

手机点胶三维轨迹生成方法

刘洋, 史金龙, 钱强, 欧镇, 白素琴

(江苏科技大学计算机学院, 江苏 镇江 212000)

摘要: 智能手机边框在制造过程中往往存在形变问题, 先前的视觉定位方法难以得到精确的边框点胶位置。针对此问题, 提出了一种新颖的手机点胶三维轨迹生成方法。首先, 使用三维线激光测量仪扫描边框, 引入基于位姿图的点云拼接算法解决边框点云的拼接问题。采用基于模板的方法, 标注模板点胶轨迹。然后, 通过基于变形图的非刚性配准算法将模板与目标边框对齐, 得到变换后的轨迹, 并提出轨迹优化策略生成精准的目标点胶轨迹。实验证明, 所提方法生成的轨迹精度可达 0.01 mm 以下, 且速度可达到在线实时测量效果。

关键词: 三维测量; 点胶轨迹; 位姿图; 非刚性配准; 轨迹优化

中图分类号: TP391

文献标志码: A

doi: 10.11959/j.issn.2096-6652.202407

Three-dimensional trajectory generation method for mobile phone dispensing

LIU Yang, SHI Jinlong, QIAN Qiang, OU Zhen, BAI Suqin

School of Computer Science and Technology, Jiangsu University of Science and Technology, Zhenjiang 212000, China

Abstract: Due to the potential issue of non-rigid deformations in the manufacturing process of mobile phone frames, most existing dispensing automation methods struggled to acquire accurate dispensing positions. For this purpose, an innovative approach of dispensing trajectory generation was proposed, which addressed the limitations associated with previous methods. Firstly, the phone frames were scanned using a 3D laser profiler, then a point cloud registration algorithm based on the pose graph was introduced to reconstruct the point clouds of frames. The template-based approach was employed, and the dispensing trajectory for template was manually annotated. Subsequently, the template was aligned with the target by a non-rigid registration algorithm based on the deformation graph, thereby the transformed trajectory was obtained. Furthermore, a trajectory refinement strategy was presented to generate the robust, accurate dispensing trajectory of the target. Extensive experiments demonstrate that the errors of the generated dispensing trajectories were consistently below 0.01 mm, and the speed can meet the real-time requirement of practical applications.

Key words: 3D measurement, dispensing trajectory, pose graph, non-rigid registration, trajectory refinement

0 引言

近年来, 智能手机厂商追逐于更加复杂化、多样化的外观, 这对手机制造工艺提出了更高的要

求。其中, 点胶封装作为保护和固定手机内部元件的重要环节, 其质量和效率直接影响着手机的整体可靠性。传统点胶封装技术存在诸多问题, 如低生产效率、不稳定的胶水用量、不均匀的胶水分布

收稿日期: 2023-12-25; 修回日期: 2024-03-02

通信作者: 钱强, qianqiang_just@just.edu.cn

基金项目: 国家自然科学基金项目 (No.51875270); 民航智慧机场理论与系统重点实验室开放基金 (No.SATS202207)

Foundation Items: The National Natural Science Foundation of China (No.51875270), Key Laboratory of Smart Airport Theory and System (No.SATS202207)

等, 这些问题在大规模生产中尤为突出^[1]。因此, 设计一种精准、鲁棒、高效的点胶自动化方法成为手机产品制造行业面临的挑战。

实现点胶自动化最重要的一点就是得到手机边框的点胶轨迹信息^[2], 包括每个点胶位置的坐标及法线, 从而引导点胶设备的机械臂^[3]移动到指定位置进行打胶, 完成点胶封装操作。

吴佳毅等^[4]使用二维相机拍摄图像, 通过标志点匹配方法将图像与模板进行校正, 根据偏差值对模拟点胶路径进行补偿, 从而得到点胶轨迹。但二维图像容易受光照影响, 缺少深度信息。贺昌兵^[5]通过三维线激光轮廓传感器测量手机边框得到点云, 根据点云投影获取边缘轮廓, 将其偏移固定距离再反投影得到三维点胶轨迹。但是, 根据点云轮廓获取点胶轨迹的方法不具备泛化性, 不适用于复杂的手机边框。陈鸿雨^[6]选用三维线激光相机测量手机边框, 提出基于多圆孔标定块的拼接算法完成点云拼接, 使用基于 3DNPFH (3D neighborhood point feature histogram)^[7]特征描述符的配准方法对模板与目标点云进行粗配准, 再通过 ICP (iterative closest point)^[8]算法^[8]精配准, 求出点胶轨迹。但该方法仅使用刚性配准算法, 实际上产品制造过程中总会存在些许瑕疵, 即使同批次的手机产品也会产生非刚性形变。因此, 手机点云很难完全对齐到目标点云, 获取到精密的点胶轨迹。

针对当前手机点胶方法由于边框微小形变而难以获取准确点胶轨迹的问题, 本文提出一种新颖的手机点胶三维轨迹生成方法, 主要工作如下。

(1) 结合基于位姿图的点云拼接算法和基于模

板变形图的非刚性配准算法, 提出一种精准高效的手机点胶轨迹生成框架。

(2) 为了提高点胶轨迹的鲁棒性, 提出了一种轨迹优化策略, 便于精准定位点胶位置。

(3) 使用三维线激光测量仪制作了多组手机边框的点云数据集, 并通过大量实验证明了本文方法的高效性。

(4) 开发了一种离线在线结合的点胶操作软件, 能够在线实时生成手机边框的点胶轨迹。

1 方法流程

本文采用基于模板的方法^[6], 系统流程如图1所示, 主要包括数据测量、模板标注和轨迹生成3个模块。

数据测量: 首先, 采用线激光测量仪对手机边框的4条边扫描, 得到4个边框点云; 接着, 基于位姿图的点云拼接算法求解出拼接矩阵, 将4个边框点云拼接为完整点云; 最后, 通过点云裁切、点云去噪和三角化等操作得到手机边框的 Mesh 模型。

模板标注: 首先, 从同批次手机边框中选取一组作为模板, 并在操作软件中读取模板的激光扫描数据; 接着, 当软件逐条显示激光线扫描的形状时, 在线条的凹槽位置处点击标注; 通过标注出所有激光线的点胶位置, 得到模板点胶轨迹。

轨迹生成: 首先, 根据模板 Mesh 模型与点胶轨迹构造变形图; 接着, 基于变形图的非刚性配准算法, 求解模板对齐到目标的变换关系, 计算模板轨迹变换后的位置, 并通过轨迹优化策略求出精准的目标点胶轨迹。

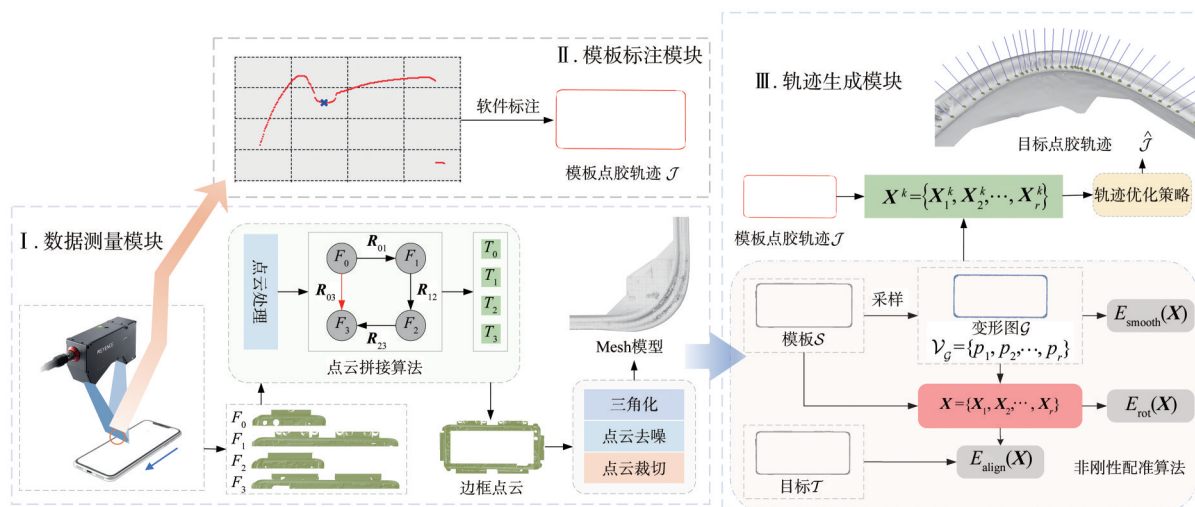


图1 系统流程

2 数据测量模块

2.1 点云拼接算法

线激光测量仪一次只能扫描手机边框的1条边,依次扫描4条边后,即可得到边框的4个点云数据 $\{F_i, i=0,1,2,3\}$;然后根据测量装置运动参数可得到4个点云的初始全局位姿 $\{T_i \in \mathbf{SE}(3), i=0,1,2,3\}$ 。初始位姿下的边框点云如图2所示,每个边框点云可分为1个直边区域和2个 R 角区域,初始位姿下相邻边框的 R 角区域相互重叠但并未完全对齐,因此还需对其进行位姿优化。

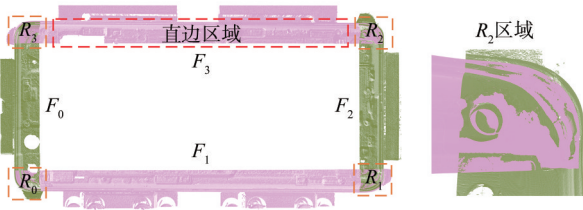


图2 初始位姿下的边框点云

为了提高点云配准的速率,对边框点云 $\{F_i\}$ 中的无关点和噪声点进行处理。首先,通过尺寸为 s_a 的体素滤波器对点云下采样;然后,以半径为 r_a 且点数阈值为 n_a 的半径滤波器去除噪声点。由于点云配准主要依赖于点云之间的部分重叠区域,即图2所示的4个 R 角区域。而边框点云中存在着大面积非重叠区域,即直边区域,通过下述操作去除这些无关点。例如对于边框 F_2 ,根据相邻边框点云 F_3 的包围框信息 $\{(x_{\min}, x_{\max}), (y_{\min}, y_{\max}), (z_{\min}, z_{\max})\}$,即可通过直通滤波确定 F_2 在 R_2 区域的部分点云。同理也可得到 F_2 在 R_1 区域的部分点云,将两部分点云作为整体,提高了点云重叠率,有利于提升配准精度。

接着,对处理后的点云 $\{F_i\}$ 进行两两配准。然而,由于漂移问题^[9],两两配准后 F_0 和 F_3 的对齐效果很差。为使边框点云在全局空间下能够相互对齐,保持高度一致性,采用位姿图优化算法进行点云拼接。如图1所示,位姿图^[10]基于图结构实现:节点由4个点云 $\{F_i\}$ 组成,每个节点存储着 F_i 的全局位姿 T_i ;两个相邻的节点 F_i 和 F_{i+1} 相连形成边,表示节点之间的约束关系。构造 F_i 和 F_{i+1} 的对应点对,通过ICP算法^[8]最小二乘线性求解 $R_{i(i+1)}$,

表示 F_i 对齐到 F_{i+1} 的相对位姿。由于边框点云首尾相交, F_0 和 F_3 也相互重叠,将其相连作为回环边,同理计算出相对位姿 R_{03} 。若把 F_0 的空间作为全局空间,则 T_0 初始化为单位矩阵,其他节点的位姿 T_i 可由相对位姿多次累积变换完成初始化。

为求解全局最优位姿 $\mathcal{T} = \{T_i, i=0,1,2,3\}$,构造损失函数 $E_{\text{reg}}(\mathcal{T})$,如式(1)所示。

$$E_{\text{reg}}(\mathcal{T}) = \sum_{i=0,1,2} f(T_i, T_{i+1}, R_{i(i+1)}) + f(T_0, T_3, R_{03}) \quad (1)$$

其中,每一项 $f(T_i, T_j, R_{ij})$ 表示全局空间中 F_i 与 F_j 的对齐损失,通过KDTree构造对应点集 $\mathcal{K}_{ij}$,计算距离误差进行表示,如式(2)所示。

$$\begin{aligned} f(T_i, T_j, R_{ij}) &= \sum_{(p,q) \in \mathcal{K}_{ij}} \|T_i p - T_j q\|^2 \approx \\ &= \sum_{(p,q) \in \mathcal{K}_{ij}} \|T_i p - T_j R_{ij} p\|^2 = \\ &= \sum_{(p,q) \in \mathcal{K}_{ij}} \|R_{ij}^{-1} T_j^{-1} T_i p - p\|^2 \end{aligned} \quad (2)$$

其中,将 F_i 与 F_j 的损失近似为 F_i 与 $R_{ij} F_i$ 的损失,保持相对位姿 R_{ij} 来优化全局位姿 T_i, T_j 。当 $T_j^{-1} T_i \approx R_{ij}$ 时,为最优解。

本文通过g2o优化库中的Levenberg-Marquardt求解器^[11]优化损失函数 $E_{\text{reg}}(\mathcal{T})$,得到最优拼接矩阵 $\mathcal{T}$ 。将其应用于 $\{F_i, i=0,1,2,3\}$,拼接即可得到完整的边框点云。

2.2 点云预处理

为了便于在非刚性配准算法中构造变形图,本文对拼接后的边框点云进行预处理,具体操作如图3所示,其中“#v”表示点云个数或Mesh的顶点个数,“#f”表示Mesh的面片个数。首先,为了去除测量过程中扫描到的无关区域,如平台的固定零件,通过多次直通滤波对点云进行裁切去除无关点,仅保留手机边框部分。然后,通过尺寸为 s_b 的体素滤波器对点云下采样;采用半径为 r_b 且点数阈值为 n_b 的半径滤波器去除离群点;对于难以去除的噪声,通过基于密度的DBSCAN算法^[12]对点云进行聚类,剔除小类别点云,即可完成去噪。经过上述操作,点云数目减少了约99.58%,大大加快了配准效率。最后,把点云的三维点投影到 $Z=0$ 平面得到二维点,通过Delaunay^[13]算法将点

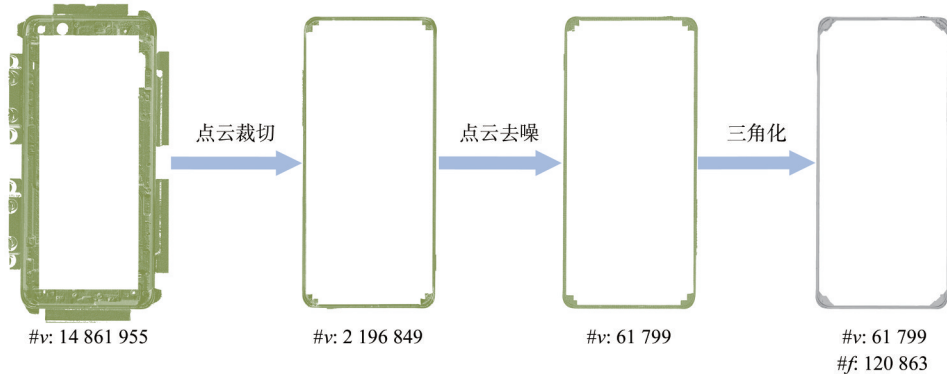


图3 边框点云预处理

连接起来生成三角网格，并映射到三维点云中，即可得到点云的Mesh模型。

3 轨迹生成模块

将模板Mesh与目标Mesh模型分别命名为 $\mathcal{S}$ 和 $\mathcal{T}$ ，其中 $\mathcal{S}$ 的顶点 $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \in \mathbb{R}^3\}$ ， $\mathcal{T}$ 的顶点 $\mathcal{U} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m \in \mathbb{R}^3\}$ 。 $\mathcal{J} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k\} \in \mathcal{V}$ ，表示所标注的模板点胶轨迹。目的是通过配准算法使 $\mathcal{S}$ 对齐到 $\mathcal{T}$ ，求出目标点胶轨迹 $\hat{\mathcal{J}}$ 。

3.1 构造变形图

对于模板 $\mathcal{S}$ ，构造变形图 $\mathcal{G} = \{\mathcal{V}_{\mathcal{G}}, \mathcal{E}_{\mathcal{G}}\}$ 表示模板顶点 $\mathcal{V}$ 的变换关系，其中 $\mathcal{V}_{\mathcal{G}} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_r\} \in \mathcal{V}$ 表示从 $\mathcal{V}$ 中采样的子集， $\mathcal{E}_{\mathcal{G}}$ 表示连接相邻节点的边。具体构造方法如下：首先， $\mathcal{V}_{\mathcal{G}}$ 初始化为空；接着，将 $\mathcal{V}$ 中所有点按照坐标轴进行排序，并优先排序模板轨迹 $\mathcal{J}$ 中点的序列位置；然后按序遍历所有点 $\mathbf{v}_i$ ，如果 $\mathcal{V}_{\mathcal{G}}$ 为空或者 $\mathbf{v}_i$ 到 $\mathcal{V}_{\mathcal{G}}$ 中所有点的最短测地距离不小于阈值 λ ，则将 $\mathbf{v}_i$ 添加到 $\mathcal{V}_{\mathcal{G}}$ 中；遍历完所有点后，即可得到变形图的点集 $\mathcal{V}_{\mathcal{G}}$ ；最后，将所有相邻节点 $\mathbf{p}_i$ 、 $\mathbf{p}_j$ 用边连接起来，得到边集 $\mathcal{E}_{\mathcal{G}}$ 。其中测地距离使用Fast marching算法^[14]进行计算。图4为变形图 $\mathcal{G}$ 的采样示意图，图中灰色网格模型即模板 $\mathcal{S}$ ，蓝色点为采样的点集 $\mathcal{V}_{\mathcal{G}}$ ，红色线条连接 $\mathcal{G}$ 中的相邻节点构成边集 $\mathcal{E}_{\mathcal{G}}$ 。

对于 $\mathcal{G}$ 中的每个节点 $\mathbf{p}_j$ ，定义一个仿射变换矩阵 $\mathbf{X}_j = (\mathbf{A}_j, \mathbf{t}_j)$ 。若模板 $\mathcal{S}$ 某一顶点 $\mathbf{v}_i$ 到 $\mathbf{p}_j$ 的测地距离 $D(\mathbf{v}_i, \mathbf{p}_j) < \lambda$ ，则可以计算出 $\mathbf{v}_i$ 受到 $\mathbf{X}_j$ 影响后的位置 $\mathbf{A}_j(\mathbf{v}_i - \mathbf{p}_j) + \mathbf{p}_j + \mathbf{t}_j$ 。对于 $\mathcal{V}$ 的每个顶点 $\mathbf{v}_i$ ，一般会受到多个节点的影响，通过式(3)计算 $\mathbf{v}_i$ 受

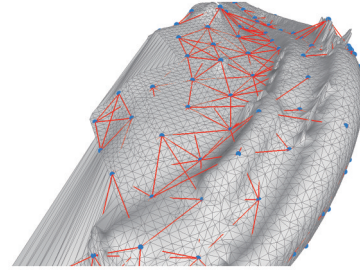


图4 变形图采样示意图

到多个节点共同影响下的变换后位置 $\hat{\mathbf{v}}_i$ ，如式(3)所示。

$$\hat{\mathbf{v}}_i = \sum_{\mathbf{p}_j \in \mathcal{L}(\mathbf{v}_i)} \omega_{ij} \cdot (\mathbf{A}_j(\mathbf{v}_i - \mathbf{p}_j) + \mathbf{p}_j + \mathbf{t}_j) \quad (3)$$

其中， $\mathcal{L}(\mathbf{v}_i) = \{\mathbf{p}_j | D(\mathbf{v}_i, \mathbf{p}_j) < \lambda\}$ 表示影响 $\mathbf{v}_i$ 的节点集合； ω_{ij} 表示与测地距离有关的权重系数，如式(4)所示。

$$\omega_{ij} = \frac{(1 - D^2(\mathbf{v}_i, \mathbf{p}_j)/\lambda^2)^3}{\sum_{\mathbf{p}_k \in \mathcal{L}(\mathbf{v}_i)} (1 - D^2(\mathbf{v}_i, \mathbf{p}_k)/\lambda^2)^3} \quad (4)$$

3.2 目标函数

令 $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_r\}$ 表示 $\mathcal{G}$ 中所有节点仿射变换的集合，通过对齐项、平滑项和旋转项构造目标函数 $E(\mathbf{X})$ ，联合优化求解 $\mathbf{X}$ ，如式(5)所示。

$$E(\mathbf{X}) = E_{\text{align}}(\mathbf{X}) + E_{\text{smooth}}(\mathbf{X}) + E_{\text{rot}}(\mathbf{X}) \quad (5)$$

$E_{\text{align}}(\mathbf{X})$ 为对齐项，表示模板 $\mathcal{S}$ 与目标 $\mathcal{T}$ 的对齐误差。对于每个变换后的顶点 $\hat{\mathbf{v}}_i$ ，通过KDTTree搜索 $\mathcal{T}$ 上的最近点 $\mathbf{u}_{\rho(i)} \in \mathcal{U}$ 形成一组对应点，通过式(6)计算距离误差得到 $E_{\text{align}}(\mathbf{X})$ ：

$$E_{\text{align}}(\mathbf{X}) = \sum_{i=1}^n \psi_{\delta_a}(\|\hat{\mathbf{v}}_i - \mathbf{u}_{\rho(i)}\|) \quad (6)$$

其中, $\psi_{\delta_a}(\cdot)$ 为 Welsch 函数^[15], 如式 (7) 所示, $\delta_a > 0$ 为自定义参数, 图 5 中红色线条为 $\delta_a = 1$ 时的函数图像。

$$\psi_{\delta_a}(x) = 1 - \exp\left(-\frac{x^2}{2\delta_a^2}\right) \quad (7)$$

相较于 L-2 损失, 使用 Welsch 函数对于噪声不会太敏感, 提高了鲁棒性, 并加快了收敛速度^[16]。

$E_{\text{smooth}}(\mathbf{X})$ 为平滑项, 保证模板 $\mathcal{S}$ 变换的平滑性。通常 $\mathcal{G}$ 上两个相邻的节点 $\mathbf{p}_i$ 、 $\mathbf{p}_j$ 在所影响共同区域内的变换应保持一致性。通过式 (8) 计算 $\mathbf{p}_i$ 、 $\mathbf{p}_j$ 的仿射变换对 $\mathbf{p}_i$ 的影响差异 $\mathbf{H}_{ij}$, 并进一步计算得到 $E_{\text{smooth}}(\mathbf{X})$, 如式 (9) 所示。

$$\mathbf{H}_{ij} = \mathbf{A}_j(\mathbf{p}_i - \mathbf{p}_j) + \mathbf{p}_j + \mathbf{t}_j - (\mathbf{p}_i + \mathbf{t}_i) \quad (8)$$

$$E_{\text{smooth}}(\mathbf{X}) = \sum_{i=1}^r \sum_{\mathbf{p}_j \in \mathcal{N}(\mathbf{p}_i)} \psi_{\delta_s}(\|\mathbf{H}_{ij}\|) \quad (9)$$

其中, r 表示节点的数目; $\mathcal{N}(\mathbf{p}_i)$ 表示 $\mathcal{G}$ 中 $\mathbf{p}_i$ 的邻域节点集合; $\delta_s > 0$ 是 Welsch 函数参数。

$E_{\text{rot}}(\mathbf{X})$ 为旋转项, 使矩阵 $\mathbf{A}_i$ 满足旋转矩阵的特性, 保证配准过程中表面的局部变换为刚体变换。用 $\mathbf{A}_i$ 与其在特殊正交群 $\mathcal{R} = \text{SO}(3)$ 中投影的偏差来表示旋转项损失, 如式 (10) 所示。

$$E_{\text{rot}}(\mathbf{X}) = \sum_{i=1}^r \|\mathbf{A}_i - \text{proj}_{\mathcal{R}}(\mathbf{A}_i)\|_F^2 \quad (10)$$

其中, $\text{proj}_{\mathcal{R}}(\cdot)$ 为在 $\mathcal{R}$ 上的投影运算; $\|\cdot\|_F$ 为矩阵的 F-范数。

3.3 替代函数

由于上述 $E_{\text{align}}(\mathbf{X})$ 和 $E_{\text{smooth}}(\mathbf{X})$ 是非线性且非凸性的, 很难直接对 $E(\mathbf{X})$ 进行优化。因此, 本文根据主化最小化 (majorization-minimization, MM) 算法^[17]思想, 每次迭代过程中构造替代函数并进行优化, 从而使目标函数收敛。如第 k 次迭代过程中, 在 $\mathbf{X} = \mathbf{X}^{(k)}$ 初始值处构造出 $E(\mathbf{X})$ 的替代函数 $\bar{E}(\mathbf{X})^{(k)}$, 对 $\bar{E}(\mathbf{X})^{(k)}$ 最小化求解 $\mathbf{X}$; 再将 $\mathbf{X}$ 作为 $k+1$ 次迭代的初始值 $\mathbf{X}^{(k+1)} = \min_{\mathbf{X}} \bar{E}(\mathbf{X})^{(k)}$, 继续构造 $k+1$ 次迭代的替代函数; 不断迭代优化 $\mathbf{X}$, 直到满足 $E(\mathbf{X})$ 的收敛条件。

本文根据文献[16]构造了 Welsch 函数的替代函数, 如式 (11) 所示。

$$\bar{\psi}_{\delta}^y(x) = \psi_{\delta}(y) + \left(\frac{1 - \psi_{\delta}(y)}{2\delta^2}\right)(x^2 - y^2) \quad (11)$$

$\bar{\psi}_{\delta}^y(x)$ 表示在初始值为 y 时的替代函数, 是一个凸性的二次函数, 当 $x=y$ 时, $\bar{\psi}_{\delta}^y(y) = \psi_{\delta}(y)$ 。Welsch 函数与替代函数如图 5 所示, $\psi_1(x)$ 为 $\delta=1$ 的 Welsch 函数图像; $\bar{\psi}_1^1(x)$ 是在初始值 $x=1$ 时的替代函数。

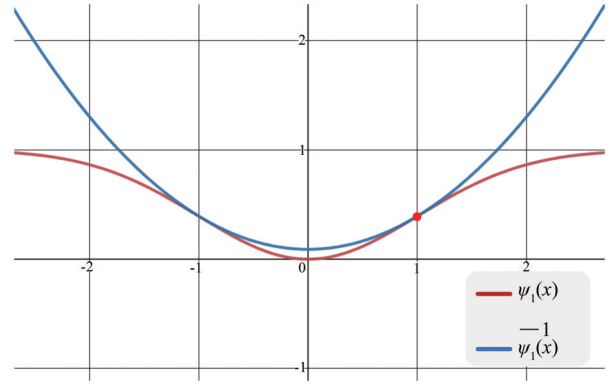


图5 Welsch函数与替代函数

将 $\bar{\psi}_{\delta}^y(x)$ 替换到对齐项式 (6) 中, 忽略其中的常数项, 得到 $E_{\text{align}}(\mathbf{X})$ 在第 k 次迭代的替代函数 $\bar{E}_{\text{align}}(\mathbf{X})^{(k)}$, 如式 (12) 所示。

$$\bar{E}_{\text{align}}(\mathbf{X})^{(k)} = \frac{1}{2\delta_a^2} \sum_{i=1}^n \exp\left(-\frac{\|\hat{\mathbf{v}}_i^{(k)} - \mathbf{u}_{\rho(i)}^{(k)}\|^2}{2\delta_a^2}\right) \|\hat{\mathbf{v}}_i - \mathbf{u}_{\rho(i)}^{(k)}\|^2 \quad (12)$$

其中, $\hat{\mathbf{v}}_i^{(k)}$ 表示初始值为 $\mathbf{X}^{(k)}$ 时, 模板顶点 $\mathbf{v}_i$ 变换后的位置; $\mathbf{u}_{\rho(i)}^{(k)}$ 是在目标顶点 $\mathcal{U}$ 中与 $\hat{\mathbf{v}}_i^{(k)}$ 最近的对应点; $\|\hat{\mathbf{v}}_i - \mathbf{u}_{\rho(i)}^{(k)}\|^2$ 是 $\|\hat{\mathbf{v}}_i - \mathbf{u}_{\rho(i)}\|^2$ 的近似值, 因为 $\mathbf{u}_{\rho(i)}$ 对于 $\mathbf{X}$ 是非线性的, 选择近似的二次函数 $\|\hat{\mathbf{v}}_i - \mathbf{u}_{\rho(i)}^{(k)}\|^2$ 进行代替。

同样, 将 $\bar{\psi}_{\delta}^y(x)$ 代入 $E_{\text{smooth}}(\mathbf{X})$ 中, 得到平滑项替代函数 $\bar{E}_{\text{smooth}}(\mathbf{X})^{(k)}$, 如式 (13) 所示。

$$\bar{E}_{\text{smooth}}(\mathbf{X})^{(k)} = \frac{1}{2\delta_s^2} \sum_{i=1}^r \sum_{\mathbf{p}_j \in \mathcal{N}(\mathbf{p}_i)} \exp\left(-\frac{\|\mathbf{H}_{ij}^{(k)}\|^2}{2\delta_s^2}\right) \|\mathbf{H}_{ij}\|^2 \quad (13)$$

在每一次迭代过程中，替换掉目标函数 $E(\mathbf{X})$ 中的对齐项与平滑项，即可得到替代函数 $\bar{E}(\mathbf{X})^{(k)}$ ：

$$\bar{E}(\mathbf{X})^{(k)} = \bar{E}_{\text{align}}(\mathbf{X})^{(k)} + \bar{E}_{\text{smooth}}(\mathbf{X})^{(k)} + E_{\text{rot}}(\mathbf{X}) \quad (14)$$

对于子优化问题 $\bar{E}(\mathbf{X})^{(k)}$ ，采用 L-BFGS^[18] 求解器来快速优化。在 L-BFGS 求解过程中，每次迭代使用 $\bar{E}(\mathbf{X})^{(k)}$ 在最近 $l+1$ 次迭代 $\mathbf{X}_{(j)}, \mathbf{X}_{(j-1)}, \dots, \mathbf{X}_{(j-l)}$ 的梯度来估计 Hessian 矩阵，并推导出梯度下降方向，找到最优值作为下一次迭代的初值。直到 $\bar{E}(\mathbf{X})^{(k)}$ 收敛，从而求出最优解 $\mathbf{X}$ 。

得到 $\mathbf{X}$ 后，继续构造 $k+1$ 次迭代的替代函数 $\bar{E}(\mathbf{X})^{(k+1)}$ ，使用 L-BFGS 求解子问题，直到原目标函数 $E(\mathbf{X})$ 满足收敛条件。

3.4 轨迹优化

为了保证获取的轨迹点位置更加准确，提出了轨迹优化策略 (trajectory refinement strategy, TRS) 对轨迹点重定位，如算法 1 所示。

算法 1: TRS

Input: $\mathcal{J} = \{c_1, c_2, \dots, c_k\}$; $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$; $\mathbf{X} = \{X_1, X_2, \dots, X_r\}$
Output: Target trajectory $\hat{\mathcal{J}} = \{\hat{c}_1, \hat{c}_2, \dots, \hat{c}_k\}$

```

1 : Initializing  $\hat{\mathcal{J}}$  to empty;
2 : Calculating the normal  $\{n_1, n_2, \dots, n_m\}$  of  $\mathcal{U}$ ; Defining Z-axis direction  $\mathbf{n}^Z = \{0, 0, 1\}$ ; Defining normal threshold  $\theta$ ;
3 : for  $c_i$  in  $\mathcal{J}$  do
4 :    $c_i^*$  ← compute transformed points by Equation (3);
5 :    $\mathcal{N}(c_i) \in \mathcal{U}$  ← search the neighbors for  $c_i$  in  $\mathcal{U}$ ;
6 :   sort( $\mathcal{N}(c_i)$ ) by the Z-axis order;
7 :   for  $u_j$  in  $\mathcal{N}(c_i)$  do
8 :     if  $n^Z \cdot n_j < \theta$  then add  $u_j$  to  $\hat{\mathcal{J}}$ ; break;
9 :   end
10 : end
11 :  $\mathcal{J} \leftarrow \hat{\mathcal{J}}$ ; Let  $\hat{\mathcal{J}}$  be empty;
12 : Add the first point  $c_0$  in  $\mathcal{J}$  to  $\hat{\mathcal{J}}$ ; delete  $c_0$  in  $\mathcal{J}$ ;
13 : while  $\mathcal{J}$  is not empty do
14 :    $c^*$  ← find the closet point for the new added point of  $\hat{\mathcal{J}}$  in  $\mathcal{J}$ ;
15 :   add  $c$  to  $\hat{\mathcal{J}}$ ; delete  $c$  in  $\mathcal{J}$ ;
16 : end
17 : for  $\hat{c}_i$  in  $\hat{\mathcal{J}}$  do
18 :    $\hat{c}_i^*$  ← the mean of neighboring points of  $\hat{c}_i$ ;
19 : end
20 : return  $\hat{\mathcal{J}}$ ;

```

首先，根据 $\mathcal{G}$ 的最优变换矩阵 $\mathbf{X}$ ，用式 (3) 求出模板轨迹 $\mathcal{J}$ 变换后的位置。由于手机边框平滑放置在设备上扫描，点胶位置通常位于手机边

框的凹槽处。因此，通过 KDTree 寻找每个轨迹点在顶点 $\mathcal{U}$ 中的最近邻点，该点的法线应垂直于 XOY 平面，且 Z 坐标应足够小，确保在凹槽位置处找到新的轨迹点 $\hat{\mathcal{J}}$ 。然后，在 $\hat{\mathcal{J}}$ 中寻找每个点的最近点，使 $\hat{\mathcal{J}}$ 中所有点按照点胶顺序排列。最后，对于 $\hat{\mathcal{J}}$ 中每个点，计算该点和前后共 $2m+1$ 个点坐标的均值，作为其新坐标，从而使轨迹更加平滑。

4 实验结果与分析

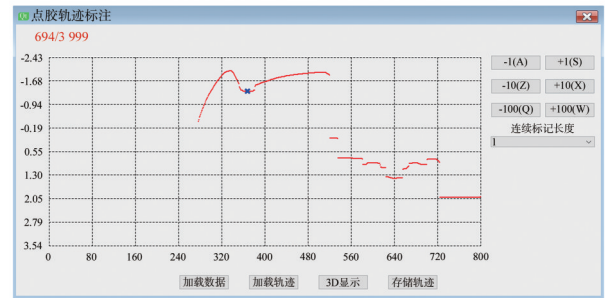
4.1 实验装置与数据集

4.1.1 实验装置

本文实验装置主要包括硬件系统和操作软件两部分，实验装置如图 6 所示。



(a) 硬件系统



(b) 操作软件的轨迹标注功能

图 6 实验装置

其中，硬件系统包括扫描系统和驱动系统。扫描系统为图 6 (a) 所示固定在上方的基恩士 LJ-X8060 线激光测量仪。激光线条的工作长度为 16 mm，每条激光线会投射出 3 200 个采样点，因此采样点的间隔距离 $\Delta x = 0.005$ mm。采样周期默认为 2 000 Hz，当物体的行进速度为 10 mm/s 时，激光线采样的间隔距离为 $\Delta y = 0.005$ mm。因此，在扫描采样时，记录下采样激光点的索引 i 和激光

线的索引 j , 即可得到采样点的坐标 (x, y, z) : $x = i \times \Delta x, y = j \times \Delta y, z$ 坐标为采样值。

驱动系统负责严格控制运动平台的旋转平移。将手机边框放置在平台上并固定, 设置每次扫描的起始点坐标和运动速度; 测量时, 平台沿 Y 轴方向上从起点到终点匀速前进运动, 同时线激光测量仪采样扫描; 扫描完一条边后, 平台顺时针旋转 90° 度并平移到下一条边的起始点, 依次对4条边进行扫描, 从而得到4个边框的点云数据, 边框扫描示意如图7所示。

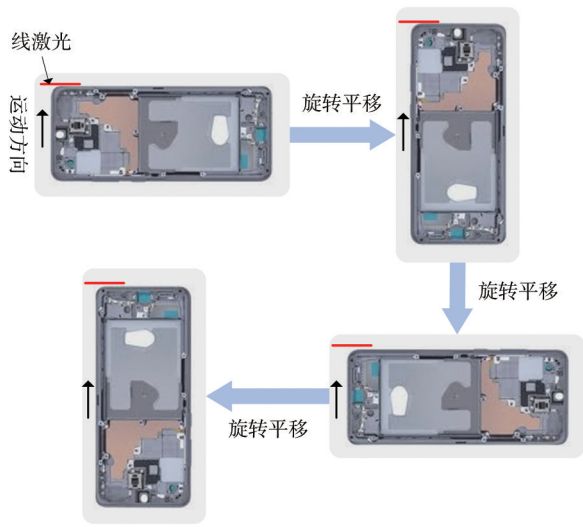


图7 边框扫描示意图

此外, 本文开发了一套离线在线结合的点胶操作软件。离线生成模板相关数据: 首先, 点云拼接算法求出模板拼接矩阵, 并生成模板 Mesh 模型; 另外, 通过软件对模板数据进行标注得到模板点胶轨迹, 图6(b)中为软件的轨迹标注功能; 最后, 通过模板 Mesh 与点胶轨迹构造出模板变形图, 节省在线时间。在线生成目标点胶轨迹: 首先, 加载相关模板数据; 在目标手机边框测量完成后, 为节省时间使用模板拼接矩阵进行拼接, 生成目标 Mesh 模型; 最后, 通过非刚性配准以及轨迹优化策略, 得到精准的目标点胶轨迹, 速度可达到实时效果。

4.1.2 数据集

实验数据如表1所示, 实验样本包含7组不同品牌、不同型号的手机边框, 涵盖了如今手机市场上常见的边框材质。每组3个, 其中1个作为该型号手机的模板, 其余2个为待测的目标边框, 分别命名为 Target1 和 Target2。通过实验装置对7组数

据(每组3个样本)进行扫描测量, 得到21组点云数据(每组4个边框点云), 即84个边框点云。

表1 实验数据

品牌	型号	数量	材质
Huawei	Mate20Pro	3	金属
Huawei	P20Pro	3	金属
Samsung	Samsung	3	金属
Xiaomi	Xiaomi9	3	金属
Realme	GTNeo2T	3	塑料
vivo	vivoS7	3	塑料
OPPO	OPPOA91	3	塑料

4.1.3 实验细节

本文采用具有 Intel Core i5-7200U CPU、8 GB RAM 的计算机, 使用 C++ 语言, 通过 PyQT 开发操作软件。

点云拼接中设置体素滤波器 $s_a = 0.1 \text{ mm}$, 半径滤波器 $r_a = 0.15 \text{ mm}$, $n_a = 25$ 。点云预处理中体素滤波器 $s_b = 0.15 \text{ mm}$, 半径滤波器 $r_b = 0.5 \text{ mm}$, $n_b = 30$ 。构造变形图时设置 $\lambda = 5 \cdot \text{length}$, 其中 length 表示模板 Mesh 的平均边长。L-BFGS 求解器中设置 $l = 5$ 。轨迹优化策略中设置法线阈值 $\theta = 0.9$, 轨迹平滑设置 $m = 1$ 。

本文通过驱动系统的运动参数计算出4个边框点云的初始全局位姿 $\{T_i, i=0, 1, 2, 3\}$, 便于点云进行位姿优化。

$$T_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad T_1 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 82 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_2 = \begin{pmatrix} -1 & 0 & 0 & 168 \\ 0 & -1 & 0 & 80 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad T_3 = \begin{pmatrix} 0 & -1 & 0 & 167 \\ 1 & 0 & 0 & -2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4.2 点云拼接实验

4.2.1 拼接精度结果与分析

本文对7组实验样本的模板数据进行点云拼接实验, 计算4个边框的配准精度。对于边框之间重叠的4个 R 角区域, 通过 KDTree 以阈值为 0.12 mm 的距离搜索对应点, 计算对应点距离的均方根误差 (root mean square error, RMSE) 来表示配准精度, 并计算出4个 R 角区域的平均对齐精度。RMSE 的计算式如下, 其中 $\mathcal{K}$ 为两个点云重叠部分构造的

对应点集。

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{K}|} \sum_{(p,q) \in \mathcal{K}} \|p - q\|^2} \quad (15)$$

表2为点云拼接实验在扫描数据集上的实验结果。本文方法在4个重叠区域中都有出色的对齐效果,7组数据的配准精度RMSE基本为0.04 mm左右。其中,vivoS7数据的平均对齐误差为0.036 mm;相对较差的为P20Pro数据,平均误差为0.043 mm。经研究发现,vivoS7手机边框比较平整,相邻两个边框点云大面积重叠,利于配准;而P20Pro边框部分有较大起伏,点云中存在更多的噪声,对配准效果有一定的影响。

表2 点云拼接精度误差

实验样本	配准精度RMSE/mm				
	R_0	R_1	R_2	R_3	平均
Mate20Pro	0.041	0.038	0.040	0.036	0.039
P20Pro	0.044	0.046	0.034	0.046	0.043
Samsung	0.041	0.037	0.037	0.040	0.039
Xiaomi9	0.045	0.041	0.038	0.043	0.042
GTneo2T	0.040	0.039	0.041	0.036	0.039
vivoS7	0.040	0.028	0.031	0.044	0.036
OPPOA91	0.041	0.042	0.045	0.040	0.042

4.2.2 对比实验

为了验证本文拼接算法的高效性,选取ICP^[19]和FGR^[20]分别对Mate20Pro模板数据的4个R角区域

域边框点云进行两两配准,计算其配准精度RMSE以及耗时,结果如表3所示。图8为不同方法在4个重叠区域配准的对比结果,其中“#n”表示在重叠区域内以最远距离为0.12 mm搜索的点对数。

表3 不同方法的点云拼接结果

方法	配准精度RMSE/mm					时间/s
	R_0	R_1	R_2	R_3	平均	
ICP	0.037	0.053	0.042	—	—	3.0
FGR	0.041	0.051	0.047	0.071	0.052	45.3
Ours	0.041	0.038	0.040	0.036	0.039	19.2

由表3和图8可以看出,本文方法使4个边框点云在全局空间中保持高质量的对齐效果,而ICP和FGR算法对于点云两两配准虽有不错的效果,但由于累计误差的存在, R_3 区域的配准精度均很差。其中ICP配准结果中 R_3 区域的两块点云之间距离过大,导致搜索最近点失败,而FGR也只能匹配到少量对应点。在时间上,FGR算法需要对点云提取FPFH特征,导致其运行速度很慢。本文方法比FGR配准效果更好,且速度更快。

4.3 轨迹生成实验

4.3.1 轨迹精度误差

为了验证所获取点胶轨迹的准确性,本文对目标手机边框的扫描数据标注出点胶位置,作为理想点胶轨迹。图9所示为Xiaomi9实验样本的点胶轨迹,其中红色点为本文方法生成的目标点胶轨迹,

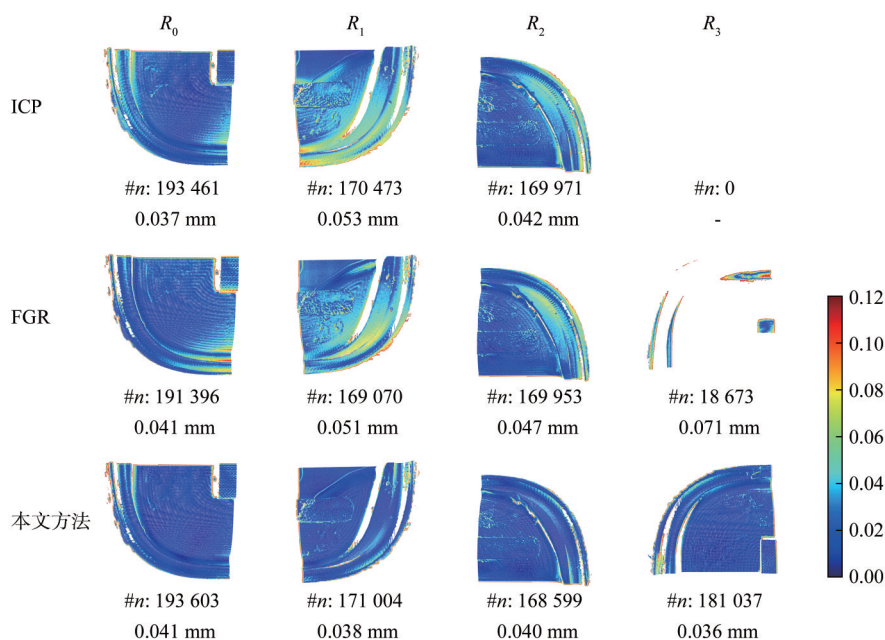


图8 不同方法在4个R角区域的配准效果对比

绿色点为标注的理想点胶轨迹。当两种点胶轨迹完全重合时，能达到最好的点胶效果。

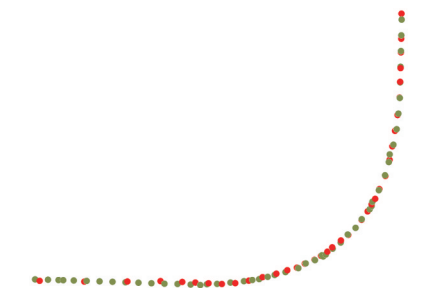


图9 目标点胶轨迹（红）与理想点胶轨迹（绿）

通过计算目标点胶轨迹与理想点胶轨迹的平均绝对误差（MAE）和最大偏差（MAX）来衡量轨迹精度，MAE的计算如式（16）所示。同时，计算模板Mesh与目标Mesh的距离误差RMSE来评估非刚性配准精度。

$$MAE = \frac{1}{\eta} \sum_{i=1}^{\eta} \|\alpha_i - \beta_i\| \quad (16)$$

式中， $\alpha_i \in \hat{\mathcal{J}}$ 表示生成的轨迹点； β_i 为理想点胶轨迹中距离 α_i 最近的点； η 表示生成轨迹点的总数。

对7组实验样本的目标边框数据进行了轨迹生成实验，轨迹生成实验结果如表4所示。本文方法使模板模型与目标模型有不错的配准精度，7组实验数据中的轨迹误差MAE基本在0.01 mm以下。效果最出色的为Xiaomi9 Target2数据，其轨迹误差仅有0.003 mm，最大偏差也只有0.025 mm。仅Mate20Pro Target1和vivoS7 Target2数据由于点云中存在较多噪声，模板模型与其配准时产生了部分错误匹配，生成的点胶轨迹精度较差。

4.3.2 对比实验

本文与N-ICP^[21]、Fast-RNRR^[16]以及FGR^[20]在Xiaomi9 Targe2实验样本上进行了对比实验。其中，N-ICP是以点对仿射变换为变形场的经典非刚性配准算法，Fast-RNRR是基于变形图的非刚性配准算法，FGR是基于特征点匹配的刚性配准算法。

将模板与目标进行配准后，使用相同的轨迹优化策略来获取轨迹点，计算模型配准精度、点胶轨迹精度以及配准时间，点胶轨迹精度对比结果如表5所示。

由表5可以看出，本文方法、Fast-RNRR以及N-ICP的配准精度与点胶轨迹精度均高于FGR，证明了非刚性配准算法的必要性，可以有效应对产品

表4 轨迹生成实验结果

实验样本		RMSE/mm	MAX/mm	MAE/mm
Mate20Pro	Target1	0.085	0.763	0.049
	Target2	0.078	0.079	0.007
P20Pro	Target1	0.056	0.071	0.007
	Target2	0.063	0.076	0.007
Samsung	Target1	0.054	0.039	0.004
	Target2	0.065	0.190	0.009
Xiaomi9	Target1	0.058	0.041	0.005
	Target2	0.054	0.025	0.003
GTneo2T	Target1	0.053	0.039	0.004
	Target2	0.065	0.042	0.005
vivoS7	Target1	0.059	0.082	0.009
	Target2	0.080	0.568	0.023
OPPOA91	Target1	0.069	0.267	0.006
	Target2	0.078	0.030	0.004

表5 点胶轨迹精度对比结果

算法	RMSE/mm	MAX/mm	MAE/mm	时间/s
FGR	0.107	0.113	0.011	276.1
N-ICP	0.081	0.081	0.005	36 526.2
Fast-RNRR	0.057	0.027	0.003	225.4
本文方法	0.054	0.025	0.003	53.5

中的形变问题；而本文与Fast-RNRR均以变形图表示变形场，不仅速度上比N-ICP更快，精度上也更优。由于本文算法采用离线在线结合模式，离线构造模板变形图，在线进行配准。因此在实际在线配准时，本文方法仅需53 s即可完成，比Fast-RNRR快很多，基本满足点胶设备的实时性要求。

4.3.3 轨迹优化策略消融实验

由于噪声的影响，模板与目标配准后得到的点胶轨迹仍存在有较大的误差。本文对扫描数据集的7组目标数据样本进行了TRS消融实验，结果如图10所示，每组数据的结果为Target1和Target2轨迹误差MAE的平均值。从图中可以看出，配准后直接计算得到的轨迹精度均较差，而使用轨迹优化策略后，7组实验数据的轨迹精度均有了明显的提升。其中OPPOA91数据的轨迹误差MAE由0.052 mm降到了0.005 mm，精度提升了9倍左右。轨迹优化策略显著提升了点胶轨迹的准确率，证明了其重要性。

基于模板的方法十分依赖于模板轨迹的标注质量，为了进一步探究模板标注质量对轨迹生成的影响，本文对标注的模板轨迹增加不同程度的噪声，

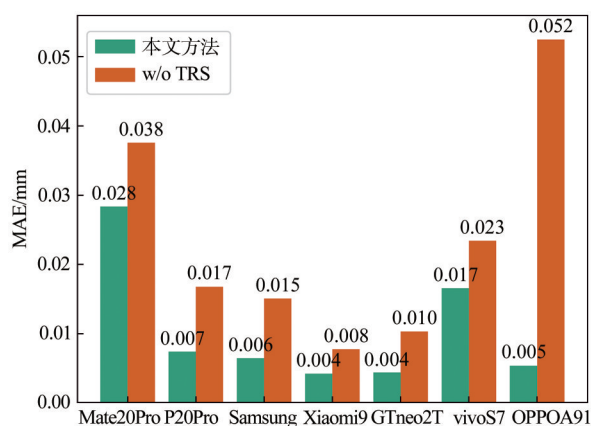


图10 轨迹优化策略在7组实验数据上的消融实验

在Xiaomi9 Target2数据上进行了TRS消融实验。本文在模板边框点云中随机选取适量点作为噪声添加到模板轨迹中，评估生成的轨迹误差，模板轨迹增加噪声的TRS消融实验如表6所示。当添加5%的噪声后，点胶轨迹的最大偏差MAX激增到5 mm左右。随着噪声不断增加，轨迹精度MAE也越来越差。模板轨迹的标注质量很大程度上影响了点胶轨迹的精度。而使用轨迹优化策略后，轨迹精度有所提升，减轻了噪声的影响，再次验证了轨迹优化策略可以有效增强点胶轨迹的鲁棒性。

表6 模板轨迹增加噪声的TRS消融实验

Noise	w/o TRS		本文方法	
	MAX/mm	MAE/mm	MAX/mm	MAE/mm
None	0.0702	0.010	0.025	0.003
5%	5.226	0.068	4.446	0.037
10%	7.321	0.107	6.472	0.055
15%	7.837	0.177	6.955	0.099
20%	5.923	0.183	6.502	0.105

5 结束语

为了实现手机产品点胶自动化，有效应对产品中的形变问题，本文提出一种高效、精准、鲁棒的点胶三维轨迹生成方法。引入基于位姿图的点云拼接算法有效解决了线激光测量仪扫描手机边框时的局限性，并通过基于变形图的非刚性配准算法以及轨迹优化策略，得到了精准的手机边框点胶轨迹。在本文的扫描数据集上验证了该方法的高效性，点胶轨迹精度误差基本在0.01 mm以下。另外，本文设计了一套离线在线结合的操作软件，在线能够实时生成手机边框的点胶轨迹。

但是，该方法仍有一些不足之处。例如测量仪扫描的点云数据中存在大量难以去除的噪声点，这会导致最终的点胶轨迹精度较差。在下一阶段的工作中，将会结合先进的点云去噪算法，进一步提高系统的泛化性和鲁棒性。

参考文献：

- [1] 程方, 张曦, 张金松. 基于机器视觉的点胶机定位系统[J]. 机械设计与制造, 2013(3): 101-104.
CHENG F, ZHANG X, ZHANG J S. A positioning system for dispensing machine based on machine-vision[J]. Machinery Design & Manufacture, 2013(3): 101-104.
- [2] 张胜奎. 智能点胶机关键技术研究[D]. 焦作: 河南理工大学, 2011.
ZHANG S K. Study on key technologies of intelligent dispensing system[D]. Jiaozuo: Henan Polytechnic University, 2011.
- [3] 李顺君, 钱强, 史金龙, 等. 基于卷积神经网络的扁体咽拭子采样机器人[J]. 计算机工程与应用, 2022, 58(15): 324-329.
LI S J, QIAN Q, SHI J L, et al. Sampling robot for tonsil swabs based on convolutional neural network[J]. Computer Engineering and Applications, 2022, 58(15): 324-329.
- [4] 吴佳毅, 陈建强, 邓宇平, 等. 基于机器视觉检测的手机显示屏点胶系统[J]. 自动化与信息工程, 2019, 40(3): 43-48.
WU J Y, CHEN J Q, DENG Y P, et al. Dispensing system of mobile screen based on machine vision inspection[J]. Automation & Information Engineering, 2019, 40(3): 43-48.
- [5] 贺昌兵. 基于三维视觉的点胶机工件定位与路径提取技术研究[D]. 成都: 电子科技大学, 2023.
HE C B. Research on workpiece positioning and path extraction technology of dispensing machine based on three-dimensional vision[D]. Chengdu: University of Electronic Science and Technology of China, 2023.
- [6] 陈鸿雨. 面向曲面中框的3D视觉引导点胶方法及系统研究[D]. 哈尔滨: 哈尔滨理工大学, 2023.
CHEN H Y. 3D visual guidance dispensing for curved middle frame method and system research[D]. Harbin: Harbin University of Science and Technology, 2023.
- [7] YOU B, CHEN H Y, LI J Y, et al. Fast point cloud registration algorithm based on 3D NPFH descriptor[J]. Photonics, 2022, 9(6): 414.
- [8] CHEN Y, MEDIONI G. Object modelling by registration of multiple range images[J]. Image and Vision Computing, 1992, 10(3): 145-155.
- [9] CHOI S, ZHOU Q Y, KOLTUN V. Robust reconstruction of indoor scenes [C]//Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2015: 5556-5565.
- [10] LU F, MILIOS E. Globally consistent range scan alignment for environment mapping[J]. Autonomous robots, 1997, 4: 333-349.
- [11] MORÉ J J. The Levenberg-Marquardt algorithm: implementation and theory[M]//WATSON G A, ed. Lecture Notes in Mathematics. Heidelberg: Springer, 1978: 105-116.
- [12] ESTER M, KRIEGER H P, SANDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]//Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. New York: ACM, 1996: 226-231.
- [13] LEE D T, SCHACHTER B J. Two algorithms for constructing a Delaunay triangulation[J]. International Journal of Computer & Information Sciences, 1980, 9(3): 219-242.

- [14] KIMMEL R, SETHIAN J A. Computing geodesic paths on manifolds[J]. Proceedings of the National Academy of Sciences, 1998, 95(15): 8431-8435.
- [15] HOLLAND P W, WELSCH R E. Robust regression using iteratively reweighted least-squares[J]. Communications in Statistics-theory and Methods, 1977, 6(9): 813-827.
- [16] YAO Y X, DENG B L, XU W W, et al. Quasi-newton solver for robust non-rigid registration[C]//Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 7597-7606.
- [17] LANGE K. Optimization[M]. Berlin: Springer Science & Business Media, 2013.
- [18] NOCEDAL J. Updating quasi-Newton matrices with limited storage[J]. Mathematics of Computation, 1980, 35(151): 773-782.
- [19] BESL P J, MCKAY N D. Method for registration of 3-D shapes[C]//Sensor Fusion Iv: Control Paradigms and Data Structures. [S.l.:s.n.], 1992: 586-606.
- [20] ZHOU Q Y, PARK J, KOLTUN V. Fast global registration[M]//Computer Vision - ECCV 2016. Cham: Springer International Publishing, 2016: 766-782.
- [21] AMBERG B, ROMDHANI S, VETTER T. Optimal step nonrigid ICP algorithms for surface registration[C]//Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2007: 1-8.



史金龙 (1976-), 男, 博士, 江苏科技大学计算机学院教授, 主要研究方向为计算机视觉、深度学习。



钱强 (1978-), 男, 江苏科技大学计算机学院讲师, 主要研究方向为机器学习、深度学习。



欧镇 (1973-), 男, 江苏科技大学计算机学院讲师, 主要研究方向为信息系统建模、系统架构设计和移动端应用。

[作者简介]



刘洋 (2000-), 男, 江苏科技大学计算机学院硕士生, 主要研究方向为计算机视觉、三维重建。



白素琴 (1976-), 女, 江苏科技大学计算机学院副教授, 主要研究方向为计算机视觉。