Box Cutter Example Application - Overview

This is a simple example application implementing the Box Cutter algorithm, as described within the SIGGRAPH 2018 paper *Box Cutter: Atlas Refinement for Efficient Packing via Void Elimination*.
http://www.cs.ubc.ca/labs/imager/tr/2018/BoxCutter/

The "example_data" directory contains three models:

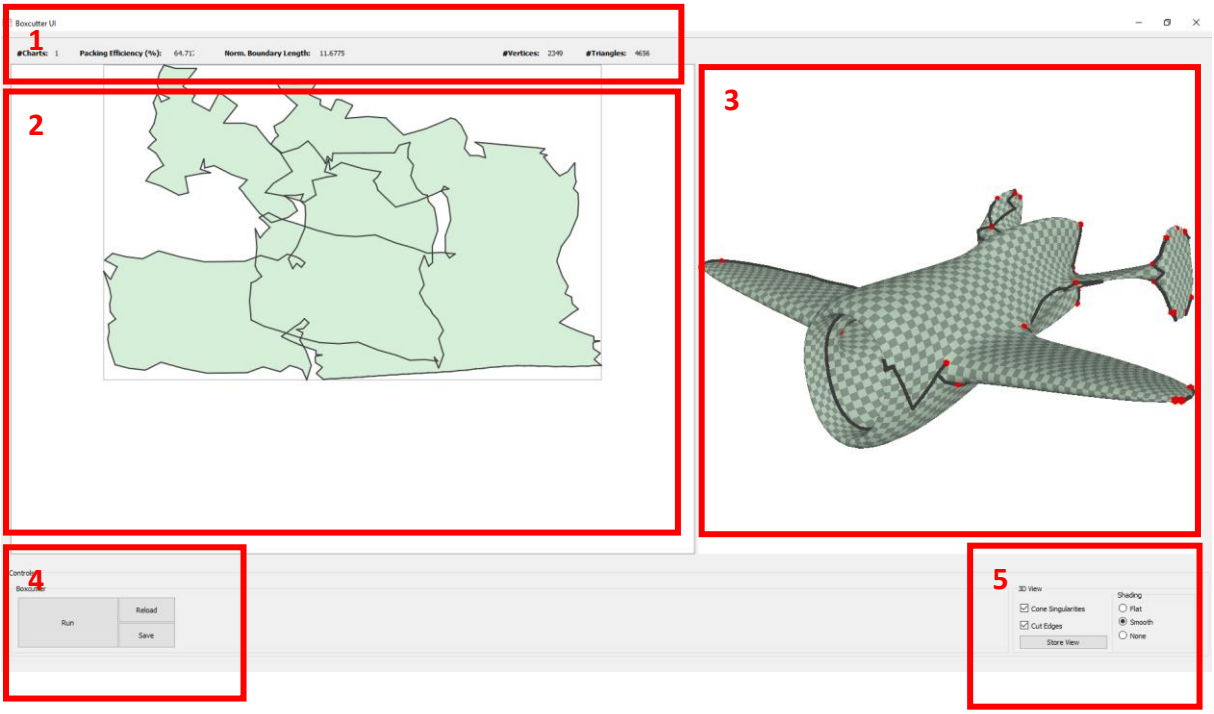
- Girl model, parameterized using *Autocuts: Simultaneous distortion and cut optimization for uv mapping, Poranne et al. 2017*
- 2D Duck model, as used in *Dapper: Decompose-and-pack for 3d printing, Chen et al. 2015*
- Aircraft model with cone singularities, parameterized using *Harmonic global parametrization with rational holonomy, Bright et al. 2017*

More data sets are available as part of the supplemental material of the paper.

Application UI

The application is started from the command line by providing the name of the OBJ file to process.

The UI is pretty simple – here's a screenshot with all parts quickly explained:



1. Statistics Area: Here you can find relevant stats of the model, including packing efficiency and UV boundary length
2. UV Display: Here, the UVs of the model are displayed
3. 3D Display: This part shows the 3D model
4. Control Buttons: These buttons trigger the main functionalities of the program; running the algorithm, re-loading the original model from disk, and saving the result (as "export.XXX", see explanation of filename suffixes below)
5. 3D Configuration Area: Here you can configure the 3D view. To store a viewpoint that should then be automatically used as the initial one (and when exporting 3D renderings in batchmode), simply press "Store View" and a file with the navigation matrix will be saved

("navigation.mtx"). It will be automatically loaded if the application is started in the respective directory.

Usage Example

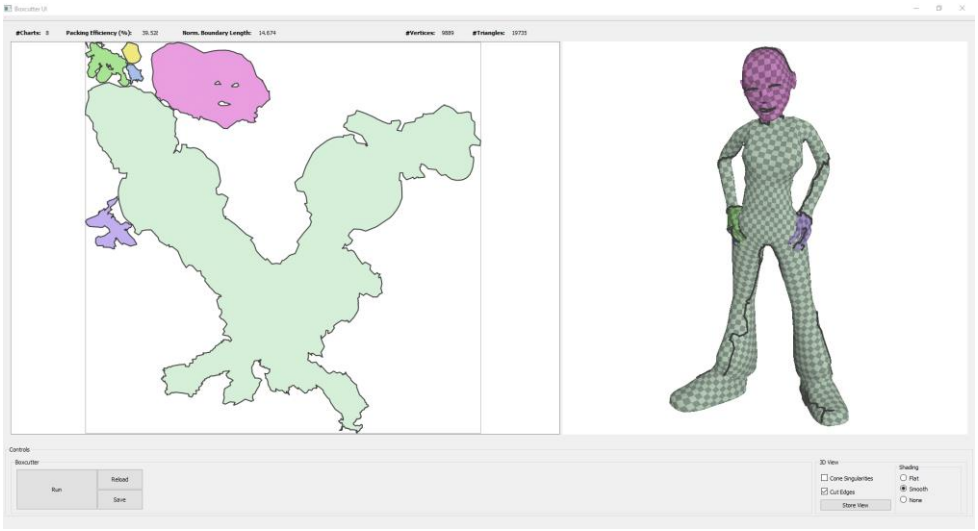
The application is quite minimalistic – it has to be invoked from the command line, there is a single **input parameter**, being the **name of the input OBJ file**, being a parameterized 3D model with or without cone singularities, and two flags:

- --square – when enabled, the algorithm is encouraged to produce square UV atlases
- --batchmode – when enabled, the program is run without a UI

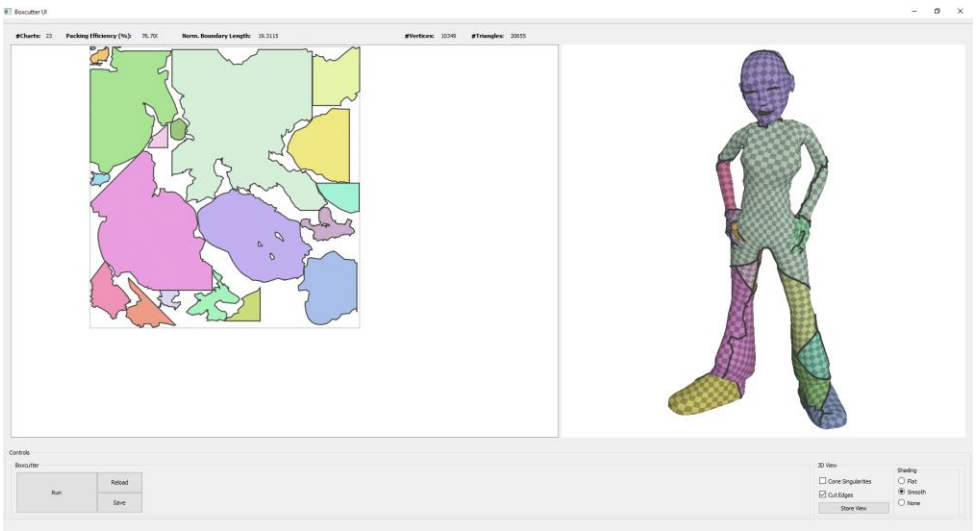
In order to optimize the girl mesh, for example, move to the directory "data\girl" and execute the following command:

```
..\..\bin\boxcutter.exe mesh.obj -square
```

This will open the Boxcutter UI, it should look like this:



You can inspect the model and press "Run" in order to run the algorithm – it will take a bit and you can see the log output in the console window. The result should look like this:



While the application is running through different stages, it will dump output their output to collections of files, which are named **<stage-name><suffix>**.

The **stage names** are (in the order of processing):

- overlap_free – overlaps have been removed
- welded – charts have been welded (for global parameterizations with cone singularities)
- bl_130 – boundary length capped to be max. 130% of the original
- bl_150 - boundary length capped to be max. 150% of the original
- bl_200 - boundary length capped to be max. 200% of the original
- input – the input data set (batchmode only)
- overlap _free_levy – overlaps removed using the method of Levy et al. (batchmode only)

The **filename suffixes** and their respective file content are:

- .json: stats in JSON format
- .obj: 3D model with respective UV coordinates
- .svg: UV layout as SVG
- _3D.png: 3D rendered result

To produce the same result without the UI, you can use the **batchmode** option:

```
..\..\bin\boxcutter.exe mesh.obj --square --batchmode
```

This will directly start the algorithm (as if you would have pressed “Run” in the UI), including a writing of the respective output files of all stages as usual, and then terminate.

If you want to see how the application welds together some seams for models with cone singularities, use the “aircraft” data set as an example.

Happy Box Cutting ☺