



SCIENTIFIC COMPUTING AND IMAGING INSTITUTE



ShapeWorks Documentation

ShapeWorksPost : Correspondence Based Shape Population Visualization Tool

Shireen Elhabian, Praful Agrawal, Riddhish Bhalodia,
Ross Whitaker

June 29, 2018

Abstract

This document gives instruction for building and usage and some brief functional overview of *shapeworkspost*, which is a post *ShapeWorks* visualization tool.

Contents

1	Introduction	3
2	Build Instructions	3
2.1	Requirements	3
2.2	Build	3
3	Usage	3
3.1	GUI Functions	5

1 Introduction

ShapeWorksPost is a visualization tool for visualizing correspondence based population shape models and their corresponding shape statistics. It takes in the set of particle files produced by *ShapeWorks* (or any other point distribution model generating algorithm) and a single pair of correspondences with associated signed distance transform. *ShapeWorksPost* converts the distance transform into mesh using marching cubes and uses it as a template mesh, or a direct mesh file as template to visualize a real time mesh deformation driven by particle variations using the algorithm presented in [2]. The visualization is based on OpenGL and powered by libIGL [1].

2 Build Instructions

2.1 Requirements

The packages required for a smooth installation and function are given as follows:

- **ITK** : Due to a marching cube bug, we need ITK-4.9 or higher, please make sure your build is without the VTKGlue
- **gcc/g++** : gcc/g++ version should be 4.9 or greater, this is due to the well known regex crash bug with gcc4.8

2.2 Build

The build follows a standard CMake build package, all the libigl components needed for the software are provided along with the source code in form of header only files. In the CMakeList you will need to edit the build path to your ITK. If the system to be built with has default gcc set to 4.8 or lower version, the cmake needs to be called as follows

```
ccmake $PATH -DCMAKE_CXX_COMPILER=g++-5 -DCMAKE_C_COMPILER=gcc-5
```

3 Usage

ShapeWorksPost is really easy to use, it takes in a xml parameter file with the following tags.

1. **point_files** List of paths to the correspondence files.
2. **num_points** Integer flag specifying the number of points on each shape.

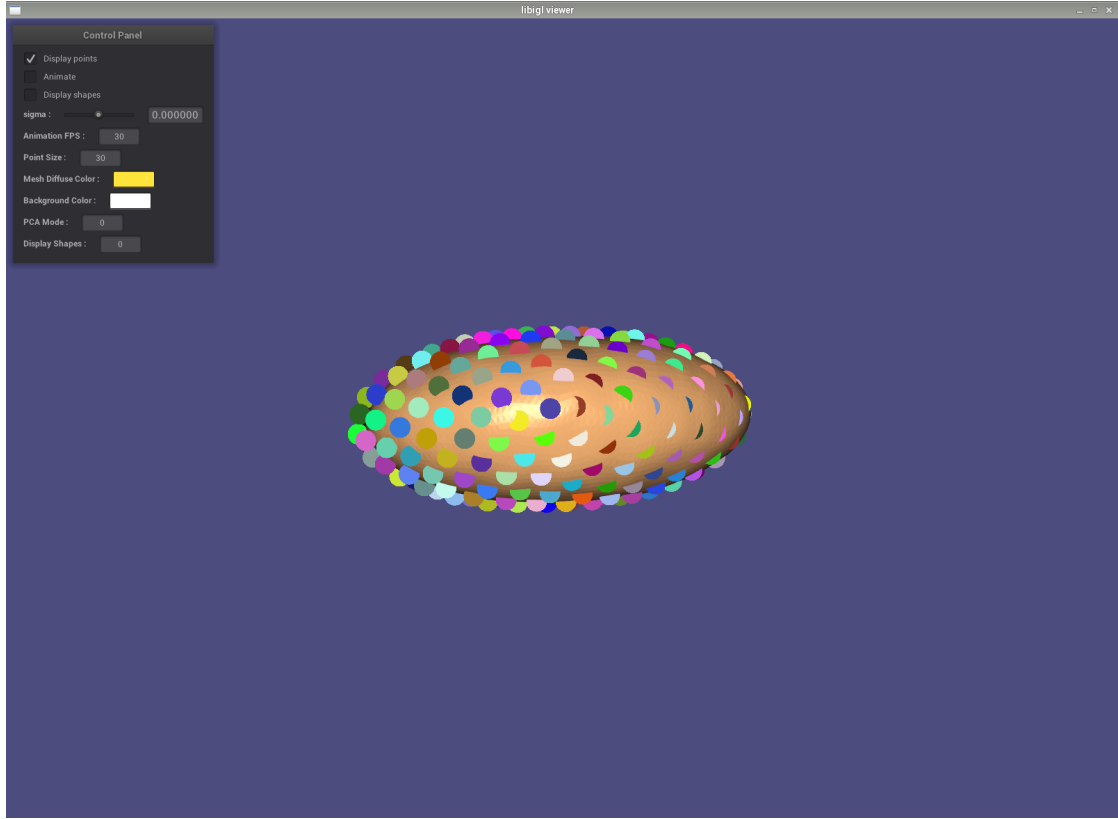


Figure 1: Screenshot of *ShapeWorksPost*

3. **rep_DT** The path to a single signed distance transform from the dataset.
4. **rep_mesh** The path to a single VTK mesh from the dataset.
5. **rep_point** The path to the correspondences corresponding to the rep_DT or rep_mesh.
6. **mesh_decimation_on** (OPTIONAL) Flag for performing mesh decimation to reduce computational overhead for very large template meshes. Recommended for meshes with more than 10000 faces. 0 – Off, 1 – On.
7. **mesh_decimation_percent** (OPTIONAL) A value between 0 and 1 which determines the fraction of the triangles to be retained after the decimation.

There is an example data set and corresponding parameter file in the source code folder.

3.1 GUI Functions

Following is the list of all functionality of the GUI:

1. Animate : This animates the variations along the current PCA mode
2. PCA Mode : Text box to select which pca mode of variations to visualize arranged in decreasing order of the eigenvalue starting from 0
3. Display Shapes : Click the Display shapes checkbox and then start going through the reconstruction of all the shapes in the data.

Figure 1, shows the basic view of the software showing the mean shape and the correspondences.

References

- [1] Alec Jacobson, Daniele Panozzo, et al. libigl: A simple C++ geometry processing library, 2017. <http://libigl.github.io/libigl/>.
- [2] Yu Wang, Alec Jacobson, Jernej Barbic, and Ladislav Kavan. Linear subspace design for real-time shape deformation. *ACM Trans. Graph.*, 34(4), 2015.