Face Recognition Based on Geometric Unified Surface Regression

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

In data analysis and general machine learning areas, it has been a long-existing question to fit a complicated, non-regular 3D surface formed by given data points. A traditional method is to fit the surface directly by neural network models to calculate the optimal parameters that minimize the loss function, generally the minimal square function. This approach is proved to be stable and accurate in most of the cases, but it suffers from the need for a large amount of calculation. In this project, we proposed an alternative approach that slice the 3D surface to 2D curves, fit each curve individually, and combine all the results together. We compared this approach to the original method to show its feasibility. Then we aim to apply it in the real world problem of facial surface analysis.

1 Introduction and Background

The motivation for this problem arises from the need for a 3-dimensional measurement of high-precision objects, including manufactured parts and facial features. The traditional and the most intuitive approach is to compute the strict maximum likelihood and fit the surface. But with very precised and elaborated surfaces, once we want to keep all features, it will take a long time to train such a huge data set.

The alternative approach we discussed here, however, will slice the surface into 2D curves and thus reduce the dimension. In this way, it will take much less computational effort and the task could be done parallel with several kernels each focusing on one curve. This kind of Subdivision schemes provide efficient algorithms for the design, representation and processing of arbitrary topology smooth surfaces. Application settings range from industrial design and animation to scientific visualization and simulation.

To explore the application of this approach, we used the data set of 3D faces and fit it with this subdivision algorithm.

2 Methods and Data

2.1 Training Method

To train both the 3D surfaces and the 2D curves, we used both neural network models. We only use two layers with `sigmoid` and the rectified linear unit functions as activation functions. We compute the least square function as our loss function. As the Gradient Descent Optimizer doesn't work very well, we finally chose to use the Adam Optimizer algorithm.

For the facial feature analysis, we used almost the same setup of neural network models in the exploring trials. But this time we used the `relu` function twice instead of the `sigmoid` function. This could be further improved by adding more layers, adjusting parameters and adding drop-out layers.

Submitted to 30th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

2.2 Training Data

For the basic training of surfaces, we created our own data based on a polynomial type function on two variables. Specifically, the function we used is:

$$z = x^2 - y^2 - 2e^{2xy}$$

We plan to extend this to more general, random data sets and surfaces. But for illustration, this is a good and easy to generate example to show our work.

For the facial surface analysis part, we used the open source data from

https://github.com/Juyong/3DFace

This dataset contains CoarseData and FineData augmented from 3131 images of 300-W with the method described in the paper 3DFaceNet: Real-time Dense Face Reconstruction via Synthesizing Photo-realistic Face Images. CoarseData is constructed by varying poses and expressions of the original images. FineData is constructed by transferring details from other images to the original images. We augment each image 30 times for both CoarseData and FineData.

3 Results and Comparison

In this section, we will discuss our results for both directly fitting 3D surface and fitting 2D curves. First, we plot the function $z = x^2 - y^2 - 2e^{2xy}$ in a 3D grid.

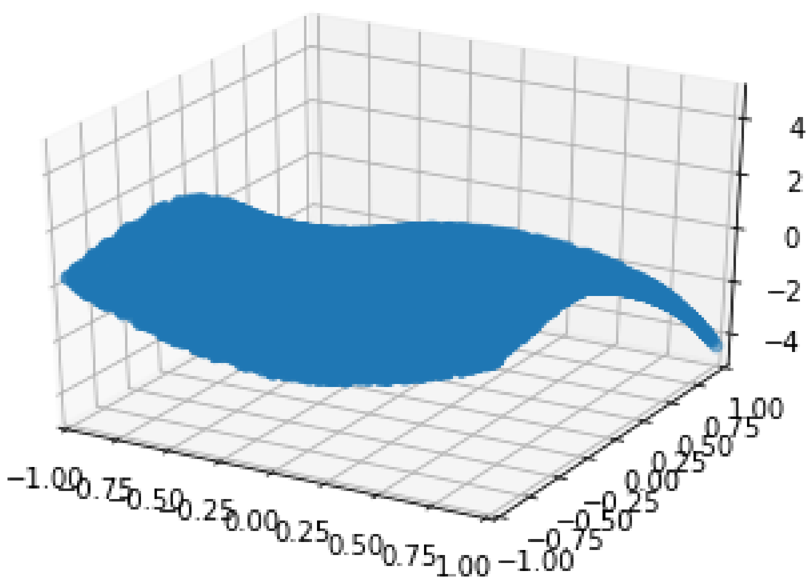


Figure 1: The original 3D surface

3.1 Result of 3D Approach

Using neural network to fit the 3D surface is straightforward. Below is the plot for loss function and the surface drew by the neural network output. From the plot, we can observe that the loss function steadily decreases and the constructed surface is very similar with the original surface.

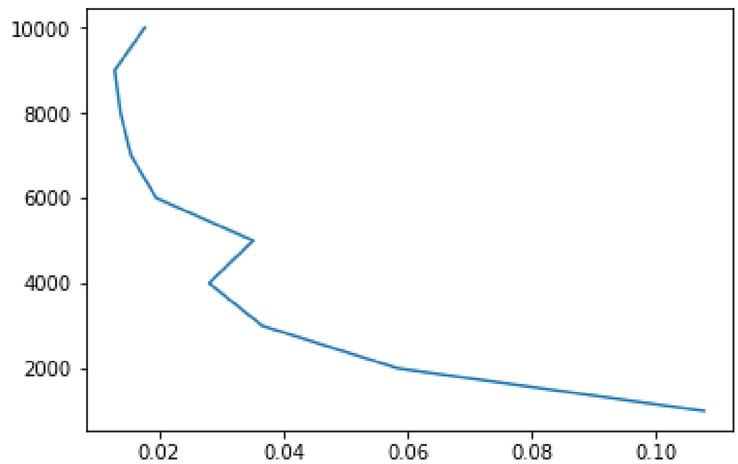


Figure 2: The cost function for 3D fitting

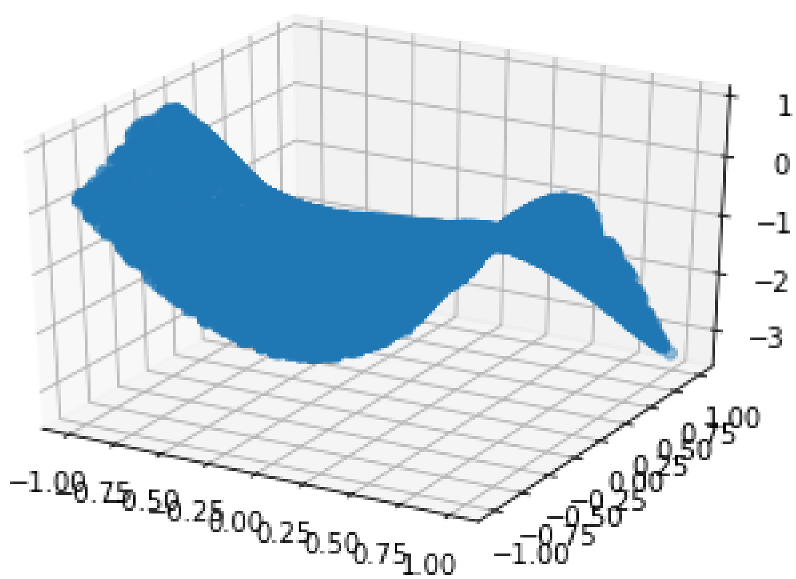


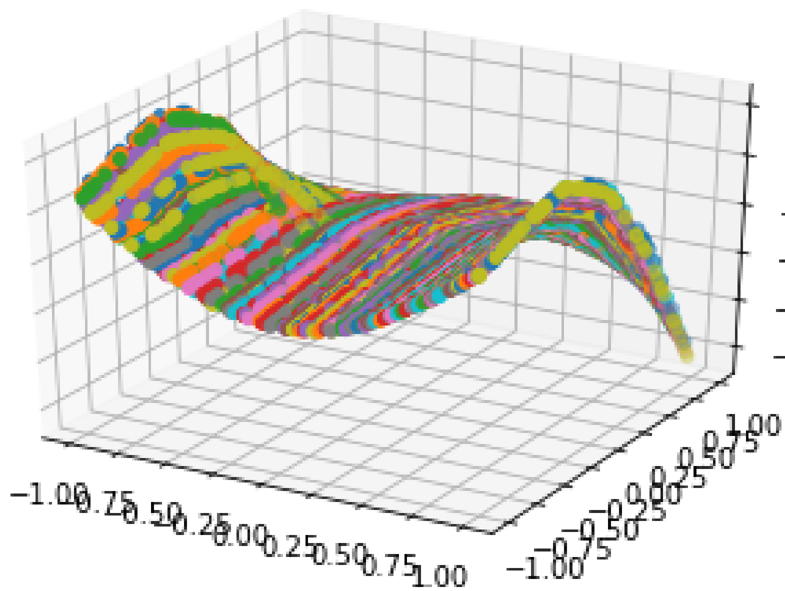
Figure 3: The reconstructed 3D surface with 3D fitting

3.2 Result of 2D Surface Regression Approach

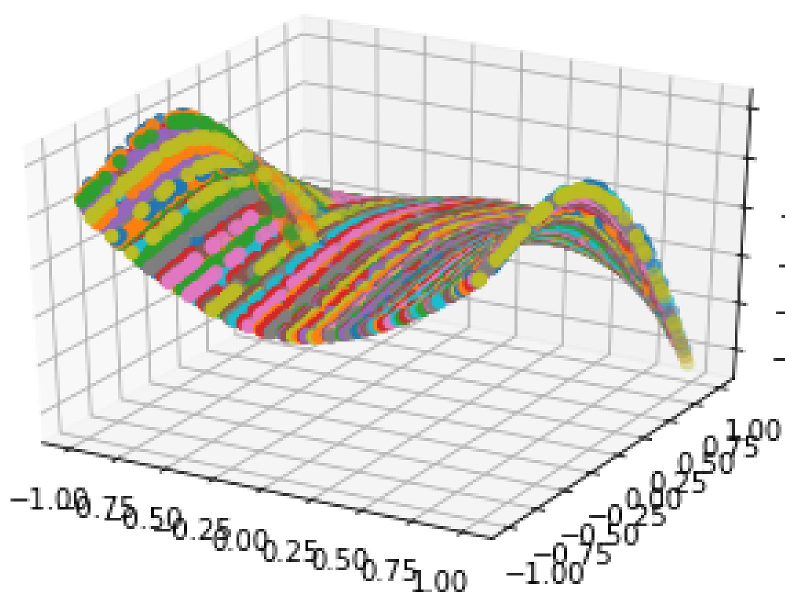
In the 2D surface regression approach, we cut the 3D surface into curves according to either the x or y axis. Then we fit each curve accordingly. The first plot is the reconstructed image with the optimal parameters, and the second one is the original surface shown in slices.

3.3 Comparison

To compare the time elapsed for those two methods, we recorded the time for each training process. The time for 3D direct fitting is 22.870622158050537s, while the time for 2D subdivision fitting needs several minutes to reach the same accuracy. This shows that indeed our subdivision approach is more efficient. Moreover, with many different curves computed at the same time, it can be easily distributed into work for several kernels.



[1]



[2]

Figure 4: 2D Slicing and Reconstruction

4 Application: Facial Surface Analysis

We randomly picked a face from the data set and turned into a parameterized 3D surface.

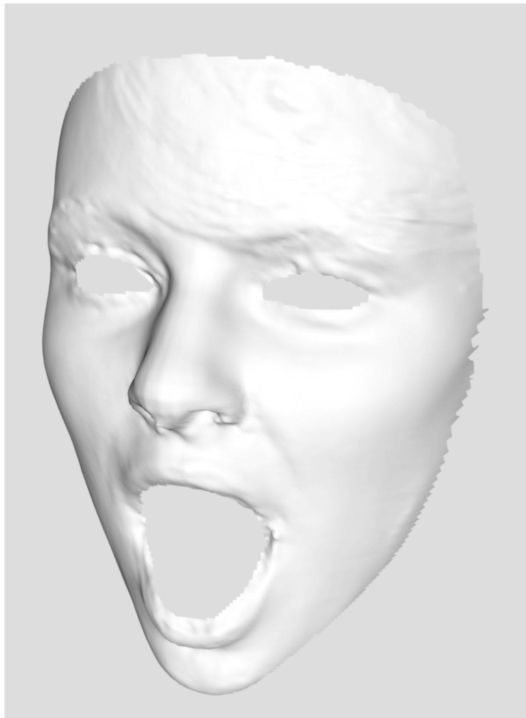


Figure 5: Original 3D model of the face

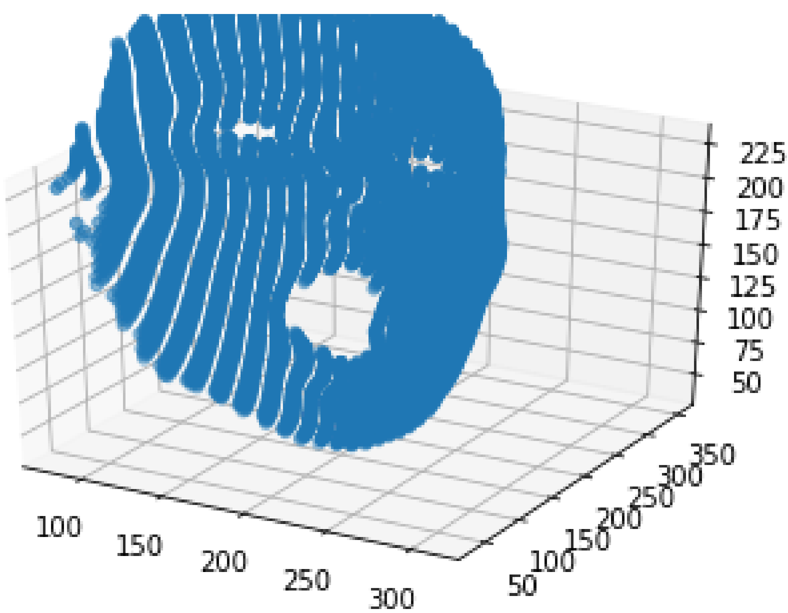


Figure 6: Sliced 3D model of the face

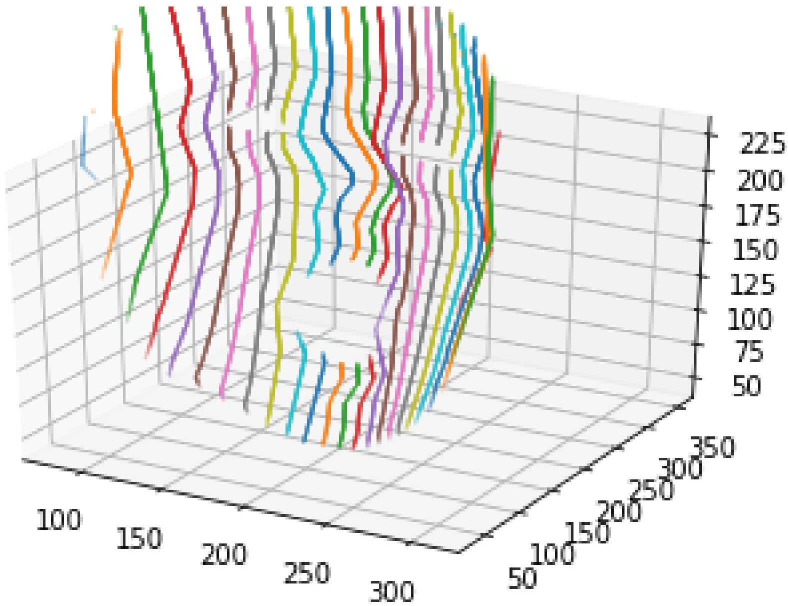


Figure 7: Reconstructed 3D model of the face

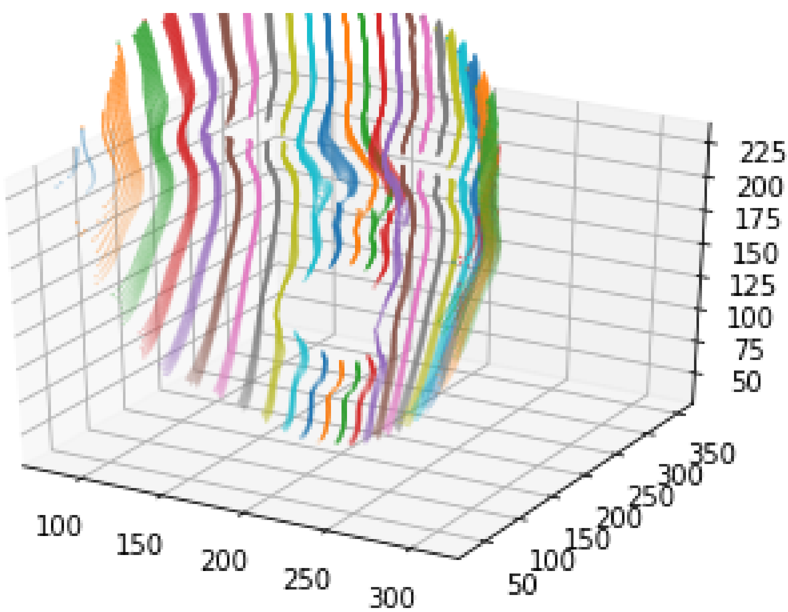


Figure 8: Parametrized 3D model of the face

If we cut data into more slides, we can get a more accurate 3D model of the face. However, it consumes longer time.

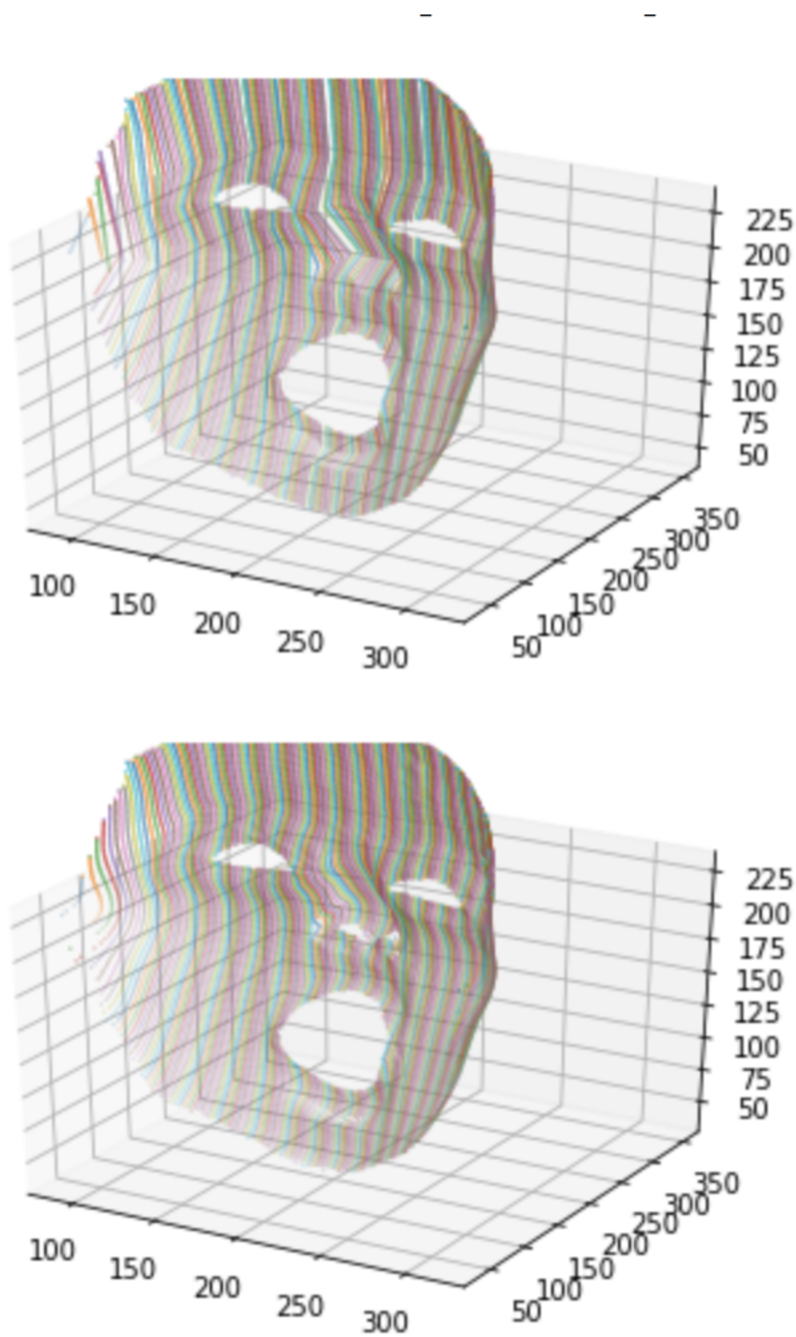


Figure 9: Reconstructed and Parametrized 3D model of the face

5 Future Plans

In the previous sections, we discussed our basic approach, its comparison with the traditional approach on an artificial function-typed surface, and on real world example of face models. The results show that this subdivision approach is promising in making the algorithm more efficient and accurate. Still, we have some further plans that might improve the applicability of this algorithm.

1. Implement the full mesh on the face 3D model including both the x and y axis. This will make our facial fitting algorithm complete.

2. Work with larger data set of faces. Extract the feature from each face and implement a simple face recognition system by matching the features.
3. Work with data set or surfaces with more complex topological structures, for example, those not of a function form. Slice the surface into function type sub-parts, run the algorithms on each part and combine the results together.
4. Generalize to higher dimensional data set.

References

[1] Kanatani, K., & Sugaya, Y. (2010). Unified Computation of Strict Maximum Likelihood for Geometric Fitting. Journal of Mathematical Imaging and Vision, 38, 1-13.

[2] Yudong Guo, Juyong Zhang, Jianfei Cai, Boyi Jiang & Jianmin Zheng (2018). CNN-based Real-time Dense Face Reconstruction with Inverse-rendered Photo-realistic Face Images. IEEE Transactions on Pattern Analysis and Machine Intelligence.