Build the Packages for Intrinsic and Extrinsic Shock Computation

Wenhan's Final Presentation of 2017 Summer Internship

A Brief Introduction to Shock Graph

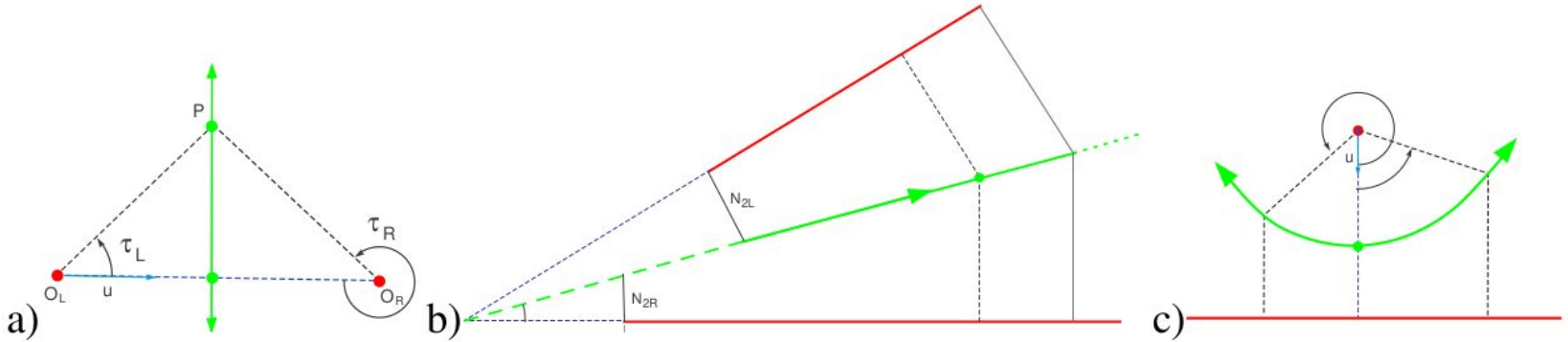
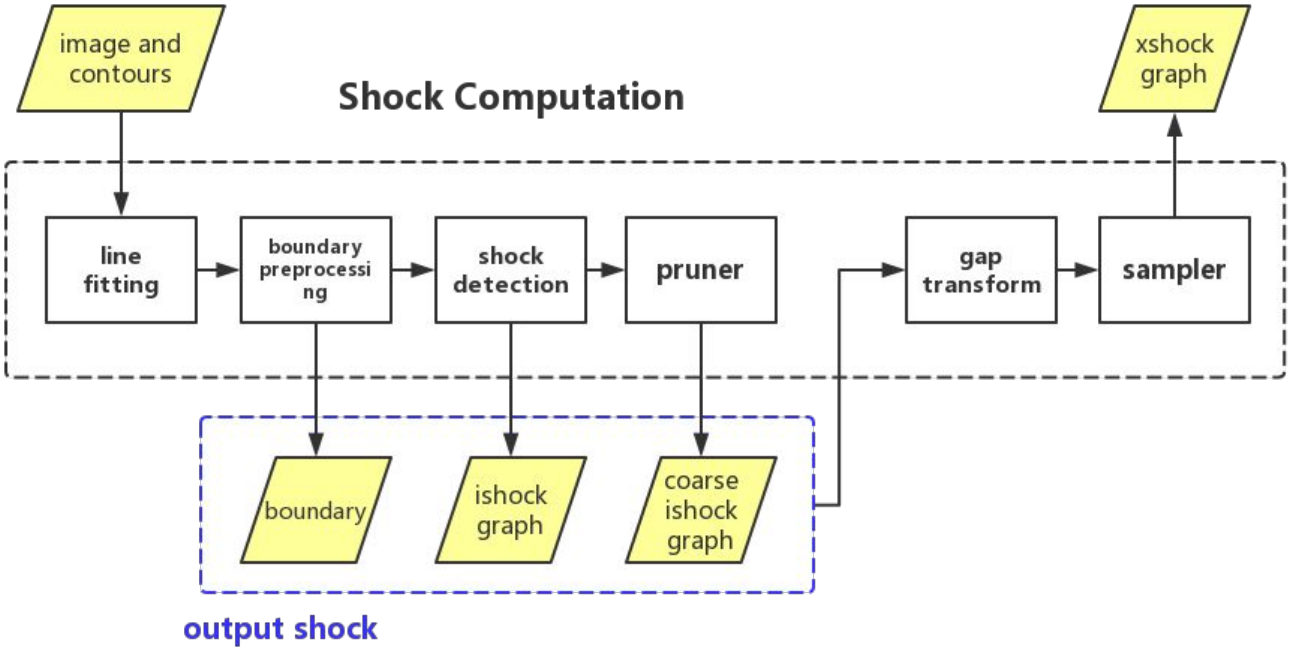


Figure 2.10: Analytic computation of shocks from polylines requires shocks from (a) point-point (b) line-line and (c) point-line pairs. Boundary sources are shown in red, shocks are shown in green, and the rays connecting boundary sources to respective shocks are shown in dashed lines. These computations assume only a pair of sources with no interaction from other sources so all shocks go off to infinity.

Initial Shock Computation Pipeline

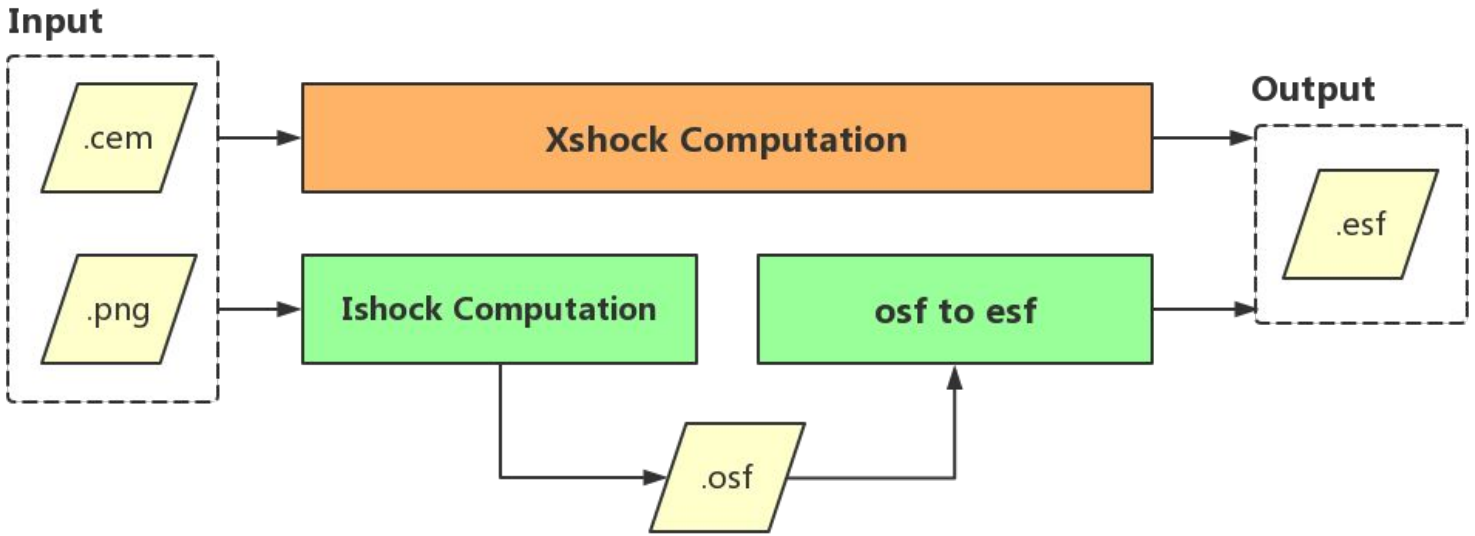


Targets

- Remove the dependencies on other parts of LEMSVXL
- Split the large packages to two parts: **Ishock Computation** and **Following Steps**, without changing the algorithm details
- Design a **text format file** to save ishock graph, which is the output of Ishock Computation part and the input of Xshock Computation part
- Test the packages to make sure the **reliability**

Targets

The final packages and pipelines should be:

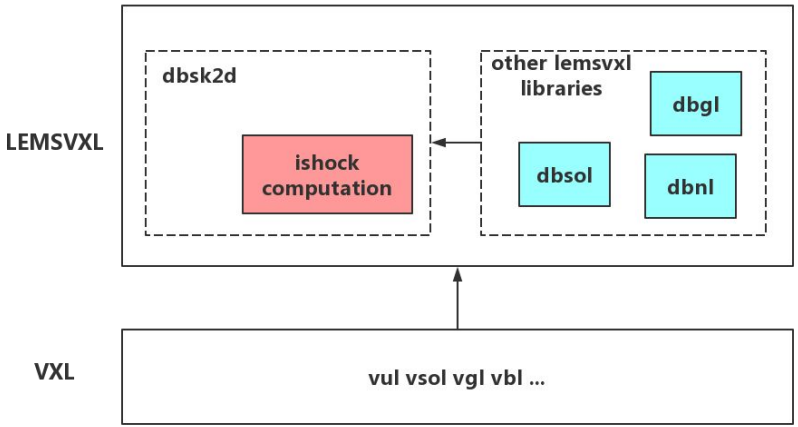


Content

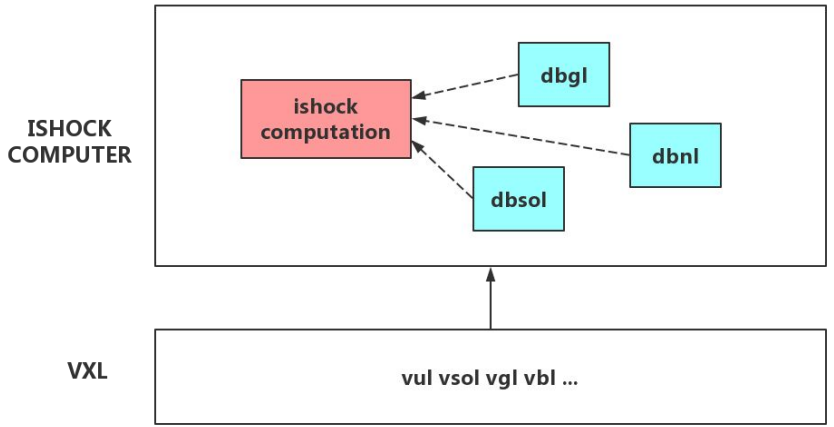
- Package Extraction
- Text File to Save Ishock Graph: .osf file
- Test Results on the Pipelines

Package Extraction

before the extraction



after the extraction



Package Extraction

STEP:

- Use tools (e.g. *doxygen*) to analyze the dependencies
- Remove the dependency on other parts of LEMSVXL
- Remove the dependency on dbsk2d library
- Finally, the **only** dependency is VXL

Advantage:

- More effective to compile
- Clearer structures
- Easier to use

New Package: Xshock Computation

- **Xshock Computation** is an **independent** pipeline for computing xshock graph. It has been removed the dependencies on LEMSVXL.
- Input: image (.png/.jpg) and contours (.cem)
- Output: extrinsic shock graph (.esf)

Input Parameters Control: .xml File

- .xml file can easily manage the input parameters for stand alone packages
- Since there are lots of parameters for the Xshock Computation, .xml is a good choice

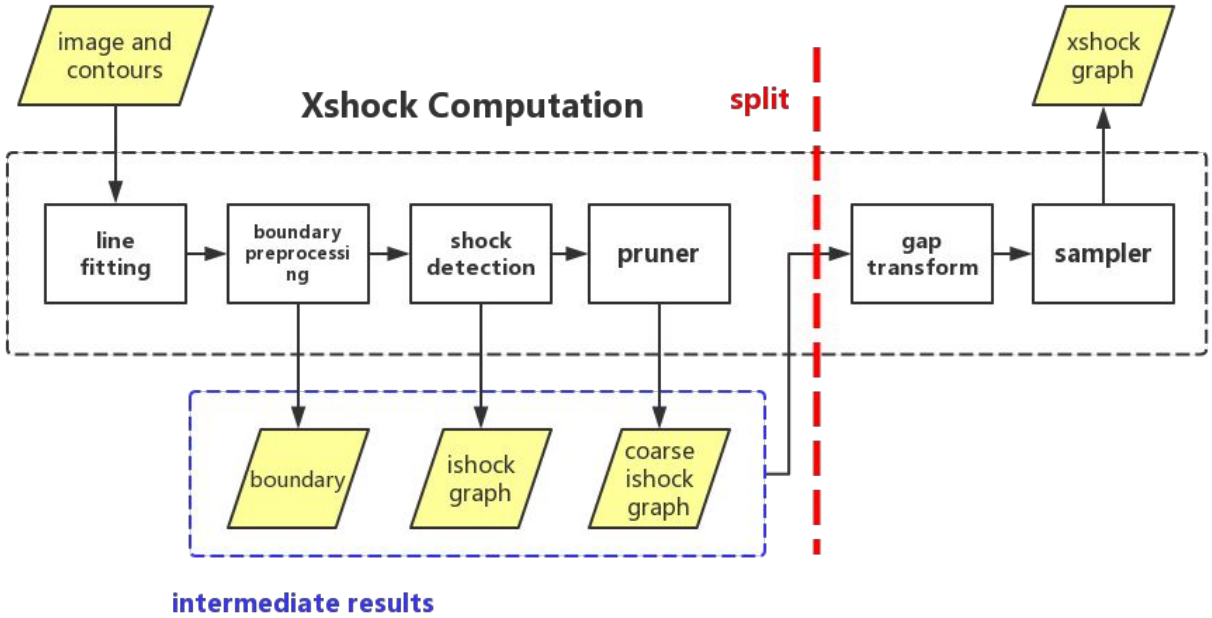
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dborl_compute_ishock><Compute_Ishock Compute_Ishock-arc_rms="0.05" Compute_Ishock-b_b
</Compute_Ishock>
<Gap_Transform Gap_Transform-alpha_app="1" Gap_Transform-alpha_cont="1" Gap_Transform-
</Gap_Transform>
<Sample_Shock Sample_Shock-both="off" Sample_Shock-inside="on" Sample_Shock-res="1">
</Sample_Shock>
<io add_bbox="on" gap_transform="on" input_contour_extention=".cem" input_image_extent
</io>
<web_directive categorization_file="categorization.xml" evaluation_file="evaluation.xm
</web_directive>
</dborl_compute_ishock>
```

Content

- Package Extraction
- Text File to Save IShock Graph
- Test Results on the Pipelines

Ishock and Xshock Graph

Output Shock, including ishock, is the **intermediate result** of Xshock Computation. Precisely, it includes boundaries, ishock and coarse ishock.



New Package: Ishock Computation

- For storing Output Shock as **files**, this stand alone package is needed.
- Input: image (.png/.jpg) and contours (.cem)
- Output: output shock graph (.osf)

New Package: osf to esf

- This package finishes other parts on Xshock Computation, e.g. gap transform, sampling, saving xshock ...
- Input: output shock graph (.osf)
- Output: extrinsic shock graph (.esf)

.osf (Output Shock) File

To save the output shock graph as text files, we need to design a format for the saver and loader. The following information is needed:

- Boundary (points and lines)
- Ishock Graph (nodes and links)
- Other information

We **do not save coarse ishock**, because when boundary and ishock is loaded, it is easy to be generated, i.e. *coarse ishock = f (boundary, ishock)*

Other information is saved for simplify the implementation of saver and loader.

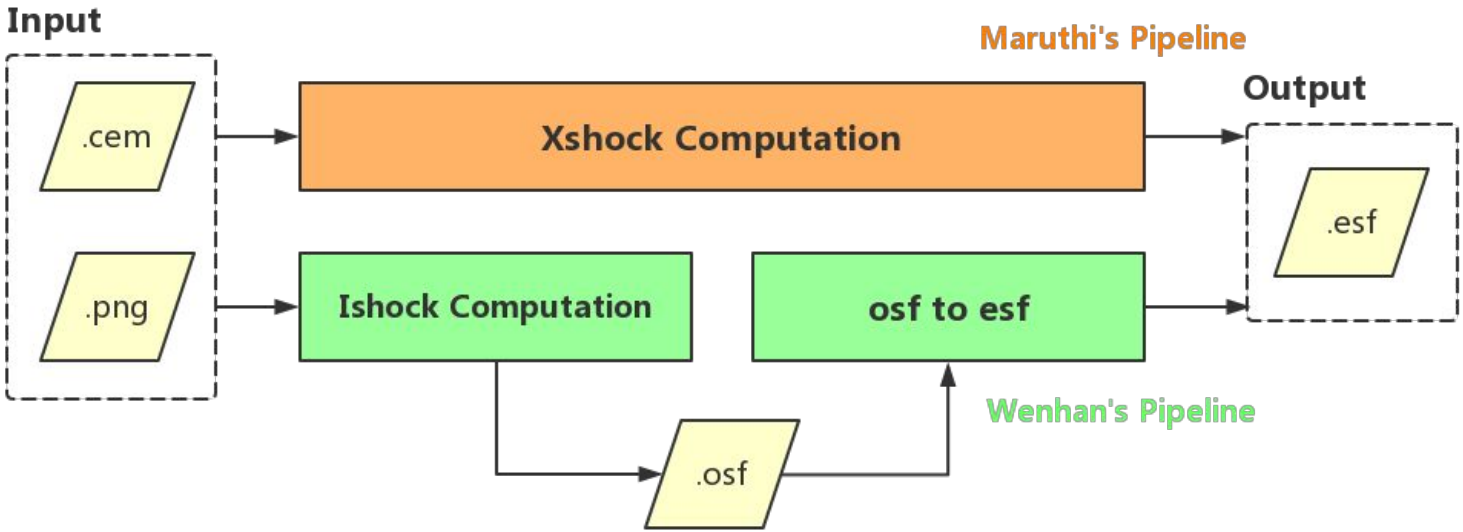
Algorithm: Generate .osf

Since there are too many structures needed to be saved, we can not save every information. Do the following steps to get **all necessary members** in .osf:

(1) Assume a set A := {necessary members, e.g <x, y>}
(2) Generate .osf file F with A
(3) Put A to the pipeline, if it **crash** or **get the wrong result**, find out the set A' := {some missing members}, A := A ∪ A', goto (2). If the results are **exactly the same**, goto (4).
(4) End

Summary of the Pipelines

Finally, we get the new packages:



Content

- Package Extraction
- Text File to Save Ishock Graph: .osf file
- Test Results on the Pipelines

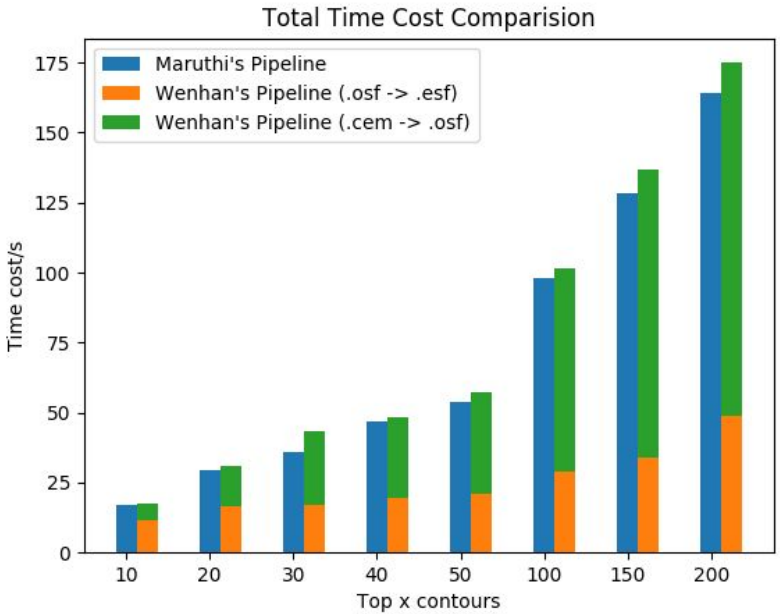
Reliability: Test Results (Crash Rate) on BSDS300

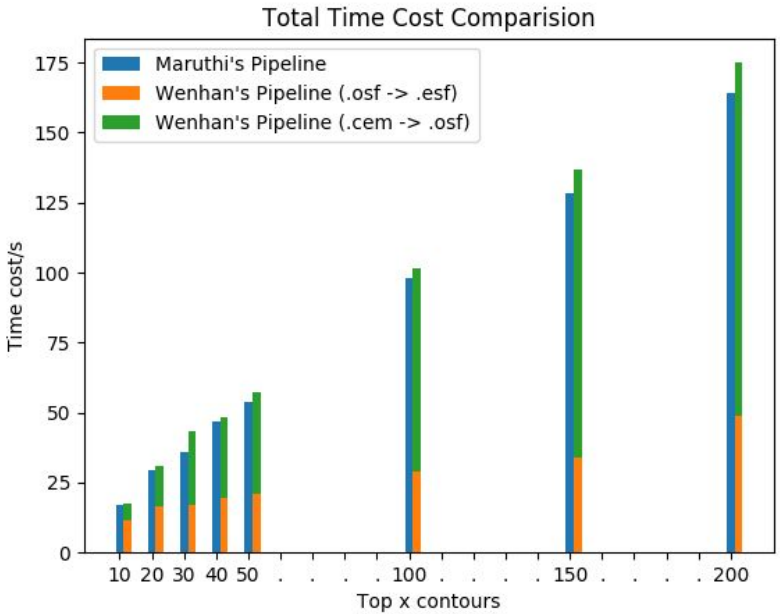
Contours	Top 10	Top 20	Top 30	Top 40	Top 50	Top 100	Top 150	Top 200
Wenhan's Pipeline	1/300	1/300	1/300	2/300	5/300	1/50	3/50	3/50
Maruthi's Pipeline	1/300	1/300	1/300	2/300	5/300	1/50	3/50	3/50

Note: Top x means there are only top x contours in .cem test files. The **same crash rate** means the bug I mentioned does not matter on "top x" tests.

Wenhan's Pipeline = Ishock Computation + osf to esf
Maruthi's Pipeline = Xshock Computation

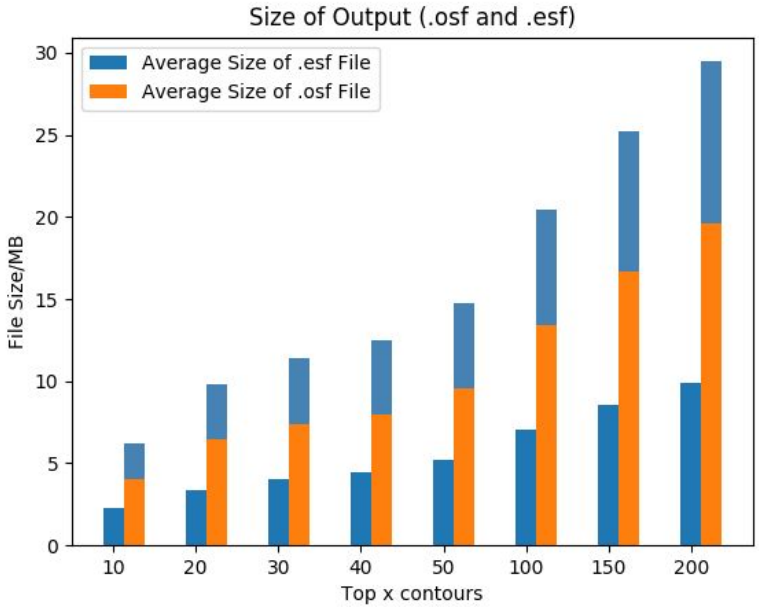
Speed

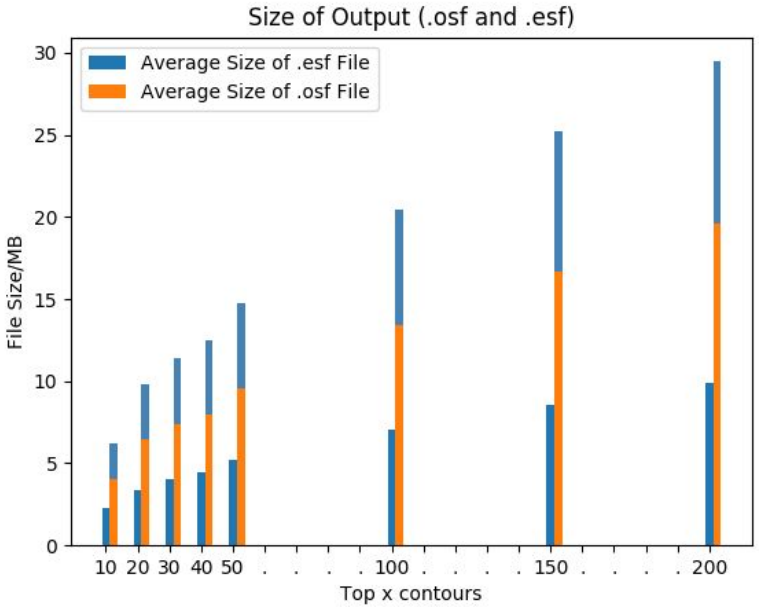




The tests are based on BSDS300 on a normal laptop. The height difference between two bars shows the time cost of .osf saver and loader.

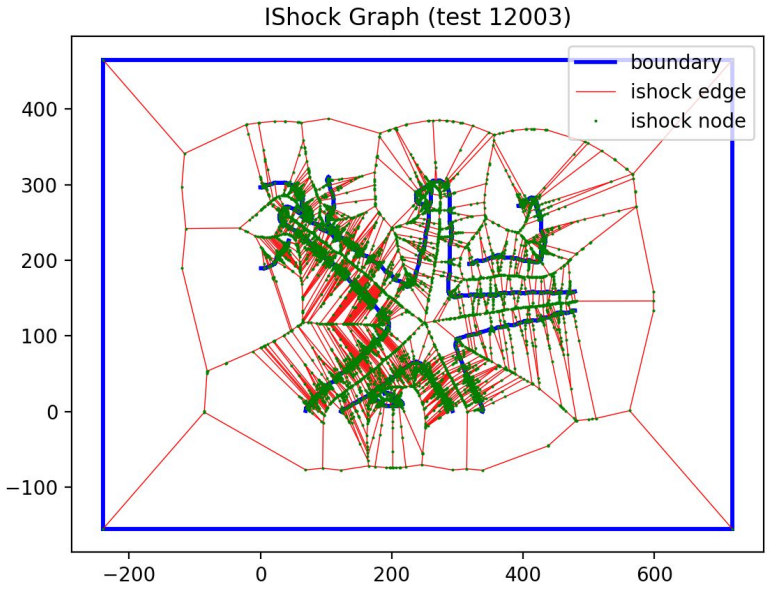
Size of Output Files

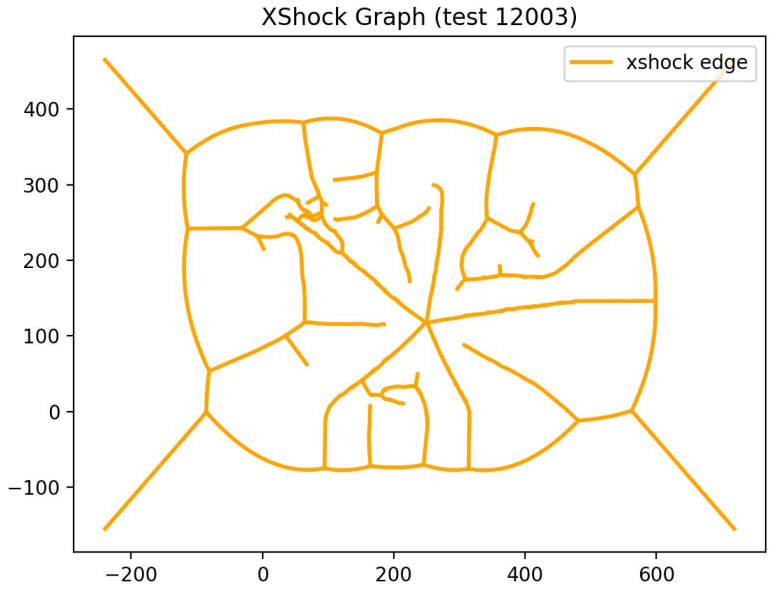




The right bars show total output size of Wenhan's Pipeline (.osf + .esf), which is 2 times larger than Maruthi's, due to the usage of .osf.

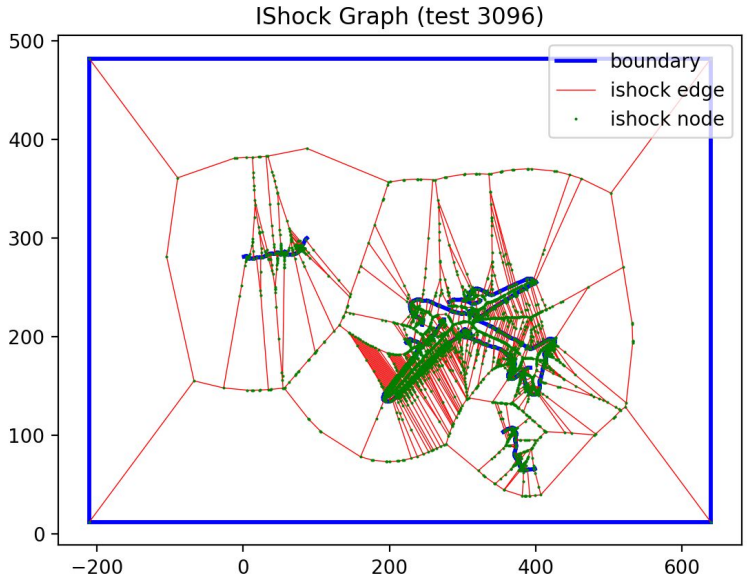
Some Examples on BSDS300

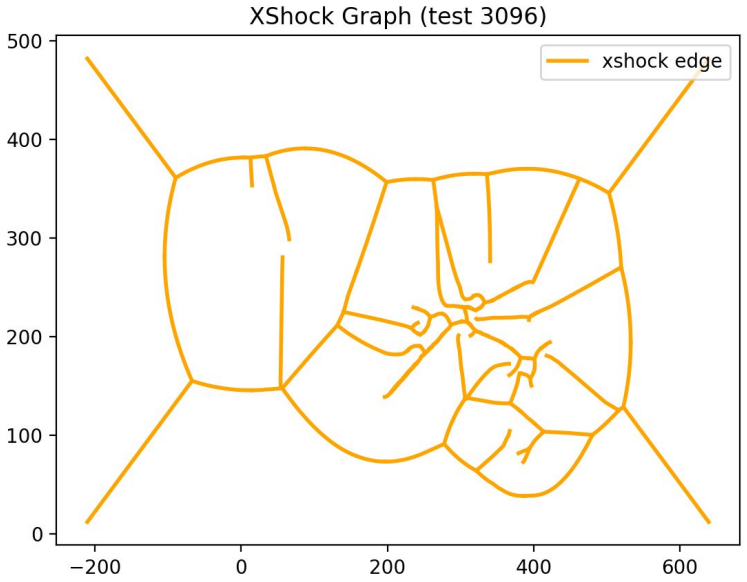




Note: Top 10 contours only.

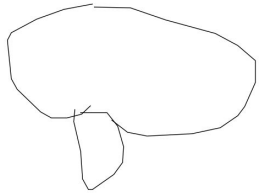
Some Examples on BSDS300

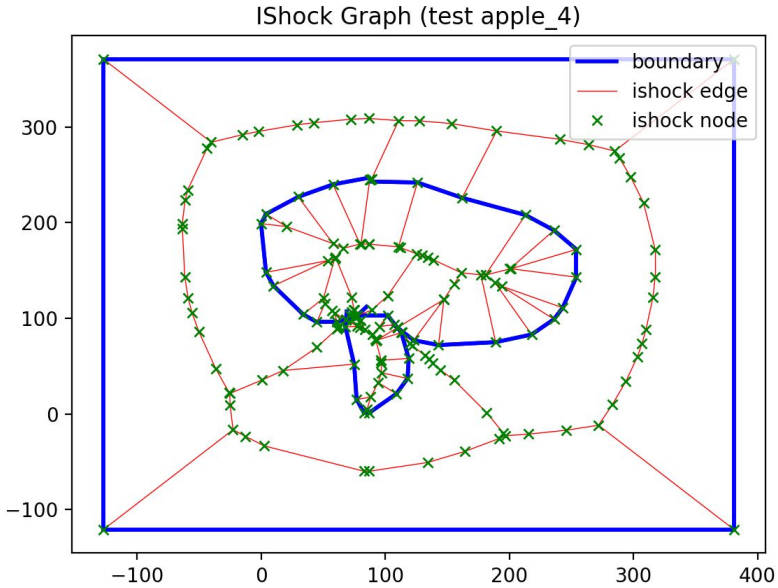


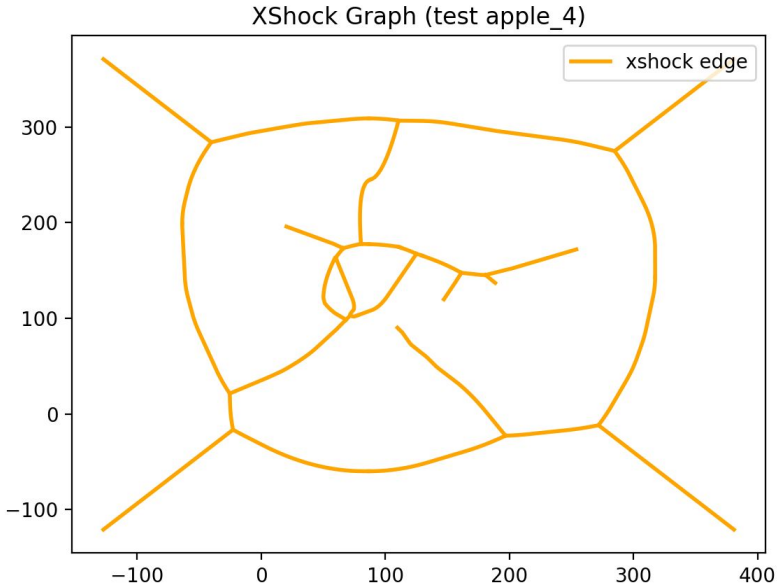


Note: Top 10 contours only.

Some Examples on Google Quick Draw

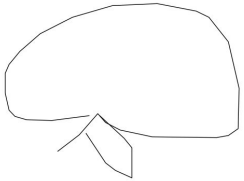


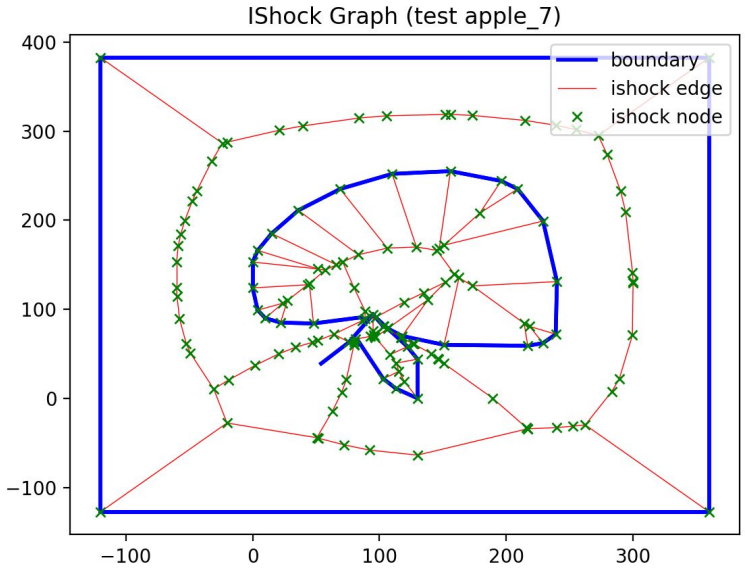


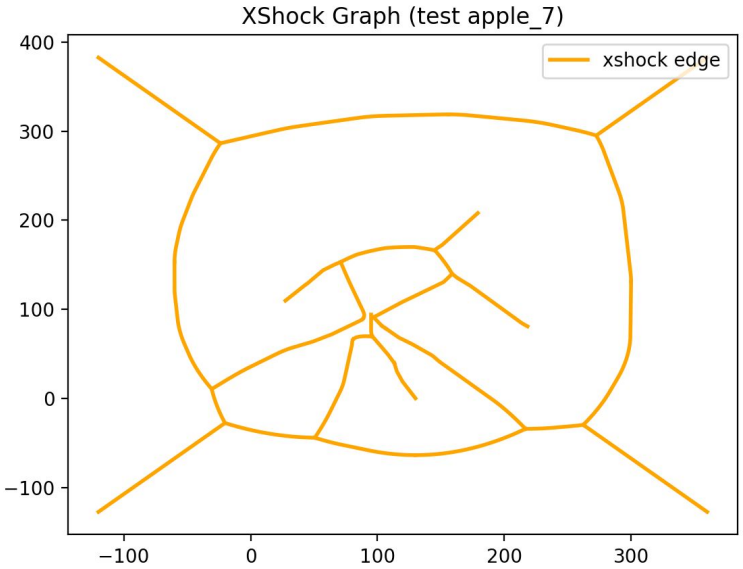


Note: Top 10 contours only.

Some Examples on Google Quick Draw

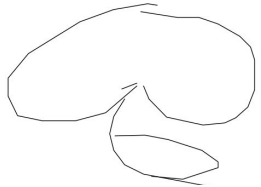


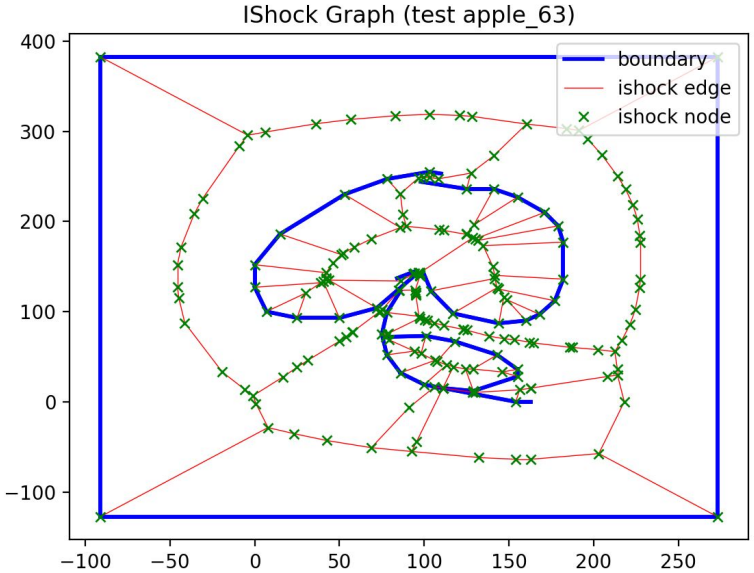


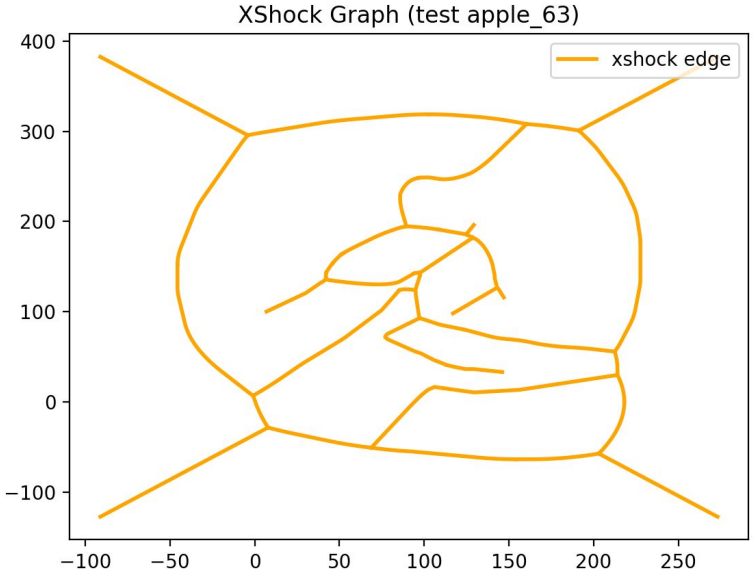


Note: Top 10 contours only.

Some Examples on Google Quick Draw







Note: Top 10 contours only.

Other Work for the Packages

- Write the User Guide and Technical Report (see .md files in GitLab)
- Make the packages compatible to latest VXL (v1.17.0)
- Use *gprof* to detect the bottleneck
- Use *valgrind* to target memory leakage (not fixed)
- Build a useful visualizer to show the results
- Build scripts to access and test the data from dataset

Future Work

- Memory leakage problem (see Progress Report slide 14, 40, 41)
- Redesign the structure for ishock graph and sampler package
- Redesign the .osf, make the number of members saved minimum
- A better visualizer (no parabola in current one)
- Test the remaining data on top 100, 150, 200… see the crash rate
- Bugs: when boundary box is off, the .esf output is strange
- ...

Thanks for watching!

by Wenhan Shi
mailto: wenhanshi2018@gmail.com