Dense Identification of 3D Facial Landmarks by Utilizing 2D as Intermediate

Tahlia Wu[1], Elwin Li[2], Paul Kronlund[3]
[1] Cupertino High School, California, CA
[2] Mountain View High School, Mountain View, CA
[3] Lycée Racine, Paris, France

Abstract

2D facial landmark identification is a well-established research area, with Dlib and MediaPipe models achieving high success rates. 3D facial landmarks identification on point clouds, however, is a less prominent research area due to the complexity of topology variation. Previous automated 3D facial landmark identification approaches include resource-intensive AI training, with the capability to recognize only ~12-68 typologically distinct features (tip of the nose/eyes). In this paper, we present an approach that identifies 468 3D landmarks. To our knowledge, our algorithm's results are the densest to date and the first to identify landmarks on topologically smooth regions (cheeks, forehead). Our approach utilizes 2D depth maps as an intermediate to circumvent the complexity of 3D data while still relying solely on spatial data. Our approach projects a 3D point cloud onto an optimized 2D depth map, utilizes existing robust 2D models, and maps landmarks back to the original 3D point cloud. After testing on the LYHM dataset, our pipeline is shown to yield high accuracy with error rates of $1.03 \pm 0.08\%$ ($\propto$=0.01), showing the potential of 2D representations in processing 3D data.

1. Introduction

3D facial landmarks are crucial for 3D human face reconstruction, which has many applications such as Face ID, animations, and Virtual Reality. A failure to accurately identify landmarks may lead to lower quality in such applications. 3D facial data is computationally complex due to inconsistencies between different point clouds. Point clouds are a list of (x, y, z) coordinates that represent the surface of a face. Each point cloud has unique scaling, point order, and number of points, which makes processing 3D faces challenging. Labeled 3D landmarks are points signifying specific facial features such as the tip of the nose or the corners of the eye. Landmarks provide a comparable point of equality across two point clouds, and they are the starting points for many processes that manipulate 3D faces. One example is ICP or iterative closest point needed for model fitting **[9]**.

For datasets without labeled landmarks, it can be labor-intensive to label thousands of samples by hand. We are interested in building an automated algorithm that identifies these landmarks. Existing related works propose using heat maps to predict the probability of a landmark in each position **[2]** or conformal (UV) mapping for a 2D representation of point clouds **[3]**. These works could only detect a limited number of landmarks with distinctive topological characteristics such as the tip of the nose or eyes. In an approach similar to ours, Li proposes an alternative strategy of mapping 3D point clouds with texture into a 2D image, similar to taking a front-view picture **[1]**. The Dlib facial landmark model **[6]** is applied to these images to detect 68 crucial landmarks on the eyes, nose, mouth, and around the face. One caveat of this approach is that it relies on the use of texture which not all point clouds contain, such as those in the FRGC dataset **[7]**.

We built upon Li's 2D-based approach to develop an algorithm that utilizes only spatial data in generating a 2D image of the point cloud. By incorporating Mediapipe **[8]**, an extremely well-researched 2D landmark detection model, we obtained high success rates in our approach. This approach also means our pipeline is easily customizable for various researchers' needs as it doesn't entail training an entirely new deep learning network.

2. Methods

Overview

Our method applies an established 2D facial landmark detection pipeline (MediaPipe) to detect the landmarks on 2D representations of a point cloud. Then we transform these landmarks back to the original 3D point cloud. Our pipeline is summarized into 4 main steps.

1) Point cloud preprocessing
2) 3D to 2D: Point Cloud to Depth Map
3) Landmark Identification on Depth Map
4) 2D to 3D: Depth Map Landmarks to Original Point Cloud

2.1 Point Cloud Preprocessing

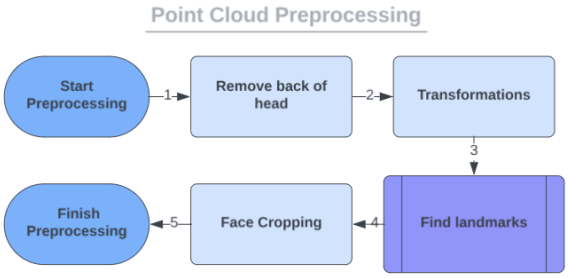


Figure 1. *A flowchart of our overall point cloud preprocessing process.*

Visibility - Removing Invisible Points

Point clouds containing invisible points from the camera angle may cause inaccurate depth map generation because the depth map treats the point cloud as one surface. For example, point clouds with the back of the heads often result in depth maps with black regions, which 2D identification pipelines (MediaPipe, Dlib) do not work on. We apply concepts from Katz's work on visibility on point clouds **[5]** to determine which points are visible from the front view–or which points are part of the frontal portion of the head.

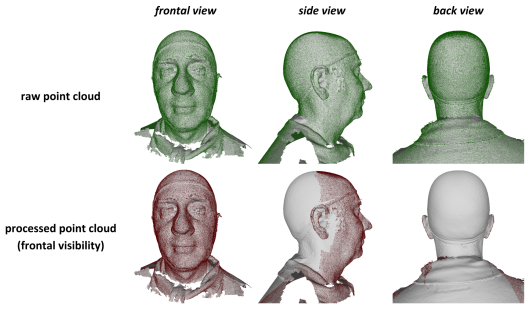


Figure 2. *A comparison between the point cloud before (green) and after (red) incorporation of visibility preprocessing.*

Transformations to Image System Coordinates

Point clouds can have a diversity of scaling and coordinate systems, so Point clouds must be scaled and translated to fit within the image coordinate system. As the image system only contains x and y, the z-axis is disregarded in these calculations. First, the point cloud is translated so that its center is at (0,0) by calculating the middle value of the range of data in each dimension and subtracting all data points by such middle values. Then to scale the image to fit

the image dimensions of 256 pixels, we convert the axis with the highest range to fit inside [-128, 128]. This optimizes the size of the face in the image while ensuring no parts of the face are cut off.

Frontal Face Cropping

Finally, point clouds may be cropped to only contain the facial portion, removing unnecessary regions in the raw point cloud (e.g. shoulders, back of the head, neck, ears, and hair). Our cropping algorithm utilizes rough landmarks for 7 points on the face (4 eye corners, the bottom tip of the nose, and 2 lip corners).

2.2 3D to 2D: Point Cloud to Depth Map

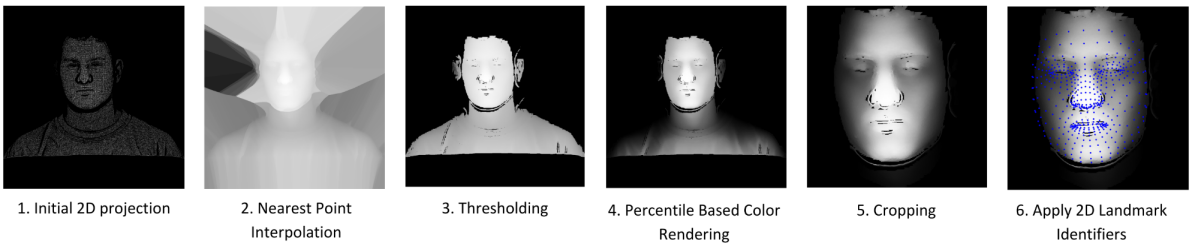


Figure 3. *A visual overview of our algorithm converting 3D point cloud data into an optimized 2D depth map for 2D landmark identification.*

The essence of the initial 3D to 2D conversion is to generate pixels that represent the surface of the face by using the z-axis value in each point for its color. Each point on the point cloud is assigned a color value (0-1) based on the z value relative to the rest of the point cloud using percentile. With this percentile-based color system, the point with the median z-value will be assigned a 0.5 color value. This is done to create a more uniform and stable spread of color values in conditions with lots of points on the extremes of the z-value range. This solves the issue of depth maps having a white hot spot in the nose region (noses are at the front of the face, and thus have high z values) and very dark in the rest of the face. This creates sharper images for an increased likelihood of landmark identification success in 2D.

However, most point clouds aren't dense enough to generate smooth depth maps. (The density of point clouds refers to the number of points in the file.) To circumvent this issue of spotty image generation, we developed a nearest point interpolation algorithm to fill in the holes in the image to create a smooth, polished look. Each pixel in the depth map image is assigned the average of the color values of the 3 closest points in the x and y plane on the input point cloud with the kd-tree structure to increase efficiency. Finally, a thresholding check is applied to ensure that color is only filled in regions that truly represent the interior of the point cloud and not the background. If the pixel is not within the x and y range of the k closest points on the point cloud, it was omitted. The value k=15 was chosen after visual confirmation. But K can be adjusted, especially with point clouds of significantly dense and sparse data, by using higher and lower k values respectively.

2.3 Landmark Identification on Depth Map

After generating an optimized depth map, a robust 2D landmark identifier is applied to identify 2D facial landmarks. We found that MediaPipe performs well on depth maps, producing 468 (x,y) landmark coordinates.

2.4 2D to 3D: Depth Map Landmarks to Original Point Cloud

To project (x,y) landmarks back to the original 3D point cloud, each landmark is correlated to the closest x and y values of the (x, y, z) points in the preprocessed point cloud. It's important to map it back to the preprocessed point

cloud to avoid landmark localization to the back of the head. The final step is to then map the 468 landmarks back into the scale of the original point cloud.

3. Results

Our pipeline was tested on 2 datasets of real capture data, the Face Recognition Grand Challenge (FRGC) and Liverpool-York Head Model (LYHM) datasets **[4][7]**.

3.1 Results on the FRGC Dataset

The FRGC dataset contains point clouds of faces with hair and shoulders, without a back of the head. As the FRGC dataset does not contain ground true landmarks for quantitative analysis, we demonstrate the success of our pipeline in Figure 4a.

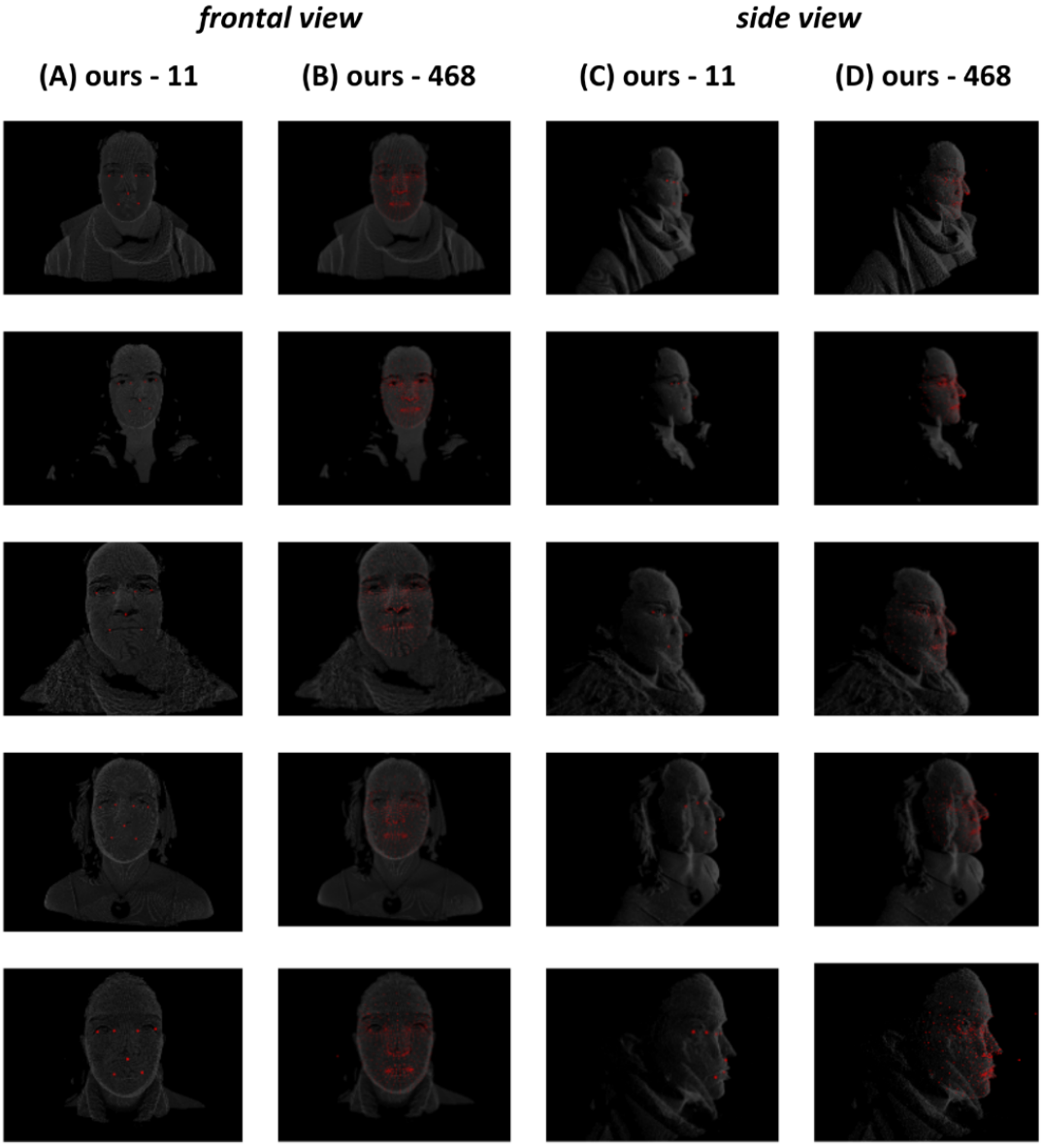


Figure 4a. *Example landmark outputs of our pipeline on faces in the FRGC dataset. Each row represents one sample with white points representing the raw point cloud and red points representing the landmarks predicted by our pipeline. Columns A and C portray an abridged 11 landmarks (refer to Table 2 below) of the 468 identified for ease of visualization. (The visual is dark because FRGC only contains raw spatial (x, y, z) pairs, unlike the mesh in the LHYM dataset.)*

3.2 Results on LHYM dataset

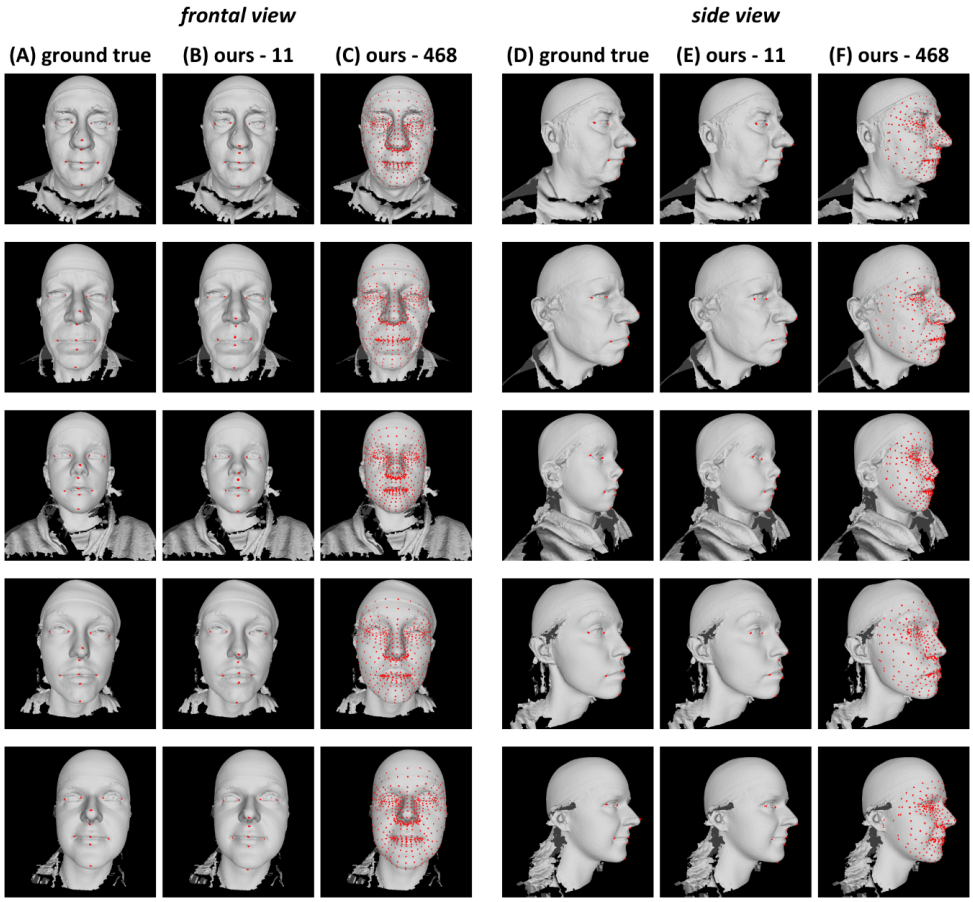


Figure 4b. *A qualitative comparison between ground true and predicted landmarks (in red) on scans from the LYHM dataset. Columns B and E visualize 11 landmarks (Table 2) identified by our pipeline which parallels the ground true landmarks in Columns A and D. Columns C and F visualize the full 468 landmarks identified by our pipeline.*

The LYHM dataset contains real capture data with dense point clouds that include regions on the back of the head and shoulders. For our evaluation, we used a subset of 129 pairs of point clouds and their labeled landmarks.

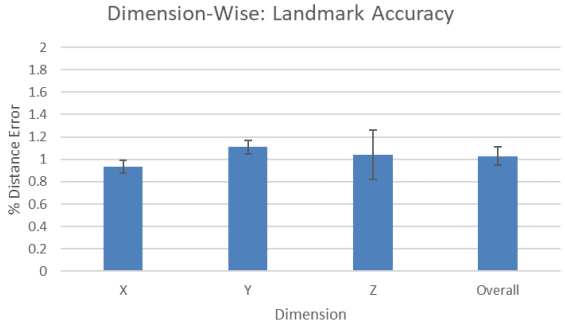
The LYHM dataset contained 68 labeled ground true landmarks, however, analysis was conducted on only 11 landmarks as not all MediaPipe landmarks have an equivalent to the 68 facial landmark coordinates. The calculations of error are based on percentage error values (Formula 1). As this error accounts for variability in point cloud size, our quantitative accuracy can be applied to point clouds of any size and density.

$$error = \frac{|predicted\ landmark\ axis\ value - true\ landmark\ axis\ value|}{axis\ range}\ x\ 100\%$$

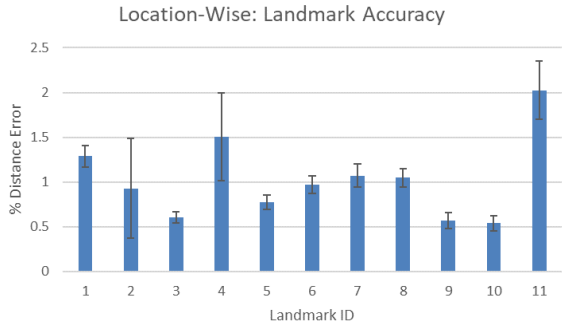
Formula 1. *The calculation for landmark identification distance error based on relative*

landmark ID	facial landmark	mp ID	dlib ID
1	left corner left eye	33	36
2	right corner left eye	133	39
3	left corner right eye	362	42
4	right corner right eye	263	45
5	bottom tip of nose	2	33
6	left corner lips	76	48
7	right corner lips	292	54
8	tip of nose	4	30
9	top center lips	0	51
10	bottom center lips	17	57
11	chin	152	8

Table 2. *A description of the 11 landmarks utilized for our quantitative accuracy analysis including ID value in our analysis and location. The right two columns contain the MediaPipe indices (mp ID) and Dlib's 68 landmark indices (dlib ID) used when comparing ground truth against our predicted landmarks.*



Graph 2. Accuracy of landmark identification with each dimension and overall. α = 0.01



Graph 3. Accuracy of landmark identification with each landmark location specified in Table 2. α = 0.01

Within the 129 identities in the LYHM sub-dataset we utilized, our pipeline returned 468 landmarks for 126, yielding a success rate of 97.7%. Our pipeline performs with consistent high accuracy performance in all 3 dimensions (Figure 5). Calculating an overall error using the arithmetic mean for the error of x, y, and z, the 0.99 confidence interval for average overall error is 1.03 ± 0.08%. As the LYHM dataset also had a diversity of different additional features, our approach is shown to be robust in handling point clouds with and without additional features.

Evaluating performance by landmark location (Figure 6), landmarks are identified fairly uniformly in accuracy. However, the chin landmark (id = 11) has a greater standard error as 6 samples produced errors in the z-axis above 10%, occurring from visibility issues. This was our highest error demographic with a mean of 3.14% z-dimension error for the chin landmark. In comparing errors across the 3 dimensions, the z-axis was shown to be more volatile and unreliable. This can be attributed to the flattening of the z dimension as we move from 3D to 2D data in our pipeline. But overall performance is still promising with a z-dimension error of 1.04%.

3.3 Ablation Studies

A. Cropping

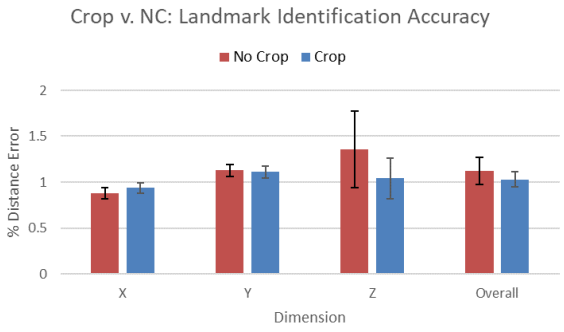


Figure 7. The overall accuracy of landmark identification between crop and no crop approaches.

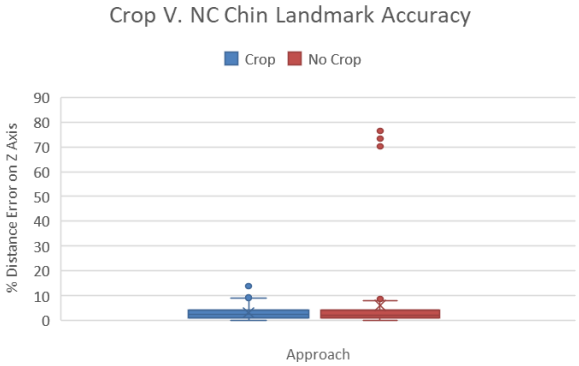


Figure 8. Box plot distribution of z-axis error for chin landmark with crop and no crop approaches.

In Figure 7, the difference in performance between cropping and no cropping is statistically insignificant although removing cropping does result in a higher average error (1.12% from 1.03% overall error). The greatest improvement occurred with eliminating high outliers for error (e.g. 80%) (Figure 8). These high outliers with the no crop algorithm occur when landmarks are identified on the back of the head, which originates from the slight errors in our preprocessing visibility algorithm. The cropping steps solve this issue by ensuring only points on the front of the face are candidates. As such, cropping can be utilized as a secondary measure of visibility.

B. Percentile in Color Generation in Depth Map

Replacing the percentile function with an approach that directly utilizes the z value (Formula 2) in the color generation step resulted in a 6.2% lower success rate in landmark identification. Success is defined as if the pipeline returns predicted landmarks. Within the 129 faces in the LYHM sub-dataset, 126/129 (97.7%) point clouds had landmarks identified with the incorporation of percentile color assignments, compared to only 118/129 (91.5%) point clouds with the more straightforward color assignment. Percentile is important in increasing the robustness of the pipeline on a wider range of point clouds.

$$color\ value\ = \frac{|z\ value\ of\ point - minimum\ z\ value\ in\ point\ cloud|}{z\ axis\ range\ in\ point\ cloud}$$

Formula 2. The alternative color assignment formula that linearly scales the range of z values into [0,1] for shades of gray.

In terms of the accuracy of landmark identification, the percentile and direct approach may have slight variations in landmark coordinates with each face, but the difference is not statistically significant (Table 3). As so, percentile in color generation does not increase accuracy, which we believe is due to MediaPipe's robust performance.

Dimension	Difference in % Accuracy
X	-0.065± 0.037
Y	-0.024 ± 0.039
Z	-0.168 ± 0.209
Overall	-0.086 ± 0.074

Table 3. Match pair confidence interval between accuracy of landmark identification between no percentile - percentile at a 99% confidence level across 118 faces at all 11 landmarks.

4. Limitations

While our method largely results in accurate landmarks, our approach of flattening the 3D data to 2D data is weak in processing areas with a large z-value range within a small (x,y) confinement–identified in a minority of nose region samples. This error occurs during the last step of mapping 2D landmarks back to 3D as the closest (x,y) points may have great variation in z value. A solution previously brought up overcomes this issue by using UV position maps instead of depth maps **[3]**. Perhaps a combination of UV position mapping and our method could result in a more robust pipeline.

As our method utilizes MediaPipe, our pipeline only works on frontally aligned point clouds. We've tried to incorporate iterative rotation of the input point cloud until MediaPipe could identify landmarks. However, this method is extremely computationally expensive and does not result in success due to failures in MediaPipe's black box. MediaPipe sometimes returns landmarks for depth maps that were not of a facial profile. To increase the robustness of our pipeline, the next step would be to incorporate a frontal face alignment algorithm during preprocessing.

5. Conclusion

3D facial landmark detection that relies solely on spatial data is challenging because of the complexity of inconsistent topologies between point clouds. Our 3D facial landmark detection algorithm[1] circumvents this challenge with 2D intermediates: projecting 3D data onto 2D depth maps and applying established 2D facial landmark detectors. Our landmark detection pipeline is accurate with error rates of 1.12% overall and robust–performing well on point clouds with the back of the heads, hair, and shoulders. Our 2D-based approach is also the most dense 3D facial landmark detector to our knowledge, generating 468 landmarks–400 more than current detectors. Our landmark detector is unique in identifying landmarks in topologically smooth regions (e.g. cheek, forehead). Outside facial landmarks, our work demonstrates the wide potential of 2D projection intermediates in manipulating complex 3D data.

Code

Our code is publicly available at github.com/cse15-sip-interns/3d_face_landmark_identification

Acknowledgements:

This research was made possible by the Science Internship Program (SIP) under the mentorship of Jiahao Luo at UCSC.

References:

[1] Li, S. (2019). 3Dface_landmark. GitHub. https://github.com/LightOfMonet/3Dface_landmark
[2] Wang, Y., Cao, M., Fan, Z., & Peng, S. (2022). Learning to Detect 3D Facial Landmarks via Heatmap Regression with Graph Convolutional Network. Proceedings of the . . . AAAI Conference on Artificial Intelligence, 36(3), 2595–2603. [DOI]
[3] Zhang, J., Gao, K., Fu, K., & Peng, C. (2020). Deep 3D Facial Landmark Localization on position maps. Neurocomputing, 406, 89–98. [DOI]
[4] H. Dai, N. E. Pears, W. Smith and C. Duncan Statistical Modeling of Craniofacial Shape and Texture. International Journal of Computer Vision (2019) [DOI][BibTeX]
[5] S. Katz and A. Tal, "On the Visibility of Point Clouds," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1350-1358, doi: 10.1109/ICCV.2015.159.
[6] The Dlib library official website, link: www.dlib.net
[7] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In IEEE Conference on Computer Vision and Pattern Recognition, 2005.
[8] MediaPipe on GitHub https://google.github.io/MediaPipe
[9] H. Mohammadzade and D. Hatzinakos, "Iterative Closest Normal Point for 3D Face Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 2, pp. 381-397, Feb. 2013, doi: 10.1109/TPAMI.2012.107.