Bridging Dimensions in Fingerprints to Advance Distinctiveness: Recovering 3D Minutiae from a Single Contactless 2D Fingerprint Image

Chengdong Dong, Ajay Kumar, *Fellow, IEEE*

Abstract—Contactless 3D fingerprint identification systems have emerged to provide more accurate and hygienic alternatives to contact-based conventional systems that acquire hundreds of millions of fingerprints everyday. However, the intricate process of acquiring 3D fingerprints presents a significant challenge, acting as a key barrier to fully unlocking the potential of 3D fingerprint biometrics. This paper introduces a novel framework to directly recover corresponding 3D minutiae template from a single contactless 2D fingerprint image. Billions of contact-based fingerprints have been acquired and employed everyday for e-governance and other applications. Seamless adoption of contactless 3D fingerprint technologies also requires advanced capabilities to accurately match 3D fingerprints with respective 2D fingerprint templates, which is currently missing in existing literature. We therefore introduce novel capabilities to accurately align minutiae templates in 3D spaces and enable compensation for the unknown perspective transformation. This capability significantly enhances the ability to accurately match 3D to 3D and 3D to 2D fingerprint templates. Furthermore, we introduce a new approach to synthesizing realistic contactless fingerprint images, resulting in the generation of a large synthetic database complete with corresponding 3D ground truths of minutiae points. Finally, we provide a detailed theoretical analysis of formulation for the uniqueness of recovered 3D minutiae templates, providing a theoretical justification for the superiority of such 3D minutiae templates over their 2D counterparts.

Index Terms—Contactless 3D Fingerprint Identification, Fingerprint Synthesis, Biometrics, Cross-Domain Fingerprint Matching.

1 Introduction

Completely contactless fingerprint identification has attracted significant attention, such systems offer a high level of hygiene and address skin deformation-related challenges with the contact-based fingerprint systems. Widely deployed fingerprint matching algorithms [1], [6] are largely based on minutiae-based features since minutia are robust to re-scaling, rotation, distortion, and noise. Such minutiae-based methods are interpretable compared with texture-based fingerprint matching methods since the minutiae matching or correspondence is provided along with the match score. Therefore, the accuracy of minutiae extraction, from the acquired fingerprint images, is critical for accurate matching, including contactless-to-contact-based (CL2CB) and contactless-to-contactless (CL2CL) fingerprint image matching. Traditional minutiae detection methods, such as in [24], rely on topological analysis and hand-crafted features for the minutiae extraction. There have been several promising attempts using deep learning-based methods [4], [20], [23], [25], [26], [27] to recover the minutiae templates from contact-based and contactless fingerprint images.

As compared to contact-based fingerprints, completely contactless fingerprints are often acquired under six degrees of freedom, under diverse finger poses, and can require complex perspective transformations to align the contactless fingerprints that are to be matched. Such match accuracy significantly degrades when two contactless fingerprints, from the same fingers, acquired under very diverse finger poses are matched. Therefore, the localization of contactless fingerprints in 3D spaces can offer a viable solution to address such problems: Related work in [21], [28] acquired contactless fingerprint images with the depth map. Therefore, 3D fingerprint templates were recovered which localized the minutiae in 3D spaces, along with the minutiae orientations using the elevation and azimuth angles. By incorporating the relative 3D measurements between the 3D minutiae, two 3D templates can be iteratively aligned, and such an approach has shown impressive results and advantages over the 2D minutiae templates. However, there are two challenges with such [21], [28] conventional methods: i) the 3D information needs to be acquired by specially designed devices or sensors which are not cost-effective, complex, and also computationally demanding, ii) the standalone performance from the 3D minutiae template matching methods, e.g. [21], [28], is quite limited and needs to be further enhanced to meet expectations for the deployments.

To address the first challenge, we design a one-stage lightweight learning-based model, abbreviated as FingerYolo3D in this paper, which can be used to recover 3D minutiae templates from a single contactless monocular fingerprint image. The proposed network architecture can also be used to extract 2D minutiae templates and such 2D minutiae extractor is abbreviated as FingerYolo2D in this paper. The 3D minutiae templates are uniquely represented using x, y, z, θ, ϕ, t, where (x, y) is the location of respective minutia in 2D space, z is the height of the minutiae at location (x, y), t is the type of the minutiae, while θ and ϕ respectively represents the azimuthal and elevational angles for the respective 3D minutia orientation. Experimental results

- *Authors are with the Department of Data Science and Artificial Intelligence, The Hong Kong Polytechnic University, Hong Kong. E-mail: chengdong.dong@connect.polyu.hk, ajay.kumar@polyu.edu.hk*

presented in this paper illustrate that the 3D minutiae templates recovered by FingerYolo3D can immensely benefit from such 3D information and offer significant advancement for the minutiae-based contactless fingerprint verification capabilities available today. The FingerYolo2D model can also achieve state-of-the-art (SOTA) performance compared with previous 2D minutiae extractors, which can validate the superiority of the design of the proposed architecture.

To effectively train the FingerYolo3D, it is necessary to acquire a large contactless fingerprint database with accurately labeled respective 3D ground truths. However, it is difficult to acquire such a large database for four reasons: i) Several legal restrictions such as from GDPR [34] prohibit large-scale collection of private databases, including those from fingerprints; ii) Current 3D fingerprint acquisition methods [21], [28], [29], [30], [31], [33] provide the depth map of the contactless fingerprint with inadvertent errors introduced during the acquisition, reconstruction or the labeling, and therefore such acquired 3D information cannot be regarded as true ground truth for 3D minutia; iii) generation of accurate ground truths, especially for 3D surfaces, is widely regarded as laborious and costly task. Therefore, this paper introduces a new approach to more accurately synthesize a large database that contains realistic contactless fingerprint images with the corresponding ground truths for the respective 3D minutiae templates.

Contactless 3D fingerprint surfaces are non-rigid surfaces that cannot be accurately described with spherical or cylindrical models. Therefore, accurate alignment of 3D minutiae templates that are often acquired from different viewpoints, largely due to the nature of contactless fingerprint imaging, cannot be accurately achieved by linear perspective transformations in 3D spaces. This is a key factor that limits match accuracy for contactless 3D fingerprints. Therefore, to address the second challenge, we introduce the Perspective Transformation Compensation Mechanism (PTCM) for 3D minutiae template matching, and this approach can achieve the SOTA performance. Our experimentation reveals that currently available 3D minutiae matching methods [21], [28] cannot outperform the SOTA 2D minutiae template methods, and such comparisons are also missing in respective references. This is largely due to the limitations of prior methods [21], [28] to avail full potential from the 3D minutiae templates. In addition, currently available methods for 3D fingerprint matching cannot be used for 3D to 2D template matching, which is critical for the success of 3D fingerprint technologies to replace large-scale deployments that have already acquired billions of 2D fingerprint templates. On the contrary, the proposed PTCM aligns the 3D minutiae templates to alleviate the influence of perspective transformation, and it is the first such method to accurately match 3D to 2D minutiae templates. The experimental results indicate that both CL2CB and CL2CL match performance benefits from the proposed PTCM.

Earlier studies [28] on the uniqueness of 3D fingerprints have made assumptions on the independence of 3D features. Such an assumption is not reasonable, although it may help to establish an overestimated upper bound on the individuality of 3D fingerprints, i.e., the probability of falsely matching randomly selected 3D fingerprints from different persons. This paper presents a systematic theoretical analysis of the accurate estimates for the uniqueness of contactless 3D fingerprints. Key contributions from this paper can be summarized as follows:

- This paper develops a *lightweight* model to recover *3D minutiae* template from a *single contactless 2D* fingerprint image. We also design a novel optimization function that can account for the cyclic nature of 3D angles, which enables our network to precisely estimate the location and orientation of 3D minutiae. Such capabilities to localize minutiae positions and orientations in 3D spaces enable highly accurate matching of contactless 2D fingerprint images using the recovered 3D minutiae templates. Rigorous experimental results in Sec. 7 indicate outperforming results, both for the 3D minutiae template and 2D minutiae template extractor introduced from this work.
- This paper for the first time develops a novel mechanism (PTCM) to accurately align and match 3D minutiae templates for both contactless 3D to contactless 3D, and contactless 3D to contact-based 2D fingerprint matching. Such new capability to align 3D minutiae templates (regardless of whether the 3D templates are acquired from the real-world 3D sensors or recovered by our proposed model) addresses a key challenge associated with estimating the perspective transformations for aligning completely contactless fingerprint images that are often acquired with arbitrary finger pose. In the PTCM module, a new tetrahedron-based algorithm is designed to remove outlier minutiae matching pairs from the minutiae correspondences that are extracted by the GNN-based module. We also provide theoretical proof (in Sec. 4) on the effectiveness and justification of the candidate tetrahedron selection strategy introduced in this algorithm. In the experiments, we align and match the 3D minutiae templates using the proposed PTCM and achieve outperforming results, both for the CL2CB and CL2CL fingerprint matching.
- Earlier research on 3D minutiae templates' individuality has overlooked the interdependence between the locations and orientations of 3D minutiae, resulting in a large overestimation of uniqueness. We present a systematic theoretical analysis to accurately estimate such real-world interdependence in the distribution of 3D minutiae. Such analysis enables us to determine the degree of performance enhancement on several 2D fingerprint databases in the public domain, validating the merit of our work in recovering 3D templates from single contactless fingerprints.

In addition, this paper also addresses technical challenges in accurately and efficiently synthesizing a large-scale database of contactless fingerprint images, along with the respective 3D ground truth minutiae templates. Accuracy of the synthesized database is vital to robustly train the 3D minutiae detector for the contactless 2D fingerprint images. Our experimental results indicate that such synthesized database is more effective in training such minutiae detection network, both for recovering 3D and 2D templates, introduced in this paper.

2 RELATED WORK

In this section, we present a brief summary of the currently available methods of fingerprint synthesis, minutiae extraction, 3D information recovery, 3D minutiae template matching. We also summarize the methods that analyze the individuality of minutiae templates.

2.1 Contactless Fingerprint Synthesis

Priesnitz *et al.* [35] propose a method to synthesize contactless fingerprints by distorting contact-based fingerprint images to simulate contactless fingerprints' ridge and valley pattern deformations. However, such distortion cannot adhere to the accurate 3D geometric perspective relationship, making it unable to provide a reliable 3D depth map for the synthesized contactless fingerprints. In addition, their synthesis method can only generate ridge-valley patterns of contactless fingerprints, which is not a realistic representation of real-world fingerprint images. Recent work in [5] uses Bezier surfaces to model 3D finger shapes. This method involves engraving ridge-valley patterns onto a 3D surface, from which contactless fingerprint images are projected using a camera projection model. However, the 3D-2D texture mapping technique used in this process is extremely time-consuming and limits the speed for generating large-scale contactless fingerprints due to the inclusion of a large number of redundant points during texture mapping. Moreover, as these contactless fingerprint images only display ridge-valley patterns without any realistic texture, the authors in [5] were obliged to adopt the Gabor filter to compensate for the domain gap between such unrealistic synthesized images and real-world contactless fingerprint images.

2.2 Minutiae Extraction of Contactless Fingerprints

Several deep neural network-based methods in the literature can detect fingerprint minutiae and outperform conventional methods such as Mindtct [24]. The patch-wise method in [20] partitions the fingerprint image into patches and predicts the minutia within each patch, which, however, ignores global features and leads to an extremely long inference time. Tang *et al.* [25] introduce FingerNet for minutiae extraction, but the performance achieved is not competitive. Reference [26] proposes a coarse-to-fine framework and generates candidate patches from the coarsely extracted minutiae map to decrease the patch count, but it remains a two-stage method with considerable time complexity. The authors in [27] propose a Fast Minutiae Extractor (FME) but still adopt a two-stage design, which motivated fellow researchers to design one-stage architectures to further enhance the extraction speed. The ContactlessMinuNet (abbreviated as CMNet) [4] is an end-to-end minutiae detection algorithm, but it is quite slow because of its inefficient U-Net architecture. Reference [23] recently proposed a framework to address the limitations discussed above. A single-stage design with a lightweight architecture was introduced, and patchwise detections were avoided. However, the neural architecture in [23] outputs the probability map instead of the actual positional results for the minutiae templates. Therefore, a complex post-processing stage is incorporated to generate the exact locations of the minutiae, which considerably reduces the total speed for the minutiae detection.

2.3 Recovery of 3D Information from Monocular 2D Contactless Fingerprint

Earlier research has shown that 3D information can be recovered from monocular contactless images using neural networks. Cui *et al.* [2] propose a neural model to predict the 3D surface normal from a single contactless fingerprint image, followed by the recovery of depth information from the estimated gradient maps. Such an approach to recover 3D information from the entire fingerprint region is computationally demanding as our objective is to recover the 3D information from the minutiae points. Instead, it is practical to only predict the 3D information on the key regions of interest. The authors in [13] propose Yolo-6D to predict the 3D bounding boxes of certain objects from a monocular image and the 3D pose can be recovered from such detected 3D bounding boxes, which motivates us to consider the task of 3D minutiae detection as a six-dimensional object detection: for each object (minutiae) to be detected, the model should generate its spatial coordinates and the orientations along x, y, z axes to construct a 6D descriptor.

2.4 Matching of 3D Minutiae Templates

There has been exciting work in [21], [28] to match fingerprint templates generated from 3D fingerprints. However, there are several limitations to such earlier work: i) These methods cannot be used to match 3D fingerprint templates with 2D fingerprint templates. Such interoperability is critical to advance contactless 3D fingerprint technologies for real-world applications; ii) these are extremely slow; and iii) These methods were introduced long back and cannot outperform the *currently* available contactless fingerprint SOTA methods that can match respective 2D fingerprints for the 3D fingerprint images. Therefore, there is a compelling need to develop advanced matchers to fully realize the potential of 3D fingerprint templates over 2D fingerprint templates.

2.5 Individuality Analysis for Minutiae Templates

The individuality of widely employed fingerprint minutiae templates to accurately identify humans has attracted several studies over the last hundred years. It's quantified as the probability of false random correspondences between two randomly matched templates from different individuals. A study in [3] is one example that provides an estimation of such individuality using 500 dpi sensors but with several assumptions on the independence of minutiae features. Zhu *et al.* [18] noted such limitations and modeled the relationship between the distributions of spatial positions and the orientations to quantify the probability of false random correspondences (PRC) more accurately. Dass *et al.* [19] introduce further enhancement in estimating the accuracy of PRC for real-world fingerprints by accounting for the influence of noise in minutiae detection. Reference [28] presented the first such estimation on the individuality of 3D fingerprints, and [32] underlines that such estimation fails to consider the inter-dependence of various minutiae features. Therefore, the estimates presented in [28] can only indicate the upper bound and are far from the expectations. A more accurate analysis of the interdependence among minutiae features on real-world 3D surfaces is necessary to generate an accurate estimation of the uniqueness of 3D fingerprint minutiae templates.

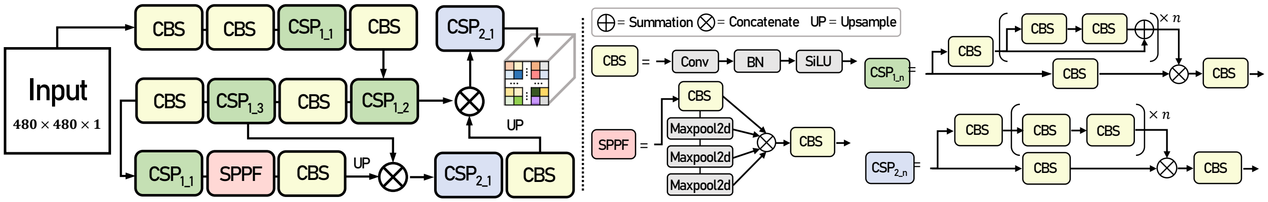

Input
480 × 480 × 1
CBS
CBS
CSP1_1
CBS
CSP2_1
CBS
CSP1_3
CBS
CSP1_2
UP
CSP1_1
SPPF
CBS
UP
CSP2_1
CBS
⊕ = Summation ⊗ = Concatenate UP = Upsample
CBS = Conv BN SiLU
SPPF = CBS Maxpool2d Maxpool2d Maxpool2d CBS
CSP1_n = CBS CBS CBS ⊕ × n CBS CBS
CSP2_n = CBS CBS CBS × n CBS CBS

Fig. 1: Network architecture for our framework to recover 3D minutiae from single monocular fingerprint image.

3 Minutiae Detection in 3D Space

A range of Yolo-based methods [9], [10], [11], [12] have recently emerged for object detection, which inspires us to address the challenges of minutiae detection in 3D spaces. Several reasons contribute to the balance between the accuracy of detection and inference speed. The introduction of appropriate loss functions forces the predicted bounding boxes to approach the target object's central location while considering the object's width and height since the incorrect prediction of the size of the bounding box will penalize the network using the enhanced loss function. The attentional mechanism in the design of such architecture contributes to the enhanced accuracy of the model significantly, especially for the objects with different scales, while the lightweight architecture with a Non-Maximum Suppression (NMS) module can eliminate redundantly detected boxes to ensure high performance and fast inference speed.

Architecture of Proposed Framework: The architecture of our FingerYolo3D network can be visualized in Fig. 1. Different from a series of popular Yolo-based architectures, our model adopts a lightweight design by removing branches for large object detection and only retaining the branch to detect tiny objects from the backbone. The FingerYolo3D receives 480×480 pixels fingerprint images as the input and partitions them into grids with the size of 6×6 pixels to predict the bounding boxes of 2D minutiae locations, confidence, minutiae type, depth, elevational angle, and azimuthal angle for minutiae in each of these grids.

The bounding box represents the 2D location of the minutiae and the average ridge-valley distance of the grid where the minutiae is located, thus it is described in three-tuple real numbers as the output feature. The confidence score occupies only one tuple whose value ranges from 0 to 1 and it reflects how likely the box contains minutiae and how accurate is the boundary box. We use ”01”, ”10” to encode the minutiae ending and minutiae bifurcation respectively, and we adopt BCE loss for minutiae type classification. In this work, we re-scale the fingerprint images into 480×480 and assume that the depth of the minutia lies in a range of 0 to 200 pixels along the z-axis. It was experimentally observed that the convergence of the model is unstable if the depth value is optimized by l_2 loss. We consider the depth prediction as a classification problem. The class identity increases by one for every 10 pixels in depth, starting from a depth of 0 pixels, and there is a total of 20 classes for the depth. We use one-hot encoding to represent the feature of depth, which is a 20-digit binary tuple for optimization by BCE loss. Both azimuth and elevation angles are encoded using a single digit. Therefore, the total size of the encoded feature template is $80 \times 80 \times 28 (3 + 1 + 2 + 20 + 1 + 1)$.

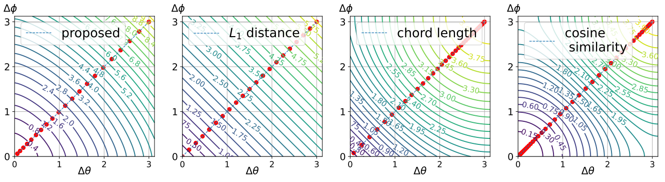
The Optimization of the Network Model: The architecture of FingerYolo3D is optimized during training using the total loss l_{total} (Eq. (1)) which is the weighted combination of five loss functions, i.e. objectness loss l_{obj} measuring the probability that a minutia exists in a proposed region of interest, classification loss l_{cls} denoting the type of detected minutia, depth l_z indicating the depth information of the detected minutia, bounding box loss l_{box} measuring how accurately the model can predict the center of the minutia and how well the predicted bounding box can localize the minutiae, and our 3D orientation loss l_o for predicting the azimuth and elevation angles of the minutia.

$$l_{total} = \lambda_{obj} l_{obj} + \lambda_{box} l_{box} + \lambda_{cls} l_{cls} + \lambda_z l_z + \lambda_o l_o \tag{1}$$

The orientation of minutia is a cyclic variable and therefore the direct prediction of this angle is not appropriate. Earlier studies that predict the orientation of 2D minutiae, several representative ways [4], [23], [27] are proposed to address this challenge. For the accurate prediction of minutiae orientations in 3D space, we propose the l_o, and can be defined in Eq. (2):

$$\begin{aligned} l_o = & w_\theta (\cos^{-1}((\cos\boldsymbol{\theta}\cos\hat{\theta} + \sin\boldsymbol{\theta}\sin\hat{\theta} - 0.5)/(1+\epsilon) + 0.5))^{1+\delta} \\ & + w_\phi (\cos^{-1}((\cos\boldsymbol{\phi}\cos\hat{\phi} + \sin\boldsymbol{\phi}\sin\hat{\phi} - 0.5)/(1+\epsilon) + 0.5))^{1+\delta} \end{aligned} \tag{2}$$

In Eq. (2), $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ respectively represent the ground truth for the azimuth and elevational orientations, $\hat{\theta}$, $\hat{\phi}$ are the respective predicted azimuthal and elevational orientations, w_θ and w_ϕ are weights to reflect the importance of θ and ϕ respectively, ϵ is set to 10^{-5} to clip the range of the item in the $\cos^{-1}(\cdot)$ to alleviate the failure during the backpropagation from $\cos^{-1}(\cdot)$ function as also stated in [23]. When FingerYolo2D is trained to extract 2D minutiae templates, we set the l_z in Eq. (1) and w_ϕ in Eq. (2) as zero. More details on loss functions are provided in Appendix G.



(a) proposed (b) L_1 (c) chord (d) cosine

Fig. 2: Comparative loss functions with their gradients.

We visualize the proposed l_o in Fig. 2 with other related loss functions for the prediction of minutia orientation. Here $\Delta\theta$ and $\Delta\phi$ respectively represent $|\boldsymbol{\theta} - \hat{\theta}|$ and $|\boldsymbol{\phi} - \hat{\phi}|$. The L_1 distance loss is $\Delta\theta + \Delta\phi$, the chord length loss is $\sqrt{(\cos\boldsymbol{\theta} - \cos\hat{\theta})^2 + (\sin\boldsymbol{\theta} - \sin\hat{\theta})^2} + \sqrt{(\cos\boldsymbol{\phi} - \cos\hat{\phi})^2 + (\sin\boldsymbol{\phi} - \sin\hat{\phi})^2}$, the cosine similarity loss is $2 - \cos\Delta\theta - \cos\Delta\phi$. Their corresponding gradients are visualized in Figs. 2a to 2d. In Fig. 2a, the w_θ and w_ϕ are set to 1 for better visualization.

The plots in Fig. 2 show that for L_1 distance and chord length distance loss functions, the gradient remains constant or increases as $\Delta\theta$ and $\Delta\phi$ decrease. This is not the desirable output for effective network training when the predicted orientation is close to the actual value. For the cosine similarity loss, the gradient nears 0 when $\Delta\theta$ and $\Delta\phi$ approach π, causing the backpropagation process to stall. On the contrary, as depicted in Fig. 2a, these drawbacks do not exist in our proposed loss function. The experimental results also validate the merit of our proposed 3D orientation loss.

Design of Dynamic Bounding Boxes: It may not be appropriate to consider the minutiae detection task as a fixed or center point detection task, primarily due to the obscure definition of the pinpoint location of minutiae since the minutiae ground truths (locations) labeled by different experts can itself be different. Such inconsistencies have been widely considered in several references [46]. Therefore, as argued in earlier studies, the exact location of minutiae ground truth should not be regarded as a point but a region: in [4], 12×12 pixels around the minutiae ground truth are labeled as a high probability region. In comparison, [23] assumes the minutiae ground truth is located in a circular region with a radius of 3 pixels. Our work is also motivated by such a consideration of the ground truth regions. We assign a square bounding box to each minutiae ground truth; the pixels within the bounding boxes are regarded as effective regions of the minutiae. We observe that the size of such an effective minutiae region is highly related to the ridge-valley distance of that region, which is ignored in previous studies [4], [23] that *fix* the region of minutiae ground truth. We therefore propose dynamic size bounding boxes for minutiae ground truth. We partition the image into 24×24 pixels patches and compute the average ridge-valley distance within each patch. The width of the square bounding box of a minutiae ground truth equals the average ridge-valley distance of the patch where the respective minutiae are located.

Inference: The candidates for 3D minutiae templates are selected from features extracted by FingerYolo3D. We remove the candidates when their predicted confidence score is lower than the threshold of $t_{conf} = 0.1$. Non-max suppression algorithm [9] is then used to eliminate the repeatedly predicted boxes for the same minutiae as per the threshold of $t_{iou} = 0.1$. Finally, from the remaining bounding boxes, we acquire the 2D location, size, depth, minutiae type, elevational and azimuthal angle of minutiae. To extract 2D minutiae by FingerYolo2D, we follow the same steps but ignore the predicted depth and elevational orientation.

4 Minutiae Matching in 3D Space

This section describes the perspective transformation compensation mechanism (PTCM) that can be used for both 3D-2D and 3D-3D minutiae template matching. There are four key steps in PTCM: i) find the matching correspondences between source and target template minutiae using a graph neural network (GNN), ii) identify and remove the outlier matching correspondences using the proposed tetrahedron based outlier matching removal algorithm, iii) compute the optimal transformation parameters from the inlier matching correspondences and align the input templates, iv) project the aligned templates on a 2D plane and use off-the-shelf 2D minutiae template matcher to compute the match score. Each of these steps is detailed in the following paragraph, and Fig. 3 visualizes the PTCM framework.

We first define some variables and minutiae representations in 3D and 2D space. In this section and Sec. 5, the locations and orientations of 3D minutia are represented by $\vec{s} \equiv (x, y, z)$, $\vec{o} \equiv (o_x, o_y, o_z)$ $s.t.$ $\vec{o}\vec{o}^T = 1$ to facilitate computational simplicity in 3D space. Considering that the common notation of 3D minutiae is $(x, y, z, \theta, \phi, t)$ [32] and also the extracted minutiae by FingerYolo3D are represented in this format, the equivalent representation of θ, ϕ to $\vec{s}, \vec{o}$ can be achieved from Eq. (3), and more details on such representation appear in Appendix E. Similarly, the common representation for 2D minutiae is (x, y, θ, t), while the locations and orientations of 2D minutia are stated in this section as $\vec{s} \equiv (x, y)$ and $\vec{o} \equiv (o_x, o_y)$, where $o_x = \cos\theta$, $o_y = \sin\theta$. We denote the 3D source template with m minutiae as $\boldsymbol{T}_1 \equiv \langle \boldsymbol{X}_1, \boldsymbol{\Theta}_1 \rangle \in \mathbb{R}^{m\times 6}$, where $\boldsymbol{X}_1, \boldsymbol{\Theta}_1 \in \mathbb{R}^{m\times 3}$ $s.t.$ $(\boldsymbol{\Theta}_1 \odot \boldsymbol{\Theta}_1)\boldsymbol{J}_{3,1} = \boldsymbol{J}_{m,1}$. The 3D target template with n minutiae is denoted as $\boldsymbol{T}_2 \equiv \langle \boldsymbol{X}_2, \boldsymbol{\Theta}_2 \rangle \in \mathbb{R}^{n\times 6}$, where $\boldsymbol{X}_2, \boldsymbol{\Theta}_2 \in \mathbb{R}^{n\times 3}$ $s.t.$ $(\boldsymbol{\Theta}_2 \odot \boldsymbol{\Theta}_2)\boldsymbol{J}_{3,1} = \boldsymbol{J}_{n,1}$. The 2D target template with n minutiae is denoted as $\boldsymbol{T}_2 \equiv \langle \boldsymbol{X}_2, \boldsymbol{\Theta}_2 \rangle \in \mathbb{R}^{n\times 4}$, where $\boldsymbol{X}_2, \boldsymbol{\Theta}_2 \in \mathbb{R}^{n\times 2}$ $s.t.$ $(\boldsymbol{\Theta}_2 \odot \boldsymbol{\Theta}_2)\boldsymbol{J}_{2,1} = \boldsymbol{J}_{n,1}$. The $\boldsymbol{J}_{i,j}$ means all-ones matrix of $i \times j$, $\odot$ denotes Hadamard product.

$$o_x = \cos\theta\cos\phi; \quad o_y = \sin\theta\cos\phi; \quad o_z = \sin\phi \tag{3}$$

Minutiae Correspondences: Earlier work [36] has incorporated GNN to locate the correspondences among the minutiae of two matched templates. However, the feature information extracted by CNN is integrated into such an approach, and therefore, their proposed method cannot find correspondence of the minutiae *without* textural information. Such an approach [36] is impractical as only the minutiae templates are stored and retrieved in most fingerprint-based systems. We therefore propose a GNN that can locate the minutiae correspondences, between 3D-3D and 3D-2D templates *without* textural features. We follow the same protocol with the kNN strategy in [36] to construct the graph from 2D minutiae templates and extend the protocol to 3D scenarios to construct the graph from 3D minutiae templates. The constructed graph is fed into GNN with three PPFConv [48] layers and two SageConv [51] layers in sequence, and we detail the forward propagation of the GNN in Eq. (5). We use $\boldsymbol{\psi}_{i,j}$ in Eq. (4) to denote the Point Pair Feature, an antisymmetric 4D descriptor, that represents the pair-wise spatial information between nodes of minutiae $\boldsymbol{p}_i = (\vec{s_i}, \vec{o_i})$ and $\boldsymbol{p}_j = (\vec{s_j}, \vec{o_j})$ sharing the same edge in the constructed graph.

$$\boldsymbol{\psi}_{i,j} = (||\vec{s_i} - \vec{s_j}||_2, \angle(\vec{o_i}, \vec{s_i} - \vec{s_j}), \angle(\vec{o_j}, \vec{s_i} - \vec{s_j}), \angle(\vec{o_i}, \vec{o_j})), \tag{4}$$

where $\angle(\mathbf{v}_1, \mathbf{v}_2) = \text{atan2}(\|\mathbf{v}_1 \times \mathbf{v}_2\|, \mathbf{v}_1 \cdot \mathbf{v}_2)$.

$$\begin{aligned}
\boldsymbol{x}_i^{k+1} &\leftarrow \text{RELU}(\boldsymbol{\Omega}_n^k(\max_{j\in\mathcal{N}(i)\cup\{i\}} \boldsymbol{\Omega}_p^k(\boldsymbol{\psi}_{i,j}))) + \tilde{\mathbf{t}}, k = 0 \\
\boldsymbol{x}_i^{k+1} &\leftarrow \text{RELU}(\boldsymbol{\Omega}_n^k(\max_{j\in\mathcal{N}(i)\cup\{i\}} \boldsymbol{\Omega}_p^k(\boldsymbol{x}_j^k \otimes \boldsymbol{\psi}_{i,j}))) + \tilde{\mathbf{t}}, k = 1, 2 \\
\boldsymbol{x}_i^{k+1} &\leftarrow \boldsymbol{W}_1^k \boldsymbol{x}_i^k + \boldsymbol{W}_2^k \operatorname*{mean}_{j\in\mathcal{N}(i)} [\boldsymbol{\sigma}^k(\boldsymbol{W}_3^k \boldsymbol{x}_j^k + \boldsymbol{b}^k)], \ k = 3, 4
\end{aligned} \tag{5}$$

In Eq. (5), the $\tilde{\mathbf{t}}$ denotes the binary feature of minutiae type that represents the type of each node of the graph, $\otimes$ denotes concatenation, $\mathcal{N}(i)$ is the set of neighbors of node i, $\boldsymbol{\Omega}_n^k$ and $\boldsymbol{\Omega}_p^k$ denote two MLP encoders in the k^{th} $(k = 0, 1, 2)$ layer of PPFConv, $\boldsymbol{W}_1^k$, $\boldsymbol{W}_2^k$, $\boldsymbol{W}_3^k$, $\boldsymbol{b}^k$ are parameters and $\boldsymbol{\sigma}^k$

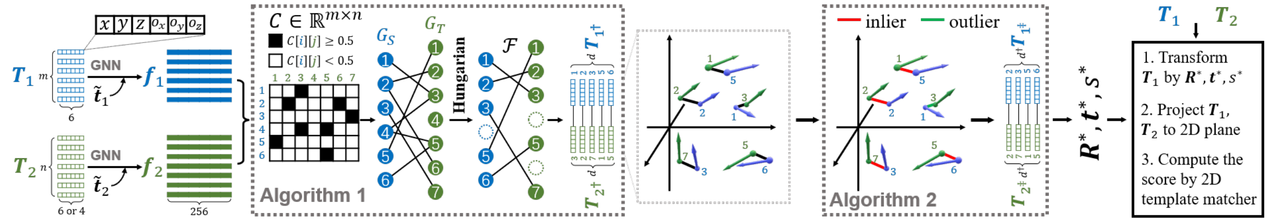


Fig. 3: Illustration for our PTCM framework, we set $m = 6$ and $n = 7$ in above figure for better visualization.

is non-linear function of the k^{th} $(k = 3, 4)$ layer of SageConv. By the GNN layers represented in Eq. (5), we acquire the high-dimensional nodes $\boldsymbol{x}_i^5 \in \mathbb{R}^{1\times 256}$ that represents the feature for each minutiae. We denote such node features of $\boldsymbol{T}_1, \boldsymbol{T}_2$ as $\boldsymbol{f}_1$, $\boldsymbol{f}_2$, where $\boldsymbol{f}_1 \in \mathbb{R}^{m\times 256}$, $\boldsymbol{f}_2 \in \mathbb{R}^{n\times 256}$.

Given the feature $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ together with the similarity matrix $\boldsymbol{C}(\boldsymbol{f}_1 \times \boldsymbol{f}_2 \to \mathbb{R}^{m\times n})$ which is computed using Eq. (6), we optimize our GNN model following the [36] by minimizing the difference between the $\boldsymbol{C}$ and the ground truth for correspondences.

$$\boldsymbol{C} = \frac{1}{2}\left(\frac{\boldsymbol{f}_1}{\sqrt{(\boldsymbol{f}_1 \odot \boldsymbol{f}_1)\boldsymbol{J}_{256,256}}} \cdot \frac{\boldsymbol{f}_2^T}{\sqrt{\boldsymbol{J}_{256,256}(\boldsymbol{f}_2^T \odot \boldsymbol{f}_2^T)}} + 1\right) \tag{6}$$

To find the one-to-one match correspondence between the features of graph nodes, [36] uses a coarse procedure to generate the results from the distance map, and therefore their selected match pairs may not be optimal. In this paper, we adopt the Hungarian algorithm [50] to ascertain the optimal one-to-one assignment $\mathcal{F} : \boldsymbol{f}_1 \to \boldsymbol{f}_2$ such that the total cost $\sum_{\boldsymbol{f}_i \in \boldsymbol{f}_1} \boldsymbol{C}(\boldsymbol{f}_i, \mathcal{F}(\boldsymbol{f}_i))$ is maximized. The details of steps are listed in Alg. 1 which output minutia $\boldsymbol{T}_{1^\dagger}$ and its match correspondence $\boldsymbol{T}_{2^\dagger}$ with the same size, [i,:] denotes the operation to get the i^{th} row of matrix. The $\boldsymbol{T}_{1^\dagger} \equiv \langle \boldsymbol{X}_{1^\dagger}, \boldsymbol{\Theta}_{1^\dagger} \rangle \in \mathbb{R}^{d\times 6}$, $\boldsymbol{T}_{2^\dagger} \equiv \langle \boldsymbol{X}_{2^\dagger}, \boldsymbol{\Theta}_{2^\dagger} \rangle \in \mathbb{R}^{d\times 6}$ (3D template) or $\in \mathbb{R}^{d\times 4}$ (2D template).

Algorithm 1 Computing Optimal One-to-One Minutia Match

```
Input: $C \in \mathbb{R}^{m\times n}$, $T_1 \in \mathbb{R}^{m\times 6}$, $T_2 \in \mathbb{R}^{n\times 6}$ (3D) or $\in \mathbb{R}^{n\times 4}$ (2D)
Output: the match pair $T_{1^\dagger}, T_{2^\dagger}$; the number of matches $d$
Initialize empty node lists $G_S, G_T$, empty edge list $E_w$
for $i$ in range $(m)$ do
    for $j$ in range $(n)$ do
        if $C[i][j] \geq 0.5$ then
            Add $node_1$ to $G_S$, set $node_1$.val $= i$
            Add $node_2$ to $G_T$, set $node_2$.val $= j$
            Add $edge$ to $E_w$
            Set $edge$.weight $= C[i][j]$
            Set $edge$.left$=node_1$, $edge$.right$=node_2$
Construct bipartite graph $\mathcal{B}$ from $(G_S, G_T, E_w)$.
Find the optimal assignment $\mathcal{F}$ from $\mathcal{B}$ using Hungarian
    algorithm [50], the number of optimal matches computed $\to$
    $d$, the edges of optimal matches $\to w_1, w_2, \ldots, w_d$
Initialize $T_{1^\dagger} \in \mathbb{O}^{d\times 6}$
Initialize $T_{2^\dagger} \in \mathbb{O}^{d\times 6}$ (3D template) or $\in \mathbb{O}^{d\times 4}$ (2D)
for $i$ in range($d$) do
    idx_s $= w_i$.left.val, idx_t $= w_i$.right.val
    $T_{1^\dagger}[i,:] = T_1[\text{idx}_s,:]$, $T_{2^\dagger}[i,:] = T_2[\text{idx}_t,:]$
```

Find the Optimal Transformation from Matches: Minutiae template with the limited number of minutia is sparse point cloud, and the orientations of minutia are critical to matching. Therefore, common point cloud registration methods cannot be directly incorporated to align the 3D-3D minutiae templates, let alone align the 3D-2D minutiae templates. In this part, we develop a tetrahedron-based inlier match selection algorithm that utilizes the locational and directional information of minutiae to remove the outlier matches between $\boldsymbol{T}_{1^\dagger}$ and $\boldsymbol{T}_{2^\dagger}$.

Let us assume that q matches are selected from the total of d matches between $\boldsymbol{T}_{1^\dagger}$, $\boldsymbol{T}_{2^\dagger}$ to find the optimal transformation parameters, and we denote the selected minutiae as $\langle \boldsymbol{X}_a^T, \boldsymbol{\Theta}_a^T \rangle$, $\langle \boldsymbol{X}_b^T, \boldsymbol{\Theta}_b^T \rangle$ respectively. The objective functions for the optimization of 3D-3D matches and 3D-2D matches are illustrated in Eq. (7) and Eq. (8) respectively. We denote the weight for the importance of orientations as ϑ, and denote the Frobenius norm of matrix by $\|\cdot\|_F$. The $\theta_x, \theta_y, \theta_z$ represent angles of roll, pitch, and yaw angle rotating operation in 3D space respectively. We compute the optimal $\boldsymbol{R}^*, \boldsymbol{t}^*, s^*$ by minimizing the objective function which is subject to the respective constraints and Powell algorithm [52] is adopted for the optimization.

$$\begin{cases} \boldsymbol{R} = e^{\theta_z \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}} e^{\theta_y \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}} e^{\theta_x \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}} \in SO(3) \\ \boldsymbol{\Upsilon} = \begin{bmatrix} s\boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_a^T & \frac{\sqrt{\vartheta}}{s}\boldsymbol{\Theta}_a^T \\ \boldsymbol{1}^T & \boldsymbol{0}^T \end{bmatrix} - \begin{bmatrix} \boldsymbol{X}_b^T & \sqrt{\vartheta}\boldsymbol{\Theta}_b^T \\ \boldsymbol{1}^T & \boldsymbol{0}^T \end{bmatrix} \\ \boldsymbol{\Phi} = \begin{bmatrix} \frac{1}{s}\boldsymbol{R}^{-1} & \frac{-1}{s}\boldsymbol{R}^{-1}\boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_b^T & s\sqrt{\vartheta}\boldsymbol{\Theta}_b^T \\ \boldsymbol{1}^T & \boldsymbol{0}^T \end{bmatrix} - \begin{bmatrix} \boldsymbol{X}_a^T & \sqrt{\vartheta}\boldsymbol{\Theta}_a^T \\ \boldsymbol{1}^T & \boldsymbol{0}^T \end{bmatrix} \\ \boldsymbol{t} \in \mathbb{R}^{3\times 1},\ \boldsymbol{X}_a, \boldsymbol{\Theta}_a, \boldsymbol{X}_b, \boldsymbol{\Theta}_b \in \mathbb{R}^{q\times 3} \\ s.t.\ (\boldsymbol{\Theta}_a \odot \boldsymbol{\Theta}_a)\boldsymbol{J}_{3,1} = \boldsymbol{J}_{q,1}, (\boldsymbol{\Theta}_b \odot \boldsymbol{\Theta}_b)\boldsymbol{J}_{3,1} = \boldsymbol{J}_{q,1} \\ \boldsymbol{R}^*, \boldsymbol{t}^*, s^* = \arg\min_{\boldsymbol{R},\boldsymbol{t},s}\ (\|\boldsymbol{\Upsilon}\|_F + \|\boldsymbol{\Phi}\|_F) \end{cases} \tag{7}$$

Scenario 3D-3D: We assume that the 3D minutia $\langle \boldsymbol{X}_a, \boldsymbol{\Theta}_a \rangle$ are transformed by $s, \boldsymbol{R}, \boldsymbol{t}$ to align the 3D minutia $\langle \boldsymbol{X}_b, \boldsymbol{\Theta}_b \rangle$, the $SO(3)$ denotes the group of all rotation matrices with respect to the origin of three-dimensional Euclidean space. We use $\boldsymbol{\Upsilon}$ in Eq. (7) to represent the weighted difference between the transformed $\langle \boldsymbol{X}_a, \boldsymbol{\Theta}_a \rangle$ and original $\langle \boldsymbol{X}_b, \boldsymbol{\Theta}_b \rangle$ that underlines the importance of locational difference and orientational difference. Symmetrically, we inverse the transform on $\langle \boldsymbol{X}_b, \boldsymbol{\Theta}_b \rangle$ to match $\langle \boldsymbol{X}_a, \boldsymbol{\Theta}_a \rangle$, and the weighted difference between original $\langle \boldsymbol{X}_a, \boldsymbol{\Theta}_a \rangle$ and transformed $\langle \boldsymbol{X}_b, \boldsymbol{\Theta}_b \rangle$ is denoted as $\boldsymbol{\Phi}$ in Eq. (7). The Frobenius norms of $\boldsymbol{\Upsilon}$, $\boldsymbol{\Phi}$ are jointly optimized.

$$\begin{cases} \boldsymbol{R}^T = \begin{bmatrix} \cos\theta_z\cos\theta_y & \sin\theta_z\cos\theta_y \\ \cos\theta_z\sin\theta_y\sin\theta_x - \sin\theta_z\cos\theta_x & \cos\theta_z\cos\theta_x + \sin\theta_z\sin\theta_y\sin\theta_x \\ \sin\theta_z\sin\theta_x + \cos\theta_z\sin\theta_y\cos\theta_x & \sin\theta_z\sin\theta_y\cos\theta_x - \cos\theta_z\sin\theta_x \end{bmatrix} \\ \boldsymbol{\Upsilon} = s\boldsymbol{R}\boldsymbol{X}_a^T + \boldsymbol{t} - \boldsymbol{X}_b^T, \boldsymbol{\Phi} = \frac{\sqrt{\vartheta}\boldsymbol{R}\boldsymbol{\Theta}_a^T}{\sqrt{\boldsymbol{J}_{2,2}(\boldsymbol{R}\boldsymbol{\Theta}_a^T \odot \boldsymbol{R}\boldsymbol{\Theta}_a^T)}} - \sqrt{\vartheta}\boldsymbol{\Theta}_b^T \\ \boldsymbol{t} \in \mathbb{R}^{2\times 1},\ \boldsymbol{X}_a, \boldsymbol{\Theta}_a \in \mathbb{R}^{q\times 3}, \boldsymbol{X}_b, \boldsymbol{\Theta}_b \in \mathbb{R}^{q\times 2} \\ s.t.\ (\boldsymbol{\Theta}_a \odot \boldsymbol{\Theta}_a)\boldsymbol{J}_{3,1} = \boldsymbol{J}_{q,1}, (\boldsymbol{\Theta}_b \odot \boldsymbol{\Theta}_b)\boldsymbol{J}_{2,1} = \boldsymbol{J}_{q,1} \\ \boldsymbol{R}^*, \boldsymbol{t}^*, s^* = \arg\min_{\boldsymbol{R},\boldsymbol{t},s}\ (\|\boldsymbol{\Upsilon}\|_F + \|\boldsymbol{\Phi}\|_F) \end{cases} \tag{8}$$

Algorithm 2 Tetrahedron-based Outlier Match Removal

```
Input: $\boldsymbol{T}_{1^\dagger}, \boldsymbol{T}_{2^\dagger}, d, t_v, t_l$
Output: Inlier minutiae matches $\boldsymbol{T}_{1^\ddagger}, \boldsymbol{T}_{2^\ddagger}$
Initialize $T_V = 0.9 \times d$, $p = 0.99$, N $= 1000$, count $= 0$
Initialize idx=0, maxNum=0, empty list $l_{\text{inlier}} = [\,]$
while count $\leq$ N and $d \geq 5$ do
    while True do
        Randomly select 4 different numbers $i_1, i_2, i_3, i_4$ from $\{1, 2, \ldots, d\}$, denote $l_{\text{idx}} = [i_1, i_2, i_3, i_4]$
        Initialize $\boldsymbol{M}_a, \boldsymbol{Z}_a \in \mathbb{0}^{4\times 3}$, initialize $\boldsymbol{M}_b, \boldsymbol{Z}_b \in \mathbb{0}^{4\times 3}$ if 3D-3D, initialize $\boldsymbol{M}_b, \boldsymbol{Z}_b \in \mathbb{0}^{4\times 2}$ if 3D-2D
        for $row$ in $l_{\text{inlier}}$ do
            $\boldsymbol{M}_a[row,:] = \boldsymbol{X}_{1^\dagger}[row,:]$,$\boldsymbol{Z}_a[row,:] = \boldsymbol{\Theta}_{1^\dagger}[row,:]$
            $\boldsymbol{M}_b[row,:] = \boldsymbol{X}_{2^\dagger}[row,:]$,$\boldsymbol{Z}_b[row,:] = \boldsymbol{\Theta}_{2^\dagger}[row,:]$
        Compute $\overline{e}$ from $\boldsymbol{M}_a$ using Eq. (9)
        if $\overline{e} > t_l$ then
            Te$_1 = \langle \boldsymbol{M}_a, \boldsymbol{Z}_a \rangle$, Te$_2 = \langle \boldsymbol{M}_b, \boldsymbol{Z}_b \rangle$; break
    set $\langle \boldsymbol{X}_a, \boldsymbol{\Theta}_a \rangle =$ Te$_1$, $\langle \boldsymbol{X}_b, \boldsymbol{\Theta}_b \rangle =$ Te$_2$, $q$=4, compute $\boldsymbol{R}^*_{te}$, $\boldsymbol{t}^*_{te}$, $s^*_{te}$ by Eq. (7) if 3D-3D (by Eq. (8) if 3D-2D)
    Initialize empty list $l_{\text{temp}} = [\,]$
    for $i$ in range($d$) do
        $\langle \boldsymbol{X}_a, \boldsymbol{\Theta}_a \rangle = \boldsymbol{T}_{1^\dagger}[i,:]$, $\langle \boldsymbol{X}_b, \boldsymbol{\Theta}_b \rangle = \boldsymbol{T}_{2^\dagger}[i,:]$
        if 3D-3D matching then
            set $q$=1, compute $\overline{v} = \|\boldsymbol{\Upsilon}\|_F + \|\boldsymbol{\Phi}\|_F$ by Eq. (7)
        if 3D-2D matching then
            set $q$=1, compute $\overline{v} = \frac{\|\boldsymbol{\Upsilon}\|_F}{2} + \frac{\|\boldsymbol{\Phi}\|_F}{2}$ by Eq. (8)
        if $\overline{v} < t_v$ then Add $i$ to $l_{\text{temp}}$
    if length of $l_{\text{temp}} >$ maxNum then
        maxNum $=$ length of $l_{\text{temp}}$; idx = count
    if length of $l_{\text{temp}} > T_V$ then
        Add $l_{\text{temp}}$ to $l_{\text{inlier}}$; break
    $\overline{r} =$ (length of $l_{\text{temp}}$) $/d$
    Add $l_{\text{temp}}$ to $l_{\text{inlier}}$; count = count + 1
    N $= log(1 - p + 10^{-8})/log(1 - \overline{r}^4 + 10^{-8})$
if $d < 5$ then
    $\boldsymbol{T}_{1^\ddagger} = \boldsymbol{T}_{1^\dagger}$, $\boldsymbol{T}_{2^\ddagger} = \boldsymbol{T}_{2^\dagger}$
else if $d \geq 5$ then
    $d^\dagger$ = length of $l_{\text{inlier}}$[idx], inlP = $l_{\text{inlier}}$[idx], $\mathbb{0}^{d^\dagger\times 6} \rightarrow \boldsymbol{T}_{1^\ddagger}$
    $\mathbb{0}^{d^\dagger\times 6} \rightarrow \boldsymbol{T}_{2^\ddagger}$ for 3D-3D, $\mathbb{0}^{d^\dagger\times 4} \rightarrow \boldsymbol{T}_{2^\ddagger}$ for 3D-2D
    for $i$ in range($d^\dagger$) do
        $\boldsymbol{T}_{1^\ddagger}[i,:] = \boldsymbol{T}_{1^\dagger}$[inlP[$i$],:], $\boldsymbol{T}_{2^\ddagger}[i,:] = \boldsymbol{T}_{2^\dagger}$[inlP[$i$],:]
```

Scenario 3D-2D: We assume that the 3D minutia $\langle \boldsymbol{X}_a, \boldsymbol{\Theta}_a \rangle$ are transformed by $s, \boldsymbol{R}, \boldsymbol{t}$ to match the 2D minutia $\langle \boldsymbol{X}_b, \boldsymbol{\Theta}_b \rangle$, we use $\boldsymbol{\Upsilon}$ in Eq. (8) to represent the difference between the transformed $\boldsymbol{X}_a$ and original $\boldsymbol{X}_b$, we use $\boldsymbol{\Phi}$ in Eq. (8) to represent the weighted orientational difference between the transformed $\boldsymbol{\Theta}_a$ and original $\boldsymbol{\Theta}_b$. The Frobenius norms of $\boldsymbol{\Upsilon}$, $\boldsymbol{\Phi}$ are jointly optimized. It should be noted that the irreversible rotation matrix $\boldsymbol{R}$ in Eq. (8) rotates 3D minutiae to align the 2D minutiae and is therefore totally different from the $\boldsymbol{R} \in SO(3)$ in Eq. (7).

Outlier Selection: If we set $q = d$ (i.e. use all match pairs) to compute the transformation parameters, it is observed that the outlier of spurious matches can adversely impact the alignment and lead to the discrepancy of the result. Therefore, the tetrahedron-based outlier match removal algorithm is developed as Alg. 2 which can remove outlier matches for both 3D-3D and 3D-2D minutiae template matching.

We iteratively select four random minutia from the source points $\boldsymbol{T}_{1^\dagger}$ to construct tetrahedron Te$_1 \equiv \langle \boldsymbol{M}_a, \boldsymbol{Z}_a \rangle$, and the average length $\overline{e}$ (computed using Eq. (9)) of edges of Te$_1$ should be longer than threshold t_l. We use $\boldsymbol{M}_1, \boldsymbol{Z}_1 \in \mathbb{R}^{4\times 3}$ to denote the locations and vertex directions of selected tetrahedron Te$_1$. In target points $\boldsymbol{T}_{2^\dagger}$, the corresponding matched points of Te$_1$ are denoted as Te$_2 \equiv \langle \boldsymbol{M}_b, \boldsymbol{Z}_b \rangle$. If the target minutiae template is 3D (2D), the $\boldsymbol{M}_b, \boldsymbol{Z}_b \in \mathbb{R}^{4\times 3}$ ($\in \mathbb{R}^{4\times 2}$). We compute the parameters $\boldsymbol{R}^*_{te}, \boldsymbol{t}^*_{te}, s^*_{te}$ to transform Te$_1$ to Te$_2$, then we use acquired $\boldsymbol{R}^*_{te}, \boldsymbol{t}^*_{te}, s^*_{te}$ to transform $\boldsymbol{T}_{1^\dagger}$. The number of *similar* minutiae pairs between the transformed $\boldsymbol{T}_{1^\dagger}$ and $\boldsymbol{T}_{2^\dagger}$ is computed. These *similar* pairs are defined by the similarity score $\overline{v}$, which is computed in Alg. 2 and falls below the predetermined threshold t_v. In this paper, we set $t_l = 80$ and $t_v = 12$. We repeat the above steps in Alg. 2 within repetition and select the minutiae matches in $\boldsymbol{T}_{1^\dagger}, \boldsymbol{T}_{2^\dagger}$ with the largest number $d^\dagger$ of *similar* pairs between transformed $\boldsymbol{T}_{1^\dagger}$ and $\boldsymbol{T}_{2^\dagger}$ as inlier matches $\boldsymbol{T}_{1^\ddagger} \in \mathbb{R}^{d^\dagger\times 6}$, $\boldsymbol{T}_{2^\ddagger} \in \mathbb{R}^{d^\dagger\times 6}$ (for 3D template) or $\boldsymbol{T}_{2^\ddagger} \in \mathbb{R}^{d^\dagger\times 4}$ (for 2D template).

$$\overline{e} = \text{mean}\sqrt{(\boldsymbol{M}\odot\boldsymbol{M})\boldsymbol{J}_{3,4}+\boldsymbol{J}_{3,4}{}^T(\boldsymbol{M}^T\odot\boldsymbol{M}^T)-2\boldsymbol{M}\boldsymbol{M}^T} \qquad (9)$$

We explain the reason for only selecting the candidate tetrahedron Te$_1$ with the average length $\overline{e}$ longer than t_l in Alg. 2. In the experiments, we observe that if the size of selected tetrahedron is small, a significant error will be introduced to $\boldsymbol{R}^*_{te}$. We analyze this phenomenon for 3D-3D matching as an example and for simplification, we assume that the scale and centroids of $\boldsymbol{M}_1$, $\boldsymbol{M}_2$ have already been aligned, the $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$ are not considered in computing $\boldsymbol{R}^*_{te}$. The optimal rotation matrix $\boldsymbol{R}^*_{te}$ between rigid transformed vertices $\boldsymbol{M}_a$, $\boldsymbol{M}_b \in \mathbb{R}^{4\times 3}$ can be computed as follows:

$$\boldsymbol{H} = \boldsymbol{M}_a^T\boldsymbol{M}_b; \boldsymbol{H} \xrightarrow{\text{SVD}} \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T; \boldsymbol{R}^*_{te} = \boldsymbol{V}\text{diag}(1,1,\text{det}(\boldsymbol{V}\boldsymbol{U}^T))\boldsymbol{U}^T \qquad (10)$$

We denote the 3D locational ground truth of source tetrahedron as $\boldsymbol{M}_{gt1}$ and that of target template as $\boldsymbol{M}_{gt2} = \boldsymbol{R}_{gt}\boldsymbol{M}_{gt1}$, where $\boldsymbol{R}_{gt}$ is for the rotation matrix ground truth. When we set $\boldsymbol{M}_a = \boldsymbol{M}_{gt1}$ and $\boldsymbol{M}_b = \boldsymbol{R}_{gt}\boldsymbol{M}_{gt1}$ in Eq. (10), the acquired $\boldsymbol{R}^*_{te}$ should be the same as $\boldsymbol{R}_{gt}$. However, in the experiments, we find that the exact location of detected 3D minutiae deviates from the ground truth of location, such deviation $\boldsymbol{\delta}$ can be modelled using Gaussian distribution $N(0, \sigma_n^2)$ and σ_n is estimated from real databases. We assume that the $\boldsymbol{\delta}$ in every extracted 3D minutiae template follows the same distribution. When we set $\boldsymbol{M}_a = \boldsymbol{M}_{gt1} + \boldsymbol{\delta}$ and $\boldsymbol{M}_b = \boldsymbol{R}_{gt}\boldsymbol{M}_{gt1} + \boldsymbol{\delta}$ in Eq. (10), the acquired $\boldsymbol{R}^*_{te}$ deviates from $\boldsymbol{R}_{gt}$, and we evaluate the deviation using angle difference $\Delta\boldsymbol{\alpha}$ by Eq. (11) between $\boldsymbol{R}^*_{te}$ and $\boldsymbol{R}_{gt}$, which is also commonly [54] adopted to evaluate the difference between rotation matrices.

$$\Delta\boldsymbol{\alpha} = |\boldsymbol{R}^*_{te}, \boldsymbol{R}_{gt}| = \arccos((\text{tr}(\boldsymbol{R}^*_{te}\boldsymbol{R}_{gt}^{-1}) - 1)/2) \qquad (11)$$

In the following part, we investigate the various distributions of error $\Delta\alpha$ introduced to $\boldsymbol{R}^*_{te}$ as the size of candidate tetrahedron Te$_1$ in Alg. 2 changes. We propose Theorem 1 to deduce Corollary 1, which ascertains the relationship between the distribution of vertices of Gaussian-distributed random tetrahedrons and the distribution of the lengths of tetrahedrons' edges.

Theorem 1. *We denote the randomly distributed Gaussian points in k-dimensional space as $X_k \sim N(0,\sigma_k^2)$, the Euclidean distances between X_k as L_k, the cumulative distribution function (cdf) of L_k as $P(l, k) = Probability(L_k \leq l)$, the probability distribution function (pdf) of L_k as $f(l, k)$, the expectation of L_k as $E(l, k)$. The $E(l, k)$ can be explicitly represented by σ_k, and the monotonic* *increasing function $h_k : \sigma_k \rightarrow E(l,k)$ is a one-to-one correspondence between $E(l,k)$ and σ_k. We can write the inverse mapping as $h_k^{-1} : E(l,k) \rightarrow \sigma_k$, which is also monotonic increasing.*

Proof. Inspired by previous studies in [49], we provide cdf $P(l,k)$, pdf $f(l,k)$ and expectation $E(l,k)$ of L_k as:

$$P(l,k) = \begin{cases} 1 - \dfrac{\Gamma\left(k/2, l^2/4\sigma_k^2\right)}{\Gamma(k/2)}, & l \geq 0 \\ 0, & l < 0 \end{cases} \tag{12}$$

$$f(l,k) = \frac{\partial}{\partial l}P(l,k) = \begin{cases} \dfrac{2^{1-k}e^{-l^2/4\sigma_k^2}\left(\frac{l}{\sigma_k}\right)^{k-1}}{\sigma_k\Gamma(k/2)}, & l \geq 0 \\ 0, & l < 0 \end{cases} \tag{13}$$

$$E(l,k) = \int_{-\infty}^{\infty} l f(l,k)\mathrm{d}l = \frac{2\sigma_k\Gamma\left(\frac{k+1}{2}\right)}{\Gamma\left(\frac{k}{2}\right)} \tag{14}$$

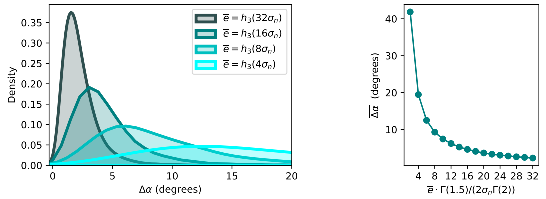
, where $\Gamma(\cdot)$ is gamma function and $\Gamma(\cdot,\cdot)$ is the lower incomplete gamma function. From Eq. (14), we can conclude that both $h_k(\cdot)$ and $h_k^{-1}(\cdot)$ are all bijective functions. □

Corollary 1. *Based on Theorem 1, we can randomly generate four three-dimensional points following the normal distribution of $N(0, h_3^{-1}(\overline{e}))$ to represent the tetrahedron whose expected average length of edges is equal to $\overline{e}$. In 3D space, $h_3(x) = 2x\Gamma(2)/\Gamma(1.5)$, $h_3^{-1}(y) = y\Gamma(1.5)/(2\Gamma(2))$.*

Based on Corollary 1, we can randomly sample 3D points following normal distribution $N(0, \sigma_t^2)$ to simulate the locational distribution of vertices of tetrahedron whose average length $\overline{e}$ of edges is $h_3(\sigma_t)$. To sample the distribution of $\Delta\alpha$, we generate a large number $\mathcal{J}$ of random tetrahedrons $\boldsymbol{M}_i \in N(0, \sigma_t^2)$, and the respective $\mathcal{J}$ random ground truth rotation matrices $\boldsymbol{R}_{gt_i} \in SO(3)$ while the σ_n of noise $\boldsymbol{\delta}$ on coordinates of vertices is fixed. We set $\boldsymbol{M}_a = \boldsymbol{M}_i + \boldsymbol{\delta}$, $\boldsymbol{M}_b = \boldsymbol{R}_{gt_i}\boldsymbol{M}_i + \boldsymbol{\delta}$ in Eq. (10) and acquire the corresponding $\boldsymbol{R}_{te_i}^*$, the $\Delta\alpha_i = |\boldsymbol{R}_{te_i}^*, \boldsymbol{R}_{gt_i}|$ is computed from Eq. (11). The samples of distribution of $\Delta\alpha$ can be represented by $\Delta\alpha_i$ $(i = 1, 2, \ldots, \mathcal{J})$ and the expectation $\overline{\Delta\alpha}$ can be estimated from the average value of $\Delta\alpha_i$.

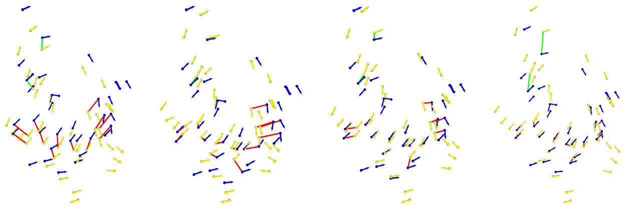
In Fig. 4a, we set $\sigma_t = 32\sigma_n, 16\sigma_n, 8\sigma_n, 4\sigma_n$ respectively and plot the corresponding distributions of $\Delta\alpha$. We observe from Fig. 4a that the error $\Delta\alpha$ tends to increase as the size (represented by $\overline{e}$) of tetrahedron decreases from $h_3(32\sigma_n)$ to $h_3(4\sigma_n)$. In Fig. 4b, the x-axis represents the magnitude of $h_3^{-1}(\overline{e}) = \overline{e} \cdot \Gamma(1.5)/(2\sigma_n\Gamma(2))$, which is proportional to $\overline{e}$, ranging from 2 to 32 with the interval of 2. As $\overline{e}$ increases, the expectation $\overline{\Delta\alpha}$ drops. It can be reasonably concluded that tetrahedrons with short average lengths of edges will introduce more significant deviation during the computing of $\boldsymbol{R}_{te}^*$, therefore, in Alg. 2, we empirically set the threshold $t_l = h_3(8\sigma_n)$ to select candidate tetrahedrons according to average length of edges to balance the speed of candidate tetrahedron selection and precise calculation of $\boldsymbol{R}_{te}^*$.

Computation of Match Score: For 3D-3D (3D-2D) matching, we calculate the final result of optimal transformation parameters $\boldsymbol{R}^*, \boldsymbol{t}^*, s^*$ between $\boldsymbol{T}_1, \boldsymbol{T}_2$ by setting $\langle \boldsymbol{X}_a^T, \boldsymbol{\Theta}_a^T \rangle = \boldsymbol{T}_{1\ddagger}$, $\langle \boldsymbol{X}_b^T, \boldsymbol{\Theta}_b^T \rangle = \boldsymbol{T}_{2\ddagger}$, $q = d^\dagger$ in Eq. (7) (in Eq. (8)) and Powell algorithm [52] is used for optimization. In Figs. 5 and 6, we visualize the progress for 3D-2D and 3D-3D alignment in different iterations of the Powell optimizer respectively. The blue (yellow) minutiae represents the source (target) template, red (green) lines denote inlier (outlier) match. The Powell optimizer converges at the 8th and correctly distinguishes the outlier and inlier matches.



(a) (b)

Fig. 4: (a) Comparative distributions of $\Delta\alpha$ with the changes in the size of tetrahedron (represented by $\overline{e}$), (b) observed degradation of average error $\overline{\Delta\alpha}$ with the increase in $\overline{e}$.



(a) iter:0 (b) iter:2 (c) iter:4 (d) iter:8

Fig. 5: Iterative alignment progression between 3D-2D.

We can use $\boldsymbol{R}^*, \boldsymbol{t}^*, s^*$ to transform $\boldsymbol{T}_1 \equiv \langle \boldsymbol{X}_1, \boldsymbol{\Theta}_1 \rangle$ and project the transformed $\boldsymbol{T}_1$ and the original $\boldsymbol{T}_2$ to 2D plane. This process for the 3D to 3D template matching can be mathematically represented in Eq. (15).

$$\begin{aligned} &\boldsymbol{X}_1^p = \mathbf{proj}_{\mathrm{x}}(({(s^*\boldsymbol{R}^*\boldsymbol{X}_1^T + \boldsymbol{t}^*\boldsymbol{J}_{1,m})}^T), \boldsymbol{X}_2^p = \mathbf{proj}_{\mathrm{x}}(\boldsymbol{X}_2) \\ &\boldsymbol{\Theta}_1^p = \mathbf{proj}_{\theta}(({(\boldsymbol{R}^*\boldsymbol{\Theta}_1^T)}^T), \boldsymbol{\Theta}_2^p = \mathbf{proj}_{\theta}(\boldsymbol{\Theta}_2) \\ &\mathbf{proj}_{\mathrm{x}}(\boldsymbol{X}) = \boldsymbol{X}\,[\boldsymbol{I}_2|\mathbf{0}]^T \\ &\mathbf{proj}_{\theta}(\boldsymbol{\Theta}) = \boldsymbol{\Theta}\,[\boldsymbol{I}_2|\mathbf{0}]^T / \sqrt{(\boldsymbol{\Theta}[\boldsymbol{I}_2|\mathbf{0}]^T) \odot (\boldsymbol{\Theta}[\boldsymbol{I}_2|\mathbf{0}]^T)\boldsymbol{J}_{2,2}} \end{aligned} \tag{15}$$

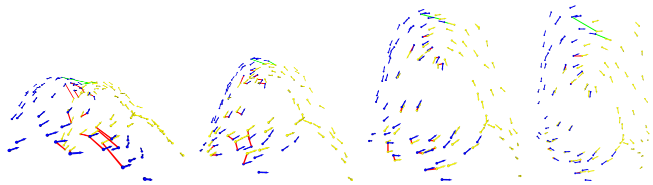
, where $\boldsymbol{I}_2$ denotes 2×2 identity matrix. For 3D-2D minutiae template matching, we can write this process in Eq. (16).

$$\begin{aligned} &\boldsymbol{X}_1^p = {(s^*\boldsymbol{R}^*\boldsymbol{X}_1^T + \boldsymbol{t}^*\boldsymbol{J}_{1,m})}^T, \boldsymbol{X}_2^p = \boldsymbol{X}_2 \\ &\boldsymbol{\Theta}_1^p = (\boldsymbol{\Theta}_1\boldsymbol{R}^{*T}) / \sqrt{(\boldsymbol{\Theta}_1\boldsymbol{R}^{*T}) \odot (\boldsymbol{\Theta}_1\boldsymbol{R}^{*T})\boldsymbol{J}_{2,2}}, \boldsymbol{\Theta}_2^p = \boldsymbol{\Theta}_2 \end{aligned} \tag{16}$$

Finally, the projected 2D templates $\langle \boldsymbol{X}_1^p, \boldsymbol{\Theta}_1^p \rangle$, $\langle \boldsymbol{X}_2^p, \boldsymbol{\Theta}_2^p \rangle$ are input to 2D minutiae template matcher and the match score is generated.

5 Uniqueness of 3D Minutiae Templates

The recovery of a 3D minutiae template from a single 2D fingerprint image is expected to significantly enhance the match accuracy for such 2D fingerprint images. Such advantage can be quantified from the uniqueness analysis of recovered 3D fingerprint templates. There has been some exciting work to quantify the individuality of 3D fingerprints using the probability of false random correspondence. The assumption of feature independence in [28] can only generate an estimate on the upper bound as such assumptions are not reasonable [18], [19], i.e. there does exist some interdependence among 3D fingerprint features, and this should be empirically estimated [32]. Therefore, our work incorporates a joint distribution mixture model for 3D minutiae templates to determine the extent of correlations between the locations and orientations of minutiae in 3D space. The probability of false random correspondences (PRC) can be more accurately



(a) iter:0 (b) iter:2 (c) iter:4 (d) iter:8

Fig. 6: Iterative alignment progression between 3D-3D.

estimated using such a model of joint feature distributions for the 3D minutiae templates.

The Joint Distribution Mixture Model: We assume that the distribution of locations of 3D minutiae follows a trivariate normal distribution $f_g^S\left(\vec{s} \mid \vec{\mu}_g, \mathbf{\Sigma}_g\right)$ and the distributions of their orientations can be modeled by Von-Mises-Fisher (VMF[1]) distribution $f_g^O\left(\vec{o} \mid \vec{\upsilon}_g, \kappa_g\right)$ shown in the Eq. (19) and can model more complex relationships as compared to those for the 2D minutiae as used in [18], [19].

$$\begin{cases} f_g^S\left(\vec{s} \mid \vec{\mu}_g, \mathbf{\Sigma}_g\right) = \frac{1}{(2\pi)^{\frac{3}{2}}\sqrt{\left|\mathbf{\Sigma}_g\right|}} e^{-\frac{1}{2}\left(\vec{s}-\vec{\mu}_g\right)^T \mathbf{\Sigma}_g^{-1}\left(\vec{s}-\vec{\mu}_g\right)} \\ f_g^O\left(\vec{o} \mid \vec{\upsilon}_g, \kappa_g\right) = \frac{\kappa_g}{4\pi \sinh \kappa_g} e^{\kappa_g \vec{\upsilon}_g^T \vec{o}} \end{cases} \tag{19}$$

We assume that the individuality of 3D minutiae templates is computed from a representative database with F contactless fingerprint images, and for *each* 3D minutiae template, we use a joint distribution mixture model to estimate the localization and directions of minutiae in 3D space. Such a joint model can estimate the nature of minutiae distribution and account for clustering tendencies and interdependence between these features in different regions of the finger surface, that is to say, for a 3D minutiae template that is composed of several components, the 3D locations with orientations follow the same distribution within the same component while varying between different components. The distribution of 3D minutiae in any η^{th} template can be estimated from Eq. (20), while G_η denotes the optimal number of components of the mixture model of the η^{th} $(\eta = 1, 2, ..., F)$ minutiae template, the τ_g represents the weight of the g^{th} $(g = 1, 2, ..., G_\eta)$ component in the mixture model, the weights of components are independently distributed with probabilities τ_1, τ_2, ..., τ_{G_η} following $\tau_j \geq 0$ and $\sum_{j=1}^{G_\eta} \tau_j = 1$, and Ξ_η denotes the unknown parameters of that mixture model including G_η, τ_g, $\vec{\mu}_g$, $\mathbf{\Sigma}_g$, $\vec{\upsilon}_g$, κ_g.

$$f\left(\vec{s}, \vec{o} \mid \Xi_\eta\right) = \sum_{g=1}^{G_\eta} \tau_g f_g^S\left(\vec{s} \mid \vec{\mu}_g, \mathbf{\Sigma}_g\right) \times f_g^O\left(\vec{o} \mid \vec{\upsilon}_g, \kappa_g\right) \tag{20}$$

To estimate the unknown parameters Ξ_η for mixture model of each 3D minutiae template, we use Expectation Maximization (EM) algorithm [53] to compute the solution similar to as in [18], [19], and the optimal number of components G_i is selected using Bayesian Information Criteria.

After a total of F joint distribution mixture models are acquired from the database, we further ascertain the class-specific phenomenon in the minutiae distribution. Reference [18] has noted that similar mixture models of 2D minutiae templates are subject to the same class, therefore mixture models belonging to different classes can vary significantly. We therefore assume that there are a total of N^* classes $C_1, C_2, \ldots, C_{N^*}$ with different proportions $|C_1|, |C_2|, \ldots, |C_{N^*}|$ for all the mixture models (model $f_i(\vec{s}, \vec{o} | \Xi_{G_i})$ for $i = 1, 2, ..., F$) in the database, and the N^* classes with their mixture models conform to the structure of hierarchical cluster tree. Therefore, considering the agglomerative hierarchical clustering in [18], a similar process is adopted to determine the optimal number N^* of all the mixture models. Then we compute the mean mixture distribution $f_{C_i}\left(\vec{s}, \vec{o} \mid \Xi_{C_i}\right)$ (abbreviated as f_{C_i}) for each class C_i in Eq. (21).

$$f_{C_i}\left(\vec{s}, \vec{o} \mid \Xi_{C_i}\right) = \frac{1}{|C_i|} \sum_{\Xi_\eta \in C_i} f\left(\vec{s}, \vec{o} \mid \Xi_\eta\right), \eta = 1, 2, \ldots, F \tag{21}$$

The steps of computing the PRC$_{3\text{d}}$ index of 3D minutiae templates from the f_{C_i} $(i = 1, 2, \ldots, N^*)$ are detailed in the following.

Estimation of PRC$_{3\text{d}}$ Index: The PRC$_{3\text{d}}$ score indicates the probability of falsely matching a pair of 3D fingerprints from the different persons. The smaller value for such PRC score between two templates indicates it is less likely to generate falsely matched results of a biometrics system which implies better individuality or the uniqueness of that biometric.

Therefore, defining the criterion for the *match* between a pair of arbitrarily selected 3D minutiae templates is important. Similar to the earlier work [18], the *match* criterion between query 3D minutiae template Q and target 3D minutiae template T is that there should be at least w pairs of matched 3D minutiae between such templates. Two minutiae are considered as matched when respective 3D minutiae $\langle \vec{s}_1, \vec{o}_1 \rangle$ and $\langle \vec{s}_2, \vec{o}_2 \rangle$ can meet both conditions $||\vec{s}_1 - \vec{s}_2||_2 \leq r_0$ and $\arccos(\vec{o}_1 \cdot \vec{o}_2) \leq d_0$ simultaneously, where the r_0 and d_0 are the empirically defined thresholds. We denote the *match* between $(\vec{s}_1, \vec{o}_1)$ and $(\vec{s}_2, \vec{o}_2)$ as $(\vec{s}_2, \vec{o}_2) \in B\left(\vec{s}_1, \vec{o}_1\right)$.

Reference [18], [19] have systematically explained that the probability $p^*(u; C_i, C_j)$ $(i, j \in 1, 2, ..., N^*)$ that u minutiae can be matched between query minutiae template with m minutia from class C_i and minutiae template with n minutia from class C_j can be modelled as Poisson distribution as follows:

$$\begin{cases} p^*(u; C_i, C_j) = e^{-\lambda(C_i, C_j)} \lambda(C_i, C_j)^u / u! \\ \lambda(C_i, C_j) = mnp(C_i, C_j) \end{cases} \tag{22}$$

where $p(C_i, C_j)$ is the probability that one randomly selected minutiae following the distribution of f_{C_i} matches one randomly selected minutiae following the distribution of f_{C_j}, and $p(C_i, C_j)$ is defined as follows while the numerical estimation of $p(C_i, C_j)$ is detailed in Appendix F.

$$\iint_{(\vec{s}_2, \vec{o}_2) \in B(\vec{s}_1, \vec{o}_1)} f_{C_i}\left(\vec{s}_1, \vec{o}_1\right) f_{C_j}\left(\vec{s}_2, \vec{o}_2\right) d\vec{s}_2 d\vec{o}_2 d\vec{s}_1 d\vec{o}_1 \tag{23}$$

[1] VMF distribution can model the distribution of random variables lies on a 3D spherical surface and extended from two-dimensional Von-Mises distribution which has been used to model the distribution of randomly distributed 2D minutiae orientations in [18], [19]. Assuming that $\vec{x}$ is p-dimensional unit vector, the probability density function is defined as:

$$f_p(\vec{x} \mid \vec{\upsilon}, \kappa) = \frac{\kappa^{\frac{p}{2}-1}}{(2\pi)^{\frac{p}{2}} \frac{p_2}{2}(\kappa)} e^{\kappa \vec{\upsilon}^T \vec{x}} \tag{17}$$

where $\kappa \geq 0$, $\|\vec{\upsilon}\| = 1$, $I_v(z) = \left(\frac{z}{2}\right)^v \sum_{k=0}^{\infty} \frac{\left(\frac{1}{4}z^2\right)^k}{k!\Gamma(v+k+1)}$. When $p = 3$, the three-dimensional random variate is distributed on a sphere:

$$f_p(\vec{x} \mid \vec{\upsilon}, \kappa) = \frac{\kappa^{\frac{1}{2}}}{(2\pi)^{\frac{3}{2}} I_{\frac{1}{2}}(\kappa)} e^{\kappa \vec{\upsilon}^T \vec{x}} = \frac{\kappa}{4\pi \sinh \kappa} e^{\kappa \vec{\upsilon}^T \vec{x}} \tag{18}$$

When $p = 2$, VMF distribution reduces to Von-Mises distribution.

In the experiments, the m is the average number of minutiae in the overlapping volume of the 3D minutiae from the whole database. The overlapping volume is the same as defined in [28], and we set $m = n$ within the same database. We can adopt the $\text{PRC}_{2\text{d}}$ index in [18] for the 3D fingerprint scenario. $\text{PRC}_{3\text{d}}(w \mid C_i, C_j)$ is the probability of observing w or more matches between query template following distribution f_{C_i} and target template following distribution f_{C_j}. The $\text{PRC}_{3\text{d}}(w \mid C_i, C_j)$ can be defined as follows:

$$\text{PRC}_{3\text{d}}(w \mid C_i, C_j) = \sum_{u \geq w} p^*(u; C_i, C_j) \tag{24}$$

The $\text{PRC}_{3\text{d}}$ value for $i, j \in 1, 2, ..., N^*$ is computed using the weighted combination of $p^*(w; C_i, C_j)$ as follows:

$$\text{PRC}_{3\text{d}}(w) = \frac{\sum_{i,j \in \{1,2,...,N^*\}} |C_i| \cdot |C_j| \cdot \text{PRC}_{3\text{d}}(w \,|\, C_i, C_j)}{\sum_{i,j \in \{1,2,...,N^*\}} |C_i| \cdot |C_j|} \tag{25}$$

6 Contactless Fingerprint Synthesis

This section explains the process of generating real-like contactless 2D fingerprint images and is organized in two stages. The first stage involves creating a 3D finger model with synthetically carved ridge valley patterns. This 3D model is then projected onto a 2D plane to obtain the ridge-valley skeletons for the contactless fingerprints, along with their corresponding 3D minutiae ground truths. The realistic contactless fingerprint images are generated in the second stage by incorporating the style transfer on these skeletons.

The 3D Finger Generation: The finger surface generation algorithm in [5] utilizes points sampled on the Bezier surface to fit the shape of real-world fingers. It is important to ensure the authenticity of ridge-valley patterns on the synthesized finger surfaces. We have empirically observed that the distance to the nearest neighbor for *any* point on the surface (i.e. the *maximum* spacing[2] of all the points) should be smaller than a specific empirical value: $0.1 \times w_{rv}$, where w_{rv} is the width of ridge-valley pattern on 3D finger surface. In this work, we empirically set w_{rv} to 0.4 mm as the height of the 3D finger surface is scaled to 2.5 cm. The surface generation method in [5] uniformly samples in the parameter domain of the Bezier surface to obtain points on the 3D finger surface. This results in a non-uniform sampling of the surface in the spatial domain. Consequently, [5] is obliged to sample dense parameters on Bezier surface to meet the maximum spacing requirement and ensure realness. However, this leads to a large number of non-uniformly sampled points and also slows down the 3D-2D texture mapping process in the subsequent steps.

Unlike the method in [5], we propose a new approach based on VGD [45] to synthesize finger surfaces. Our method synthesizes surfaces that meet the maximum spacing requirement using significantly fewer points. The specifics of this method are outlined in Alg. 3. We employ a sixteenth-order Bezier surface with 17×17 control points to model the 3D finger surface instead of concatenating four eight-order Bezier surfaces in [5], which requires that the respective G_1 continuity conditions must be strictly followed. Such constraints for G_1 continuity require extra computational resources, which offsets the performance benefits gained from using concatenated sub-surfaces of a lower order. In Alg. 3, we denote the control points as $P \in \mathbb{R}^{m \times n}$ where m, n represent the numbers for the order in two directions of Bezier surface, and we set $m = n = 16$ in this paper. The Bezier surface P_ζ and $P_{\zeta'}$ are generated using Eq. (26) where the 3D points of $P_{\zeta'}$ are represented as $(x_1, x_2, ..., x_{\zeta'})$, $(y_1, y_2, ..., y_{\zeta'})$, $(z_1, z_2, ..., z_{\zeta'})$ respectively.

$$P(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} P_{ij} B_j^n(u) B_i^m(v); u, v = 0, \frac{1}{\zeta}, \frac{2}{\zeta}, \ldots, \frac{\zeta}{\zeta} \tag{26}$$

, the $B_i^n(t)$ is the Bernstein polynomial defined in Eq. (27):

$$B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \tag{27}$$

Algorithm 3 Uniform 3D Fingerprint Surface Generation

```
Input: Controlling points $P$ of Bezier finger surface;  Required
    maximum point cloud spacing $\rho$;
Output: Uniformed 3D finger surface $\mathcal{S}$
Generate Bezier surface $P_\zeta$ of $\zeta \times \zeta$ points from $P$, $\zeta$ is
    empirically set to 100 in this paper
Compute the maximum point cloud spacing $\rho_\zeta$ of $P_\zeta$
Generate Bezier surface $P_{\zeta'}$ of $\zeta' \times \zeta'$ points from $P$, $\zeta' =$
    $\beta \times \zeta \times \rho_\zeta / \rho$, $\beta$ is empirically set to 4 in this paper
$x_{\max}, x_{\min} = \max(x_1, x_2, ..., x_{\zeta'}), \min(x_1, x_2, ..., x_{\zeta'})$
$y_{\max}, y_{\min} = \max(y_1, y_2, ..., y_{\zeta'}), \min(y_1, y_2, ..., y_{\zeta'})$
$z_{\max}, z_{\min} = \max(z_1, z_2, ..., z_{\zeta'}), \min(z_1, z_2, ..., z_{\zeta'})$
Compute the dimensions of voxel grid with width of $\rho$ along
    with $x, y, z$ axes:
    $N_x = (x_{\max} - x_{\min})/\rho$, $N_y = (y_{\max} - y_{\min})/\rho$, $N_z = (z_{\max} - z_{\min})/\rho$
Compute voxel index $\mathcal{I}_i$ for each point $(x_i, y_i, z_i)$ in $P_{\zeta'}$:
    $\mathcal{I}_i = \lfloor (x_i - x_{\min})/\rho \rfloor + \lfloor (y_i - y_{\min})/\rho \rfloor \times N_y + \lfloor (z_i - z_{\min})/\rho \rfloor \times N_x \times N_y$
Compute the average coordinates and surface normals of
    points with the same voxel index in $P_{\zeta'}$ and acquire $\mathcal{S}$
```

Once the 3D finger surface is generated by the Alg. 3, the 3D-2D mapping algorithm introduced in [5] is used to ascertain the relationship between the synthesized 2D texture of ridge-valley pattern with the uniformed 3D finger surface, and the ridge-valley pattern are then engraved on the 3D surface according to this relationship. Such a mapping algorithm significantly benefits from the proposed uniform surface generation approach, and the computational complexity is greatly reduced as now fewer points

[2] The point cloud spacing refers to the average distance between a point and its adjacent points. If the distribution of points in a region is sparse, the spacing between two points in that region will be large, and the ridge-valley pattern in that region will be far from those in real-world fingerprints.

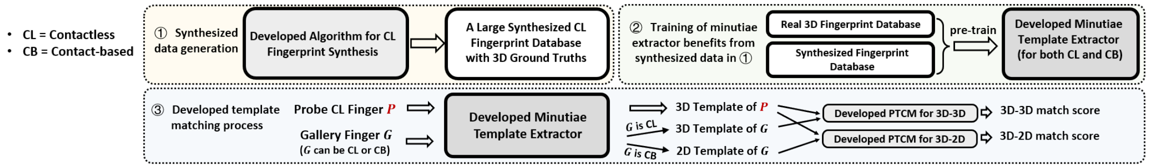

• CL = Contactless
• CB = Contact-based
① Synthesized data generation
Developed Algorithm for CL Fingerprint Synthesis
A Large Synthesized CL Fingerprint Database with 3D Ground Truths
② Training of minutiae extractor benefits from synthesized data in ①
Real 3D Fingerprint Database
Synthesized Fingerprint Database
pre-train
Developed Minutiae Template Extractor (for both CL and CB)
③ Developed template matching process
Probe CL Finger P
Gallery Finger G
(G can be CL or CB)
Developed Minutiae Template Extractor
3D Template of P
G is CL
3D Template of G
G is CB
2D Template of G
Developed PTCM for 3D-3D
Developed PTCM for 3D-2D
3D-3D match score
3D-2D match score

Fig. 7: The overview of the experimental evaluation of the developed methods.

are processed. Then we project the 3D engraved fingers on 2D imaging plane and acquire the ridge valley skeletons of contactless fingerprints following the camera projection algorithm similar to as in [5]. During this step, the ground truths for the respective 3D minutiae templates are simultaneously and automatically generated. The details of these steps are provided in Appendix H.

Motivated by [38] that adopts CycleGAN [17] in generating real-like contact-based fingerprint impressions from the ridge-valley skeleton of 2D contact-based fingerprints, generating realistic contactless fingerprint images in our approach also use CycleGAN architecture. During training, the ridge-valley skeleton images are considered the source domain, while the real fingerprint images are regarded as the target domain. We use Nested UNet [8] as the generator's backbone, replacing all deconvolution modules with up-sampling modules followed by a convolutional layer. This modification, recommended in [7], helps to alleviate the unwanted artifacts of checkerboard pattern in the generated fingerprint images. We use the pre-trained generator from the source domain to the target domain for inference. Ridge-valley skeleton images serve as the input for generating a large database of realistic contactless fingerprints.

7 EXPERIMENTS AND RESULTS

The overview of our experimental evaluation of the developed methods is presented in Fig. 7.

7.1 Databases and Protocols

This section provides brief details on the databases and protocols used in our experimentation. We use the same pre-processing steps as [5], including ROI detection, segmentation, and re-scaling on fingerprint images. All the fingerprint images are re-scaled except in the FVC database to ensure that the average ridge distances remain about 8 pixels for such 480×480 images. We further adopt contrast-limited adaptive histogram equalization for contactless fingerprint images to enhance the image quality.

FVC Database: FVC2002 [40] and FVC2004 [41] are utilized for training and evaluating the FingerYolo2D. We follow the same protocol as in [23], [27] and use the FVC database that provides 2D minutiae ground truth labels from fingerprint experts [39]. It contains four sub-databases: FVC2002-DB1A, FVC2002-DB3A, FVC2004-DB1A, and FVC2004-DB3A. As in [23], half of the subjects are randomly selected for training in each sub-database and another half are used for the evaluation. The size of fingerprint images is *not* re-scaled.

CFPose Database [20]: The 200 images from the last 20 fingers in this database are used to train the style transfer model of fingerprint synthesis. The 1200 images from the first 120 fingers are used to evaluate the matching performance (CL2CL) of framework (FingerYolo3D + PTCM) using the same protocol as in [5], [20].

PolyU C2CL Database [16]: The FingerYolo2D is trained using the same protocol as in [4], [23] i.e. using the contactless fingerprints in session 1 and session 2 whose identity numbers range from 1 to 136 (total 1440 images). To generate the ground truth labels, the minutiae templates are firstly extracted using VeriFinger and then we manually refine the labels, i.e. remove the spurious minutiae, and label the missing minutiae. The remaining 1200 impressions from 200 identities in session 1 are used to evaluate match performance (CL2CB) using FingerYolo2D with the same protocol as in [4] where 200 genuine and 39800 imposter pairs are generated. We follow the same protocol in [4], [23] where 3000 genuine and 19900 imposter pairs are generated to evaluate match performance (CL2CL) from FingerYolo2D.

PolyU 3D Database [21]: We generate the ground truths for 3D minutiae templates from this database by: i) extract the 2D minutiae templates using VeriFinger, ii) manually refine the acquired templates, iii) compute the surface normal from depth ground truth and generate the 3D ground truth (z and ϕ) from the depth and surface normal. There are 1920 (160 identities, 6 images in session-1 and 6 images in session-2 for each identity) contactless fingerprints with ground truths of 3D minutiae templates and 1920 contact-based fingerprints with ground truths of 2D minutiae templates that exist in both session 1 and session 2, we refer to these data as PolyU3D-*cross*. There are 1056 (176 identities, 6 images in session-1 for each identity) contactless fingerprints with ground truths of 3D minutiae templates and 1056 contact-based fingerprints with ground truths of 2D minutiae templates that only exist in session 1, we refer to these data as PolyU3D-*single*. All the contactless fingerprints with 3D minutiae ground truths in PolyU3D-*cross* are selected to train FingerYolo3D. The performance of PTCM on the real database is evaluated using PolyU3D-*cross* and more details are provided in Sec. 7.3. We evaluate the minutia detection accuracy (evaluated by the mean F1 score) from FingerYolo3D using PolyU3D-*single* database.

ZJU Database [22]: We follow the same protocol in [23] to use this database for training FingerYolo2D. This database is also used to evaluate match performance (both CL2CB and CL2CL) using (FingerYolo3D + PTCM) framework with the same protocol in [5].

ISPFDv2 Database [14]: This database is used to evaluate match performance (both CL2CB and CL2CL) for (FingerYolo3D+PTCM) framework with the same protocol in [5].

UWA Database: The 600 contactless fingerprint images from the first 100 identities in this database are used to train style transfer model of fingerprint synthesis. We use the same protocol as in [4], [23] to train and evaluate FingerYolo2D. This database is also used to evaluate match performance (for both CL2CB and CL2CL) from (FingerYolo3D+PTCM) framework with the same protocol as used in [5].

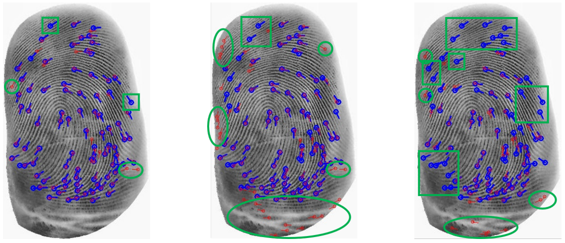
The IIIT-D Multi-Sensor (MOLF) Database [44]: This database is used to train FingerYolo2D and more details are provided in Appendix C.

Synthesized Contact-based Database [42]: This database is used to train FingerYolo2D and more details can be found in Appendix C.

Synthesized Contactless Database (Sec. 6): The synthesized images from the proposed method, with the 3D minutiae ground truths of this database, are used to train FingerYolo3D. The 3D minutiae ground truths from this database are also used to train the GNN model for the PTCM. More details on this database are provided in Appendix C.

7.2 Detection of 2D Minutiae and Matching

We can visualize representative samples for detecting 2D minutiae on contactless fingerprint images in Fig. 8, where three different 2D minutiae detection methods are utilized. All the detected minutia are shown in red, while the ground truths are marked in blue color. The green rectangle identifies the missing minutiae, and the green ellipse shows the falsely detected minutiae. It can be observed from Fig. 8b, that the VeriFinger detector is prone to generate falsely detected minutia as compared with the other two methods, and our proposed FingerYolo2D can more robustly detect 2D minutiae with the least number of missing and spuriously detected minutiae as compared to those from the VeriFinger [6] and CMNet [4].



(a) **ours** (b) COTS [6] (c) CMNet [4]

Fig. 8: Minutiae detection results from different methods.

Tab. 1 presents comparative analysis on the accuracy of detecting 2D minutiae templates using the FingerYolo2D where the respective ground truth minutiae were manually labeled by the experts in the four sub-databases of FVC database. The accuracy of detecting the minutiae is evaluated from the mean F1 scores of the *correctly detected* minutia. Similar to as in [23], [27], we also evaluate the detection accuracy of these methods using the stated criterion to predict the minutia within the 12 pixels spatial and 20° of orientation tolerance to the respective ground truth minutia. Following [23], we adopt a stricter criterion for the tolerance that the distance between predicted minutia and the ground truth should be within 6 pixels while the orientation difference is within 10° to be considered a correct match. It can be observed from such results in Tab. 1 that the proposed FingerYolo2D method can achieve superior minutiae detection performance under both criteria compared to the methods in [23], [27].

TABLE 1: Comparative 2D Minutiae Extraction Accuracy.

Method Database	[27][1]	[23][1]	[23][2]	Proposed[1]	Proposed[2]
FVC02-DB1A	0.879	0.910	0.880	0.927	0.908
FVC02-DB3A	0.854	0.868	0.853	0.877	0.865
FVC04-DB1A	0.845	0.878	0.829	0.894	0.849
FVC04-DB3A	0.821	0.833	0.818	0.856	0.830
Average (above)	0.849	0.875	0.839	0.889	0.862

[1] using default criterion (±12 pixels and ±20°)
[2] using stricter criterion (±6 pixels and ±10°)

TABLE 2: Comparative 2D minutiae extraction (CL2CB).

Method	Database	AUC	EER	Database	AUC	EER
mindtct [24]	PolyU C2CL	57.58%	43.62%	UWA	77.69%	28.26%
MinutiaeNet [26]		50.78%	50.00%		56.23%	45.87%
VeriFinger [6]		84.78%	22.32%		86.43%	20.32%
CMNet [4]		90.91%	14.60%		93.90%	11.64%
FingerYolo2D		**97.39%**	**5.86%**		**95.20%**	**9.93%**

TABLE 3: Comparative 2D minutiae extraction (CL2CL).

Method	Database	AUC	EER	Database	AUC	EER
mindtct [24]	PolyU C2CL	58.91%	36.85%	UWA	81.84%	4.28%
MinutiaeNet [26]		93.03%	13.35%		79.74%	26.34%
VeriFinger [6]		98.16%	2.99%		95.44%	9.02%
CMNet [4]		99.33%	1.94%		98.24%	4.28%
Extractor [23]		99.25%	1.90%		98.91%	3.89%
FingerYolo2D		**99.81%**	**0.70%**		**99.26%**	**1.69%**

The minutiae-based fingerprint verification (for both CL2CB and CL2CL) performance using the 2D minutiae templates extracted by FingerYolo2D and other 2D minutiae extraction methods is also presented. We use the MCC [1] to match the extracted 2D minutiae templates, Such comparative performance presented in Tabs. 2 and 3 for both CL2CB and CL2CL matching indicate that our proposed model can offer superior performance than previous 2D minutiae template extraction methods.

7.3 Detection of 3D Minutiae and Matching

The visualization of an estimated 3D minutiae template, along with the respective contactless fingerprint image, appears in Fig. 9. This figure presents such visualization from four different view angles. The estimated 3D minutia are shown using the blue pyramids while the projected 2D minutia on the contactless fingerprint image appear in red, and the projection relationship is annotated using the dotted lines. The 3D minutia are connected using Delaunay triangulation for better visualization in 3D space, such triangulated mesh coarsely consists of a surface that approximates the shape of the finger.

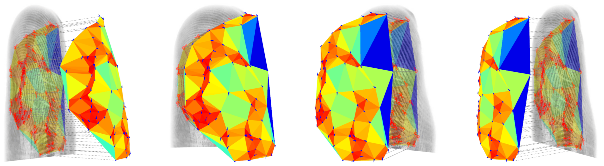
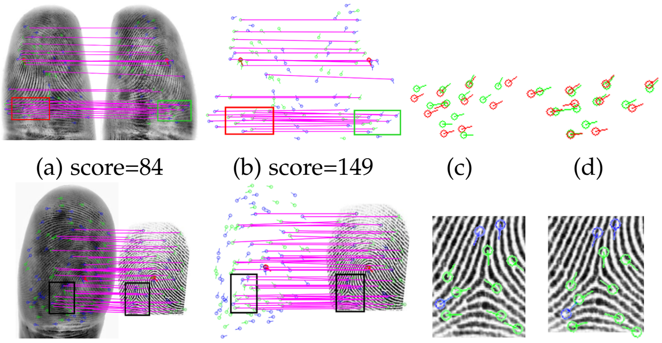


Fig. 9: Multiple views for a predicted 3D minutiae template.

Detection Accuracy for 3D Minutiae: The 3D minutiae detection accuracy using the proposed FingerYolo3D detector is estimated by comparing the extracted 3D minutiae templates with the 3D minutiae ground truths of all the 1056 contactless fingerprints in PolyU3D-*single* of PolyU 3D database [21] and this accuracy is quantified using the mean F1 score. The criterion for the correct match between the predicted 3D minutiae and the ground truth is defined using the tolerance i.e. whenever the predicted minutia is within 12 pixels of the respective ground truth minutia and the orientational difference in 3D space is within 20°. The mean F1 score from such evaluation using PolyU3D-*single* is 0.816.

PTCM Evaluation: Fig. 10a presents a sample visualization of a pair of original 3D-3D minutiae templates and it is visualized using its 2D projection. In Fig. 10b we use PTCM to align the same 3D-3D templates and project to 2D. It can be observed that with the use of PTCM, the respective match score generated by VeriFinger increases from 84 to 149. We especially highlight challenging (cross-view) regions with red and green boxes in Fig. 10a and present in Fig. 10c the corresponding visualization of minutia from these two regions using the same red and green color. Fig. 10d shows the compensated minutiae matches from Fig. 10b, where the minutiae are better aligned as compared to those in Fig. 10c. Similar visualization for a pair of 3D-2D minutiae templates (3D template is visualized in 2D) is shown in Fig. 10e. The minutiae templates in Fig. 10f are compensated from Fig. 10e using PTCM and the respective match score by VeriFinger increases from 238 to 363. We enlarge the respective regions in black rectangles in Fig. 10e and project the 3D minutiae on the corresponding contact-based fingerprint in Fig. 10g. We perform same steps for pair in Fig. 10f and generate Fig. 10h. It can be observed from Fig. 10h that CL2CB minutiae correspondences are better aligned than Fig. 10g.


(a) score=84 (b) score=149 (c) (d)

(e) score=238 (f) score=363 (g) (h)

Fig. 10: Visualization for effectiveness of the PTCM

The quantitative analysis for PTCM using the comparative ROC curves is presented Fig. 11 and respective EER, and TAR@FAR=10^{-4} values are summarized in Tab. 4. These experiments were performed to ascertain the effectiveness of PTCM for the 3D minutiae template matching using the PolyU3D-*cross* dataset. Two off-the-shelf 2D minutiae template matchers (MCC [1], VeriFinger [6]) and one 3D-3D minutiae template matcher (tetrahedron based matcher [21]) are selected as the baseline for the for comparisons. It is worth noting that the tetrahedron-based method [21] can only be used for 3D-3D matching since this method is not able to match 3D to 2D templates. For the MCC and VeriFinger matcher, if the template pair is matched without PTCM, we retain only 2D information for such templates and input the acquired 2D templates to the matcher for computing the match scores. If the PTCM is used for matching, we recover the projected 2D minutiae templates from the input template pair as detailed in Sec. 4, and input such templates to compute the match scores. The evaluation protocol can be summarized as follows.

- 3D-2D one-session matching: we perform all-to-all 3D to 2D template matching using session-1 of PolyU3D-*cross* dataset. Therefore, we generate 5760 (160*(6*6)) genuine and 915840 ((160*159)*(6*6)) imposter match scores.
- 3D-3D one-session matching: we perform 3D to 3D template matching under all-to-all protocols using session-1 of PolyU3D-*cross* dataset. Therefore 2400 (160*(6*5)/2) genuine and 457920 ((160*159/2)*(6*6)) imposter match scores are generated.
- 3D-2D two-session matching: we perform all-to-all matching between 3D template of session-1 and 2D template of session-2 using PolyU3D-*cross* dataset. Therefore 5760 (160*(6*6)) genuine and 915840 ((160*159)*(6*6)) imposter match scores are generated.
- 3D-3D two-session matching: we perform all-to-all 3D to 3D template matching between session-1 and session-2 of PolyU3D-*cross* dataset. We generate 5760 (160*(6*6)) genuine and 915840 ((160*159)*(6*6)) imposter match scores.

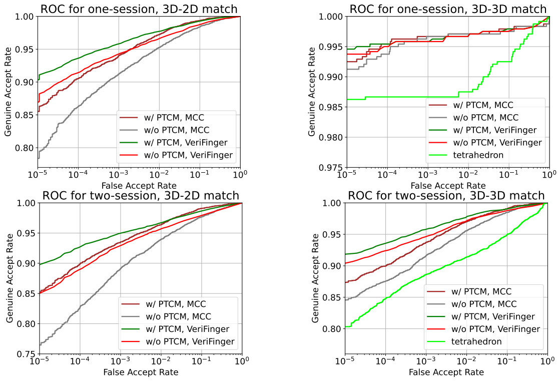


Fig. 11: Comparative results for *real* 3D minutiae templates.

TABLE 4: Performance of PTCM on PolyU3D-*cross*.

scenario	PTCM	matcher	EER	TAR@FAR
one-session, 3D-2D	✔	MCC [1]	**1.875%**	90.521%
one-session, 3D-2D	✔	VeriFinger [6]	1.925%	**93.698%**
one-session, 3D-2D	✗	MCC [1]	3.119%	86.267%
one-session, 3D-2D	✗	VeriFinger [6]	2.636%	91.510%
one-session, 3D-3D	✔	MCC [1]	**0.316%**	**99.542%**
one-session, 3D-3D	✔	VeriFinger [6]	0.352%	**99.542%**
one-session, 3D-3D	✗	MCC [1]	**0.316%**	99.375%
one-session, 3D-3D	✗	VeriFinger [6]	0.352%	99.500%
one-session, 3D-3D	-	tetrahedron [21]	1.256%	98.625%
two-session, 3D-2D	✔	MCC [1]	**2.345%**	89.861%
two-session, 3D-2D	✔	VeriFinger [6]	2.609%	**92.604%**
two-session, 3D-2D	✗	MCC [1]	3.735%	82.656%
two-session, 3D-2D	✗	VeriFinger [6]	2.609%	88.906%
two-session, 3D-3D	✔	MCC [1]	2.133%	89.965%
two-session, 3D-3D	✔	VeriFinger [6]	**1.709%**	**93.524%**
two-session, 3D-3D	✗	MCC [1]	3.025%	87.535%
two-session, 3D-3D	✗	VeriFinger [6]	2.186%	92.361%
two-session, 3D-3D	-	tetrahedron [21]	6.093%	84.740%

The comparative results presented in this section indicate that the PTCM can help to enhance the performance of the 3D-3D and 3D-2D minutiae template matching, for both intra-session and cross-session tasks, and its ability to utilize the 3D information of minutiae is significantly better than those using the tetrahedron-based matcher [21].

Evaluation for (FingerYolo3D+PTCM): To evaluate the performance of FingerYolo3D and PTCM framework on databases without the known ground truth for 3D minutiae, (FingerYolo3D + PTCM) framework is considered to recover 3D minutiae templates from the monocular contactless fingerprints followed by PTCM to compute the match score from such recovered 3D minutiae templates. The performance evaluation using (FingerYolo3D + PTCM) for CL2CB matching performance on three databases is presented in Tab. 5 and Fig. 12. The ROC, CMC, and FPIR-FNIR plots, along with the EER, TAR@FAR=10^{-4} and Rank-1 accuracy, are reported. For comparisons, we provide the performance from other frameworks that use different minutiae extraction and template matching methods. Such experiments are organized under four groups: A_1, A_2, A_3, A_4, and explained here. In group A_1, VeriFinger extracts 2D templates from the contactless fingerprints. In group A_2, the 3D information of templates extracted from contactless fingerprints by

FingerYolo3D is removed, and resulting 2D templates are used for contactless fingerprint matching. In group A_3, 3D information of templates extracted from contactless fingerprints by FingerYolo3D is firstly removed. The 3D ellipsoid modeling [20] is then used to recover respective 3D minutiae features, followed by the compensation method in [20] to acquire the compensated minutiae templates projected on 2D plane for the matching. In group A_4, FingerYolo3D is used to extract 3D minutiae templates, and then our PTCM is employed to acquire the compensated minutiae templates that are projected to 2D. In groups A_1, A_2, A_3, and A_4, VeriFinger extracts 2D templates from contact-based fingerprints, and VeriFinger matches the CL2CB templates.

TABLE 5: Comparative Results of CL2CB Matching (in %).

Database	Index	A_1	A_2	A_3	A_4
ZJU	EER	4.008	2.900	2.435	**2.196**
	TAR@FAR	86.002	90.085	90.530	**92.094**
	Rank-1	91.566	94.023	94.519	**95.581**
ISPFDv2	EER	5.421	2.802	2.384	**1.771**
	TAR@FAR	83.146	88.423	89.513	**91.356**
	Rank-1	91.810	94.381	95.238	**96.762**
UWA	EER	11.906	18.586	11.869	**10.479**
	TAR@FAR	56.76	58.927	61.488	**66.282**
	Rank-1	67.704	69.679	72.523	**75.736**

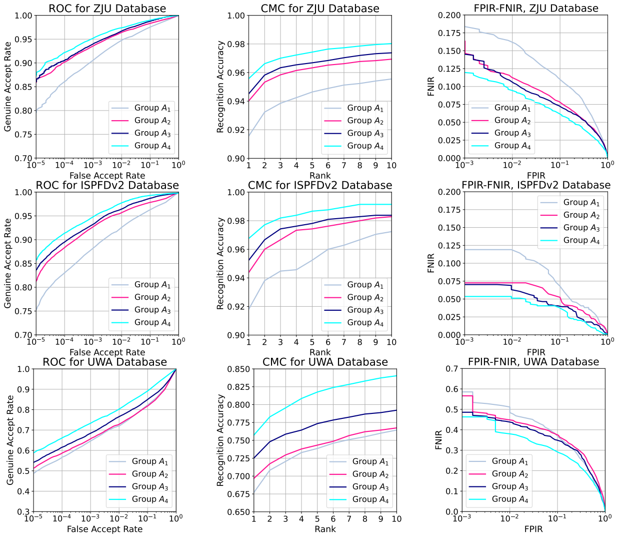


Fig. 12: Comparative results for the CL2CB matching.

The CL2CL match performance from (FingerYolo3D + PTCM) is evaluated on four databases. These results are presented in Fig. 13 and summarized in Tab. 6, which also provides the performance of other frameworks that use different minutiae extraction and template matching methods. The CL2CL match experiments are organized into five groups, i.e., B_1, B_2, B_3, B_4, and B_5. The contactless fingerprint matching methods in groups B_1, B_2, B_3, and B_5 are the same as those in groups A_1, A_2, A_3, and A_4, respectively. In group B_4, we use FingerYolo3D for 3D minutiae template extraction, and the tetrahedron-based matcher [21] is used to compute match scores of extracted 3D templates. The VeriFinger is used to compute the match score for the templates in groups B_1, B_2, B_3, and B_5.

TABLE 6: Comparative Results for CL2CL Matching (in %).

Database	Index	B_1	B_2	B_3	B_4	B_5
CFPose	EER	5.077	2.901	2.738	3.443	**2.053**
	TAR@FAR	86.481	88.037	89.611	82.167	**92.574**
	Rank-1	92.130	93.148	93.889	90.833	**97.037**
ZJU	EER	0.876	0.798	0.644	1.770	**0.502**
	TAR@FAR	97.968	98.382	98.678	92.992	**98.974**
	Rank-1	99.404	99.173	99.592	96.547	**99.735**
ISPFDv2	EER	2.937	2.096	2.147	3.139	**1.556**
	TAR@FAR	95.302	96.603	96.698	92.317	**97.492**
	Rank-1	98.111	98.889	98.889	97.667	**99.444**
UWA	EER	21.156	23.141	26.471	29.585	**19.950**
	TAR@FAR	28.782	36.647	39.558	22.021	**43.842**
	Rank-1	62.851	68.624	74.096	59.739	**77.159**

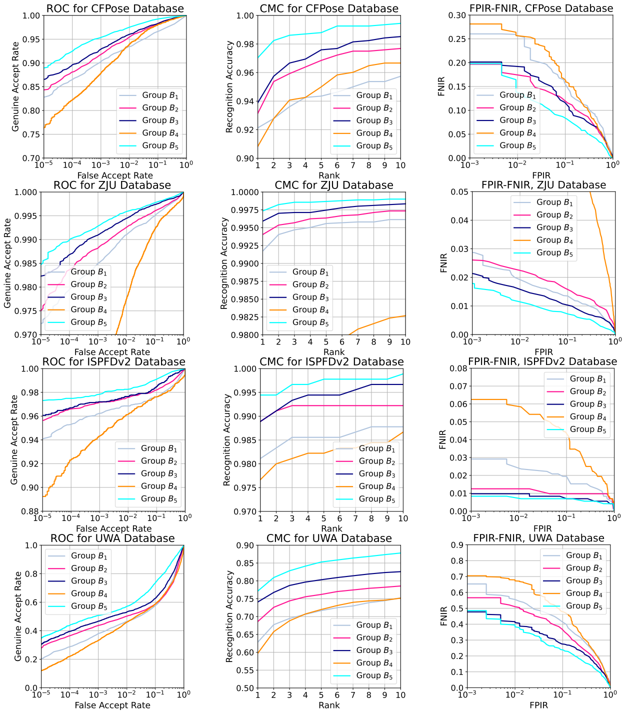


Fig. 13: Comparative results for the CL2CL matching.

To evaluate the effectiveness of the CL2CB and CL2CL matching, we can compare the performance from A_2 vs. A_3 vs. A_4 and B_2 vs. B_3 vs. B_5. It can be observed that both the 3D recovery method in [20] and our proposed FingerYolo3D can improve the matching performance. Therefore, in both cases, the 3D information recovered by these two methods is effective, and the proposed framework to recover 3D information and compensate 3D templates (with PTCM) outperforms the framework introduced in [20]. Further, the 3D information is recovered similarly for B_4 and B_5, while exploited by two different matching methods. The performance of the tetrahedron-based matcher lags far behind our proposed PTCM mechanism. It can be observed that the performance for group B_4 is even below B_2, where the 3D information is discarded, and [21] has not provided comparisons with the other 2D minutiae matchers, such as MCC [1] or the VeriFinger [6]. Due to space constraints, performance comparisons with other methods [2], [33] are provided in Appendix A and Appendix B, respectively.

7.4 Inference Speed for FingerYolo3D Model

The inference time for the different minutiae extraction models is presented in Tab. 7. It can be observed from Tab. 7 that even with the use of GTX1050 mobile, which is quite outdated than the currently popular GPU cards used in other deep learning-based algorithms, our algorithm still reaches the fastest speed. In addition, our model offers superior performance over a popular COTS when evaluated on the same CPU, which further validates the extremely high speed of our proposed model.

TABLE 7: Comparative Analysis for the Inference Speed.

Method	Device	Test Database	Time (s)
VeriFinger [6]	i9-10850K	FVC2004	1.07
Proposed	i9-10850K	FVC2004	**0.00625**
MinutiaeNet [26]	GTX1060	FVC2004	1.2 [27]
FME [27]	GTX1060	FVC2004	0.03 [27]
Extractor [23]	GTX1050Ti(Mobile)	FVC2004	0.025 [23]
Proposed	GTX1050(Mobile)	FVC2004	**0.0115**
VeriFinger [6]	i9-10850K	UWA	1.11
Proposed	i9-10850K	UWA	**0.00698**
MinutiaeNet [26]	TITAN Xp	UWA	1.2 [4]
CMNet [4]	TITAN Xp	UWA	0.86 [4]
Extractor [23]	GTX1050Ti(Mobile)	UWA	0.083 [23]
Proposed	GTX1050(Mobile)	UWA	**0.0122**

7.5 Comparative Analysis for Fingerprint Synthesis

Tab. 8 presents comparative analysis from generating the finger surfaces using the method in [5] and from the proposed method while ensuring that the maximum point cloud spacing between these two methods remains the same and is 64, 32, 16, and 8μm (micrometer), respectively. To ensure fair comparison, the height of 3D finger model is re-scaled to 2.5 cm. We compare the number of points, the speed of surface generation, and the speed of 3D-2D ridge-valley pattern mapping of finger surfaces generated by these two methods under the same maximum point cloud spacing. This analysis indicates that when the maximum point cloud spacing is the same, our proposed method generates fewer points on the surface using slightly more time than [5]. However, it saves more time in the following step for the 3D-2D mapping algorithm than using [5] and therefore benefits the total time required for the 3D finger synthesis.

TABLE 8: Comparison for 3D Finger Generation Methods.

Method	[5]				**This Paper**			
max. point spacing	64μm	32μm	16μm	8μm	64μm	32μm	16μm	8μm
number of points on surface ($\times 10^3$)	14.3	56.1	198.19	986.3	2.5	10.0	31.93	108.2
speed of surface generation (seconds)	0.010	0.025	0.087	0.438	0.018	0.052	0.185	0.667
speed of 3D-2D mapping (seconds)	1.41	4.44	13.96	27.41	0.70	1.03	3.06	9.03

The first-row images in Fig. 14 are the samples from the profile-view, top-view, and frontal view of the 3D finger surface generated using the method in [5] and contains 3600 points. The 3D finger surface generated by the proposed method is also shown in Figs. 14e to 14g and also contains 3600 points. Although the number of points of the meshgrid displayed in the first row and the second row of Fig. 14 are the same, the maximum point cloud spacing for the first finger surface is much larger than the second finger surface since the points in the first surface are unevenly distributed as highlighted in Fig. 14d and the points in the second surface are evenly distributed as shown in Fig. 14h.

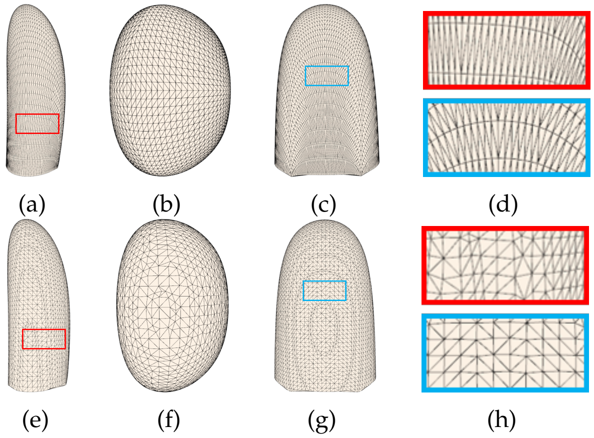


Fig. 14: Comparative visualization of results from original Bezier surface and uniform Bezier 3D finger surface modeling.

7.6 Individuality Analysis using PRC

This section presents experimental results for comparative individuality analysis, using PRC_{2d} index for the contactless 2D minutiae templates and the PRC_{3d} index for respective 3D minutiae templates recovered from the contactless 2D fingerprints. The PRC_{2d} index and PRC_{3d} index, using the ISPFDv2 database and with $m = n = 52$ setting, are respectively summarized in Tab. 9a and 9b; we fix the threshold w to 12 and change the values of r_0 and d_0. The threshold r_0 ranges from 8 to 15 in steps of 1, and d_0 ranges from $15.3°$ to $21.6°$ in steps of $0.9°$; with a larger r_0 and larger d_0, we use more relaxed conditions for considering two minutiae as matched. It can be observed from Tabs. 9a and 9b that when the match conditions are relaxed, the probability of falsely matching two randomly generated minutiae templates increases. We compute the order of magnitude of $\frac{\text{PRC}_{2d}}{\text{PRC}_{3d}}$ and denote this number in red on the lower right in Tab. 9b. It can be observed that with the reduction in r_0 and d_0, i.e. when the matching criterion gets stricter, the ratio of PRC2d to PRC3d increases. Tab. 9c provides the PRC_{2d} and PRC_{3d} scores from ISPFDv2 database, and with the increase in w while fixing the values of r_0 and d_0. This trend also indicated that as the matching criterion gets stricter, the value of $\frac{\text{PRC}_{2d}}{\text{PRC}_{3d}}$ increases. This can explain the theoretical advantage of 3D minutiae templates over 2D minutiae templates, using enhanced fingerprint uniqueness, especially when the matching criteria are more stringent, which is the case for most high-security deployment applications.

The comparative results for the PRC_{2d} and PRC_{3d} scores using five different fingerprint databases are presented in Tab. 10. These comparisons on the estimated uniqueness of 2D and 3D minutiae templates are also presented using the prior method in [28]. It can be observed from Tab. 10 that the recovered 3D minutiae templates can offer superior uniqueness over 2D templates as (x, y, z, θ, ϕ) has two higher degrees of freedom than (x, y, θ), which can significantly lower the possibility $p(C_i, C_j)$ of occurring a false match between two randomly selected minutia. The degradation of $p(C_i, C_j)$ further leads to the decrease in the parameter $\lambda(C_i, C_j)$ for the Poisson distribution which exponentially influences the final PRC score. Here, we use the same criterion as in [28] with the setting of $r_0 = 15, d_0 = 21.6°$ to compute the PRC_{2d} and PRC_{3d}, respectively, for the 2D and 3D minutiae templates. The overestimated results in Tab. 10 reveal that [28] ignored the interdependence between the features, and the model in our work has estimated the correlation between the distribution of minutiae location and orientation in 3D space, leading to a more precise estimation of the 3D fingerprint uniqueness.

TABLE 9: Individuality Analysis Using Comparative PRC Scores between 2D and Recovered 3D Minutiae Template

(a) PRC_{2d} Score for ISPFDv2 Database ($m = n = 52, w = 12$)

	$d_0 = 15.3°$	$d_0 = 16.2°$	$d_0 = 17.1°$	$d_0 = 18.0°$	$d_0 = 18.9°$	$d_0 = 19.8°$	$d_0 = 20.7°$	$d_0 = 21.6°$
$r_0 = 8$	5.60×10^{-9}	6.11×10^{-9}	8.37×10^{-9}	1.29×10^{-8}	1.84×10^{-8}	4.74×10^{-8}	6.78×10^{-8}	1.27×10^{-7}
$r_0 = 9$	5.64×10^{-8}	6.80×10^{-8}	1.75×10^{-7}	2.44×10^{-7}	3.77×10^{-7}	5.99×10^{-7}	7.47×10^{-7}	8.52×10^{-7}
$r_0 = 10$	3.58×10^{-7}	9.75×10^{-7}	1.10×10^{-6}	2.13×10^{-6}	2.55×10^{-6}	3.37×10^{-6}	5.27×10^{-6}	8.62×10^{-6}
$r_0 = 11$	1.91×10^{-6}	4.77×10^{-6}	7.06×10^{-6}	9.02×10^{-6}	1.70×10^{-5}	1.99×10^{-5}	3.01×10^{-5}	3.83×10^{-5}
$r_0 = 12$	1.07×10^{-5}	3.14×10^{-5}	3.77×10^{-5}	7.31×10^{-5}	9.88×10^{-5}	1.09×10^{-4}	1.25×10^{-4}	2.38×10^{-4}
$r_0 = 13$	5.74×10^{-5}	1.05×10^{-4}	1.66×10^{-4}	2.34×10^{-4}	2.62×10^{-4}	4.22×10^{-4}	5.04×10^{-4}	6.15×10^{-4}
$r_0 = 14$	2.35×10^{-4}	3.60×10^{-4}	5.95×10^{-4}	7.18×10^{-4}	8.59×10^{-4}	1.68×10^{-3}	2.11×10^{-3}	3.00×10^{-3}
$r_0 = 15$	7.46×10^{-4}	1.32×10^{-3}	1.54×10^{-3}	2.21×10^{-3}	3.30×10^{-3}	4.01×10^{-3}	5.05×10^{-3}	6.86×10^{-3}

(b) $\text{PRC}_{\text{3d}(n)}$ Score for ISPFDv2 ($m = n = 52, w = 12$), red number (n) on lower right denotes order of magnitude of $\frac{\text{PRC}_{\text{2d}}}{\text{PRC}_{\text{3d}}}$

	$d_0 = 15.3°$	$d_0 = 16.2°$	$d_0 = 17.1°$	$d_0 = 18.0°$	$d_0 = 18.9°$	$d_0 = 19.8°$	$d_0 = 20.7°$	$d_0 = 21.6°$
$r_0 = 8$	$3.03 \times 10^{-21}_{(12)}$	$5.50 \times 10^{-21}_{(12)}$	$2.64 \times 10^{-20}_{(11)}$	$1.22 \times 10^{-19}_{(11)}$	$4.74 \times 10^{-19}_{(10)}$	$8.47 \times 10^{-19}_{(10)}$	$9.55 \times 10^{-19}_{(10)}$	$5.74 \times 10^{-18}_{(10)}$
$r_0 = 9$	$2.10 \times 10^{-19}_{(11)}$	$1.72 \times 10^{-19}_{(11)}$	$1.19 \times 10^{-18}_{(11)}$	$1.90 \times 10^{-18}_{(11)}$	$7.50 \times 10^{-18}_{(10)}$	$3.01 \times 10^{-17}_{(10)}$	$6.36 \times 10^{-17}_{(10)}$	$5.89 \times 10^{-16}_{(9)}$
$r_0 = 10$	$1.44 \times 10^{-17}_{(10)}$	$3.21 \times 10^{-17}_{(10)}$	$5.46 \times 10^{-17}_{(10)}$	$7.64 \times 10^{-17}_{(10)}$	$1.57 \times 10^{-16}_{(10)}$	$3.38 \times 10^{-16}_{(9)}$	$8.72 \times 10^{-16}_{(9)}$	$1.20 \times 10^{-15}_{(9)}$
$r_0 = 11$	$7.68 \times 10^{-17}_{(10)}$	$9.03 \times 10^{-17}_{(10)}$	$2.72 \times 10^{-16}_{(10)}$	$6.27 \times 10^{-16}_{(10)}$	$1.42 \times 10^{-15}_{(10)}$	$6.15 \times 10^{-15}_{(9)}$	$1.03 \times 10^{-14}_{(9)}$	$1.93 \times 10^{-14}_{(9)}$
$r_0 = 12$	$1.56 \times 10^{-16}_{(10)}$	$1.31 \times 10^{-15}_{(10)}$	$7.91 \times 10^{-15}_{(9)}$	$1.23 \times 10^{-14}_{(9)}$	$4.39 \times 10^{-14}_{(9)}$	$6.01 \times 10^{-14}_{(9)}$	$9.46 \times 10^{-14}_{(9)}$	$2.89 \times 10^{-13}_{(8)}$
$r_0 = 13$	$3.51 \times 10^{-15}_{(10)}$	$5.08 \times 10^{-14}_{(9)}$	$1.03 \times 10^{-13}_{(9)}$	$3.52 \times 10^{-13}_{(8)}$	$6.36 \times 10^{-13}_{(8)}$	$9.48 \times 10^{-13}_{(8)}$	$1.18 \times 10^{-12}_{(8)}$	$4.51 \times 10^{-12}_{(8)}$
$r_0 = 14$	$2.72 \times 10^{-13}_{(8)}$	$3.18 \times 10^{-13}_{(9)}$	$5.53 \times 10^{-13}_{(9)}$	$1.33 \times 10^{-12}_{(8)}$	$5.81 \times 10^{-12}_{(8)}$	$8.70 \times 10^{-12}_{(8)}$	$1.11 \times 10^{-11}_{(8)}$	$3.84 \times 10^{-11}_{(7)}$
$r_0 = 15$	$7.72 \times 10^{-13}_{(8)}$	$3.15 \times 10^{-12}_{(8)}$	$6.88 \times 10^{-12}_{(8)}$	$8.49 \times 10^{-12}_{(8)}$	$2.71 \times 10^{-11}_{(8)}$	$3.26 \times 10^{-11}_{(8)}$	$5.23 \times 10^{-11}_{(7)}$	$1.67 \times 10^{-10}_{(7)}$

(c) PRC_{2d} and PRC_{3d} Scores for ISPFDv2 Database for Different w Thresholds ($m = n = 52, r_0 = 15, d_0 = 21.6°$)

w	10	12	14	16	18	20	22	24
PRC_{2d}	3.88×10^{-2}	6.86×10^{-3}	1.24×10^{-3}	1.03×10^{-4}	2.15×10^{-5}	1.64×10^{-6}	6.43×10^{-8}	5.34×10^{-9}
PRC_{3d}	6.64×10^{-8}	1.67×10^{-10}	1.29×10^{-12}	1.76×10^{-14}	3.72×10^{-17}	1.01×10^{-18}	3.39×10^{-22}	1.06×10^{-23}

TABLE 10: Comparative PRC_{2d} and PRC_{3d} Scores from Different Contactless Fingerprint Databases ($r_0 = 15, d_0 = 21.6°, w = 12$)

Database	m, n	PRC_{2d} [18]	PRC_{3d} (**ours**)	PRC_{2d} [28]	PRC_{3d} [28]
CFPose [20]	$55, 55$	8.33×10^{-3}	4.59×10^{-10}	2.46×10^{-6}	3.56×10^{-16}
PolyU 3D [21]	$47, 47$	6.56×10^{-3}	5.28×10^{-10}	4.38×10^{-7}	1.02×10^{-16}
ZJU [22]	$63, 63$	1.15×10^{-4}	2.40×10^{-11}	3.95×10^{-6}	4.55×10^{-16}
ISPFDv2 [14]	$52, 52$	6.86×10^{-3}	1.67×10^{-10}	4.98×10^{-6}	2.35×10^{-15}
UWA [15]	$42, 42$	7.01×10^{-5}	3.46×10^{-13}	1.42×10^{-8}	3.43×10^{-18}

8 Ablation Study

The ablation studies can help to further validate and understand the contributions from the synthesized fingerprint database, the introduction of the dynamic bounding box, and 3D orientation loss during the network training of our 3D/2D minutiae extraction architecture.

Synthesized Contactless Fingerprints: We performed experiments to ascertain the contribution from our synthesized database for the extraction of 2D minutiae, and these results are summarized in Tab. 11. In this table, the baseline results in the group "w/o syn." are from Tabs. 2 and 3, where the synthesized database was not used to train FingerYolo2D, and we added the synthesized database, besides the training dataset, on the baseline to train FingerYolo2D in the group "w/ syn.". Similarly, Tab. 12 summarizes the experimental results performed to evaluate the contribution of our synthesized database for the 3D minutiae extraction. For the results of Tab. 12, we only used the PolyU 3D database [21] to train FingerYolo3D in the group "w/o syn." and this represented the baseline results. In the group "w/ syn.", the performance on ISPFDv2 and CFPose database is respectively from Tabs. 5 and 6, where both the synthesized database and PolyU 3D database [21] are used to train FingerYolo3D. These experimental results indicate that our synthesized fingerprint database can enhance the performance of the proposed architecture to recover 2D and 3D minutiae for the CL2CB and CL2CL matching tasks.

TABLE 11: Effectiveness of Synthetic Data for FingerYolo2D

Index	Task	w/ syn.	w/o syn.	Task	w/ syn.	w/o syn.
AUC	UWA(	**96.22**%	95.20%	UWA(	**99.43**%	99.26%
EER	CL2CB)	**9.16**%	9.93%	CL2CL)	**1.47**%	1.69%

TABLE 12: Effectiveness of Synthetic Data for FingerYolo3D

Task	CL2CB task (ISPFDv2)			CL2CL task (CFPose)		
Index	EER	TAR@FAR	Rank-1	EER	TAR@FAR	Rank-1
w/ syn.	**1.771**%	**91.356**%	**96.762**%	**2.053**%	**92.574**%	**97.037**%
w/o syn.	2.722%	88.304%	94.286%	2.615%	89.481%	94.259%

Dynamic Bounding Box for Minutiae Detection: To evaluate the effectiveness of the dynamic bounding box approach for the FingerYolo2D and FingerYolo3D, we summarize the comparative results in Tab. 13 using the detection accuracy (evaluated by mean F1 score) of FingerYolo2D and FingerYolo3D. These two detectors were trained with the proposed dynamic bounding box and other three scenarios where the width w of the ground truth bounding box is fixed at $0.75 \times l_{rv}$, $1.0 \times l_{rv}$, $1.25 \times l_{rv}$, where l_{rv} denotes the average length between ridge and valley. These results indicate that the performance of the proposed dynamic bounding box is superior to the other approach where the width of such bounding box remains fixed.

3D orientation loss L_o: Since this loss function is designed to accurately predict the 3D minutiae orientations from FingerYolo3D, we compare the 3D minutiae template detection accuracy for the PolyU3D-*single* database when the

TABLE 13: Ablation Study for Dynamic Bounding Box.

Model	Method / Database	**Dynamic**	$0.75{\times}l_{rv}$	$1.0{\times}l_{rv}$	$1.25{\times}l_{rv}$
FingerYolo2D	FVC02-DB1A	**0.927**	0.904	0.915	0.918
	FVC02-DB3A	**0.877**	0.875	0.870	0.872
	FVC04-DB1A	**0.894**	0.871	0.873	0.880
	FVC04-DB3A	**0.856**	0.839	0.851	0.845
	Average (above)	**0.889**	0.872	0.878	0.880
FingerYolo3D	PolyU3D-*single*	**0.818**	0.789	0.801	0.804

FingerYolo3D trained by the proposed L_o with the scenario when the FingerYolo3D trained by other orientation loss functions. The mean F1 scores from the proposed loss, L_1 loss, chord loss, and cosine loss are, respectively, 0.818, 0.794, 0.783, and 0.767. These results indicate the effectiveness of the proposed loss function.

9 CONCLUSIONS AND FURTHER WORK

This paper[3] has introduced the first attempt to recover a 3D minutiae template from a *single* contactless fingerprint image. We propose a neural model with lightweight architecture (FingerYolo3D) to recover such 3D minutiae templates. This model can simultaneously predict 3D coordinates, 3D orientations, and the type of respective minutia. To this end, this paper also proposes a dynamic bounding box scheme to enable the model to automatically learn the size of the effective region at the respective minutiae point. In addition, the newly introduced orientation loss can help the model predict the azimuthal and elevational orientations of the 3D minutiae more accurately. Our extensive experimental presented in Sec. 7 of this paper indicates that such an architecture can also accurately extract 2D minutiae templates (FingerYolo2D) and achieve SOTA performance.

The approach introduced in Sec. 6 to synthesize completely contactless 3D fingerprint images can generate more realistic finger surface texture and their 3D minutiae ground truths, including respective coordinates and directions. Experimental results presented in Sec. 8 reveal that the extensive synthetic contactless fingerprint database generated following the proposed method benefits the minutiae extraction model, and experimental results in Sec. 7 show that the 3D information prediction model trained by such synthesized database can help to enhance the minutiae matching performance.

This paper has also presented the first attempt to *simultaneously* address the problem of 3D to 3D and 3D to 2D fingerprint minutiae template matching, whose success is critical for the broader adoption of 3D fingerprint technologies. To fully avail the potential from 3D information, we introduced pose invariant PTCM to compensate for the perspective transformation of the 3D minutiae templates, yielding a discernible performance surge of 3D minutiae template matching. Comparative evaluation of the proposed PTCM in Sec. 7 with prior 3D to 3D minutiae template matching method [21] using a *real* 3D minutiae template database indicates superior match performance from PTCM. It was also observed that the use of the FingerYolo3D to recover 3D minutiae templates from contactless fingerprints first and then followed by PTCM to match the extracted 3D templates can offer much better performance than directly matching the 2D minutiae templates extracted from contactless fingerprints. This observation can help establish our framework's merit (FingerYolo3D+PTCM) in fully utilizing the recovered 3D information from the monocular contactless 2D fingerprints.

We also evaluated the uniqueness of the recovered 3D minutiae templates using their probability of false random correspondences. A statistical model was developed in this paper to more precisely compute such $\text{PRC}_{3\text{d}}$ scores from the observed distribution of minutiae templates in the *available* 3D space. Experimental results presented in Sec. 5 reveal that the probability of falsely matching 3D minutiae templates is several orders of magnitude lower than that of 2D minutiae templates. This observation can theoretically validate the merit of our approach for contactless fingerprint matching and also generate a more realistic upper bound on the uniqueness of 3D fingerprints.

Further enhancement of this work should attempt to introduce more realistic rendering algorithms, such as in [37], to render the finger skin of the synthesized 3D finger mesh grids to provide a more realistic illumination profile for contactless fingerprint images. A controllable diffusion model can be designed to accurately control the appearances of the synthesized contactless fingerprints following the style transfer steps. Further efforts to enhance the framework's speed (FingerYolo3D+PTCM) can help meet stringent requirements for the large-scale deployment of contactless fingerprint technologies.

[3] Algorithm(s) detailed in this paper are part of *US Patent No. 63/774,168.*

REFERENCES

[1] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutia cylinder-code: A new representation and matching technique for fingerprint recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2128–2141, 2010.

[2] Z. Cui, J. Feng, and J. Zhou, "Monocular 3d fingerprint reconstruction and unwarping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 8679–8695, 2023.

[3] S. Pankanti, S. Prabhakar, and A. K. Jain, "On the individuality of fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1010–1025, 2002.

[4] Z. Zhang, S. Liu, and M. Liu, "A multi-task fully deep convolutional neural network for contactless fingerprint minutiae extraction," *Pattern Recognition*, vol. 120, p. 108189, 2021.

[5] C. Dong, A. Kumar, "Synthesis of Multi-View 3D Fingerprints to Advance Contactless Fingerprint Identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13134–13151, Nov. 2023.

[6] Neurotechnology verifinger SDK 12.2, Accessed: Jul 2023. [Online], Available: https://www.neurotechnology.com/verifinger.html.

[7] C. Dong, A. Kumar, and E. Liu, "Think twice before detecting gan-generated fake images from their spectral domain imprints," in *Proc. CVPR 2022*, pp. 7865–7874.

[8] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2018, pp. 3–11.

[9] Ultralytics. Yolov5, Accessed: Jul 2023. [Online], Available: https://github.com/ultralytics/yolov5.

[10] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[11] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. CVPR 2023*, pp. 7464–7475.

[12] X. Xu, Y. Jiang, W. Chen, Y. Huang, Y. Zhang, and X. Sun, "Damo-yolo: A report on real-time object detection design," *arXiv preprint arXiv:2211.15444*, 2022.

[13] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *Proc. CVPR*, 2018, pp. 292–301.

[14] A. Malhotra, A. Sankaran, M. Vatsa, and R. Singh, "On matching finger-selfies using deep scattering networks," *IEEE IEEE Trans. Biom. Beh. & Identity Sc.*, vol. 2, no. 4, pp. 350–362, 2020.

[15] W. Zhou, J. Hu, I. Petersen, S. Wang, and M. Bennamoun, "A benchmark 3d fingerprint database," in *Proc. 11th Intl. Conf. FSKD*. IEEE, 2014, pp. 935–940.

[16] C. Lin and A. Kumar, "Matching contactless and contact-based conventional fingerprint images for biometrics identification," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 2008–2021, 2018.

[17] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. ICCV 2017* , pp. 2223–2232.

[18] Y. Zhu, S. C. Dass, and A. K. Jain, "Statistical models for assessing the individuality of fingerprints," *IEEE Tran. Info. Foren. & Sec*, vol. 2, no. 3, pp. 391–401, 2007.

[19] S. C. Dass, "Assessing fingerprint individuality in presence of noisy minutiae," *IEEE Tran. Info. Foren. & Sec.*, vol. 5, no. 1, pp. 62–70, 2010.

[20] H. Tan and A. Kumar, "Towards more accurate contactless fingerprint minutiae extraction and pose-invariant matching," *IEEE Tran. Info. Foren. & Sec*, vol. 15, pp. 3924–3937, 2020.

[21] C. Lin and A. Kumar, "Tetrahedron based fast 3d fingerprint identification using colored leds illumination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3022–3033, 2017.

[22] S. A. Grosz, J. J. Engelsma, E. Liu, and A. K. Jain, "C2cl: Contact to contactless fingerprint matching," *IEEE Tran. Info. Foren. & Sec.*, vol. 17, pp. 196–210, 2021.

[23] Y. Feng and A. Kumar, "Detecting locally, patching globally: An end-to-end framework for high speed and accurate detection of fingerprint minutiae," *IEEE Tran. Info. Foren. & Sec.*, vol. 18, pp. 1720–1733, 2023.

[24] K. Ko, "User's guide to nist biometric image software (nbis)," 2007.

[25] Y. Tang, F. Gao, J. Feng, and Y. Liu, "Fingernet: An unified deep network for fingerprint minutiae extraction," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2017, pp. 108–116.

[26] D.-L. Nguyen, K. Cao, and A. K. Jain, "Robust minutiae extractor: Integrating deep networks and fingerprint domain knowledge," in *2018 International Conference on Biometrics*. IEEE, 2018, pp. 9–16.

[27] B. Zhou, C. Han, Y. Liu, T. Guo, and J. Qin, "Fast minutiae extractor using neural network," *Pattern Recognition*, vol. 103, p. 107273, 2020.

[28] A. Kumar and C. Kwong, "Towards contactless, low-cost and accurate 3d fingerprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 681–696, 2015.

[29] S. Huang, Z. Zhang, Y. Zhao, J. Dai, C. Chen, Y. Xu, E. Zhang, and L. Xie, "3d fingerprint imaging system based on full-field fringe projection profilometry," *Optics and Lasers in Engineering*, vol. 52, pp. 123–130, 2014.

[30] F. Liu and D. Zhang, "3d fingerprint reconstruction system using feature correspondences and prior estimated finger model," *Pattern Recognition*, vol. 47, no. 1, pp. 178–193, 2014.

[31] R. D. Labati, A. Genovese, V. Piuri, and F. Scotti, "Toward unconstrained fingerprint recognition: A fully touchless 3-d system based on two views on the move," *IEEE transactions on systems, Man, and cybernetics: systems*, vol. 46, no. 2, pp. 202–219, 2015.

[32] A. Kumar, *Contactless 3D fingerprint identification*. Springer, 2018.

[33] X. Yin, Y. Zhu, and J. Hu, "3d fingerprint recognition based on ridge-valley-guided 3d reconstruction and 3d topology polymer feature extraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 1085–1091, 2019.

[34] IT Governance Privacy Team, *EU general data protection regulation (GDPR)–an implementation and compliance guide*. IT Governance Publishing, 2017, https://doi.org/10.2307/j.ctt1trkk7x.

[35] J. Priesnitz, C. Rathgeb, N. Buchmann, and C. Busch, "Syncolfinger: Synthetic contactless fingerprint generator," *Pattern Recognition Letters*, 2022.

[36] Y. Shi, Z. Zhang, S. Liu, and M. Liu, "Towards more accurate matching of contactless fingerprints with a deep geometric graph convolutional network," *IEEE IEEE Trans. Biom. Beh. & Identity Sc.*, vol. 5, no. 1, pp. 29–38, 2022.

[37] E. d'Eon, D. Luebke, and E. Enderton, "Efficient rendering of human skin," in *Proceedings of the 18th Eurographics conference on Rendering Techniques*, 2007, pp. 147–157.

[38] A. B. V. Wyzykowski, M. P. Segundo, and R. de Paula Lemes, "Level three synthetic fingerprint generation," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 9250–9257.

[39] M. Kayaoglu, B. Topcu, and U. Uludag, "Standard fingerprint databases: Manual minutiae labeling and matcher performance analyses," *arXiv preprint arXiv:1305.1443*, 2013.

[40] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "Fvc2002: Second fingerprint verification competition," in *2002 International Conference on Pattern Recognition*, vol. 3. IEEE, 2002, pp. 811–814.

[41] R. Cappelli, D. Maio, D. Maltoni, J. L. Wayman, and A. K. Jain, "Performance evaluation of fingerprint verification systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 3–18, 2005.

[42] K. Bahmani, R. Plesh, P. Johnson, S. Schuckers, and T. Swyka, "High fidelity fingerprint generation: Quality, uniqueness, and privacy," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 3018–3022.

[43] Clarkson Fingerprint Generator, Accessed: Jul 2023. [Online], Available: https://github.com/keivanB/Clarkson_Finger_Gen.

[44] A. Sankaran, M. Vatsa, and R. Singh, "Multisensor optical and latent fingerprint database," *IEEE access*, vol. 3, pp. 653–665, 2015.

[45] S. Liu, M. Zhang, P. Kadam and C.-C. J. Kuo, "3D Point Cloud Analysis: Traditional Deep Learning and Explainable Machine Learning Methods," *Springer*, 2021.

[46] A. Malhotra, A. Sankaran, M. Vatsa, R. Singh, K. B. Morris, and A. Noore, "Understanding ace-v latent fingerprint examination process via eye-gaze analysis," *IEEE IEEE Trans. Biom. Beh. & Identity Sc.*, vol. 3, no. 1, pp. 44–58, 2021.

[47] A. T. Wood, "Simulation of the von mises fisher distribution," *Communications in statistics-simulation and computation*, vol. 23, no. 1, pp. 157–164, 1994.

[48] H. Deng, T. Birdal, and S. Ilic, "Ppfnet: Global context aware local features for robust 3d point matching," in *Proc. CVPR*, 2018.

[49] B. Thirey and R. Hickman, "Distribution of euclidean distances between randomly distributed gaussian points in n-space," *arXiv preprint arXiv:1508.02238*, 2015.

[50] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[51] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[52] M. J. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *The computer journal*, vol. 7, no. 2, pp. 155–162, 1964.

[53] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007.

[54] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proc. CVPR*, 2018, pp. 8601–8610.

APPENDIX A
COMPARATIVE PERFORMANCE WITH [22]

The method in [22] recovers the 3D information from the monocular contactless 2D fingerprints and correspondingly unwarps images. These unwarped fingerprint images, instead of minutiae templates, are fed into VeriFinger to compute the match score, adopting a totally different framework as compared to the method in this paper. Despite these differences, we still performed comparative performance evaluation since both of the methods recover 3D information from single 2D image. These results are summarized in Tab. S1 using the EER index for the CL2CB and CL2CL matching. In order to ensure fair comparisons, we adhere to the same matching protocol as in [22] during such experiments.

Table S1: Comparative performance of our method with [22].

Method	Raw	Unwarp [22]	**Ours**
CL2CB	5.44%	5.00%	**0.20%**
CL2CL	7.77%	5.95%	**3.15%**

It is important to note that the reconstructed 3D information in [22] can only be fully exploited when the pose rotation is provided as ground truth. When the evaluation dataset, such as PolyU C2CL, does not include information on the finger poses, their unwarping algorithm only shows incremental improvement for the fingerprint identification task. In contrast, our proposed framework does not require the finger pose, as the PTCM module provides accurate pose alignment for given template pairs. This is one of the significant advantage of our method and is also the reason for superior results in Tab. S1.

APPENDIX B
COMPARATIVE PERFORMANCE WITH [33]

The study [33] employs an iterative process to traverse each point of a 3D minutiae template, generating a minutiae cylinder code that is topologically aligned for each minutiae point. These codes are then amalgamated to form the extracted representation of 3D minutiae template. This method is quite impressive, but it can be considered as an incremental enhancement of the 2D MCC [5] representation incorporating the 3D information. Despite its failure to incorporate 3D information in template matching, we present comparisons with [33] to demonstrate the superior performance of our 3D minutiae template matching algorithm.

Table S2: Comparative Performance of our method with [33].

Performance	EER	TAR@FAR=10^{-4}	Rank-1 Acc
Method [33]	0.68%	98.3%	97.6%
Ours	**0.36%**	**99.4%**	**99.2%**

APPENDIX C
ADDITIONAL DETAILS ON DATABASES AND PROTOCOLS

The IIIT-D Multi-Sensor (MOLF) Database [44]: Following [23], the first 200 fingerprint images from the first 5 subjects are used for training and the ground truths are first detected by VeriFinger and then manually checked for any needed correction.

Synthesized Contact-based Database [42]: Same as in [23], we use the 50048 synthetic fingerprint images provided in the released code [43] for paper [42]. To generate minutiae ground truths, we detect the minutia by VeriFinger and remove those spurious minutiae points whose quality scores are under a fixed threshold value.

Synthesized Contactless Database (Our Method in Section 6): Following the method detailed in Section 6 of this paper, we generated 1000 different fingerprint identities of 3D finger model. For each 3D finger model, we generate 10 skeletons of contactless fingerprint images as observed from -45° to +45° along the y-axis with a step of 10°. The 3D minutiae ground truths are generated along with the skeletons (ridge structures) of contactless fingerprint images. Therefore, there are a total of 10,000 skeletons of contactless fingerprint images with 3D minutiae ground truths that belong to 1000 different identities. For the 5000 skeletons of contactless fingerprint images with identities ranging from 1 to 500, we generate 5000 realistic contactless fingerprint images in CFPose database style by style transferring model introduced in this paper. For the 5000 skeletons of contactless fingerprint images with identities ranging from 501 to 1000, we generate 5000 realistic contactless fingerprint images in UWA database style by the style transfer using similar approach. These 10,000 synthesized fingerprint images, paired with 3D minutiae ground truths, are used to train the 3D minutiae detection model FingerYolo3D. To ensure fair comparison, we follow the same training protocol as in [23] to train the FingerYolo2D and therefore we don't use this synthesized database to train FingerYolo2D.

For training the GNN model which extracts the feature from the graph of minutiae template in the proposed PTCM, we use all the generated 10,000 3D minutiae ground truths belonging to 1000 identities in this synthesized database. We automatically generate a total of $45000(1000 \times 10 \times 9/2)$ pairs of the ground truth of genuine matching correspondence from the synthesized database to train the GNN. In order to enhance the robustness of the GNN model, we introduce noise to the 3D minutiae templates by randomly adding spurious 3D minutiae points, removing $10\% \sim 40\%$ 3D minutiae points, applying random transformation, adding Gaussian noise to the locations of 3D minutiae, adding noises which follows Von-Mises-Fisher distribution to the orientations of 3D minutiae.

APPENDIX D
MORE DETAILS OF INFERENCE SPEED

We evaluated the inference speed of the proposed FingerYolo3D model implemented using Pytorch running on a GTX1050 Mobile GPU (2GB memory). We convert the Pytorch model into ONNX format to compare the inference speed of the our FingerYolo3D with the VeriFinger on i9-10850K CPU. When we use the CPU to run our model, the running time for FVC2004 database is 6.25ms (5.9ms for inference and 0.35ms for non-max suppression), the running time for UWA database is 6.98ms (6.6ms for inference and 0.38ms for non-max suppression). When we use

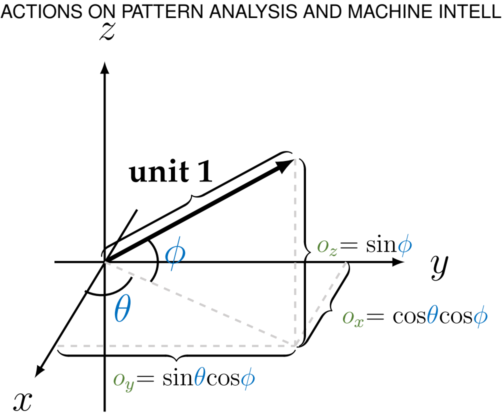


Figure S1: Illustration of angular conversion.

GTX1050 Mobile to run our proposed model, the running time for FVC2004 database is 11.5ms (7.9ms for inference and 3.6ms for non-max suppression), the running time for UWA database is 12.2ms (8.3ms for inference and 3.9ms for non-max suppression).

Appendix E
Angular Conversion

In section 4 of the paper, to facilitate computations in 3D space, we used (x, y, z, o_x, o_y, o_z) representation for the typical notations of 3D minutiae $(x, y, z, \theta, \phi, t)$ in many references. This transformation for equivalent representation used in this paper can be achieved Eq. (S1) and the illustration for such angular conversion is illustrated in Fig. S1.

$$o_x = \cos\theta\cos\phi; \quad o_y = \sin\theta\cos\phi; \quad o_z = \sin\phi. \qquad \text{(S1)}$$

Appendix F
Computing $p(C_i, C_j)$ for Induviduality Model

Algorithm S1 Calculation of $p(C_i, C_j)$

```
Input: $f_{C_i}(\vec{s}, \vec{o} \mid \Xi_{C_i})$, $f_{C_j}(\vec{s}, \vec{o} \mid \Xi_{C_j})$
Output: $p(C_i, C_j)$
Generate the $\boldsymbol{T}_{C_i} \in \mathbb{R}^{\mathcal{N}\times 6}$ following $f_{C_i}(\vec{s}, \vec{o} \mid \Xi_{C_i})$
Generate the $\boldsymbol{T}_{C_j} \in \mathbb{R}^{\mathcal{N}\times 6}$ following $f_{C_j}(\vec{s}, \vec{o} \mid \Xi_{C_j})$
initialize $num = 0$
for $k$ in range($\mathcal{N}$) do
    $(\vec{s}_{k_i}, \vec{o}_{k_i}) = \boldsymbol{T}_{C_i}[k, :]$; $(\vec{s}_{k_j}, \vec{o}_{k_j}) = \boldsymbol{T}_{C_j}[k, :]$
    if $(\vec{s}_{k_j}, \vec{o}_{k_j}) \in B(\vec{s}_{k_i}, \vec{o}_{k_i})$ then $num = num + 1$
$p(C_i, C_j) = \frac{num}{\mathcal{N}}$
```

We describe the numerical method to compute $p(C_i, C_j)$ in this section. In Alg. S1, given the minutiae distribution of class C_i denoted as $f_{C_i}(\vec{s}, \vec{o} \mid \Xi_{C_i})$, where Ξ_{C_i} includes G_{C_i} and $\{\tau_g, \vec{\mu}_g, \boldsymbol{\Sigma}_g, \vec{v}_g, \kappa_g\}$ $(g = 1, 2, ..., G_{C_i})$, we sample a large number $\mathcal{N} = 10^7$ of 3D minutiae points following $f(\vec{s}, \vec{o} \mid \Xi_{C_i})$, the sampled minutiae points are denoted as $T_{C_i} \in \mathbb{R}^{\mathcal{N}\times 6}$. We use Box-Muller Transform algorithm to sample $\mathcal{N} \times \tau_g$ 3D coordinates of minutiae denoted as $\boldsymbol{X}^g_{C_i} \in \mathbb{R}^{(\mathcal{N}\times\tau_g)\times 3}$ following trivariate normal distribution $N(\vec{\mu}_g, \boldsymbol{\Sigma}_g)$. We use the method [47] to sample $\mathcal{N} \times \tau_g$ 3D orientations of minutiae denoted as $\boldsymbol{\Theta}^g_{C_i} \in \mathbb{R}^{(\mathcal{N}\times\tau_g)\times 3}$ following Von-Mises-Fisher distribution vMF$(\vec{v}_g, \kappa_g)$. We horizontally concatenate the $\boldsymbol{X}^g_{C_i}$ and $\boldsymbol{\Theta}^g_{C_i}$ to acquire $\boldsymbol{T}^g_{C_i} \in \mathbb{R}^{(\mathcal{N}\times\tau_g)\times 6}$, then we vertically concatenate $\boldsymbol{T}^g_{C_i}$ $(g = 1, 2, ..., G_{C_i})$ to acquire $\boldsymbol{T}_{C_i} \in \mathbb{R}^{\mathcal{N}\times 6}$.

We follow the same steps to sample $\mathcal{N}$ 3D minutiae points following $f(\vec{s}, \vec{o} \mid \Xi_{C_j})$ and acquire $\boldsymbol{T}_{C_j} \in \mathbb{R}^{\mathcal{N}\times 6}$. Then we compute $p(C_i, C_j)$ using Alg. S1.

Table S3: Details of minutiae extraction backbone.

index	from	n	params	module	arguments
0	-1	1	3520	Conv	[3, 32, 6, 2, 2]
1	-1	1	18560	Conv	[32, 64, 3, 2, 0]
2	-1	1	18816	C3	[64, 64, 1]
3	-1	1	73984	Conv	[64, 128, 3, 2, 0]
4	-1	2	115712	C3	[128, 128, 2]
5	-1	1	295424	Conv	[128, 256, 3, 2, 0]
6	-1	3	625152	C3	[256, 256, 3]
7	-1	1	1180672	Conv	[256, 512, 3, 2, 0]
8	-1	1	1182720	C3	[512, 512, 1]
9	-1	1	656896	SPPF	[512, 512]
10	-1	1	131584	Conv	[512, 256, 1, 1, 0]
11	-1	1	0	Upsample	[2, 'nearest']
12	[-1, 6]	1	0	Concat	[1]
13	-1	1	361984	C3	[512, 256, 1, 0]
14	-1	1	33024	Conv	[256, 128, 1, 1]
15	-1	1	0	Upsample	[2, 'nearest']
16	[-1, 4]	1	0	Concat	[1]
17	-1	1	90880	C3	[256, 128, 1, 0]
18	[17]	1	11610	Detect	described in paper

Appendix G
Additional Details on Neural Network Training

G.1 Minutiae Detecting Task

Backbone Design: In Tab. S3, we provide a comprehensive summary of the parameter details for each layer of the minutiae extraction network presented in this paper. The 'index' column identifies the layer's index, while a value of '-1' in the 'from' column indicates that the layer is connected to the preceding one. We also specify the number of blocks and parameters for each layer. For the Conv module, the arguments listed correspond to the input channels, output channels, kernel size, stride, and padding, respectively. In the case of the 'C3' module (a CSP bottleneck with three convolutional layers), the arguments denote the input channels, output channels, and the repetition count of the hidden layers within the Bottleneck implementation. Regarding the SPPF module, its arguments indicate the input and output channels. Lastly, for the Upsample operation, the arguments provided represent the upsampling scale factor and the method employed for upsampling.

Training Strategy: We utilize SGD for optimization, setting the learning rate to 1×10^{-3}. The batch size is configured to 32. The initial three epochs serve as warm-up epochs, during which the learning rate is set to 0.1. The network is trained over 150 epochs, and optimal checkpoints are selected based on the lowest loss achieved on the training dataset. For data augmentation during training, we apply random rotations within the range of -30° to 30°, random rescaling from 0.75 to 1.25, random pixel shifts ranging from -50 to 50 pixels, and adjustments to brightness and contrast.

Anchor Box Settings: As in [10], during the network training, anchor boxes consist of a set of predefined boxes with fixed height and width, which are used to capture the scale and aspect ratio of the object to be detected. The shape and size of anchor boxes are always selected by clustering algorithms according to the sizes of objects in the training dataset. Different from popular Yolo architectures such as Yolov5, which provides nine different anchor templates for objects in various scales and shapes, we only provide three squared sizes of anchor templates in our experiments to fit the context of minutiae templates: 6×6, 9×9 and 12×12 pixels. During training, we assigned the anchor box, which is the most similar size to the current minutiae ground truth, as the one responsible for predicting the respective minutiae's 2D location and width.

Details on Loss Functions: In the total loss function, the $\lambda_{obj}, \lambda_{box}, \lambda_{cls}, \lambda_z, \lambda_o$ are set to 1.0, 0.05, 0.25, 0.15, and 0.15 respectively for all the experiments in this paper. We define the l_{obj}, l_{cls} and l_{box} following the same definition as in Yolov5 [10], the l_{obj} and l_{cls} are BCE loss, the l_{box} is CIOU loss. For 3D features prediction, the l_z is BCE loss, and we set w_θ, w_ϕ and δ to 1.5, 0.5 and 0.2 for l_o since we observed during the network training that the weight of θ should be larger to help the convergence.

G.2 GNN for Correspondence Finding

In Tab. S4, we list the parameters of the developed GNN for finding the minutiae correspondence.

Table S4: Details of GNN Backbone.

Layer	Description	Configuration
mlp1_1	Sequential MLP	FC(4, 128) , BN(128) , ReLU() , FC(128, 196) , BN(196) , ReLU() , FC(196, 196)
mlp1_2	Sequential MLP	FC(196, 196) , BN(196) , ReLU() , FC(196, 256) , BN(256) , ReLU() , FC(256, 256)
conv1	PPFConv	local_nn=mlp1_1, global_nn=mlp1_2
mlp2_1	Sequential MLP	FC(260, 256) , BN(256) , ReLU() , FC(256, 256) , BN(256) , ReLU() , FC(256, 256)
mlp2_2	Sequential MLP	FC(256, 256) , BN(256) , ReLU() , FC(256, 256) , BN(256) , ReLU() , FC(256, 256)
conv2	PPFConv	local_nn=mlp2_1, global_nn=mlp2_2
mlp3_1	Sequential MLP	FC(260, 256) , BN(256) , ReLU() , FC(256, 256) , BN(256) , ReLU() , FC(256, 256)
conv3	PPFConv	local_nn=mlp3_1, global_nn=None
SAGElayer0	SageConv	SAGEConv(256, 256)
bn0	BatchNorm1d	BN(256)
SAGElayer1	SageConv	SAGEConv(256, 256)
bn1	BatchNorm1d	BN(256)

We employ SGD to optimize the training of this architecture. Specifically, the learning rate, weight decay, and momentum are configured to 1×10^{-3}, 1×10^{-4}, and 0.9, respectively. The batch size is set to 64 with a total of 20 epochs, while the optimal checkpoints are selected based on the minimal loss observed on the training dataset. For data augmentation, our methodology includes applying random rotations within a range of -30° to 30° and random rescaling factors from 0.75 to 1.25. Additionally, to further enhance model robustness, we randomly eliminate up to 30% of the matching correspondences from the ground truths and introduce random shifts to the coordinates of the template's minutiae points.

Appendix H
Additional Details on Fingerprint Database Synthesis

This section provides additional details on the method for synthesizing realistic contactless 3D fingerprint images with their minutiae ground truths. These details are summarized in the following Sec. H.1 to Sec. H.5.

H.1 Generating Controlling Points from Real 3D Finger Surface

Given real finger surfaces, we initially label the control points of a Bezier surface to approximate these surfaces using an iterative brute-force global search algorithm. This process ensures that the points on the generated Bezier surface achieve the minimum Chamfer Distance (CD) relative to the points on the real finger surface.

Given that the brute-force search method is computationally intensive, we introduce a neural network, denoted as $\boldsymbol{\omega}$ consisting of three MLP layers to predict the optimal controlling points P of the given point cloud. We sample 20 finger surfaces from the PolyU3D database [R8] and use the brute-force search method described above to label the ground truth control points for these samples. We sample 20 finger surfaces from the PolyU3D database [R8] and use the brute-force search method described above to label the ground truth control points for these samples. To generate a larger dataset of controlling points, we apply principal component analysis (PCA) to the 20 sets of controlling points and compute the most representative basis of controlling points, then linear combination of the basis of controlling points is adopted to acquire a large number of diverse controlling points. We denote the generated control points as $\{P_{gt_1}, P_{gt_2}, \dots\}$. The corresponding Bezier surfaces $\{\mathcal{B}_1, \mathcal{B}_2, \dots\}$ are generated from $\{P_{gt_1}, P_{gt_2}, \dots\}$. For each set of control points, we sample 1024 points from the corresponding Bezier surface (if the number of points of the given point cloud is less than 1024, up-sample the point cloud to acquire 1024 points). The sampled point clouds are denoted as $\{\boldsymbol{X}_{gt_1}, \boldsymbol{X}_{gt_2}, \dots\}$. To ensure the robustness of the model $\boldsymbol{\omega}$, random Gaussian noise $\boldsymbol{\epsilon}$ is added to the point clouds' locations, generating $\boldsymbol{X}_{gt_i} \leftarrow \boldsymbol{X}_{gt_i} + \boldsymbol{\epsilon}$.

The controlling points prediction model takes data with the shape of $\mathbb{R}^{3\times 1024}$ as input, and it outputs the predicted 3D locations of controlling points with the shape of $3 \times 17 \times 17$. We input the $\boldsymbol{X}_{gt_i}$ into the model and acquire the predicted controlling points $P'_i = \boldsymbol{\omega}(\boldsymbol{X}_{gt_i})$. We feed $\{(\boldsymbol{X}_{gt_1}, P_{gt_1}), (\boldsymbol{X}_{gt_2}, P_{gt_2}), \dots\}$ into the model $\boldsymbol{\omega}$ and optimize the model using the square loss to minimize the l_2 distance between the $\{P_{gt_1}, P_{gt_2}, \dots\}$ and the predicted controlling points $\{P'_1, P'_2, \dots\}$.

After the parameters $\boldsymbol{\omega}$ is acquired, we use this pre-trained model $\boldsymbol{\omega}$ to compute the controlling points $\{P_{\text{real}_1}, P_{\text{real}_2}, \dots\}$ from the point clouds $\{\mathcal{S}_{\text{real}_1}, \mathcal{S}_{\text{real}_2}, \dots\}$ of finger surfaces sampled from the real 3D finger database.

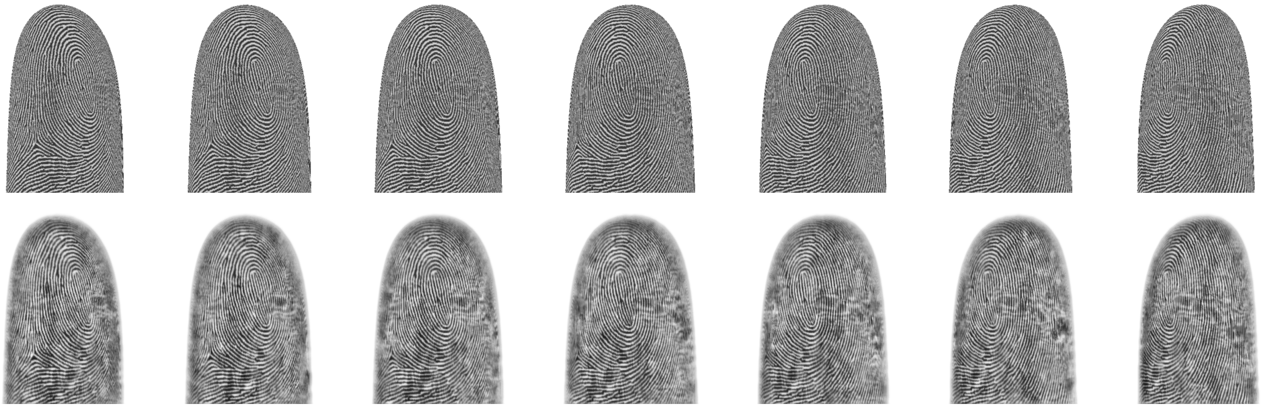


Figure S2: Visualization for the fingerprints following style transfer.

H.2 Generating Bezier Surface from P

Given a set of control points P, the Bezier surface is generated as follows:

$$P(u,v) = \sum_{i=0}^{M}\sum_{j=0}^{N} B_i^M(u)B_j^N(v)P_{ij}; u \in [0,1], v \in [0,1], \tag{S2}$$

where $B_a^b(t) = \frac{b!}{a!(b-a)!}t^a(1-t)^{b-a}$.

The surface normal $N(u,v)$ for the Bezier surface $P(u,v)$ is computed using the following Eqs. (S3) to (S5). These are utilized during the steps (in H4 discussed below) of 3D ridge engraving and the computation for the 3D minutiae ground truth.

$$N(u,v) = \text{norm}(\frac{\partial}{\partial u}P(u,v) \times \frac{\partial}{\partial v}P(u,v)), \tag{S3}$$

where $\text{norm}(\vec{n}) = \frac{\vec{n}}{\|\vec{n}\|}$.

$$\begin{aligned} \frac{\partial}{\partial u}P(u,v) &= \sum_{i=0}^{M}\sum_{j=0}^{N} \frac{\partial B_i^M(u)}{\partial u} B_j^N(v)P_{ij} \\ \frac{\partial B_i^M(u)}{\partial u} &= \frac{M!}{i!(M-i)!}\{iu^{i-1}(1-u)^{M-i} \\ &\quad +(-1)u^i(M-i)(1-u)^{M-i-1}\} \end{aligned} \tag{S4}$$

$$\begin{aligned} \frac{\partial}{\partial v}P(u,v) &= \sum_{i=0}^{M}\sum_{j=0}^{N} B_i^M(u)\frac{\partial B_j^N(v)}{\partial v} P_{ij} \\ \frac{\partial B_j^N(v)}{\partial v} &= \frac{N!}{j!(N-j)!}\{jv^{j-1}(1-v)^{N-j} \\ &\quad +(-1)v^j(N-j)(1-v)^{N-j-1}\} \end{aligned} \tag{S5}$$

H.3 3D-2D Textural Mapping

We use the interpolated Bezier ISOMAP [1] to compute the 3D-2D textural mapping relationship for uniform Bezier surface, and the steps for such computations are summarized in in Alg. S2. The variable P in this algorithm represents the control points to generate the Bezier surface. The ρ_{sparse} and ρ_{dense} denote the required maximum point cloud spacing, respectively, for the sparse and dense Bezier surfaces, where $\rho_{sparse} \gg \rho_{dense}$. The dense point cloud of 3D finger surface $\mathcal{S}_{dense}^{3D}$ and its projected 2D points $\mathcal{S}_{dense}^{2D}$ are acquired. Each row of $\mathcal{S}_{dense}^{3D}$ and $\mathcal{S}_{dense}^{2D}$ has a one-to-one correspondence for 3D-to-2D textural mapping. The surface normal of $\mathcal{S}_{dense}^{3D}$ is denoted as N_{dense}^{3D} in this algorithm.

Algorithm S2 Bezier 3D Finger Surface Synthesis

```
Input: $P$, $\rho_{sparse}$, $\rho_{dense}$
Output: $\mathcal{S}_{dense}^{3D}$, $\mathcal{S}_{dense}^{2D}$
Generate sparse uniformed point cloud of finger surface $\mathcal{S}_{sparse}^{3D}$ from $P$, $\rho_{sparse}$ using Alg. 1 in this paper
Generate dense uniformed point cloud of finger surface $\mathcal{S}_{dense}^{3D}$ from $P$, $\rho_{dense}$ using Alg. 1 in this paper
Input $\mathcal{S}_{sparse}^{3D}$, $\mathcal{S}_{dense}^{3D}$ to interpolated Bezier ISOMAP algorithm in [1] and acquire $\mathcal{S}_{dense}^{2D}$
```

H.4 3D Ridge Engraving

The ridge-valley patterns on the acquired 3D finger surface $\mathcal{S}_{dense}^{3D}$ are engraved similarly to those in [1]. First, we align the $\mathcal{S}_{dense}^{2D}$ with the 2D ridge-valley pattern image of the synthesized contact-based fingerprint. After alignment, we generate the boolean matrix g' of the same size as $\mathcal{S}_{dense}^{3D}$: if the 2D coordinate ($\mathcal{S}_{dense}^{2D}[0][i]$,$\mathcal{S}_{dense}^{2D}[1][i]$) is aligned with a pixel that belongs to a ridge, we set the corresponding ($g'[0][i]$,$g'[1][i]$,$g'[2][i]$)=$(1,1,1)$, else if the coordinate ($\mathcal{S}_{dense}^{2D}[0][i]$,$\mathcal{S}_{dense}^{2D}[1][i]$) is aligned with a pixel that belongs to a valley, we set the corresponding ($g'[0][i]$,$g'[1][i]$,$g'[2][i]$)=$(0,0,0)$. Then, the ridge engraving is implemented using Eq. (S6) to generate the engraved point cloud $\mathcal{S}_{\text{eng}}$ for the respective 3D finger surface.

$$\mathcal{S}_{\text{eng}} = \mathcal{S}_{dense}^{3D} + N_{dense}^{3D} \times g' \times \eta \tag{S6}$$

where η controls the depth of engraved patterns and is empirically set to 2×10^{-6}.

H.5 Synthesis of Contactless Fingerprint Image with Respective Minutiae Ground Truth

The ground truth labels of the 3D minutiae for the synthesized 3D finger surfaces are generated using the same method as detailed in section III of [1]. We simulate the multiple views of the acquired finger surface $\mathcal{S}_{\text{eng}}$ in 3D space by rotating it from -45° to +45° along the y-axis with a step of 10°, and the 3D ground truth of minutiae is also rotated simultaneously. For each of the view angles, the 3D finger surface from the world coordinate system is projected onto the camera coordinate system to acquire the skeleton image of the contactless fingerprint. For each of the skeleton images, we use the style transfer model discussed in section 6 of this paper to generate a more realistic contactless fingerprint image. Therefore, there are ten realistic contactless fingerprints with minutiae ground truths for each of the 3D finger surfaces.

Fig. S2 shows several samples of these images. The first row displays the synthesized skeletons, while the second row presents the corresponding contactless fingerprint images following the style transfer.

References

[1] C. Dong, A. Kumar, "Synthesis of Multi-View 3D Fingerprints to Advance Contactless Fingerprint Identification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 3, pp. 1085–1091, 2023.

[2] C. Lin and A. Kumar, "Matching contactless and contact-based conventional fingerprint images for biometrics identification," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 2008–2021, 2018.

[3] H. Tan and A. Kumar, "Towards more accurate contactless fingerprint minutiae extraction and pose-invariant matching," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3924–3937, 2020.

[4] C. Lin and A. Kumar, "Tetrahedron based fast 3d fingerprint identification using colored leds illumination," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 3022–3033, 2017.

[5] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutia cylinder-code: A new representation and matching technique for fingerprint recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2128–2141, 2010.

[6] Z. Zhang, S. Liu, and M. Liu, "A multi-task fully deep convolutional neural network for contactless fingerprint minutiae extraction," *Pattern Recognition*, vol. 120, p. 108189, 2021.

[7] Neurotechnology verifinger SDK 12.2, Accessed: Jul 2023. [Online], Available: https://www.neurotechnology.com/verifinger.html.

[8] C. Dong, A. Kumar, and E. Liu, "Think twice before detecting gan-generated fake images from their spectral domain imprints," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7865–7874.

[9] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2018, pp. 3–11.

[10] Ultralytics. Yolov5, Accessed: Jul 2023. [Online], Available: https://github.com/ultralytics/yolov5.

[11] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[12] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.

[13] X. Xu, Y. Jiang, W. Chen, Y. Huang, Y. Zhang, and X. Sun, "Damo-yolo: A report on real-time object detection design," *arXiv preprint arXiv:2211.15444*, 2022.

[14] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 292–301.

[15] A. Malhotra, A. Sankaran, M. Vatsa, and R. Singh, "On matching finger-selfies using deep scattering networks," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, no. 4, pp. 350–362, 2020.

[16] W. Zhou, J. Hu, I. Petersen, S. Wang, and M. Bennamoun, "A benchmark 3d fingerprint database," in *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. IEEE, 2014, pp. 935–940.

[17] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[18] S. Pankanti, S. Prabhakar, and A. K. Jain, "On the individuality of fingerprints," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 8, pp. 1010–1025, 2002.

[19] Y. Zhu, S. C. Dass, and A. K. Jain, "Statistical models for assessing the individuality of fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 391–401, 2007.

[20] S. C. Dass, "Assessing fingerprint individuality in presence of noisy minutiae," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 62–70, 2010.

[21] S. A. Grosz, J. J. Engelsma, E. Liu, and A. K. Jain, "C2cl: Contact to contactless fingerprint matching," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 196–210, 2021.

[22] Z. Cui, J. Feng, and J. Zhou, "Monocular 3d fingerprint reconstruction and unwarping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[23] Y. Feng and A. Kumar, "Detecting locally, patching globally: An end-to-end framework for high speed and accurate detection of fingerprint minutiae," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1720–1733, 2023.

[24] K. Ko, "User's guide to nist biometric image software (nbis)," 2007.

[25] Y. Tang, F. Gao, J. Feng, and Y. Liu, "Fingernet: An unified deep network for fingerprint minutiae extraction," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2017, pp. 108–116.

[26] D.-L. Nguyen, K. Cao, and A. K. Jain, "Robust minutiae extractor: Integrating deep networks and fingerprint domain knowledge," in *2018 International Conference on Biometrics (ICB)*. IEEE, 2018, pp. 9–16.

[27] B. Zhou, C. Han, Y. Liu, T. Guo, and J. Qin, "Fast minutiae extractor using neural network," *Pattern Recognition*, vol. 103, p. 107273, 2020.

[28] A. Kumar and C. Kwong, "Towards contactless, low-cost and accurate 3d fingerprint identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 681–696, 2015.

[29] S. Huang, Z. Zhang, Y. Zhao, J. Dai, C. Chen, Y. Xu, E. Zhang, and L. Xie, "3d fingerprint imaging system based on full-field fringe projection profilometry," *Optics and Lasers in Engineering*, vol. 52, pp. 123–130, 2014.

[30] F. Liu and D. Zhang, "3d fingerprint reconstruction system using feature correspondences and prior estimated finger model," *Pattern Recognition*, vol. 47, no. 1, pp. 178–193, 2014.

[31] R. D. Labati, A. Genovese, V. Piuri, and F. Scotti, "Toward unconstrained fingerprint recognition: A fully touchless 3-d system based on two views on the move," *IEEE transactions on systems, Man, and cybernetics: systems*, vol. 46, no. 2, pp. 202–219, 2015.

[32] A. Kumar, *Contactless 3D fingerprint identification*. Springer, 2018.

[33] X. Yin, Y. Zhu, and J. Hu, "3d fingerprint recognition based on ridge-valley-guided 3d reconstruction and 3d topology polymer feature extraction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 3, pp. 1085–1091, 2019.

[34] I. G. P. Team, *EU general data protection regulation (gdpr)–an implementation and compliance guide*. IT Governance Ltd, 2020.

[35] J. Priesnitz, C. Rathgeb, N. Buchmann, and C. Busch, "Syncolfinger: Synthetic contactless fingerprint generator," *Pattern Recognition Letters*, 2022.

[36] Y. Shi, Z. Zhang, S. Liu, and M. Liu, "Towards more accurate matching of contactless fingerprints with a deep geometric graph convolutional network," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 5, no. 1, pp. 29–38, 2022.

[37] E. d'Eon, D. Luebke, and E. Enderton, "Efficient rendering of human skin," in *Proceedings of the 18th Eurographics conference on Rendering Techniques*, 2007, pp. 147–157.

[38] A. B. V. Wyzykowski, M. P. Segundo, and R. de Paula Lemes, "Level three synthetic fingerprint generation," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 9250–9257.

[39] M. Kayaoglu, B. Topcu, and U. Uludag, "Standard fingerprint databases: Manual minutiae labeling and matcher performance analyses," *arXiv preprint arXiv:1305.1443*, 2013.

[40] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "Fvc2002: Second fingerprint verification competition," in *2002 International Conference on Pattern Recognition*, vol. 3. IEEE, 2002, pp. 811–814.

[41] R. Cappelli, D. Maio, D. Maltoni, J. L. Wayman, and A. K. Jain, "Performance evaluation of fingerprint verification systems," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 1, pp. 3–18, 2005.

[42] K. Bahmani, R. Plesh, P. Johnson, S. Schuckers, and T. Swyka, "High fidelity fingerprint generation: Quality, uniqueness, and privacy," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 3018–3022.

[43] Clarkson Fingerprint Generator, Accessed: Jul 2023. [Online], Available: https://github.com/keivanB/Clarkson_Finger_Gen.

[44] A. Sankaran, M. Vatsa, and R. Singh, "Multisensor optical and latent fingerprint database," *IEEE access*, vol. 3, pp. 653–665, 2015.

[45] S. Liu, M. Zhang, P. Kadam and C.-C. J. Kuo, "3D Point Cloud Analysis: Traditional Deep Learning and Explainable Machine Learning Methods," *Springer*, 2021.

[46] A. Malhotra, A. Sankaran, M. Vatsa, R. Singh, K. B. Morris, and A. Noore, "Understanding ace-v latent fingerprint examination process via eye-gaze analysis," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 1, pp. 44–58, 2021.

[47] A. T. Wood, "Simulation of the von mises fisher distribution," *Communications in statistics-simulation and computation*, vol. 23, no. 1, pp. 157–164, 1994.

[48] H. Deng, T. Birdal, and S. Ilic, "Ppfnet: Global context aware local features for robust 3d point matching," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[49] B. Thirey and R. Hickman, "Distribution of euclidean distances between randomly distributed gaussian points in n-space," *arXiv preprint arXiv:1508.02238*, 2015.

[50] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[51] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[52] M. J. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *The computer journal*, vol. 7, no. 2, pp. 155–162, 1964.

[53] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007.

[54] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8601–8610.