3D FACE RECONSTRUCTION FROM HARD BLENDED EDGES

Yueming Ding\textsuperscript{1} and P.Y. Mok\textsuperscript{1,2}

\textsuperscript{1}The Hong Kong Polytechnic University, Inst. Textiles & Clothing, Hong Kong
\textsuperscript{2}Laboratory for Artificial Intelligence in Design, Hong Kong

ABSTRACT
3D face reconstruction from 2D images is an important research topic because it supports a wide range of applications, such as face recognition, animations, games, and AR/VR systems. 3D face reconstruction from contour features is a challenging task, because traditional edge detection algorithms produce a lot of noises, which are prone to making the reconstruction model trapped in a local optimum or even being degraded. With the development of deep learning, a lot of researcher introduce neural network into contour detection, which can extract relatively clear contours compared with previous methods. In this article, we employ a hard blended face contour feature from neural network and canny edge extractor for face reconstruction. Our method not only improves the 3D face model reconstruction accuracy on synthesis images, but performs more accurately and robustly on in-the-wild images under blurriness, makeup, occlusion and ill-illumination conditions.

KEYWORDS
3D Face Reconstruction, Feature Extraction, Deformable Model

1. INTRODUCTION
3D face reconstruction based on 2D images is a well-explored subject in computer vision, which has been widely applied in animation, virtual try-on systems as well as in face identification (Pavlakos et al., 2019). Traditional face reconstruction approach is pixel-wise analysis-by-synthesis (Blanz & Vetter, 1999), in which the distance between real image and synthetic image from face model is minimized pixel-by-pixel directly. A less direct approach is feature-based reconstruction (Huber et al., 2015), in which the similarity between features extracted from the input and estimated face model is minimized instead of computing pixel-wise difference. In contrast, the feature-based reconstruction is comparatively more robust and computational efficient (Egger et al., 2020).

In the feature-based reconstruction, landmark feature is widely used in initialization stage and important in a lot of state-of-art works. However, there are several deficiencies on landmark features. One of disadvantages is that the locations of landmarks are easily drifted or occluded in large pose conditions. Another disadvantage is that different annotation strategies employ different number of landmarks, which make it hard to unify different dataset in reconstruction. In contrast, edge features represent more detailed geometric information about the face shape and with less limitation. However, the traditional edge detectors, such as canny, are sensitive to illumination, makeup and occlusion. As a result, their generated edge features contain a lot of noises, which are prone to making face reconstruction trapped at local optimum.

Instead of using traditional edge detectors, we employ a neural network to help us extract facial contours for face reconstruction. The key contributions of this paper are as follows:
- We introduce facial contour heatmap features extracted from deep neural networks into the edge-based 3D face reconstruction.
- We propose the hard blended edges which combine edges generated from the features from neural network and from canny edge detector.
- Our 3D face reconstruction method achieve great accuracy and robust on the synthetic facial image dataset and in-the-wild image dataset.
2. RELATED WORK

Landmark and contour are two most common sparse facial features being used for the task of face reconstruction. Landmark features represent the sparse facial geometric information, which have been widely used in face alignment, face recognition as well as in face reconstruction (Blanz & Vetter, 1999; Schönborn et al., 2013). In recent years, the deep neural network achieves significant performance in landmark prediction. For instance, Chen et al. (2014) proposed a cascaded neural network to detect landmarks. Zhu and Ramanan (2012) used a multi-task network to detect landmarks and classify facial attributes. In contrast with deterministic landmarks, heatmap landmarks provide a likelihood representation for landmark locations. The heatmap landmark representation is first used by Newell et al. (2016) in human pose landmark detection, who proposed an hourglass network to extract both local and global features for human pose estimation. Yang et al. (2017) extended hourglass network into face landmark detection.

However, as mentioned before, there exist several limitations on landmark features. Thus, a lot of researches introduce contour or edge information into face reconstruction. Blanz and Vetter (1999) used boundary features in face model fitting, and establish an active shape model. Moghaddam et al. (2003) used multi-view silhouettes to fit the 3DMM model (Paysan et al., 2009). Romdhani and Vetter (2005) formulated edge distance cost as a mixed energy function in optimization. Keller et al. (2007) showed that the cost function of face reconstruction based on contour and edge is not differentiable or continuous because of the occlusion and noise on those features. Fitzgibbon (2003) use the soft boundary on face reconstruction, which is robust but with relatively less accuracy. Thus, because of the challenge of obtaining clean edges or contours, edge or contour based face reconstruction is still an open question (Egger et al., 2020). In one of the state-of-arts, Bas et al. (2016) found that reconstructing face from hard edges performance better than from soft edges.

Based on previous researches, face reconstruction based on edge or contour features is limited by traditional edge detectors. A lot of researchers focus on extracting object contours from images using deep neural networks for segmentation and object detection purposes (Shen et al., 2015; Yu et al., 2017). The LAB model (Wu et al., 2018) focus on extracting face contour heatmaps to assist landmark location on in-the-wild images, which can extract relatively clean face contour heatmaps from in-the-wild images on different conditions. We will introduce this framework to extract facial contour features on the face model reconstruction.

Except landmark and contour features, there are other features used in face reconstruction. Huber et al. (2015) reconstruct face models from images based on scale-invariant feature transform (SIFT) local features. Zhu et al. (2016) proposed a 3D reconstruction solution based on dense features. Shang et al. (2020) used the multi-view dense features on 3D face reconstruction. Other researchers reconstruct the 3D face model from the face shading (Zhu et al., 2015), because shadow and illumination provide clues to the human face shape. However, those dense features will spend more computing source and lead optimization less prone to convergence.

3. METHOD

![Figure 1. 3D Face Reconstruction from Blended Contour](image)

Figure 1. 3D Face Reconstruction from Blended Contour
Figure 1 shows an overview of our method, which includes four stages. In the first stage, we extract the facial contour heatmaps from the LAB neural network which predicts landmark location with the assist of contour information (Wu et al., 2018). In the second stage, we convert the LAB heatmap features, representing the likelihood of facial contours, into the LAB hard edges. In the third stage, we fuse the LAB hard edges with the hard edges detected from canny edge detector to generate the blended hard edges. In the final stage, the parameters of 3D Morphable Model (Paysan et al., 2009) will be predicted based on the blended hard edges.

3.1 Boundary

There are two types of boundary defined in the literature: occluding contours and texture edges (Keller et al., 2007). The occluding contour represents the information related to face shape, which separates the face from background. The texture edges are segmented edges related to the texture information, which are sensitive to make up and illumination. In Figure 2, the red lines represent the occluding contours, and the blue edges represent the texture edges. In this article, we use the term ‘boundaries’ to represent edges extracted from 2D images. The boundaries, including both occluding contours and texture edges, are usually segmented.

![Figure 2. Texture Edges and Occluding Contours](image)

The occluding contour from a 3D model is generated through determining of the variation of the mean vertex normal from adjacent faces as well as the visibility of those vertices. On a 3D face model, if the normals of the adjacent faces are more than 90 degree, the vertices along the connecting edges are on the occluding contours. Then, we use z-buffer, also known as depth buffer, to filter the visibility of the vertices on the occluding contours. Finally, the 2D coordinates of vertices on the occluding contours will be obtained through orthographic projection.

3.2 Blended Hard Edges

In our 3D face reconstruction method, we make use of the LAB framework (Wu et al., 2018) to assist the contour extraction. The LAB framework predicts facial landmark coordinates from input face images. The LAB framework consists of two parts. The first part, shown as part (a) in Figure 1, includes several hourglass networks for generating facial contour heatmaps. Hourglass network structure is good at extracting both local and global information. The second part, shown as part (b) in Figure 1, is for landmark detection, which consists of several convolutional layers and fully connected layers. The inputs to part (b) include the extracted facial heatmaps from part (a) as well as the original input image. In the training process, the differences between ground truth landmarks and the predicted landmarks are back propagated to update the entire network framework. Since Gaussian heatmaps are fuzzy and of high variance, Wu et al. (2018) thus introduced the generative adversarial training strategy to generate relatively clear heatmaps by putting more focuses on boundary areas.

We use face heatmaps generated from the part (a) from the LAB framework (Wu et al., 2018) to generate facial contour features, which are of size $13 \times 64 \times 64$, representing the Gaussian distribution of face contours. According to Bas et al. (2016), soft edges (i.e. Gaussian contours heatmaps) are not as effective as hard edges for 3DMM reconstruction. Thus, we transform the soft contours into hard edges. To do so, we first upsample the features be of the same size as the input images by bilinear interpolation, i.e. a size of $13 \times 384 \times 384$. Next, since the face contour heatmap are non-convex, we segment those contour to convex subsets. Another challenge is that the Gaussian contours heatmaps are anisotropic in the horizontal (row) and vertical (column) directions. For example, the contours of the chin shows larger variances in the horizontal direction compared with those in the vertical direction. To address this, we conduct the same processes in both the row and the column directions. For example, in the row direction, we calculate the gradient for each
pixel in each column and record coordinates from those pixels with continuous non-zero gradients in convex subset in row. Because each segment is convex in the row subset, we can select coordinates with top 3 pixel value easily to generate the coordinate set. Then, we combine coordinate sets from convex subsets both in the row and the column directions to generate a hard LAB edges, as detailed in Figure 3.

![Diagram](image)

Figure 3. Generate hard blended edges from LAB features and canny edges

Moreover, since the LAB framework only considers the face part without other information, such as the shape of ears and neck, we therefore blend texture edges obtained from canny edge detector with the LAB hard edges in our face reconstruction task. After detecting the bounding box of the LAB hard edges, we use it to splice the canny edge map and combine hard canny edges with the LAB facial edges to obtain the blended edges. The details of the hard edges transformation and the blended edge generation are shown in Figure 3.

### 3.3 Optimization

An iterative edge-points optimization is employed to predict the 3DMM shape parameters. In the first step, the pose and shape parameters are initialized through landmark alignment. After initiation, we generate the 2D coordinates of vertices on the occluding contours from the 3D face model by the method described in Section 3.1. Then, the K-Nearest Neighbor algorithm is used to find the corresponding points on the detected edges of the 2D image for each vertex on the occluding contours. Next, the pose and shape parameters are iteratively updated by minimizing the distance between points on edges of the 2D image and on occluding contour from the predicted 3D model. In the final stage of the optimization, the pose parameters is fixed, the shape parameters are optimized until the error is less than the threshold. A hybrid loss function will be employed combined the weighted landmark and edge loss function, and the penalty of shape parameters.

**Landmark Loss function** The landmark loss is used both in the initialization and in the formal training stage. We calculate the squared distance between landmarks predicted from 2D image and the corresponding projected pre-defined vertices on 3D face model.

\[
E_{lmk} = \frac{1}{N} \sum_{n=1}^{N} \left\| v_{i_{lmk}}^{n} - P(\hat{v}_{i_{lmk}}^{n}) \right\|^2
\]

where \( P(\cdot) \) represents orthographic projection operation; \( v_{i_{lmk}}^{n} \) denotes \( N \) landmark points predicted from input image; \( v_{i_{lmk}}^{n} \) are the corresponding predefined points on 3D mesh.

**Contour loss function** The contour loss employs square distance to calculate the distance between the points on the detected facial edges of the 2D image and the points on the projected occluding contours from a 3D face model.

\[
E_{edge} = \frac{1}{N_c} \sum_{j \in C} \left\| v_{j_{cont}}^{n} - P(\hat{v}_{j_{edge}}^{n}) \right\|^2
\]
where \( v_j^{\text{cont}} \) represent vertex on the occluding contour of the predicted mesh model, \( j \in C \) and \( C \) is the number of vertices on the contour; \( v_j^{\text{edge}} \) denotes the point on the input image, which corresponds to the vertex \( j \) on the occluding contours.

**Penalty loss function** The shape parameters of the 3DMM model is expected to follow a normal distribution (Egger et al., 2020). To prevent the face shape from diverging, a regularization term is utilized to normalize the face shape model.

\[
E^p = \sum_{k=1}^{N} \alpha_k^2
\]

where \( \alpha \) is vector of shape parameters, which correspond to \( N \) principal components about the shape of the face model.

**Hybrid loss function** A hybrid loss function is used in the final optimization stage, which combines the weighted landmark loss function, the contour loss function, and the penalty function.

\[
E = \omega_1 E^{\text{lmk}} + \omega_2 E^{\text{edge}} + \omega_p E^p
\]

where \( \omega_1 \) represents the weight of landmark cost function \( E^{\text{lmk}} \), \( \omega_2 \) represents the weight of edge cost function \( E^{\text{edge}} \), and \( \omega_p \) represents the weight of penalty function \( E^p \).

## 4. EXPERIMENTS

### 4.1 Database and Evaluation Metric

**Synthetic image dataset** The synthetic image dataset is generated from the Basel Face Model. As the method in (Bas et al., 2016), we randomly generate 10 groups of shape and texture parameters to obtain 10 ground true face shape and appearance models, which represent 10 identities. Then, synthetic images are rendered from each mesh under 17 degrees \( (0^\circ, \pm 10^\circ, \pm 20^\circ, \pm 30^\circ, \pm 40^\circ, \pm 50^\circ, \pm 60^\circ, \pm 70^\circ, \pm 80^\circ) \) through orthographic projection. Experiments are evaluated on the synthetic image dataset.

**In-the-wild image dataset** WFLW dataset (Wu et al., 2018) named wider facial dataset in-the-wild is employed as the in-the-wild image dataset in our experiment. This database has ground true landmarks with 98 points. Because the dataset contains rich attribute annotations, such as occlusion, illumination, blurriness and makeup, we can evaluate the robustness of our method under different image conditions. However, there is no ground true 3D data in this dataset, so an indirect evaluating metric is used when experiments are conducted on this dataset.

**Evaluation metrics** The performances of different methods on synthetic dataset are evaluated through calculating the mean vertex Euclidean distance between ground true models and predicted models after Procrustes analysis. We employ the ground true landmarks in WFLW as our ground true values. For each estimated face model, we use K Nearest Neighbor search algorithm to find the \( k \) closet corresponding points among the vertices on the occluding contour.

### 4.2 Comparison with Existing Methods

#### 4.2.1 Evaluation on Synthetic Images

Our proposed method is compared with hard canny edges based method of Bas et al. (2016) on synthetic images. Three experiments were conducted under the synthetic image dataset. In the first experiment, we conducted experiment on soft LAB features which were generated from face contour heatmap features extracted from LAB framework. In the second experiment, we transferred those soft LAB features into hard LAB edges by the method described in Section 3.2, then fit the 3D morphable model to those hard LAB edges. In the third experiment, we obtained 3D model from blended edges, by blending the global canny edges with our hard LAB features.
Figure 4. The edge features and 3D face reconstructions on synthetic dataset. From the left to right: inputs, hard canny edges, hard LAB edges, hard blended edges and face reconstructions.

The qualitative examples from synthetic image dataset are shown in Figure 4. As images in the second column shown, even though there exists few ill-illumination and blurriness condition in those synthetic images, canny edge detector still generate a lot of noise. Compared with canny edges, LAB edges contain less noise. Our proposed hard blended edges not only contain relatively less noise but provide more geometric information. The quantitative results are reported in Table 1. We show our results in synthetic dataset over all pose angles of the evaluation metrics. Our blended edges method performs well in most of the pose angles and achieves 1.60 mean distance (mm). This shows that using the blended edges improve the performance of face reconstructions from contour images.

### Table 1. Average Euclidean vertex distance (mm) on synthetic dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Rotation angle</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0° ±10° ±20° ±30° ±40° ±50° ±60° ±70° ±80°</td>
<td></td>
</tr>
<tr>
<td>Hard canny edges</td>
<td>2.36 2.29 2.01 2.06 2.04 2.24 1.80 1.91 2.10 2.09</td>
<td></td>
</tr>
<tr>
<td>Proposed (soft LAB features)</td>
<td>2.21 2.13 2.19 2.56 3.10 3.63 3.11 3.57 3.07 2.84</td>
<td></td>
</tr>
<tr>
<td>Proposed (hard LAB edges)</td>
<td>2.15 1.98 2.05 2.02 2.44 2.83 2.60 2.40 2.34 2.31</td>
<td></td>
</tr>
<tr>
<td>Proposed (hard blended edges)</td>
<td><strong>1.59</strong> <strong>1.64</strong> <strong>1.60</strong> <strong>1.46</strong> <strong>1.50</strong> <strong>1.45</strong> <strong>1.43</strong> <strong>1.68</strong> <strong>2.08</strong> <strong>1.60</strong></td>
<td></td>
</tr>
</tbody>
</table>

#### 4.2.2 Evaluation on In-the-Wild Images

We extend our experiments and evaluations using in-the-wild images. We compare our reconstruction method based on hard blended edges with the methods based on hard canny edges (Bas et al., 2016) using in-the-wild images from WFLW dataset (Wu et al., 2018). The results are shown in Table 2. We use Normalized Mean Error (NME) as metric, which measure the distance between ground true landmarks and their corresponding vertices from the predicted 3D model. We conducted our experiments on images from WFLW dataset with attributes of blurriness, illumination, makeup and occlusion, respectively. The results show that our blended edges method perform more robustly in these conditions.

### Table 2. Normalized Mean Error (NME) on in-the-wild dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Blurriness</th>
<th>Illumination</th>
<th>Makeup</th>
<th>Occlusion</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard canny edges</td>
<td>22.29</td>
<td>22.57</td>
<td>27.37</td>
<td>27.31</td>
<td>24.88</td>
</tr>
<tr>
<td>Proposed (hard LAB edges)</td>
<td>21.89 21.96 21.96</td>
<td>27.02</td>
<td>24.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed (hard blended edges)</td>
<td><strong>21.72</strong> <strong>21.26</strong> <strong>21.16</strong></td>
<td><strong>26.74</strong></td>
<td><strong>24.28</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The qualitative results are shown in Figure 5. We select images with makeup, blurriness, occlusion and ill-illumination annotations, respectively. The canny edges detected from in-the-wild images are shown in the second row in Figure 5. Under the makeup and occlusion conditions, a lot of noises are generated. The fourth row shows the reconstructed face models from Canny edges. Our proposed hard blended edges are shown in the third row, which show less noises and more clear contours. The corresponding 3D face reconstruction using blended edges are shown in the last row, and it can be shown the predicted models have better fitting to the faces in the inputs.
5. CONCLUSION

We present a 3D face reconstruction method based on hard blended edges in this paper. The experiments show that the hard LAB edges represent relatively clean facial boundary information. In the contrast, the canny edges provide global information, such as the shape of head and neck, but contain a lot of noise. The combination of the two kinds of feature improve the reconstruction accuracy on synthesis image dataset significantly. According to the experiment on the in-the-wild images, the canny edges are more sensitive to illumination, makeup and blurriness, which cause the reconstruction easily to be affected. Our proposed hard blended edges can improve the accuracy and robustness of the 3D face reconstruction. In the future, we will explore the integration of other contour detection neural networks, and extend our method to multi-view contours reconstruction.

ACKNOWLEDGEMENT

The work described in this paper was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Grant Numbers 152161/17E & 152112/19E). This work was also partially funded by the Laboratory for Artificial Intelligence in Design (Project Code: RP1-1), Hong Kong Special Administrative Region.
REFERENCES


