。

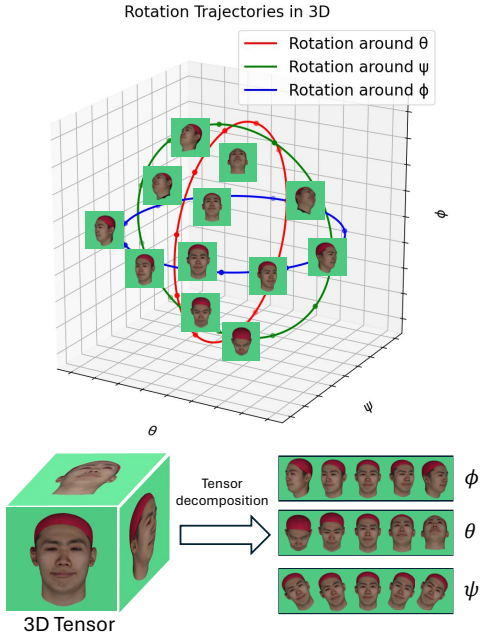


Figure 1. 将我们生成的数据集中样本的头部姿态变化分解为独立的子空间的概述。

## 1. 介绍

头部姿态估计（HPE）是指确定一个人的头部相对于相机或全局坐标系统的方向，有时还包括位置的过程 [1, 18] 。精确的 HPE 对于各种应用来说至关重要，包括驾驶员辅助（监控注意力）、人机交互（增强界面）、虚拟现实（改善透视渲染）、监控（行为分析）和定向广告（跟踪视觉注意力） [2] 。

表达人头姿态估计（HPE）的方法有多种。其中最流行的一种表示法是使用欧拉角。这种表示通常由三个角度组成：偏航，即绕垂直轴旋转（例如，头部的左右移动）；俯仰，即绕横轴旋转（例如，头部的上下移动）；以及滚转，即绕纵轴旋转（例如，头部的侧向倾斜）。虽然脸部图像被视为高维数据样本，但一个关键的观察是，其各种姿态可以定义一个有限制的姿态变化所约束的低维流形。因此，头部姿态估计可以被视为一个维度降低和流形学习的问题 [25, 9] 。

降维技术一般可以被分为线性、多线性和非线性方法。其中最广泛使用的线性方法是主成分分析 (PCA) [27] 和线性判别分析 (LDA) [20] 。早期研究 [33, 30] 已经利用 PCA 和 LDA 用于人脸识别任务 [13, 12] 。另一个值得注意的线性方法是保持局部性投影 (LPP) [6] ，以及它的变体 [19, 24] ，专注于在面部数据集中保持局部关系，同时捕捉其内在流形结构。然而，由于姿态流形的非线性，线性方法难以准确揭示其基础流形。

另一方面，多线性分析通过提供一个数学框架扩展了线性方法，该框架非常适合于多模式数据问题中的降维。在这些情况下，数据被表示为多维张量，采用多线性分解将张量分解为其正交模式矩阵。诸如 Isomap [22] 和局部线性嵌入（LLE） [11] 等方法已被研究用于学习由方向参数定义的潜在流形结构。虽然这些技术可以学习数据的低维表示，但它们产生的流形是隐式定义的。这使得很难施加准确建模由旋转变化引起的固

有结构的特定约束。

在本文中，我们提出了一种基于非线性张量的方法，该方法基于多线性分解 [8, 31]，以学习由 3D 旋转定义的流形。特别地，我们的方法能够将高维人脸图像的底层结构映射到低维姿态流形上，从而实现对每个欧拉角旋转作为独立流形的学习。为此，我们对数据描述符（如面部标志点）进行多线性分解（即 Tucker 分解），以分离旋转因子（如偏航、俯仰、滚动），并获得一组子空间。图 1 展示了对 $\mathbb{R}^3$ 张量进行这种分解的可视化。这些子空间的主成分对应于围绕一个欧拉角的姿态变化，并定义由独特的正弦参数控制的连续曲线。张量分解是一个计算代价高的操作，需要大量的处理时间和内存，使其不适合实时推理。为克服此限制，我们引入了一个有三个多层感知机（MLP）头的编码器，每个 MLP 头负责预测一个欧拉角。我们的方法的优势在于，我们仅对输入数据的有限子集执行分解，然后训练一个轻量级编码器和 MLP 头以在实时中实现等效的推理结果。

为了进行张量分解，训练集中的输入数据必须具有姿态一致性。这意味着，为了正确分解张量，必须为偏航、俯仰和滚转的每种可能组合提供一个输入样本以填充张量。据作者所知，没有现有的 HPE 数据集具备这一特性。虽然可以通过对缺失组合进行插值来填充张量，但训练集中姿态配置的分布可能导致张量主要由插值特征填充，从而导致不准确的特征向量。因此，特征向量可能主要反映插值过程引入的随机波动，而不是捕捉数据的主要特征。在我们的实验中，我们通过在 FaceScape 数据集中 ˜ [32] 的三维模型的内在旋转和渲染相应的二维图像，为每个姿态组合生成一个姿态一致的数据集来解决这个问题。这种方法不仅解决了姿态一致性问题，还生成了高度精确的 HPE 数据集注释，与其他数据集中容易出错的手动注释形成对比。我们使用生成的数据集进行分解和训练编码器进行姿态估计实验。我们采用 MediaPipe Face Mesh Toolkit ˜ [21] 从生成的人脸图像中提取面部标志。我们在两个流行的公开 HPE 数据集上评估我们的方法，分别是 BIWI ˜ [10] 和 AFLW2000 ˜ [36]，并证明通过训练三个 MLP 头可以实现角度估计，而不是通过 ˜ [9] 中描述的余弦函数的耗时优化。这既实现了最先进的准确性，又实现了实时性能。

本文的其余部分安排如下。第 2 节简要回顾了现有的头部姿态估计方法。第 3 节概述了张量分解方法，特别强调了高阶奇异值分解（HOSVD）。在第 4 节，我们介绍了我们的 NLML-HPE 框架，该框架建模由于旋转引起的数据变化，以实现实时姿态估计。第 ?? 节介绍了在两个广泛使用的 HPE 数据集以及我们生成的数据集上进行的实验。第 ?? 节讨论了我们方法的潜在局限性。最后，第 5 节总结了本文的内容。

## 2. 相关工作

人体姿态估计是计算机视觉中广泛研究的任务，其历史中有非常多样的方法。因此，我们仅关注与流形学习相关的方法，因为对所有 HPE 方法的全面审查超出了本文的范围。

基于流形的方法的主要思想是考虑对头部姿态变化的基本结构进行建模。在事实上，由改变面部方向生成的所有面部图像集本质上是一个三维流形（忽略或补偿其他类型的图像变化，如比例、光照等变化），然而，它嵌入在高维的图像空间中。由于该流形在特征定义的环境空间中具有固有的非线性，研究人员调查了非线性流形学习方法，例如局部线性嵌入、Isomap、同步子流形嵌入和同胚流形分析。2014 年，Takallou 和 Kasaei 提出了一种用于头部姿态估计的非线性张量模型，使用多线性分解来捕捉身份、姿态和像素信息，重点放在偏航旋转上。他们的方法将查询图像投影到姿态和身份子空间中，生成姿态参数，并与统一的姿态流形进行验证以获得最终估计。Sundararajan 和 Woodard 引入了一种基于特征的流形嵌入，用于在无约束图像中进行头部姿态估计。他们解决了学习反映仅由头部姿态引起的变化的相似性核，同时忽略其他变化来源的挑战。为实现这一目标，他们利用身份不变的几何模糊特征来学习相似性核。后来在 2018 年，Hong 等人提出了一个整合多模态数据和多任务学习的人脸姿态估计新框架。他们通过结合流形正则化卷积层（MRCL）来增强传统卷积神经网络，后者在低秩空间中学习神经元输出之间的关系。

最接近我们工作的研究是 Derkach 等人的研究，他们利用多线性分解将姿态变化（偏航、俯仰和滚动）分解到不同的子空间，有效地建模了由这些旋转产生的底层 3D 流形。他们的方法包括两个阶段：(i) 训练阶段，他们进行 HOSVD 来分解填充的张量并优化三角参数；(ii) 测试阶段，通过计算 $\underset{w^{(\text{yaw})},w^{(\text{pitch})},w^{(\text{roll})}}{\arg\min} \|x - \hat{x}\|$ 来优化输入人脸图像的变化角度，其中 $\hat{x}$ 通过计算复杂的 Einstein 求和操作获得。该方法不适合实时应用。相比之下，我们提出了一种方法，该方法使用轻量级编码器和三个 MLP 头来即时获得输入图像的头部姿态角度。

## 3. 技术背景：多线性分解

多线性分解是指将多维数据分解为更简单的低维成分的过程。多线性分解的一种常用方法是张量分解，比如塔克分解或 CANDECOMP/PARAFAC (CP)。高阶 SVD (HOSVD) 是一种塔克分解方法，其中因子矩阵是通过对张量的每个展平切片（模）应用奇异值分解 (SVD) 得到的 [4, 7]。特别地，张量通常被称为 n 路数组或 n 模矩阵。例如，向量和张量可以分别视为一阶和二阶张量。为了增强理解，我们首先回顾标准的 SVD。

给定矩阵 $A \in \mathbb{R}^{m \times n}$，其奇异值分解（SVD）如下：

$$A = U\Sigma V^T = \sum_{k=1}^{r} \sigma_k u_k v_k^T = \sum_{k=1}^{r} \sigma_k u_k \otimes v_k \quad (1)$$

即对 $A$ 的元素 $A_{ij}$，其中 $\otimes$ 表示张量（或外）乘积。$\boldsymbol{x} \otimes \boldsymbol{y} \triangleq \boldsymbol{x}\boldsymbol{y}^T$; $\Sigma$ 是一个对角矩阵，其对角线上是 $A$ 的

非零奇异值（$A^TA$ 的特征值的平方根）；$u_k$ 和 $v_k$ 分别是矩阵 $U$ $(m \times r)$ 和 $V$ $(n \times r)$ 的正交列，其中 $v_k$ 是 $A^TA$ 的特征向量，而 $u_k = Av_k/\sigma k$ [4] 。

在处理二维数据集 $A_{ij}$ 时，奇异值分解（SVD）是很有帮助的，该数据集自然表示为矩阵 $A$ 。在许多应用中，包括我们的应用中，我们必须处理多维数据。特别是，我们的目标是使用一个五阶张量来建模依赖于五个因素的数据：特征、身份以及围绕正交空间轴的三个旋转（偏航、俯仰、滚动）。这种方法使我们能够有效捕捉这些因素之间的复杂相互依赖关系。

奇异值分解（SVD）可以通过多种方式推广到高阶张量或多路数组。在本文中我们使用的方法是所谓的 Tucker/HOSVD 分解，如图 2 (a) 所示。

给定一个五阶张量 $T \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4 \times I_5}$ ，Tucker 分解可以表示为：

$$T = \sum_{J_1=1}^{I_1} \dots \sum_{J_5=1}^{I_5} G_{J_1 J_2 \dots J_5} a_{J_1}^{(1)} \otimes a_{J_2}^{(2)} \otimes \dots \otimes a_{J_5}^{(5)} \quad (2)$$

或者使用模乘法 [8] 表示为：

$$T_{j_1 j_2 \dots j_5} = \sum_{J_1=1}^{I_1} \dots \sum_{J_5=1}^{I_5} G_{J_1 J_2 \dots J_5} \times_1 A_{j_1 J_1}^{(1)} \times_2 \dots \times_5 A_{j_5 J_5}^{(5)} \quad (3)$$

其中，$G \in \mathbb{R}^{J_1 \times J_2 \times J_3 \times J_4 \times J_5}$ 是核心张量，用于捕捉各个组件之间的相互作用，$A^{(n)} \in \mathbb{R}^{I_n \times J_n}$ 是与每个模式 $n$ 相关联的因子矩阵（即主成分），表示原始张量维度如何与低维核心张量相关。n 模乘法（$\times_n$ ）是张量代数中的一种运算，在特定模式（维度）下，通过矩阵乘以张量。它将矩阵-向量和矩阵-矩阵乘法推广到高阶张量。因此，$G$ 与 $A$ 的 n 模乘法记为 $G \times_n A$ 产生一个新张量 $W \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_{n-1} \times I_n \times J_{n+1} \times \dots \times J_N}$ ，并元素级地定义为：

$$(G \times_n A)_{J_1 \dots J_{n-1} I_n J_{n+1} \dots J_N} = \sum_{j_n=1}^{J_n} G_{j_1 j_2 \dots j_N} A_{i_n j_n} \quad (4)$$

HOSVD 等价于将 SVD 应用于通过展开（也称为矩阵化或扁平化）$T$ [4] 得到的每个因子矩阵 $A^n$ 。展开三阶张量的一个例子如图 2 （b）所示。

## 4. 提出的方法

为了训练如图 ?? 所示的 NLML-HPE 方法，首先需要用包含 $N_{id}$ 个主体（身份）的训练集的 $N$ 个样本 $x_n \in \mathbb{R}^{D_f}$ 填充一个五维张量 $T \in \mathbb{R}^{N_{id} \times D_y \times D_p \times D_r \times D_f}$ ，其中 $x_n$ 指的是一个用来表示特征（在我们的案例中为从人脸图像中提取的 3D 标志点）的 $D_f$ 维向量。这些 $x_n$ 是通过内在旋转从 facescape 数据集 [32] 中选择姿态的 300 个主体的 3D 模型生成的。因此，每个 $x_n$ 根据其身份和定义其相应旋转角度（即，偏航、俯仰和滚动）度数的 3 个值进行标记。这些旋转角可以分

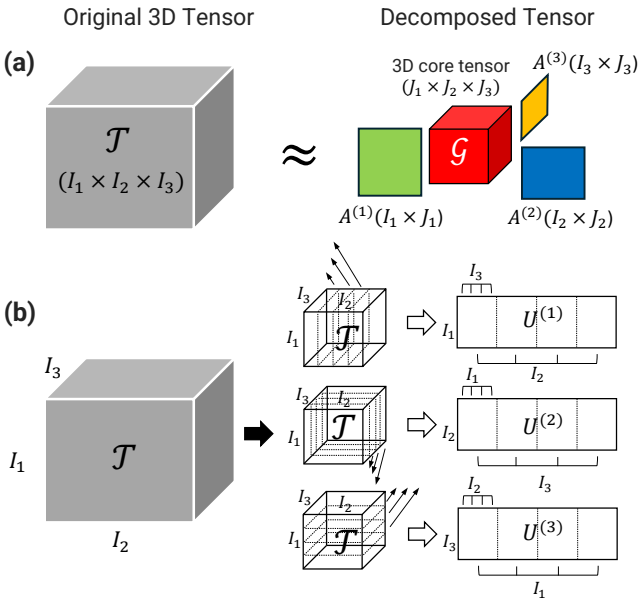


Figure 2. (a) 一个 3D 张量分解的描述。(b) 将 $(I_1 \times I_2 \times I_3)$ -张量 $T$ 展开为 $(I_1 \times I_2 I_3)$ -矩阵 $\boldsymbol{U}^{(1)}$ 、$(I_2 \times I_3 I_1)$ -矩阵 $\boldsymbol{U}^{(2)}$ 和 $(I_3 \times I_1 I_2)$ -矩阵 $\boldsymbol{U}^{(3)}$ $(I_1 = I_2 = I_3 = 4)$ 。

别离散化为 $D_y$ 、$D_p$ 和 $D_r$ 个箱。在填充张量时，我们通过首先计算它们的重心（$\boldsymbol{C}$ ），然后如公式 5 所示计算缩放因子（$s$ ），并将所有标志点除以该因子来标准化提取的标志点（$L$ ）。这确保了所有人脸图像具有相同的缩放和平移。

$$\begin{aligned} L &= \{(x_i, y_i, z_i)\}_{i=1}^n \\ \boldsymbol{C} &= \left( \frac{1}{n} \textstyle\sum_{i=1}^n x_i, \frac{1}{n} \sum_{i=1}^n y_i, \frac{1}{n} \sum_{i=1}^n z_i \right) \\ \|\boldsymbol{L}_i - \boldsymbol{C}\|^2 &= (x_i - C_x)^2 + (y_i - C_y)^2 + (z_i - C_z)^2 \\ s &= \sqrt{\frac{1}{n} \textstyle\sum_{i=1}^n \|\boldsymbol{L}_i - \boldsymbol{C}\|^2} \\ \hat{L}_i &= \left( \frac{x_i - C_x}{s}, \frac{y_i - C_y}{s}, \frac{z_i - C_z}{s} \right) \end{aligned} \quad (5)$$

一旦张量 $T$ 被填充，我们可以使用等式 3 对其进行分解，如下所示：

$$T = G \times_1 A^{(id)} \times_2 A^{(y)} \times_3 A^{(p)} \times_4 A^{(r)} \times_5 A^{(f)} \quad (6)$$

每个因子矩阵 $A^{(*)}$ 跨越对应给定因子的子空间。因此，它的行可以被解释为在该因子子空间内跨每个参数代表数据行为的向量。它的列可以被视为沿着张量对应模式（维度）的数据整体结构或模式的组成成分（或因子向量）。例如，矩阵 $A^{(id)}$ 的行标记为 $a^{(*)}$ ，捕捉到表征对象 $x_n \in \mathbb{R}^{D_f}$ 形状的独特特征。同时，矩阵

$A^{(y)}, A^{(p)}, A^{(r)}$ 中的每行包含确定对象绕每个轴按特定角度旋转的系数。由于 $G$ 与 $A^{(f)}$ 的 n 模乘积可以被解释为将形状和旋转后的对象映射到特征空间中，它们可以组合成一个辅助变量 $\boldsymbol{W} = G \times_1 A^{(f)}$ [28] 。

分解张量 $T$ 后，可以重构一个样本 $x$ ，表示为 $\bar{x}$ ，如下所示：

$$x \simeq \bar{x} = \boldsymbol{W} \times a^{(id)} \times a^{(y)} \times a^{(p)} \times a^{(r)} \tag{7}$$

其中 $\{ a^{(id)}, a^{(y)}, a^{(p)}, a^{(r)} \}$ 是来自因子矩阵 $A^{(id)}, A^{(y)}, A^{(p)}, A^{(r)}$ 的行向量。因此，给定一个测试样本 $x \in \mathbb{R}^{D_f}$ ，可以通过理论上最小化重构误差估计出旋转角度 [26] 。

$$\underset{a^{(id)}, a^{(y)}, a^{(p)}, a^{(r)}}{\operatorname{argmin}} \| x - \boldsymbol{W} \times a^{(id)} \times a^{(y)} \times a^{(p)} \times a^{(r)} \| \tag{8}$$

这是一个最小化问题，我们需要同时优化四个向量，以准确估计姿态估计中的偏航、俯仰和滚转。虽然存在各种方法 [3] 来解决此最小化问题，但实现精确结果具有挑战性，并且获得的解决方案往往不能保持不同子空间的流形结构。因此，必须对最小化结果进行约束，以与训练样本所固有定义的流形结构对齐。使用方程 6 来分解一个张量，我们得到三个不同的矩阵：$A^{(y)}$ 、$A^{(p)}$ 和 $A^{(r)}$ ，分别跨越绕偏航、俯仰和滚转的旋转子空间。根据 [9] ，如果我们绘制这些矩阵列的值，它们大致形成一个可以通过余弦函数很好逼近的螺旋曲线。因此，它们揭示了旋转子空间的系数遵循单峰流形结构。基于这一观察，可以对旋转系数应用显式约束，使得它们可以直接在潜在的旋转流形上联合估计。因此，方程 8 可以重写如下：

$$\begin{aligned} &\underset{\omega^{(id)}, \omega^{(y)}, \omega^{(p)}, \omega^{(r)}}{\operatorname{argmin}} \| x - \hat{x} \| \\ &\hat{x} = \boldsymbol{W} \times a^{(id)} \times f^{(y)}(\omega^{(y)}) \times f^{(p)}(\omega^{(p)}) \times f^{(r)}(\omega^{(r)}) \end{aligned} \tag{9}$$

，其中 $f^{(*)} : \mathbb{R} \to \mathbb{R}^{D_*}$ 是以角度 $\omega^{(*)}$ 为输入并输出系数向量 $a^{(*)}$ 的三角函数。$D_*$ 表示给定子空间的维度数，对应于每个因子矩阵中的列数。参数 $\omega^{(*)}$ 和 $a^{(*)}$ 都是需要优化的变量。因此，$f^{(*)}$ 将表示为一个由余弦参数化的实函数向量，如下所示：在其中，$\alpha_j^{(*)}, \beta_j^{(*)}, \gamma_j^{(*)}$ 和 $\varphi_j^{(*)}$ 是控制三角波行为的正弦参数。需要注意的是，对于每个旋转子空间，每一个子空间的维度都将关联一组独特的参数。这些正弦参数的值是通过傅里叶变换初始化的，并通过优化以下最小化问题获得：

其中，$a_{ij}^{(*)}$ 是由经过训练的张量分解得到的矩阵 $U^{(y)}, U^{(p)}, U^{(r)}$ 的元素。项 $\omega_i^{(*)}$ 表示的是用于构建张量的离散旋转角度的第 i 个区间对应的角度值。

求解方程 9 中的优化问题非常耗时，不适用于实时姿态估计。相反，我们训练了一个轻量级编码器，将展平的人脸标志点（1404 个特征）映射到 9 个目标变量。编码器的架构是一个全连接的前馈神经网络，由六个线性层组成，其大小逐渐减小：1024、512、256、128、64 和 9。在每个中间层之后应用 ReLU 激活函数，倒数第二层使用 Tanh 激活。9 个输出被分为三个集合，每个集合包含三个，分别代表预测的因子向量 $\hat{a}^{(y)}, \hat{a}^{(p)}, \hat{a}^{(r)}$ ，对应于偏航、俯仰和滚动。因子矩阵 $A^{(y)}, A^{(p)}, A^{(r)}$ 中的因子向量作为此映射的真实值。我们还训练了三个 MLP 头，每个头以其中一个因子向量为输入，预测相应的欧拉角。所有三个 MLP 头共享相同的架构。为了生成细粒度的因子向量（即，具有非常小角度间隔的向量），我们使用预训练的正弦参数 $(\alpha_j^{(*)}, \beta_j^{(*)}, \gamma_j^{(*)}, \varphi_j^{(*)})$ 。这些细粒度的因子向量作为训练 MLP 头的真实数据。

在测试过程中，编码器和 MLP 头按顺序串联成一个单一的模型，能够实现给定人脸图像的实时头部姿态预测。该方法相较于现有方法的一个关键优势在于其能够通过有限的训练数据学习从面部特征到姿态潜变量的映射。这得益于通过张量分解对旋转子空间的准确近似。因此，即使在小数据集的情况下，也可以训练一个紧凑而有效的编码器用于该映射。此外，与以往的方法不同，我们的模型能够提供实时的头部姿态估计。我们在 Core i7-13700K CPU 上评估了其计算效率，并与最先进方法进行了比较。正如表 2 所示，我们的方法每帧仅需 $1.38 \times 10^{-3}$ 秒来预测姿态，而 TokenHPE [35] 和 6DRepNet [14] 分别需要 $16.88 \times 10^{-3}$ 和 $17.59 \times 10^{-3}$ 秒/帧。

我们通过创建一个姿态一致的数据集开始我们的实验，其中旋转角度以 10 度为间隔进行离散化。具体来说，偏航角从 $-50°$ 到 $+50°$ ，俯仰角从 $-40°$ 到 $+40°$ ，滚转角从 $-30°$ 到 $+30°$ 。正如在 Sec. ?? 中讨论的，我们通过结合这些角度，利用 Facescape 数据集 [32] 中的 300 个对象的 3D 模型进行内部旋转来生成我们的数据集。我们定义了这些界限是由于 mediapipe 面部网格在极端姿态下检测面部标记的限制。这可以通过将给定的旋转角乘以正面面部的面部标记（即，当偏航、俯仰和滚动都为 $0°$ 时）来补偿。然而，我们的实验表明，这种乘法非常复杂，可能不会导致与 mediapipe 提取的标记姿态相同的结果。一个更好的替代方案可能是使用 transformers [29] 提取特征，并在张量中填充较少的角度界限限制。

在执行分解后，我们利用因子矩阵的前三个主要分量（即维度），记为 $a_j^{(y)}, a_j^{(p)}, a_j^{(r)}$ 至 $1 \leq j < D_*$ 来训练编码器。这些维度共同捕捉了大约 95 % 的总能量（数据方差的大部分），而其余的维度主要代表噪声。

为了降低张量分解的计算成本，我们将因子矩阵中的旋转角度间隔设为 10 度。然而，这种粗略的间隔可能无法提供足够的表示，来准确地将编码器的潜在姿态映射到 MLP 头中的相应欧拉角。为了解决这个问题而不改变张量分解，我们在方程 ?? 中拟合余弦函数到每个因子向量，优化正弦参数 $(\alpha_j^{(*)}, \beta_j^{(*)}, \gamma_j^{(*)}, \varphi_j^{(*)})$ 。利用这些参数，我们生成具有更细角度间隔（例如，0.01 度）的新因子矩阵。然后使用这些更精细的因子矩阵来高效训练 MLP 头。

### 4.1. 数据集和评估

我们通过在我们生成数据集的 70 % 上进行训练来进行实验，该数据集是使用 FaceScape 数据集 [32] 合成的。对于测试，我们使用两个流行的公共基准数据集：AFLW2000 [37] 和 BIWI [10]，以及我们生成数据集的剩余 30 %。

FaceScape 数据集 [32] 是一个大规模的 3D 人脸模型集合，旨在用于 3D 人脸重建及相关领域的研究。该数据集包括从 938 个主体捕获的 18,760 个带纹理的 3D 人脸模型，每个主体均展示 20 种特定的表情。

AFLW2000 数据集 [37] 数据集包含 AFLW 数据集 [18] 的前 2000 张图像。它具有多样化的面部外观和背景设置。BIWI 数据集 [10] 包含 20 个人（6 名女性和 14 名男性）的 15,678 张图像，其中 4 个人出现两次。头部姿势范围大约在偏航 ±75° 和俯仰 ±60° 之间。

为了与最新技术进行比较，我们采用欧拉角的平均绝对误差（MAE）和旋转矩阵向量的平均绝对误差（MAEV）作为评估指标，并采用与 TokenHPE [35] 和 6DRepNet [14] 中提到的相同测试设置。

我们在相同的实验条件下，使用 MAE 和 MAEV 作为度量，针对公共基准测试评估 NLML-HPE 与最先进方法相比的表现。对于早期模型，我们报告原论文中的 MAE，并在可用时报告 MAEV。MAEV 由 Cao 等人提出，旨在解决欧拉角的不连续性。表中展示了在 AFLW2000 和 BIWI 数据集上的结果。虽然我们的方法是在自定义数据集上训练的，但其他所有方法都在 300W-LP 数据集上训练。我们的方法在 MAE 和 MAEV 度量上显示出强大的泛化能力，始终实现具有竞争力的性能。在 AFLW2000 数据集上，NLML-HPE 实现了 3.08 的平均 MAE，优于 Dlib (14.19)、3DDFA (20.07)、HopeNet (6.19) 和 FSA-Net (5.36)，并在滚动估计方面表现出特别准确（明显优于 FSA-Net）且接近 TokenHPE。虽然 NLML-HPE 略落后于 TokenHPE (2.56) 和 6DRepNet (2.61)，但它在所有错误类型中保持了一致的性能。在基于向量的评估中，NLML-HPE 实现了 4.78 的 MAEV，优于 TriNet (6.31)、HopeNet (6.85) 和 FSA-Net (6.77)，并且与 TokenHPE (3.97) 和 6DRepNet (4.02) 保持竞争力，突显其在几何上一致且稳健的 3D 姿态估计能力。

在 BIWI 数据集上，NLML-HPE 保持了强劲表现，其 MAE 为 3.85，略低于 6DRepNet (3.75)，但领先于 TriNet (4.47) 约 13.8 %，并在大多数早期方法上表现出改善。与 TokenHPE (3.93) 相比，NLML-HPE 在俯仰角估计上表现更好（5.29 对 5.33），在偏航角精度上也具有竞争力（3.58 对 4.06），表明在各个角度上的预测能力平衡。在基于向量的评估中，NLML-HPE 获得 MAEV 为 6.00，超过了 TriNet (6.64) 和 FSA-Net (6.40)，并且接近 6DRepNet (5.65)。这些结果突显了 NLML-HPE 在姿势估计上的稳健性和一致性，尽管使用了更少的训练数据和更简单的架构，但在方向差异上较小且具有竞争力的准确性。

Table 2. 与我们验证集上 SOTA 方法在欧拉角平均绝对误差、向量误差和每帧计算时间的比较。Y: 偏航, P: 俯仰, R: 滚转, L: 左, D: 下, F: 前, TPF: 每帧时间。

| Method | Euler angles errors | | | | Vector errors | | | | TPF (ms) |
|---|---|---|---|---|---|---|---|---|---|
| | Y | P | R | MAE | L | D | F | MAEV | |
| TokenHPE [35] | 9.0 | 6.1 | 6.3 | 7.1 | 11.0 | 8.4 | 11.2 | 10.2 | 16.88 |
| 6DRepNet [14] | 28.0 | 13.6 | 6.3 | 16.0 | 30.2 | 13.8 | 32.3 | 25.4 | 17.59 |
| NLML-HPE (ours) | 3.4 | 3.5 | 2.4 | 3.1 | 4.4 | 4.5 | 5.0 | 4.6 | 1.16 |

将我们的模型与 SOTA 方法在验证集上的表现进行比较，如表格 2 所示，NLML-HPE 在欧拉角和向量误差指标上都优于这些方法，特别是在推广到未见数据的能力上。NLML-HPE 的 MAE 为 1.8，显著低于 TokenHPE 的 16.3 和 6DRepNet 的 29.0，突显出其在预测欧拉角方面的高精确度。这种表现也反映在 MAEV 中，NLML-HPE 获得了 3.0 的值，而 TokenHPE 和 6DRepNet 则分别为 25.7 和 37.9，展示了其精确估计 3D 姿势的强大能力。

虽然 TokenHPE 和 6DRepNet 在常用的数据集如 AFLW2000 和 BIWI 上表现良好，这些数据集的训练

Table 1. 在 AFLW2000 和 BIWI 数据集上，比较欧拉角和向量平均绝对误差的 SOTA 方法

| Method | Euler angles errors | | | | | | | | Vector errors | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AFLW2000 | | | | BIWI | | | | AFLW2000 | | | | BIWI | | | |
| | Yaw | Pitch | Roll | MAE | Yaw | Pitch | Roll | MAE | Left | Down | Front | MAEV | Left | Down | Front | MAEV |
| 3DDFA [36] | 4.71 | 27.08 | 28.43 | 20.07 | 5.50 | 41.90 | 13.22 | 20.20 | 30.57 | 39.05 | 18.52 | 29.38 | 23.31 | 45.00 | 35.12 | 34.47 |
| Dlib [17] | 8.50 | 11.25 | 22.83 | 14.19 | 11.86 | 13.00 | 19.56 | 14.80 | 26.56 | 28.51 | 14.31 | 23.12 | 24.84 | 21.70 | 14.30 | 20.28 |
| HopeNet [23] | 5.31 | 7.12 | 6.13 | 6.19 | 6.00 | 5.88 | 3.72 | 5.20 | 7.07 | 5.98 | 7.50 | 6.85 | 7.65 | 6.73 | 8.68 | 7.69 |
| FSA-Net [34] | 4.96 | 6.34 | 4.77 | 5.36 | 4.56 | 5.21 | 4.56 | 4.28 | 6.75 | 6.21 | 7.34 | 6.77 | 6.03 | 5.96 | 7.22 | 6.40 |
| QuatNet [15] | 3.97 | 5.62 | 3.92 | 4.50 | 2.94 | 5.49 | 4.01 | 4.15 | - | - | - | - | - | - | - | - |
| HPE [16] | 4.80 | 6.18 | 4.87 | 5.28 | 3.12 | 5.18 | 4.57 | 4.29 | - | - | - | - | - | - | - | - |
| TriNet [5] | 4.36 | 5.81 | 4.51 | 4.89 | 3.11 | 5.09 | 5.20 | 4.47 | 6.16 | 5.95 | 6.82 | 6.31 | 6.58 | 5.80 | 7.55 | 6.64 |
| TokenHPE [35] | 2.68 | 3.41 | 1.59 | 2.56 | 4.06 | 5.33 | 2.41 | 3.93 | 3.38 | 3.90 | 4.63 | 3.97 | 5.21 | 5.71 | 7.06 | 6.00 |
| 6DRepNet [14] | 2.79 | 3.39 | 1.65 | 2.61 | 3.43 | 5.22 | 2.61 | 3.75 | 3.47 | 3.87 | 4.71 | 4.02 | 4.77 | 5.72 | 6.48 | 5.65 |
| NLML-HPE (ours) | 3.06 | 4.23 | 1.96 | 3.08 | 3.58 | 5.29 | 2.67 | 3.85 | 4.02 | 4.77 | 5.53 | 4.78 | 5.34 | 6.03 | 6.63 | 6.00 |

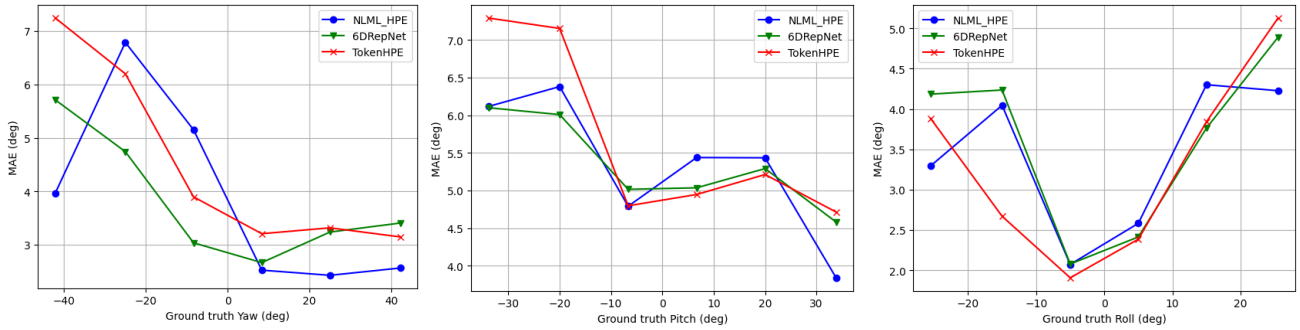


Figure 3. BIWI 数据集上角度区间的误差分析

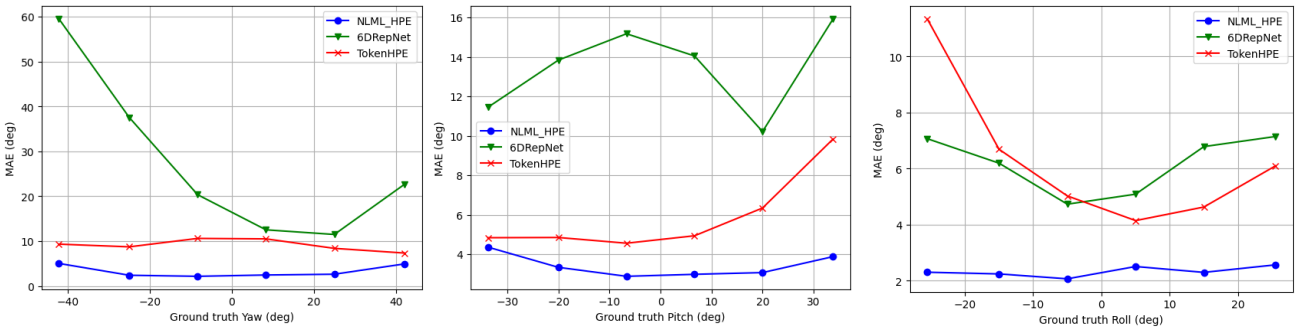


Figure 4. 验证集上角度区间的误差分析

和验证数据之间通常共享模式，但它们在有效泛化到多样化数据方面表现挣扎，而 NLML-HPE 有效地克服了这一挑战，使其更适合实际场景。我们对我们提出的 NLML-HPE 方法及 SOTA、TokenHPE 和 6dRepNet 在 BIWI 数据集和我们的验证集上的表现进行了误差分析。这些结果分别在图 3 和图 4 中进行了说明。在这些图中，偏航角范围 $[-50°, +50°]$ 被均匀地分为 16.67° 的区间，俯仰角范围 $[-40°, +40°]$ 被分为 13.33° 的区间，滚转角范围 $[-30°, +30°]$ 被分为 10° 的区间。

观察这些图表，第一个显著问题是 6dRepNet 和 TokenHPE 在应用到我们生成的数据集时的泛化问题，这从图 4 中的高预测误差可以看出。虽然 TokenHPE 报告的俯仰角和偏航角的平均绝对误差（MAE）最接近我们的结果，且与中心的偏差较小，但对于其他角度尤其是极端姿态时，误差显著增加。相比之下，我们的方法，即使在一个有限的自定义生成集上训练，在所有欧拉角度上都实现了近乎一致的 MAE，如同图 3 所示，甚至在极端姿态下以更低的 MAE 超过了一些状态的方法。这表明我们的 NLML-HPE 方法更适合真实世界的场景，因为它结合了轻量级架构并在未见数据上具有较低的预测误差。

如在第 ?? 节所讨论的，我们的实验在偏航范围为 $[-50°, +50°]$ 、俯仰在 $[-40°, +40°]$ 、以及滚转在 $[-30°, +30°]$ 的情况下进行。由于我们的特征提取器的限制，它在很大偏离正面姿态的情况下难以准确捕捉面部标志，因此我们限制了预测范围。然而，这种限制可以通过使用一种能够在极端姿态中提取特征的更准确的特征提取器来克服。

## 5. 结论

在本文中，我们提出了一种基于非线性流形学习的新颖头部姿态估计算法。该方法将三维空间中的姿态估计问题映射为学习独立流形的底层结构的任务。我们通过应用多线性分解，将姿态变化的因素分离到不同的子空间中，每个子空间分别对应于偏航、俯仰和滚动的效果。基于这些子空间定义为一个连续曲线的观察结果，我们使用三角函数对其进行建模。通过最小化该函数，我们得到的解总是与旋转参数的变化一致。随后，我们训练一个深度编码器和三个 MLP 头，使用这些解作为真实值进行实时姿态估计。实验结果表明，尽管 NLML-HPE 仅在一个有限的、定制生成的数据集上进行了训练，它在常用数据集上仍然达到了接近于最新技术水平的表现。在未来工作中，我们计划扩展我们的方法，使用能够捕捉所有旋转角度下面部特征的更准确的特征提取器。

本研究由欧盟资助（GA 101119800 - EMERALD）。

## References

[1] A. F. Abate, C. Bisogni, A. Castiglione, and M. Nappi. Head pose estimation: An extensive survey on recent techniques and applications. Pattern Recognition, 127:108591, 2022.

[2] A. Asperti and D. Filippini. Deep learning for head pose estimation: A survey. SN Computer Science, 4(4):349, 2023.

[3] A. Bakry and A. Elgammal. Untangling object-view manifold for multiview recognition and pose estimation. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13, pages 434–449. Springer, 2014.

[4] G. Bergqvist and E. G. Larsson. The higher-order singular value decomposition: Theory and an application [lecture notes]. IEEE signal processing magazine, 27(3):151–154, 2010.

[5] Z. Cao, Z. Chu, D. Liu, and Y. Chen. A vector-based representation to enhance head pose estimation. In Proceedings of the IEEE/CVF Winter Conference on applications of computer vision, pages 1188–1197, 2021.

[6] J. Chen, B. Li, and B. Yuan. Face recognition using direct lpp algorithm. In 2008 9th International Conference on Signal Processing, pages 1457–1460. IEEE, 2008.

[7] P. Comon. Tensors: a brief introduction. IEEE Signal Processing Magazine, 31(3):44–53, 2014.

[8] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. SIAM journal on Matrix Analysis and Applications, 21(4):1253–1278, 2000.

[9] D. Derkach, A. Ruiz, and F. M. Sukno. Tensor decomposition and non-linear manifold modeling for 3d head pose estimation. International Journal of Computer Vision, 127(10):1565–1585, 2019.

[10] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool. Random forests for real time 3d face analysis. International journal of computer vision, 101:437–458, 2013.

[11] Y. Fu and T. S. Huang. Graph embedded analysis for head pose estimation. In 7th International Conference on Automatic Face and Gesture Recognition (FGR06), pages 6–pp. IEEE, 2006.

[12] M. Ghafourian, J. Fierrez, L. F. Gomez, R. Vera-Rodriguez, A. Morales, Z. Rezgui, and R. Veldhuis. Toward face biometric de-identification using adversarial examples. In 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), pages 723–728. IEEE, 2023.

[13] M. Ghafourian, J. Fierrez, R. Vera-Rodriguez, A. Morales, and I. Serna. Otb-morph: One-time biometrics via morphing. Machine Intelligence Research, 20(6):855–871, 2023.

[14] T. Hempel, A. A. Abdelrahman, and A. Al-Hamadi. Toward robust and unconstrained full range of rotation head pose estimation. IEEE Transactions on Image Processing, 33:2377–2387, 2024.

[15] H.-W. Hsu, T.-Y. Wu, S. Wan, W. H. Wong, and C.-Y. Lee. Quatnet: Quaternion-based head pose estimation with multiregression loss. IEEE Transactions on Multimedia, 21(4):1035–1046, 2018.

[16] B. Huang, R. Chen, W. Xu, and Q. Zhou. Improving head pose estimation using two-stage ensembles with top-k regression. Image and Vision Computing, 93:103827, 2020.

[17] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1867–1874, 2014.

[18] K. Khan, R. U. Khan, R. Leonardi, P. Migliorati, and S. Benini. Head pose estimation: A survey of the last ten years. Signal Processing: Image Communication, 99:116479, 2021.

[19] G.-F. Lu, Z. Lin, and Z. Jin. Face recognition using regularised generalised discriminant locality preserving projections. IET computer vision, 5(2):107–116, 2011.

[20] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Face recognition using lda-based algorithms. IEEE Transactions on Neural networks, 14(1):195–200, 2003.

[21] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, et al. Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172, 2019.

[22] B. Raytchev, I. Yoda, and K. Sakaue. Head pose estimation by nonlinear manifold learning. In Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., volume 4, pages 462–466. IEEE, 2004.

[23] N. Ruiz, E. Chong, and J. M. Rehg. Fine-grained head pose estimation without keypoints. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 2074–2083, 2018.

[24] K. R. Soundar and K. Murugesan. Preserving global and local information–a combined approach for recognising face images. IET computer vision, 4(3):173–182, 2010.

[25] H. M. Takallou and S. Kasaei. Head pose estimation and face recognition using a non-linear tensor-based model. IET Computer Vision, 8(1):54–65, 2014.

[26] J. Tenenbaum and W. Freeman. Separating style and content. Advances in neural information processing systems, 9, 1996.

[27] M. A. Turk, A. Pentland, et al. Face recognition using eigenfaces. In CVPR, volume 91, pages 586–591, 1991.

[28] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In Computer Vision—ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part I 7, pages 447–460. Springer, 2002.

[29] A. Vaswani. Attention is all you need. Advances in Neural Information Processing Systems, 2017.

[30] M. Wan, G. Yang, Z. Lai, and Z. Jin. Feature extraction based on fuzzy local discriminant embedding with applications to face recognition. IET computer vision, 5(5):301–308, 2011.

[31] M. Wang, Y. Panagakis, P. Snape, and S. Zafeiriou. Learning the multilinear structure of visual data. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4592–4600, 2017.

[32] H. Yang, H. Zhu, Y. Wang, M. Huang, Q. Shen, R. Yang, and X. Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In Proceedings of the ieee/cvf conference on computer vision and pattern recognition, pages 601–610, 2020.

[33] J. Yang, D. Zhang, A. F. Frangi, and J.-y. Yang. Two-dimensional pca: a new approach to appearance-based face representation and recognition. IEEE transactions on pattern analysis and machine intelligence, 26(1):131–137, 2004.

[34] T.-Y. Yang, Y.-T. Chen, Y.-Y. Lin, and Y.-Y. Chuang. Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1087–1096, 2019.

[35] C. Zhang, H. Liu, Y. Deng, B. Xie, and Y. Li. Tokenhpe: Learning orientation tokens for efficient head pose estimation via transformers. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8897–8906, 2023.

[36] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 146–155, 2016.

[37] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li. High-fidelity pose and expression normalization for face recognition in the wild. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 787–796, 2015.