3D Partial Face Recognition under Mask Occlusion

Alex Li
yuhuili@stanford.edu

Abstract

The presence of occlusion has been a long-standing challenge for 3D Face Recognition. This paper investigates the effect of mask occlusion on the KMTS-TPWCRC 3D Partial Face Recognition framework proposed by Lei et al and evaluates the accuracy in a face identification task. To address the lack of access to 3D face databases, Basel Face Model is used to generate a set of face meshes, followed by ray tracing with and without the mask to simulate face scans, forming a dedicated dataset for this work. The results show a significant decrease in the overall identification rate in the presence of an N95 mask, which impacts the quality of the initial KMTS descriptor extraction with its large coverage of the face and undermines the locality constraints of the data representation in the classification stages.

1. Introduction

Many smartphones today have built-in 3D Face Recognition [2], but the presence of occlusion often poses challenges for these systems and forces the user to fallback on an alternative authentication method such as manually entering a password. Use cases such as unlocking device and authorizing large payments certainly require a high level of security [13], but some use cases such as approving low-value transactions for transits have more tolerance and should succeed even when some parts of the face are occluded. Furthermore, in the current state of COVID-19 pandemic, an occlusion-tolerant 3D Face Recognition system can reduce the likelihood of shoulder surfing attack [9] when a non-critical authentication action is performed.

In this project, we will explore the effect of mask occlusion on the identification rate of a 3D Partial Face Recognition method proposed by Lei et al. as the KMTS-TPWCRC framework [10]. Its Keypoint-based Multiple Triangle Statistics (KMTS) operate on local features that utilize interesting information from the neighborhood of the detected keypoints, and unlike global features, they can tolerate missing or corrupted data and so are less sensitive to occlusions. Its classification is performed by a Two-Phase Weighted Collaborative Representation Classification (TP-WCRC), which only requires a single sample per subject. In the original paper, the authors evaluated their method on datasets whose face scans only had small regions occluded or corrupted, but masks bring a unique challenge as they cover a large area of face – the nose in particular. Due to the lack of existing datasets, we will synthesize faces from a 3D Morphable Face Model [7] for evaluation.

2. Related Work

Based on the feature extraction technique, 3D Face Recognition can be categorized into conventional methods and deep learning-based methods. Conventional methods can be further divided into ones that utilize local features such as a LBP and SVM-based method proposed by Shi et al. [20], ones that process the entire face holistically such as a 3DLBP and KDA method proposed by Peter et al. [17], and hybrid methods such as the one proposed by Bagchi et al. [3] that utilizes surface normals.

To handle partial face or occlusions, some approaches would attempt to restore the missing parts and remove the occlusion before performing recognition. Colombo et al.'s [5] scheme analyzes the eigenface approximation of a face to detect occlusion, then removes these regions and restores the face via the Gappy PCA algorithm. However, such additional data processing phase can be time consuming. Alternatively, Sparse Representation, also known as Sparse Approximation, are also an effective approach for handling occlusion in both 2D and 3D, as shown in Wright et al.'s work [22] and Guo et al.'s work [8], but these would require large number of training images per subject and would not usually be feasible in real world applications. Lei et al.'s contribution in their KMTS-TPWCRC approach is its effective recognition using only a single sample per subject, its tolerance with both holistic and partial face data, and its more efficient computation using Collaborative Representation.

Since most of these prior work were evaluated on well-known datasets (see Section 3), their testing may be biased towards certain types of occlusions. Specifically, these occlusions are usually small and so are not representative of masked faces. Therefore, in this paper, we implement and evaluate the performance of Lei et al's KMTS-TPWCRC

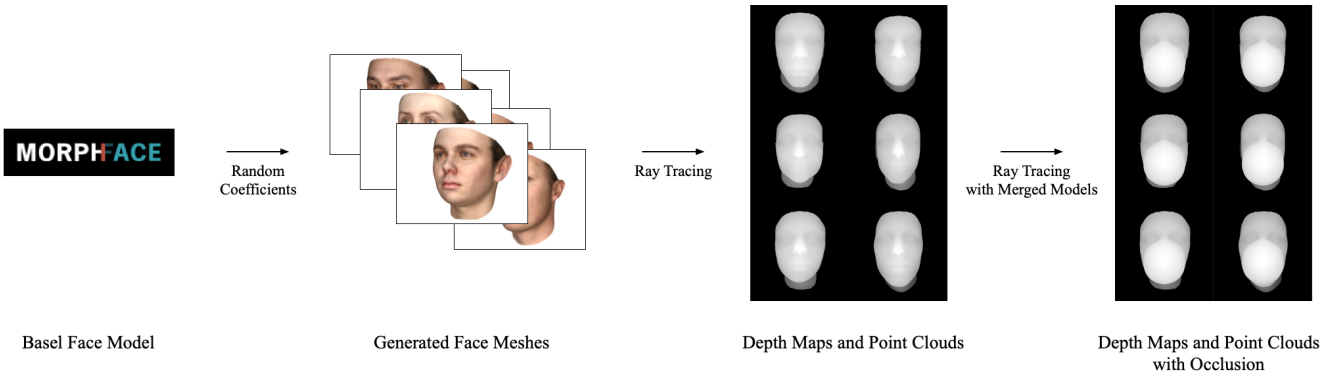


Figure 1: Dataset creation pipeline

face recognition method in the presence of masks.

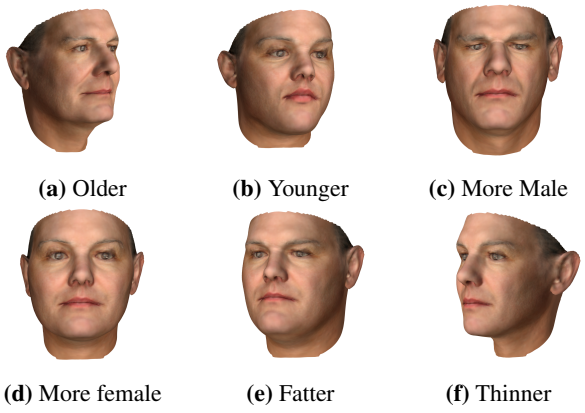
3. Dataset

In 3D Face Recognition research, the acquisition of 3D face data is a considerable challenge as they are far less readily available as 2D face images, and this is primarily due to the need of specialized stereo vision hardware. These systems could be divided into active acquisition and passive acquisition [23] such that the former actively emit a laser or a structured light [11] to determine the shape of the object and the latter typically use 2D images taken from different angles [19] to perform triangulation. In the recent years, there have been an uptrend of available 3D measuring systems in the consumer markets that can perform 3D object acquisition in great precision and at high speed [4], but 3D face data are generally considered sensitive Personally Identifiable Information (PII) stored at some dedicated secure subsystem and not exportable off the devices.

Nonetheless, 3D face datasets are still essential to the advancement of 3D Face Recognition. Many research institutions have established their own databases. Some prominent ones are Bosphorus [18] and UMBDB [6], and they contain not just facial scans of human subjects, but scans with different types of occlusions, notably suitable for this project. However, these datasets are extremely restrictive and are only available to formal research groups.

3.1. Basel Face Model

To overcome the lack of accessible datasets and the difficulty in gathering real faces with masks, we have to craft our own dataset, and so we turn to a proxy. A 3D Morphable Face Model is a power tool that can generate synthetic faces from random model coefficients. Basel Face Model [15] is constructed from 3D scans of 100 male and 100 female faces, and the results of different parameters that tune facial characteristics are shown in Figure 2. We use it to create 64 random 3D face models with different facial characteristics as the base dataset. Each face is a textured triangle mesh, and it is converted into other representations.



(a) Older **(b)** Younger **(c)** More Male

(d) More female **(e)** Fatter **(f)** Thinner

Figure 2: 3D faces generated from the Basel Face Model

3.2. Occlusion Placement

Once the base dataset containing all faces is constructed, we synthesize occlusions by placing additional objects on the faces. One type of object is masks, but cloth and surgical masks are not rigid, and fitting them on faces is a challenging task due to cloth deformation. Thus, we restrict to only using a rigid N95 mask model, which is to be merged with each face to create a masked version of the base dataset as shown in Figure 3.

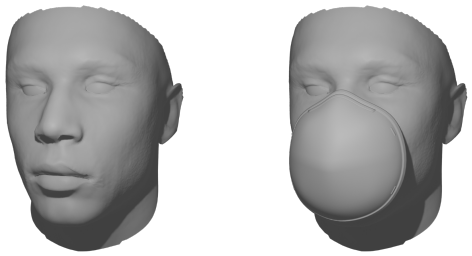


Figure 3: Face with and without a N95 mask as occlusion

3.3. Simulated Face Scans via Ray Tracing

Since each face is a triangle mesh up to this point, we can directly convert them into point clouds, but we want the simulated face scans to not deviate too far from real-life active stereo reconstructions. Therefore, we ray trace the models, and this is advantageous because ray tracing simulates the fixed resolution of a structured light source and the region of the mesh not hit by the rays are not included in the final point cloud. Then by using the known camera matrix, we can compute the 3D coordinates of all points from the depth map to create the point cloud.

More specifically, we construct a ray tracer that emits 1000×1000 rays with a 60° field of view from some camera location. For now, the camera is directly in front of the face. For each ray R described by $R = O + kD, \quad \forall k \in \mathbb{R}^+$ where $O \in \mathbb{R}^3$ denotes the origin and $D \in \mathbb{R}^3$ denotes the direction, it is intersected with every triangle $T = \{v_0, v_1, v_2\}$ of the face mesh. If there is at least one ray-triangle intersection, the 3D point at $O + k_0 D$ for the smallest $k = k_0$ is added to the point cloud.

When this bare-bone single threaded ray tracer is put into work, each "scan" would require approximately 8 minutes on a 2.3GHz Intel i9 processor because each mesh has a large number of triangles (50k vertices and 100k faces), and our image resolution is quite high. To improve, we utilize an open source octree data structure implementation [16] and write the algorithms to build and traverse the octree to accelerate this stage of the dataset creation. This results in a $120\times$ speedup where each face scan would only take 4 seconds, vastly cutting down the wait time when the face meshes or scans need to be regenerated due to minor tweaks in the project details. Two sample depth maps from this simulated face scan process are shown in Figure 4.

In addition to scans from the front, we also generate a set where each face rotated 5 and 15 degrees around the vertical axis. All angles are scanned with and without masks.

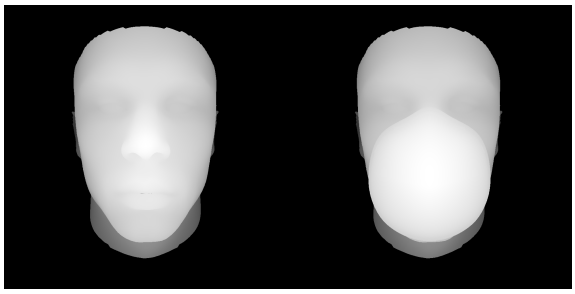


Figure 4: Ray traced depth map of a face

3.4. Summary

In short, we start out with 64 face generated using random coefficients from the Basel Face Model. Each face mesh is then combined with an occlusion object, a mask, to form a new mesh. We then perform simulated face scans, from front, from a 5° angle, and from a 15° angle, using a ray tracer accelerated by octree to produce the depth maps and point clouds. The depth maps and the associated point clouds serve as the final 3D Face dataset for the project. This pipeline is shown in Figure 1.

4. Method

For this project, we use the local feature-based conventional method that utilize traditional algorithms to extract features from Lei et al. [10]. Local feature-methods are generally more robust with missing or unexpected data due to occlusion. More specifically, we evaluate the KMTS-TPWCRC framework's performance on mask-occluded faces.

4.1. Keypoint Detection

Keypoint-based methods take two steps to obtain the representation of a face: keypoint detection and feature descriptor construction [21]. Keypoints are salient points that contain interesting geometric information, and we detect them using the algorithm proposed by Mian et al. in [12] as follows.

Given the depth map of a face such that each pixel corresponds to a 3D point that in whole form a point cloud $F = [x_i \ y_i \ z_i]^T$ where $i = 1, \ldots, n$ and n is the number of points, we sample the depth map at uniform intervals in the two dimensions – approximately 4mm in the world space assuming a weak perspective plane at the face. At each sample point p, we construct a sphere of radius r centered at p to obtain a local patch of points $L_j = [x_j \ y_j \ z_j]^T$ where $j = 1, \ldots, n_p$ and n_p is the number of points cropped by the sphere. The mean m and covariance matrix C of L are defined as

$$m = \frac{1}{n_p} \sum_{j=1}^{n_p} L_j \tag{1}$$

$$\begin{aligned} C &= \mathbb{E}[(L - m)(L - m)^T] \\ &= \mathbb{E}[LL^T] - E(Lm^T) - E(mL^T) + E(mm^T) \\ &= \mathbb{E}[LL^T] - mm^T - mm^T + mm^T \\ &= \frac{1}{n_p} \sum_{j=1}^{n_p} L_j L_j^T - mm^T \end{aligned} \tag{2}$$

The covariance matrix C is square and symmetric, and so it has 3 real eigenvalues and 3 real orthogonal eigenvectors [14]. We can perform Principal Component Analysis (PCA) via eigendecomposition to obtain eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and the corresponding eigenvectors v_1, v_2, v_3 where $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and $Cv_i = \lambda_i v_i$.

Let A be the matrix such that row i is v_i. We can apply Hotelling transform to align L with its principal axis [1] by

$$L'_j = A(L_j - m) \quad j = 1, \ldots, n_p \tag{3}$$

where L'_j is in a new coordinate system with origin at the centroid of the points in L and axes in the direction of the eigenvectors of C.

Let L'_{jx} and L'_{jy} be defined as the x and y components of L'_j. Then define

$$\delta = [\max_j(L'_{jx}) - \min_j(L'_{jx})] - [\max_j(L'_{jy}) - \min_j(L'_{jy})] \tag{4}$$

which measures the difference between the first and second principal axes of L. According to [12], δ is non-negative and proportional to the variation in the depth of L', or 0 if it is planar or spherical. Therefore, we set a threshold $t = 4$ such that if $\delta \geq t$, p is taken as a keypoint.

4.2. Feature Descriptors

Each keypoint p now needs to be represented by some feature descriptor, and we use the Keypoint-based Multiple Triangle Statistics (KMTS).

Let T be the triangle formed by the keypoint p and two other points q_a, q_b in L. Four types of geometric information can be extracted as follows,

- A: the angle formed by line segments p, q_a and p, q_b.
- C: the radius of the circle circumscribing T.
- D: the Euclidean distance between q_a and q_b.
- N: the angle between the z-axis and line that passes through q_a and q_b.

If L has n points except the keypoint itself, $\binom{n}{2} = \frac{1}{2}n(n-1) = O(n^2)$ triangles are generated based on the KMTS descriptor scheme.

After normalizing and concatenating the histogram of m bins to form a histogram-based descriptor, each patch L is represented by the KMTS descriptor with length $4m$. We follow $m = 30$ from the original paper. An example of the four histogram statistics from a detected keypoint is shown in Figure 5.

4.3. First-phase Classification

The original paper proposed a two-phase scheme. The first stage computes the class-based probability estimation

$$p = \mathrm{p}(Y|D) = [\mathrm{p}(Y|D_1), \mathrm{p}(Y|D_2), \ldots, \mathrm{p}(Y|D_C)]^T \tag{5}$$

where $Y = [y_1, y_2, \ldots, y_n]$ contains the KMTS descriptors of an input face, and D_c contains the KMTS descriptors of a known face in the gallery. In other words, p is a vector with C entries for a total of C known individuals, and each value

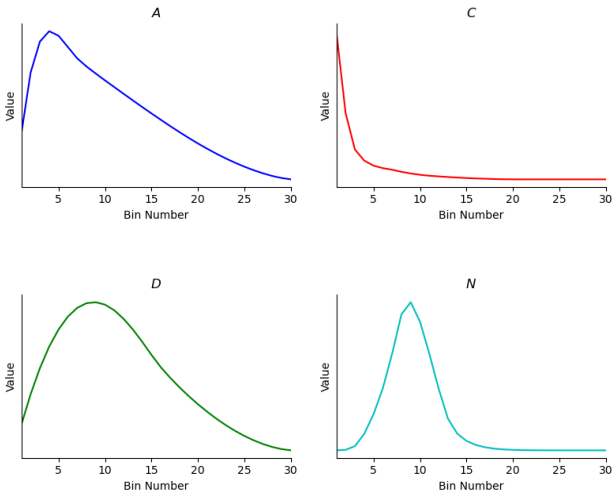


Figure 5: The four histogram-based descriptor for a detected keypoint

represents the prior probability of classifying the input face to a known face.

Let $d_c = \sum_{i=1}^{n} \min_j (\|y_i, (D_c)_j\|)$ be the sum of the minimum Euclidean distance of each descriptor to the set of all descriptors of a known individual c, and let $d_{\min} = \min_i (d_i)$. The probability vector p can be computed as

$$p_c = \mathrm{p}(Y|D_c) = \exp\left(\frac{d_c/d_{\min}}{\epsilon}\right) \tag{6}$$

where ϵ as chosen to be -1 is a constant.

4.4. Second-phase Classification

The second stage classification begins with a K-Nearest Neighbor (KNN) search based on Euclidean distance. For each descriptor y of the input face, we extract a subset of descriptors from cth individual as $\hat{D_c} = [(\hat{D_c})_1, (\hat{D_c})_2, \ldots, (\hat{D_c})_K]$, which are the K descriptors closest to y. Thus we obtain a subset of descriptors of all C individuals as $\hat{D} = [\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_C]$.

The Weighted Collaborative Representation Classification (WCRC) process computes the coding coefficient vector of a descriptor y_i of the input face with the loss function

$$\hat{x_i} = \arg\min_{x_i} \left\| \hat{D}x_i - y_i \right\|^2 + \lambda \left\| p^T x_i \right\|^2 \tag{7}$$

Unfortunately, the original paper seems to have a shape mismatch in $p^T x_i$, so we take an educated guess and repeat the values in p K times to match the length of x.

Without the l_1-norm in the traditional Sparse Representation methods that require expensive optimization of l_1-norms, the solution for x_i here can actually be obtained

analytically. To minimize the right hand side, we take the derivative with respect to x_i

$$\begin{aligned}&\nabla_{x_i}((\hat{D}x_i - y_i)^T(\hat{D}x_i - y_i) + \lambda(p^T x_i)^T(p^T x_i))\\ &= \nabla_{x_i}(x_i^T\hat{D}^T\hat{D}x_i - 2(\hat{D}^T y_i)^T x_i + \lambda x_i^T pp^T x_i)\\ &= 2\hat{D}^T\hat{D}x_i - 2\hat{D}^T y_i + 2\lambda pp^T x_i\end{aligned}$$

Then setting it to zero to obtain

$$(\hat{D}^T\hat{D} + \lambda pp^T)x_i = \hat{D}^T y$$

and so $\hat{x_i}$ has a closed form solution

$$\hat{x_i} = (\hat{D}^T\hat{D} + \lambda pp^T)^{-1}\hat{D}^T y \tag{8}$$

Finally, the class specific representation residual is computed as

$$r_c(Y) = \frac{1}{n}\sum_{i=1}^{n}\frac{\left\|y_i - \hat{D}_c\pi_c(\hat{x_i})\right\|}{\|\pi_c(\hat{x_i})\|} \tag{9}$$

where π_c is the function that selects coefficients corresponding to the cth individual. For the face identification task, the identity $I(Y)$ of an input face Y is computed as

$$I(Y) = \arg\min_c(r_c(Y)) \tag{10}$$

and note the original paper incorrectly stated $\arg\max$.

5. Results

5.1. Keypoint Detection

A sample output of the keypoint detection algorithm described in Section 4.1 is shown in Figure 6, where each red point is a detected keypoint. As the depth map shows, the second face has generally flatter features compared to the first face, resulting in much fewer useful keypoints, especially around the eye region. The third face has much more pronounced features near the brow ridge, resulting in many more valuable keypoints outside the masked area.

There are also interesting observations from the masked depth map. The mask itself would not only make the algorithm classify most of the points sampled within its area as keypoints, but also cause those sampled around the mask to be considered as well. The latter can be explained by the fact that δ in the keypoint detection algorithm is decided based on variation in depth, and the extruded volume due to the mask causes a large variation and pushes over the threshold for those points near the mask boundary.

Similar observations on the correlation between facial features and the detected keypoints as well as the effect of mask occlusion can be made from the 15° scans, as shown in Figure 7. However, note that the keypoints detected are

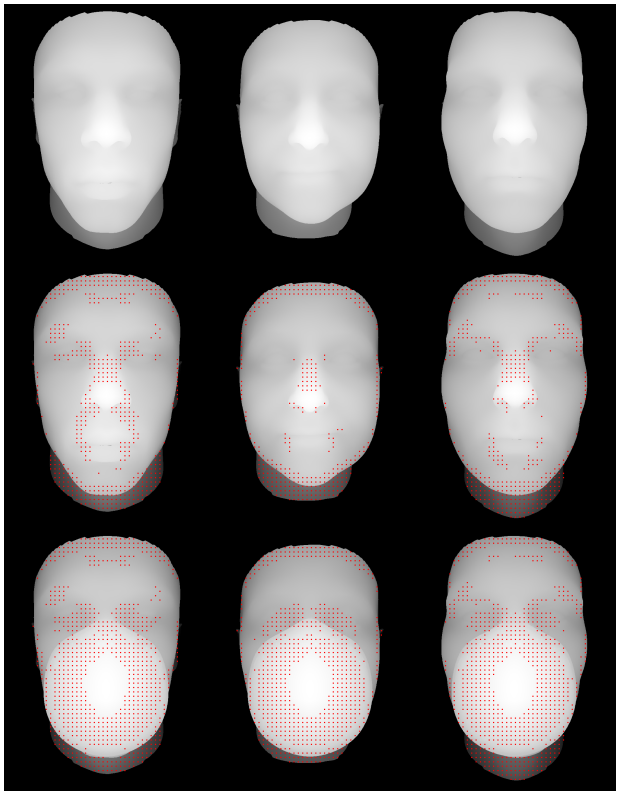


Figure 6: Detected keypoints (front)

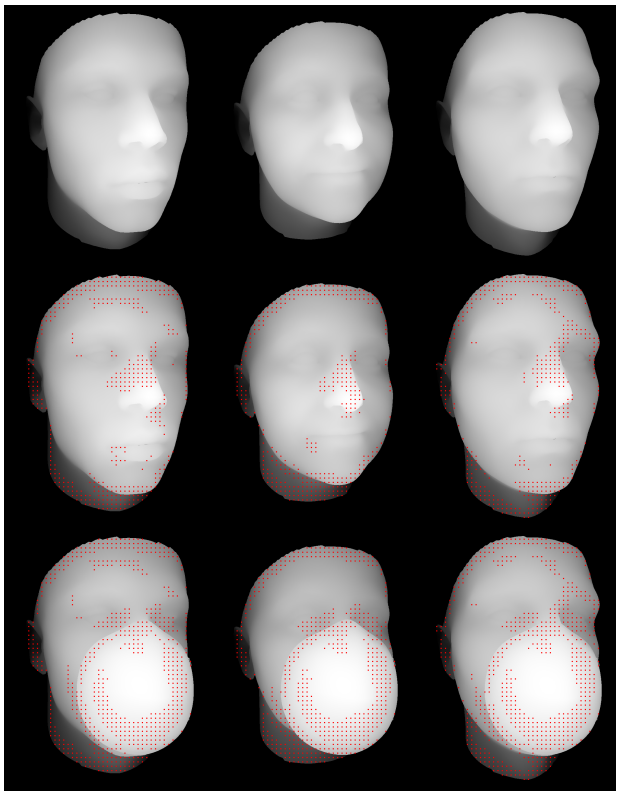


Figure 7: Detected keypoints (15°)

not quite at the same places from these two angles, and the number of keypoints are not identical.

In all scans, there are many keypoints being picked near the boundary of the faces. The ones near the foreheads are likely artifacts because our head meshes have a sharp edge at the top, the ones on the neck are likely due to an abrupt change in depth from the neck to the chin, and the ones on the ears are expected due to the variations in shape.

5.2. Classification

The performance of a Face Identification task, which is a one-to-n identification system, can be measured by a Cumulative Match Curve (CMC). In particular, each individual has only a single scan in the gallery. For each input face, we perform the classification using the TPWCRC framework against a gallery of faces to produce the residuals $r_c(Y)$ for each individual c in the gallery, and rank these from small to large. Then the CMC top-k accuracy of a single input face is given by

$$Acc_k = \begin{cases} 1 & \text{if top-}k\text{ ranked gallery samples} \\ & \text{contain the input face} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Then the CMC curve is constructed by averaging the shifted step function 11 across all input faces.

In this paper, the frontal view without mask scans are used as the gallery, and we evaluate the other variations of our datasets by examining the CMC Curve with varying K, the parameter in the KNN search.

5.2.1 Frontal view with mask

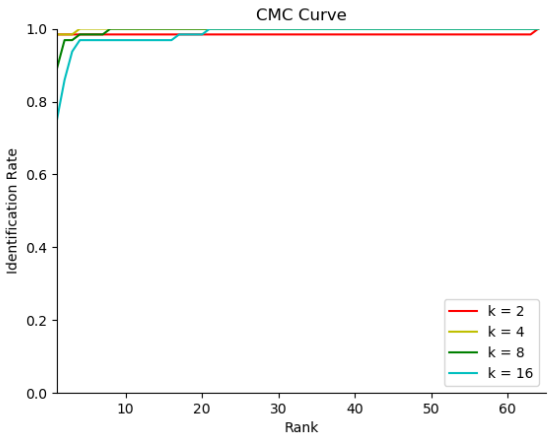


Figure 8: CMC Curve for frontal view with mask

Since the gallery is the frontal view *without* mask, most descriptors from an input face are identical matches to the descriptors of the corresponding face in the gallery. This allows the WCRC coding $\hat{x}$ to be highly sparse, usually resulting in a single coefficient of 1 corresponding to same feature descriptor. Moreover, this greatly minimizes the loss function $\left\|\hat{D}x_i - y_i\right\|^2$ and leads to a small residual, which is the key to a confident identification. As shown in Figure 8, even when only using two closest descriptors per gallery individual, $K = 2$, we could produce near perfect identification rate. Notice as the number of neighbors increases, it doesn't add much besides introducing some noise to the optimization problem.

5.2.2 5° view without mask

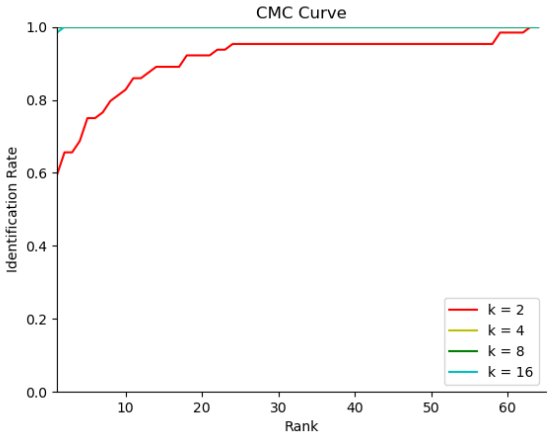


Figure 9: CMC Curve for 5° view without mask

On the other hand, the deficiency of using a small K is immediately evident when face scans are not taken at the same angle as those in the gallery, as shown in Figure 9. $K = 2$ correctly identifies a face only 60% of the time, correctly ranks a face at top 10 only 80% of the time, and its identification rate plateaus out only after approximately Rank-20. But once we increase K to 4 or greater, we obtain near perfect performance, which is a great baseline as it shows that the method is still effective when the detected keypoints are slightly shifted.

5.2.3 5° view with mask

Adding a mask to the 5-degree view drastically worsened the CMC Curve, as shown in Figure 10. Identification is correct less than 50% of the time, and near perfect identification rate only occurs around Rank-25, which is pretty bad given that the dataset size is only 64.

As analyzed earlier in Section 5.1, there are large number of keypoints detected on the mask using Mian et al's method from [12]. The performance degradation due to

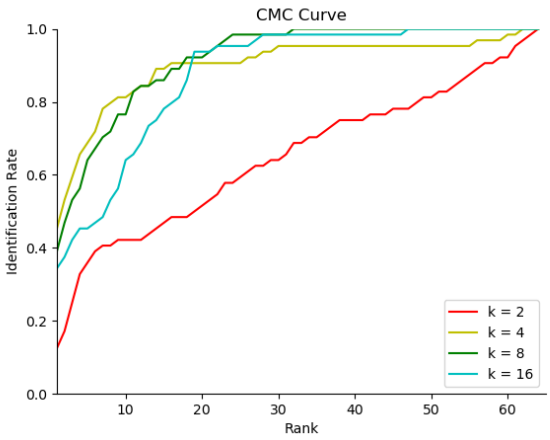


Figure 10: CMC Curve for 5° view with mask

mask is likely contributed by the large number of extra keypoints on the mask itself as well as the decreased number of individual-specific keypoints from the faces, resulting in the data representation becoming global.

5.2.4 15° view without mask

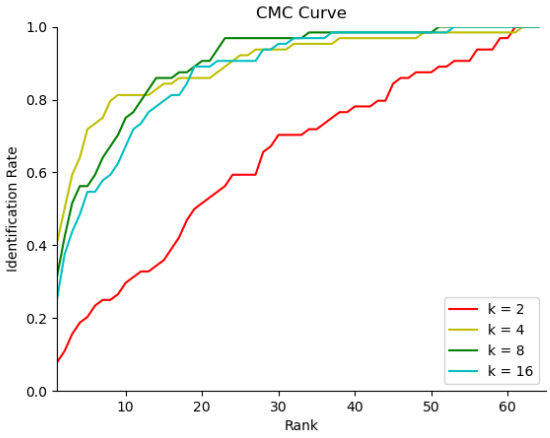


Figure 11: CMC Curve for 15° view without mask

For the 15 degree without mask, we get similar results, as shown in Figure 11. In the case of $K = 2$, it doesn't perform much better than randomly guessing, which would be reflected as a diagonal line.

The best CMC Curve here, $K = 8$, unfortunately plateaus out at near perfect identification rate only at around Rank-25, which is similar to the 5° with mask scans from earlier. In comparison, Lei et al. were able to achieve near perfect accuracy on most well known datasets at lower ranks and with larger gallery sizes. There are a number of possible reasons for this difference beyond the expected impact due to a larger occlusion area. First, some parameters are not explicitly given out in the original paper – ϵ controls the strength of the exponential function in the first-phase classification, and it directly influences the regularization factor in the WCRC coding optimization step. Second, different datasets may lead to varying performance of the algorithm, and in particular, we are only able use a 3D Morphable Face Model instead of real 3D face scans. Additionally, our gallery size is only 64, which may be small and susceptible to greater proportion of noise.

5.2.5 15° view with mask against the gallery

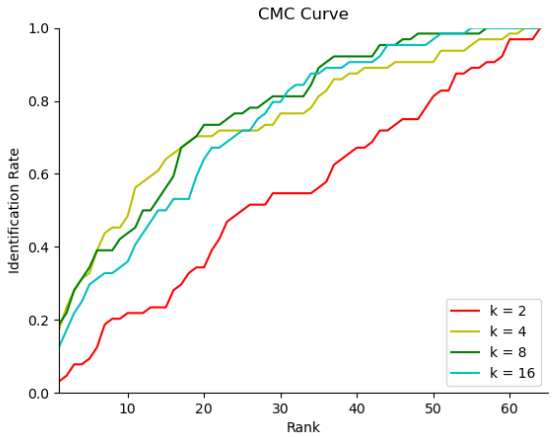


Figure 12: CMC Curve for 15° view with mask

As expected, masked version of the 15° view would not achieve better identification rate than the maskless version, as shown in Figure 12. Notably, the $K = 2$ curve is almost identical to a diagonal line, and the remaining ones are also moving significantly towards the diagonal.

Just as discussed earlier, the changes in keypoint distribution likely have heavily interfered with the class-based prior computation in the first-phase, as well as the WCRC coding process in the second-phase. One possible improvement would be to tune the parameter t to reject more patches from being considered keypoints, but it would reject more points globally, not just in the masked region. Another possible solution is to remove these outliers, but that would require an additional stage to classify the occluded regions and incur more computation.

5.3. Limitations

There are a few notable limitations of the results, primarily due to the synthesized dataset. First, the static 3D

Morphable Face Model cannot generate different expressions for a particular face, thus the scans of an individual are all taken on the identical mesh. This may have inflated the identification rate in some cases. Second, the mask used as the occlusion object is a rigid N95 mask, which is not representative of other types of masks being used in terms of the shape and coverage, leading to potentially more keypoints on the mask and different keypoints around the mask than the average and hinder the accuracy. Additionally, Basel Face Model was trained on from mostly Europeans in their 20s [15], and so the meshes may be biased and not representative of the facial features of people from different regions and people of different ages.

6. Conclusion

In this paper, we have successfully implemented Lei et al's KMTS-TPWCRC method [10] and explored the effect of mask occlusion on the performance of 3D Face Recognition using the Basel Face Model. Indeed, it is an excellent approach that requires only a single training sample per individual, but the large occlusion area of the mask impacts the quality of the extracted KMTS descriptors, breaks the locality constraint of the data representation, and significantly reduces the identification rate of the algorithm.

There are many possible future directions in extending this work to improve the results. For instance, it would be worthwhile to explore how the mask area can be segmented and excluded out before feeding the input face to the framework, because it can very likely boost the accuracy, considering the near-perfect results in the frontal view with mask dataset and the 5° view without mask dataset. At the same time, it would be interesting to try to optimize the parameter t for facial features in the eye region to obtain higher quality input to the two-phase classification.

Moreover, in terms of completeness, the limitations of using the Basel Face Model can be avoided by using a variety of research databases of 3D face scans collected from different demographics, and different types of mask shapes can be experimented in synthesizing the mask-occluded point cloud. Certainly, it would be even better to collect real world dataset with subjects wearing masks. Also, as indicated in the original paper, the class specific representation residual $r_c(Y)$ can be used for a face verification task, in other words, deciding whether the input face *is* a particular individual. This would require tuning an additional parameter for the decision problem, and its performance can be evaluated using the Receiver Operating Characteristic (ROC) curve.

A. Source Code

The source code of this project is available at https://github.com/yuhuili/Stanford-CS231A-Project

References

[1] S. Annadurai and R. Shanmugalakshmi. *Fundamentals of Digital Image Processing*, page 66–70. Pearson, 2010.

[2] Apple. About face id advanced technology, Sep 2021.

[3] P. Bagchi, D. Bhattacharjee, and M. Nasipuri. 3d face recognition using surface normals. *TENCON 2015 - 2015 IEEE Region 10 Conference*, 2015.

[4] A. Breitbarth, T. Schardt, C. Kind, J. Brinkmann, P.-G. Dittrich, and G. Notni. Measurement accuracy and dependence on external influences of the iphone x truedepth sensor. *Photonics and Education in Measurement Science 2019*, 2019.

[5] A. Colombo, C. Cusano, and R. Schettini. Gappy pca classification for occlusion tolerant 3d face detection. *Journal of Mathematical Imaging and Vision*, 35(3):193–207, 2009.

[6] A. Colombo, C. Cusano, and R. Schettini. Umb-db: A database of partially occluded 3d faces. *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011.

[7] B. Egger, W. A. Smith, A. Tewari, S. Wuhrer, M. Zollhoefer, T. Beeler, F. Bernard, T. Bolkart, A. Kortylewski, S. Romdhani, and et al. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics*, 39(5):1–38, 2020.

[8] Z. Guo and Y.-y. Fan. Sparse representation for 3d face recognition. In *2013 Fourth World Congress on Software Engineering*, pages 336–339, 2013.

[9] H. Khan, U. Hengartner, and D. Vogel. Evaluating attack and defense strategies for smartphone pin shoulder surfing. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018.

[10] Y. Lei, Y. Guo, M. Hayat, M. Bennamoun, and X. Zhou. A two-phase weighted collaborative representation for 3d partial face recognition with single sample. *Pattern Recognition*, 52:218–237, Apr 2016.

[11] M. Levoy, J. Ginsberg, J. Shade, D. Fulk, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, and et al. The digital michelangelo project. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, 2000.

[12] A. S. Mian, M. Bennamoun, and R. Owens. Keypoint detection and local feature matching for textured 3d face recognition. *International Journal of Computer Vision*, 79(1):1–12, 2007.

[13] A. Misra, R. K. Dev, and M. M. Rajasekaran. Secured payment system using face recognition technique. *4th International Conference on the Science and Engineering of Materials: ICoSEM2019*, Oct 2020.

[14] M. Negahban. *The mechanical and thermodynamical theory of plasticity*, page 126–126. Taylor & Francis, 2013.

[15] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2009.

[16] S. Perreault. Octree c++ class template.

[17] M. Peter, J.-L. Minoi, and I. H. Hipiny. 3d face recognition using kernel-based pca approach. *Lecture Notes in Electrical Engineering*, page 77–86, 2018.

[18] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3d face analysis. *Lecture Notes in Computer Science*, page 47–56, 2008.

[19] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[20] L. Shi, X. Wang, and Y. Shen. Research on 3d face recognition method based on lbp and svm. *Optik*, 220:165157, 2020.

[21] S. Soltanpour, B. Boufama, and Q. Jonathan Wu. A survey of local feature methods for 3d face recognition. *Pattern Recognition*, 72:391–406, 2017.

[22] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[23] S. Zhou and S. Xiao. 3d face recognition: A survey. *Human-centric Computing and Information Sciences*, 8(1), 2018.