DeepSketch2Face: A Deep Learning Based Sketching System for 3D Face And Caricature Modeling

Xiaoguang Han    Chang Gao    Yizhou Yu
The University of Hong Kong
Motivation:
Interactive 3D Face Modeling Remains Challenging

- 3D modeling using existing tools (e.g., Zbrush and Maya) is labor-intensive and time-consuming.
Our Goal:
Sketch-Based 3D Face and Caricature Modeling

• A sketching system for *amateur users* to create a 3D *face or caricature model* with a complicated shape and expression in *minutes*
Related work:
3D Modeling Based on Curve Handles

- Fibermesh: Designing Freeform Surfaces with 3D Curves (Nealen et al. 2007a)
- A Sketch-Based Interface for Detail-Preserving Mesh Editing (Nealen et al. 2007b)
- Sketched lines only provide information for sparse control

[Nealen et al. 2007a] [Nealen et al. 2007b]
Related work:
Deep Learning Based Model Inference from Sketches

- Deep learning helps infer parameters of procedural models for fast urban \((\text{Nishida et al. 2016})\) or man-made object \((\text{Huang et al. 2016})\) modeling.

- The generated model is not exactly the same as the sketched one.
We combine *deep learning based model inference* and *handle-based deformation* together.

- Sketched lines help determine the *depth* of vertices according to complex correlations learned by our *deep regression network*.
- and also serve as 2D position constraints for key feature lines.
User Interface: Overview

- **Three interactive modes** for coarse-to-fine 3D face modeling
User Interface: Initial Sketching Mode

- This mode allows completely unconstrained drawing and erasing.
- The 3D model is updated by deep learning based model inference only without deformation.
User Interface: Follow-up Sketching Mode

- The user can refine the 2D sketch by erasing and redrawing lines.
- This mode integrates *model inference* and *deformation* to generate a 3D model better matching the sketched lines.
User Interface: Gesture-Based 3D Refinement

- A **gesture-based UI** designed for shape refinement using handle-based Laplacian mesh deformation [Sorkine et al. 2004]
Problem: 3D Face Inference from Sketches

• To approximate the non-linear mapping from a 2D sketch to the vertices of a 3D face model
Database Construction: 3D Models

- A face database expanded from FaceWarehouse (Cao et al. 2014)
  - Identity: 4 levels of face exaggeration (Sela et al. 2015)
  - Expression: A subset from FaceWarehouse plus a new set defined by an artist
- 150 identities × 4 levels of exaggeration × 25 expressions (15,000)
Database Construction: 2D Sketches

• Major contours: 2D projections of pre-defined feature lines on 3D models (red and blue)

• Suggestive contours (DeCarlo et al. 2003) for wrinkle lines (dark)

• Augmentation: random noise for viewing parameters, random line removal and deformation

• Hand-drawn sketches from artists
Bilinear Morphable Representation

- Bilinear encoding for 600 (=150 × 4) identities × 25 expressions
- Each face model is represented by an identity vector $u$ (50-d) and an expression vector $v$ (16-d)

\[ V = C \times_2 u^T \times_3 v^T \]  

[Vlasic et al. 2005]
Network Architecture: Overall

- **Pixel-level** and **shape-level** input
- **Two independent branches** of fully-connected layers for u and v
- Bilinear output and a **vertex loss layer**
Network Training

- Stage I: **classifier training** (identity and expression classification)
Network Training

• Stage I: **classifier training** (identity and expression classification)
• Stage II: **u-v regression**
Network Training

- Stage I: **classifier training** *(identity and expression classification)*
- Stage II: **u-v regression**
- Stage III: **fine-tuning** the complete network with the vertex loss layer
Results of Model Inference

- It takes \textbf{50ms} on average on a 3.4GHz Intel processor with a GeForce Titan X GPU.
Results of Model Inference

- It takes **50ms** on average on a 3.4GHz Intel processor with a GeForce Titan X GPU.
Results of Model Inference

- It takes **50ms** on average on a 3.4GHz Intel processor with a GeForce Titan X GPU.
Implementation: Handle-Based Laplacian Deformation

- **Curve handles** are *predefined* on a template mesh and *transferred* to any model inferred by our deep regression network.
- Deformation is performed directly by solving a linear system.
Implementation:

Gestures and Gesture Classification

• 10 different pen gestures are defined and mapped to 10 operations.

• We use a CNN to achieve highly accurate gesture recognition to ensure fluency of interaction.

• Our network achieves 96% accuracy.
Result Gallery
User Studies on the Interface

• **Goal:** Our system *vs.* Deformation-only system (*skip initial sketching*)

• **Deformation-only system:** deep learning based model inference is disabled

• **Stage I Tasks:**
  a) Each participant was given a 2D portrait or caricature as reference, and asked to create a 3D model with a similar shape and expression;
  b) The participant was asked to repeat the same task twice using the above two systems;
  c) A modeling session terminates after 15 minutes or the participant becomes satisfied.
Stage I: User Experience

- 12 amateur users were invited (8 men and 4 women).
- All participants agreed that our system generates better results.
- None of the participants managed to finish early using deformation-only interface while they spent on average 10 minutes to complete the task using our system.

Reference Image  Deformation-only System  Our System
Stage II: Evaluation

- **38 additional subjects** were invited to compare the results from Stage I.
- Each participant was asked to *choose the model that looks more natural and better resembles the sketch*.
- Our results received **82%** votes.

![Graph showing the results of the evaluation](image)
Comparisons on 3D Model Inference

- **PixelShapeCNN**: Our final network
- **PixelCNN**: The network with pixel-level input only
- **ShapeNN**: A regression network takes shape-level input only
- **PixelCNN-wrinkle**: *PixelCNN* trained on the sketch images without wrinkle lines
- **PixelCNNSingle**: A simplified *PixelCNN* which has a single stack of 3 fully connected layers to infer both u and v vectors
- **MMfitting**: Morphable model fitting to minimize the errors between projections of curve handles and the 2D sketches
Comparisons on 3D Model Inference

• Our final network outperforms all other variants

<table>
<thead>
<tr>
<th>network</th>
<th>mean error (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PixelShapeCNN</td>
<td><strong>2.04</strong></td>
</tr>
<tr>
<td>PixelCNN</td>
<td>2.22</td>
</tr>
<tr>
<td>PixelCNN-Wrinkle</td>
<td>2.63</td>
</tr>
<tr>
<td>ShapeNN</td>
<td>3.36</td>
</tr>
<tr>
<td>PixelCNNSingle</td>
<td>7.83</td>
</tr>
<tr>
<td>MMfitting</td>
<td>6.06</td>
</tr>
</tbody>
</table>
Summary and Contributions

• A novel sketching system is proposed for 3D face and caricature modeling.

• A CNN based deep regression network is designed for inferring 3D face models from 2D sketches.

• A significantly expanded face database with diverse identities, expressions and levels of exaggeration is also constructed for training and testing.
Limitations and Future work

- Our system generates unnatural results when given inconsistent exaggeration of facial parts.
- Our system is not able to infer facial details from sketches.
Review and Rethinking

• Review: Why face? (animals, human body, garment etc.)
  - the amount of user interaction / how to build the database
• Review: Why caricature?
• Review: Why frontal view sketch?
• Review: Why three modes?
• Review: Experiences on network training
  - we need a baseline firstly / tuning the data (e.g. expression set tuning)

• Rethinking: Data-driven vs. User interaction
Thank You!

Q&A
Network Architecture: Shape-Level Input (Q&A)

- **2D bilinear encoding**: a vector of \((50+16)\) dimensions
- **High-level** global shape information
Implementation: 
Gestures and Gesture Classification (Q&A)

• 10 different pen gestures are defined and mapped to 10 operations.
• We use a CNN to achieve highly accurate gesture recognition to ensure fluency of interaction.
• 10000 images were collected, 9000 for training and 1000 for testing.
  – Our network achieves 96% accuracy.
Comparison with ZBrush

• A skilled artist (>2 years ZBrush experience) was recruited and asked to create a 3D model in 10 minutes that looks like a reference model.

The reference model created in our system by an amateur user

The model created in Zbrush by a skilled artist
More Results on Model Inference (Q&A)