CMPT-764 COURSE PROJECT

Krut Patel - Mohammad Nourbakhsh - Rahul Moorthy - Wei Wang

Table of Contents

Summary

- Focus of our project.
- Mainly composed of three modules: Parser, Mixer and Scorer
- Platform we chose (Python + VTK).

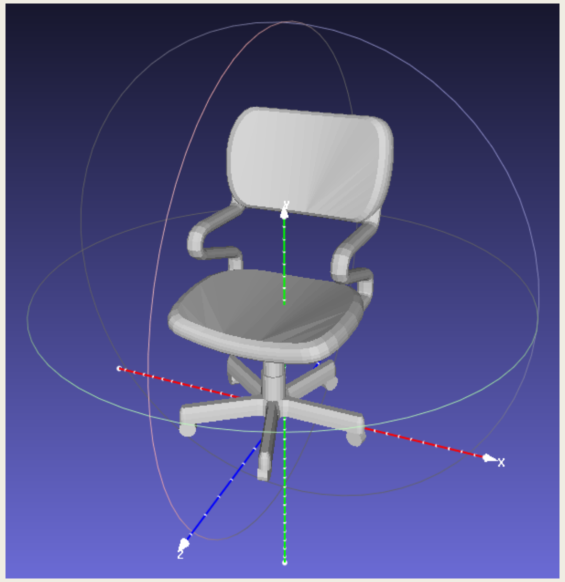
Parser

- Read the obj mesh file and converted it into data structures to be used later.
  - *List of vertices, faces, edges.*
  - *Min max distance of each axis and the bounding box.*
  - *Internal flags to mark display mode, normalized or not etc.*

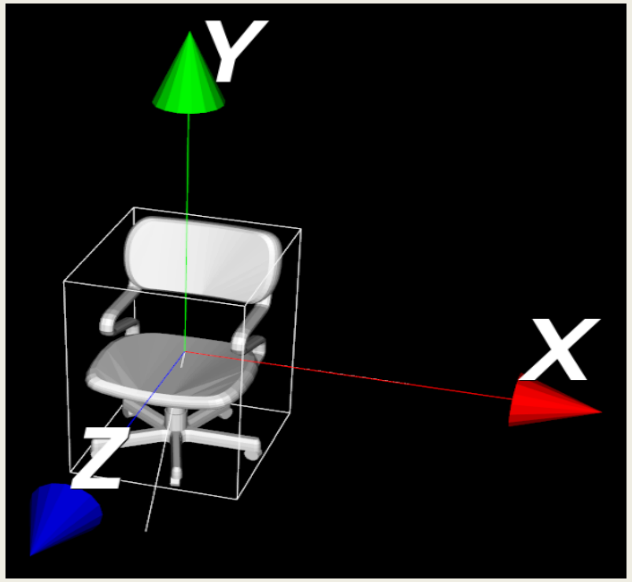
- Offer functionalities to help other team members to implement their work.
  - *Ability to display mesh in 3D and/or voxel.*
  - *Ability to compute and display mesh legend such as axis, bounding box etc.*
  - *Normalize mesh origin and resize all vertices between 0 and 1.*
  - *Convert mesh to voxel representation with given resolution and size.*
  - *Analyze principle components.*
  - *Project top, left, and front view based on voxel with depth enabled.*
  - *Detect collision between meshes based on bounding box and voxel.*

Parser

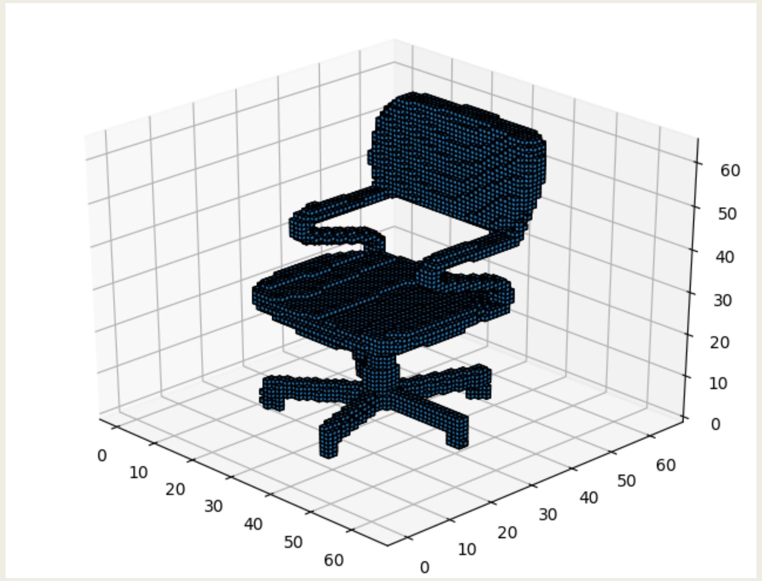
- Voxel Representation



MeshLab



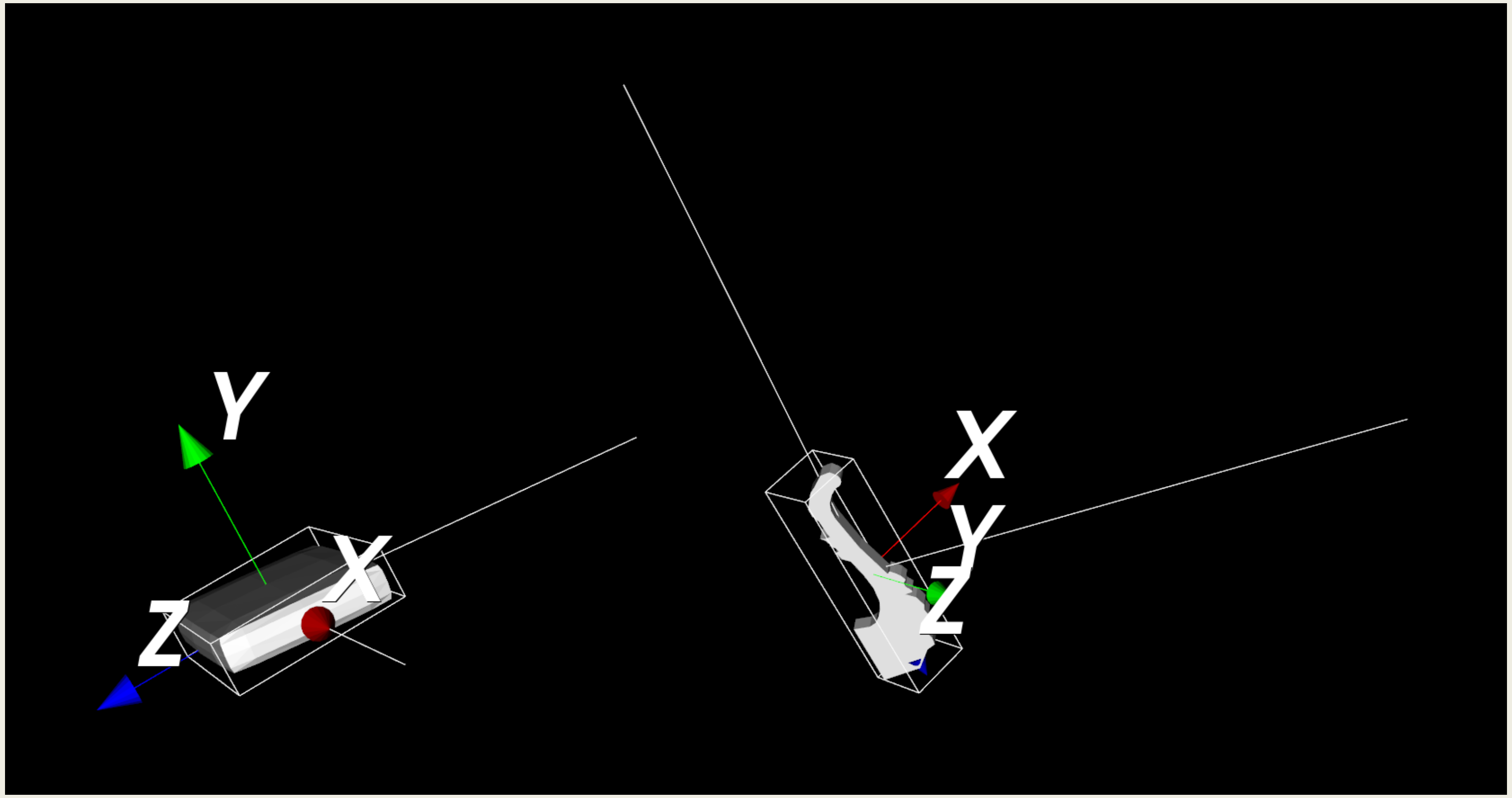
Preview of our results in VTK



Preview of our Voxels in Matplotlib

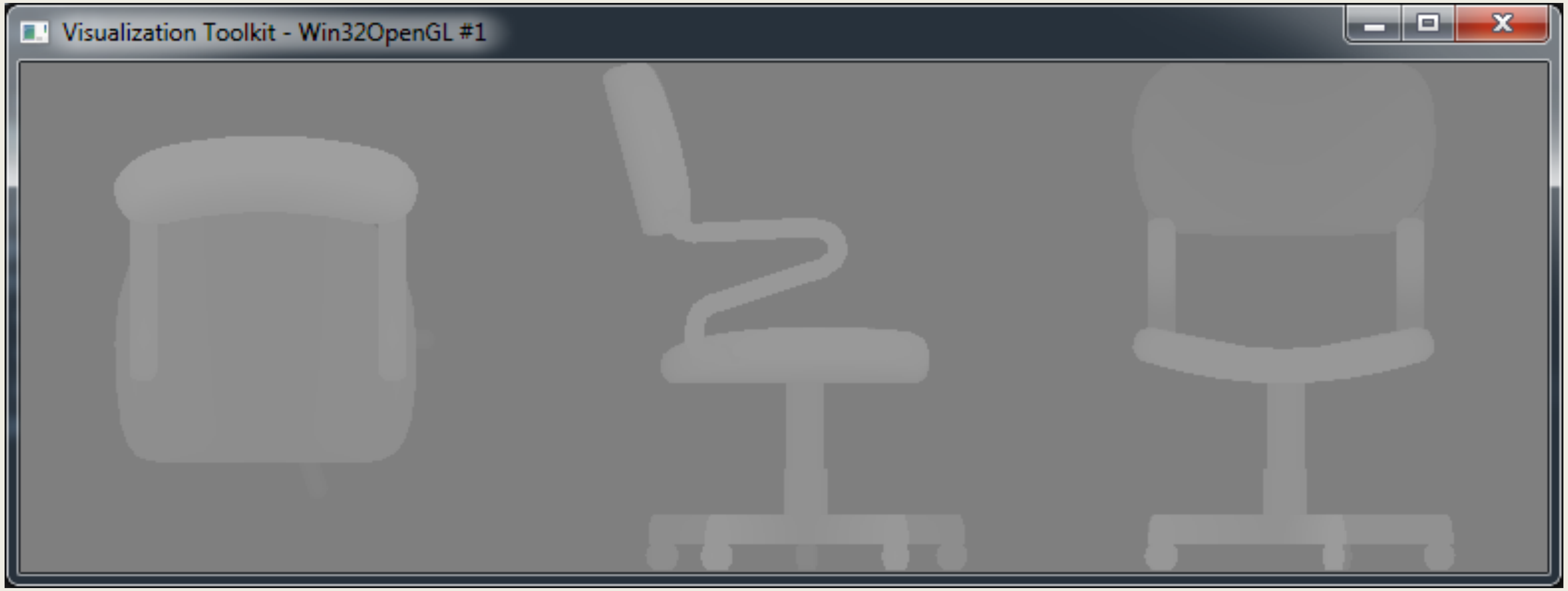
Parser

- Utilizing PCA



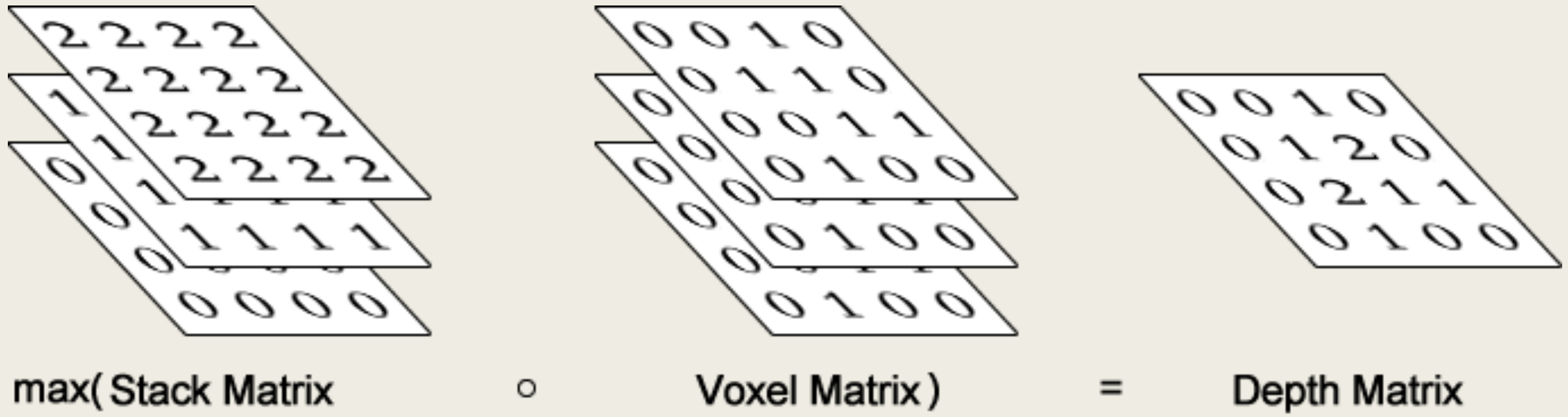
Parser

- Top, Left, and Front Views.



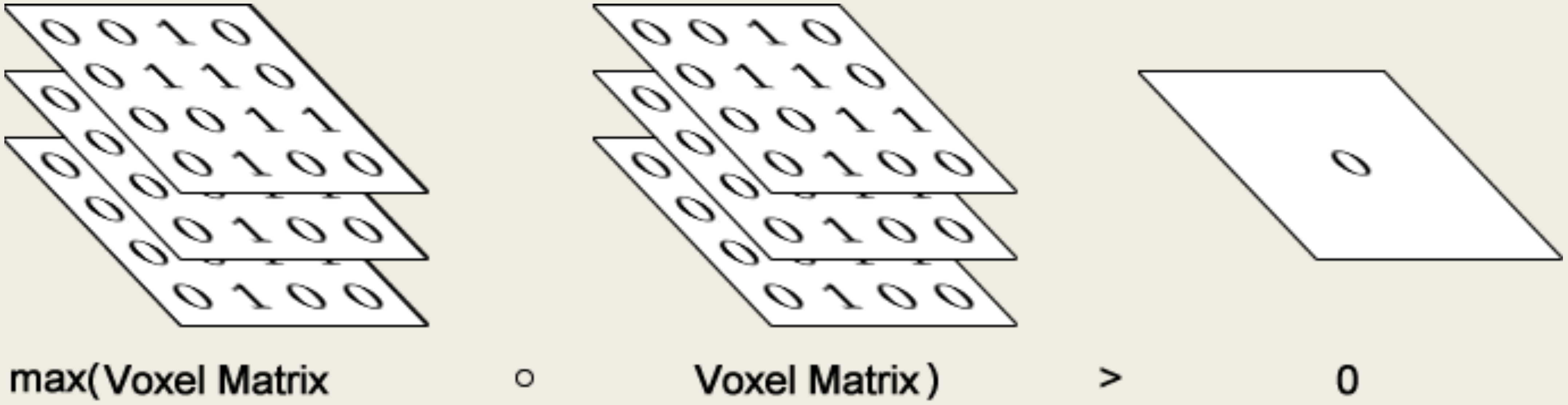
Parser

- Top, Left, and Front Views.



Parser

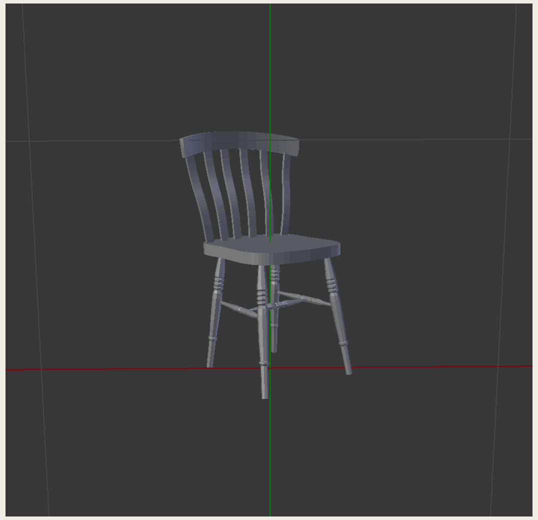
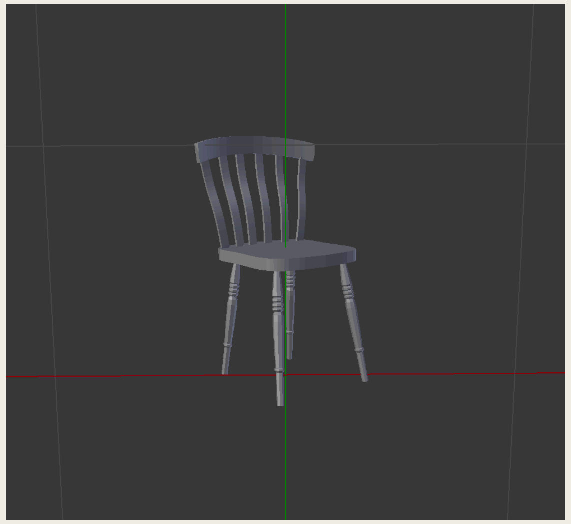
- Collision Detection



Mixer

- Dataset for the Mixer:

- Discarded chair parts that are not either a leg, seat or back.

- For chairs with armrest, we combined the armrest with the seat as an entire mesh.

Mixer

- Only *4-legged* chairs are considered as part of input to the mixer. Chairs with either 1 or 2 legs are discarded. For instance:

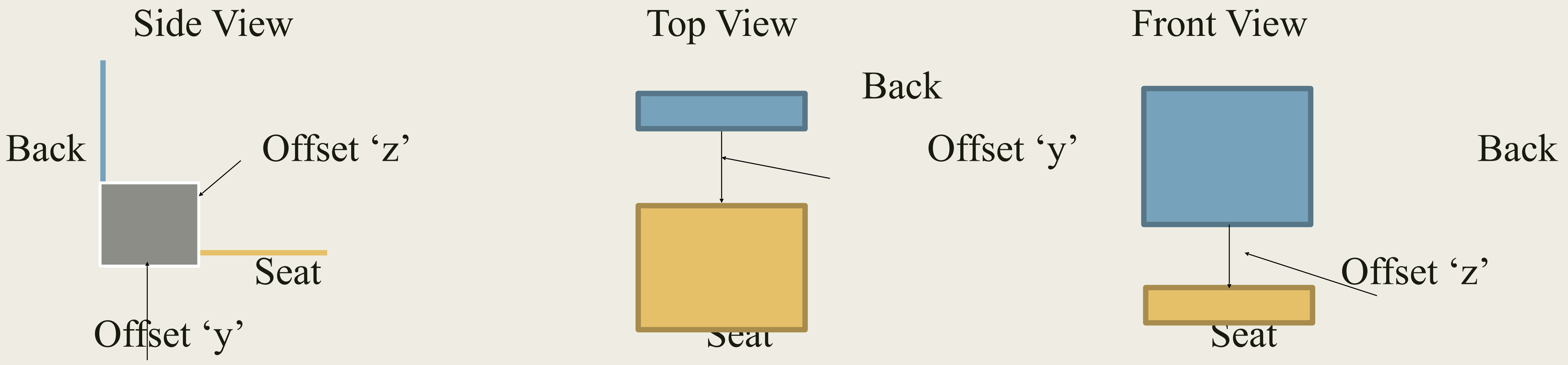
Mixer

- Implemented Algorithm :

– We consider seat as the main node and connect the legs and back of the chair to it. Also we normalize the parts w.r.t its original chair.

A. For the Back of the Chair:

1. We scale the back to the size of the seat. Then, initialize the back at some offset ‘y’ and ‘z’



2. We iteratively move the back close to the seat by a step-size across step ‘z’ and step ‘y’

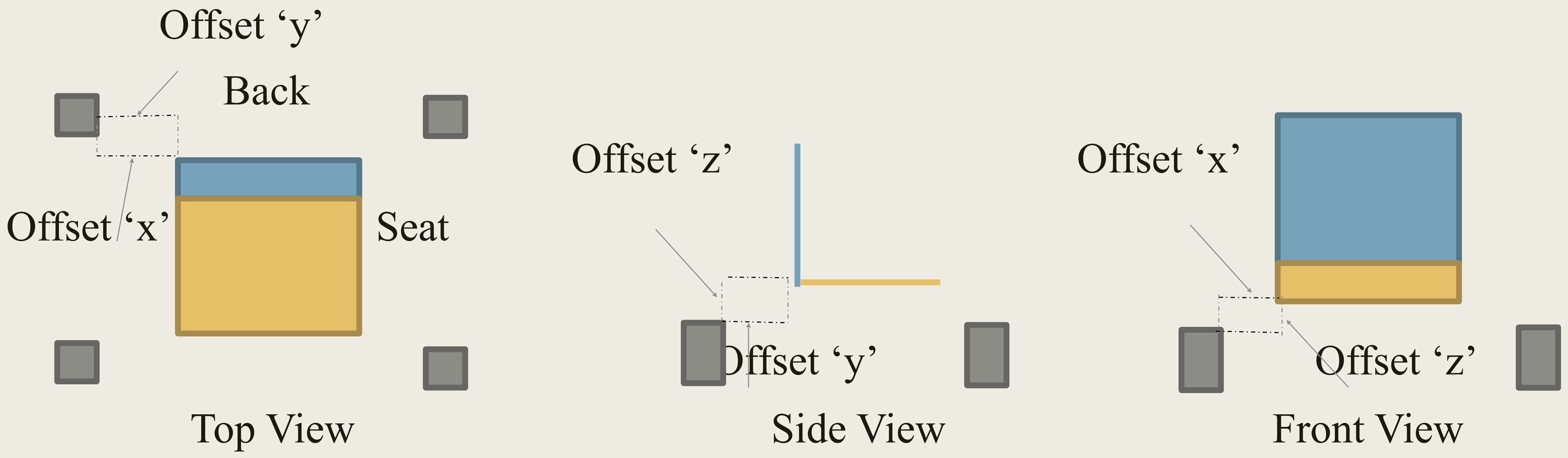
Mixer

3. After each iteration, we voxelize the back and seat and check for its intersection

4. Repeat step 2 and 3 until collision condition is met.

B. For the Legs of the Chair:

1. Initialize all legs at some offset x, offset y and offset z from each corner of the seat.



Mixer (Contd)

2. Iteratively move legs closer to the seat by offset x, offset y and offset z.

i. We don’t apply step z to other legs if either of the one have collided.

ii. If front legs (i.e front left and front right) any of them have collided, set the collision for the other leg as true. Same applies for back legs as well.

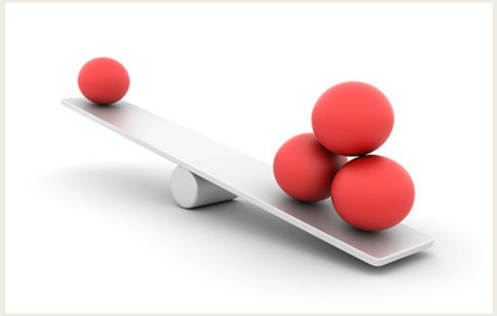
3. After each iteration check for the collision condition after voxelizing the legs and seat.

4. Repeat steps 2 and 3.

Repeat the entire process until legs and back collide/intersect with the seat. Post which, we get the projection after voxelizing the entire combined model and save it to images.

Scorer

- Improvements:
  - *Changes made to the dataset:*
    - *Dataset was heavily unbalanced (unequal number of positive and negative samples). Handled this scenario in the loss function.*
    - *Order of images in positive and negative classes was inconsistent. Appropriate scripts were implemented to keep the order intact.*
  - *Concatenating three shots of images into a single one and passing this as input to the network.*
  - *Dividing the dataset to train and test. (80% and 20% respectively)*



Scorer

- Improvements (Contd):

- *Changed the network structure as :*
    - ***conv + ReLU + maxpool + 10*(resblock : conv + batchnorm + ReLU + conv + batchnorm) + maxpool + conv + ReLU + fc + ReLU + dropout + fc + logSoftmax***

    - *This network structure helped us handle vanishing gradients by utilizing resnet blocks (skip connections).*

- *Weighted CrossEntropy was used as the loss function.*
    - *Considered a weighted objective function (**weights:1/len(negativeSamples, 1/positiveSamples)),** which helped in resolving the imbalanced dataset.*

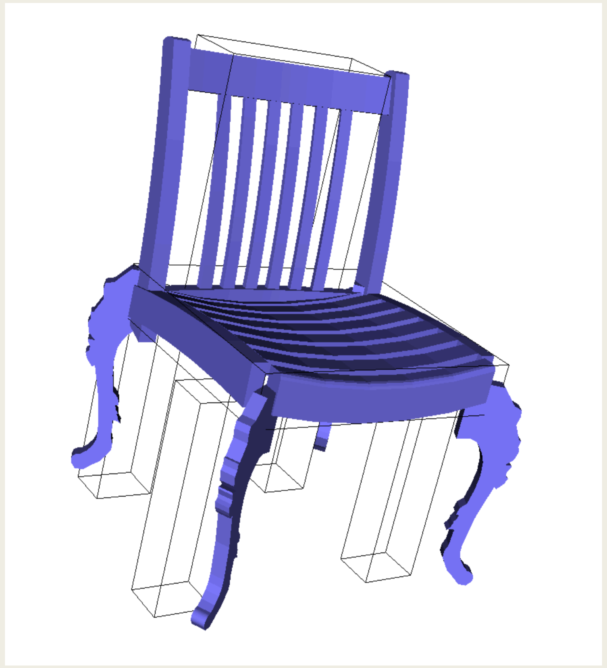
Scorer

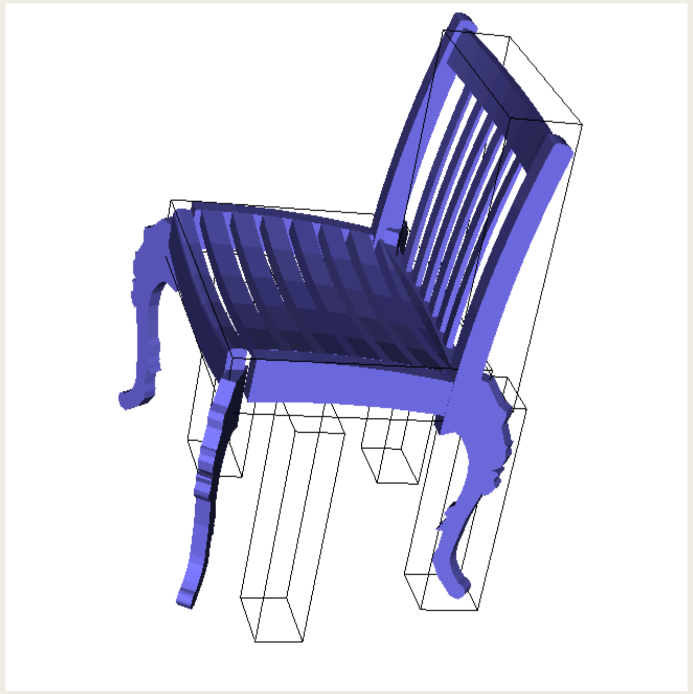
- Improvements (Contd):

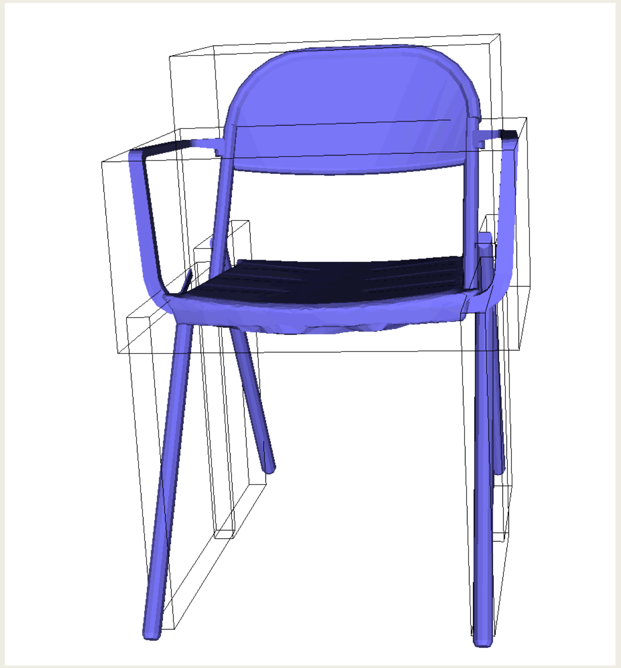
- Adam was used the optimizer (with lr : 0.0001)

- Currently, we are reporting precision and recall (precision: 0.9977, recall : 0.9992)

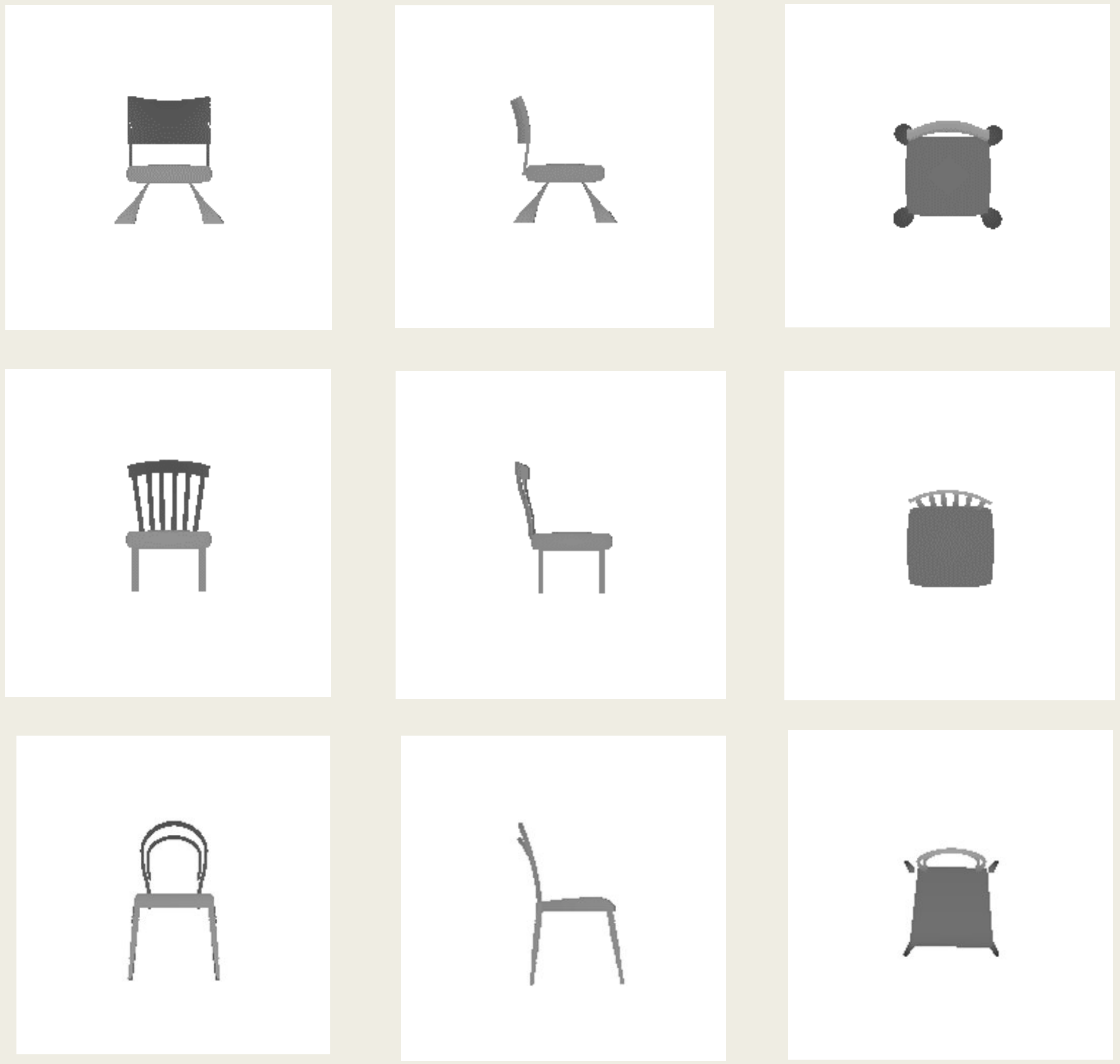
Results



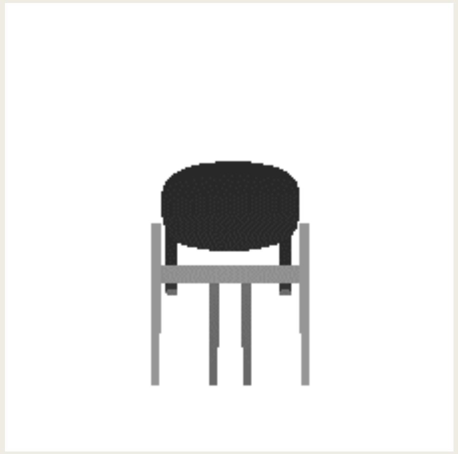


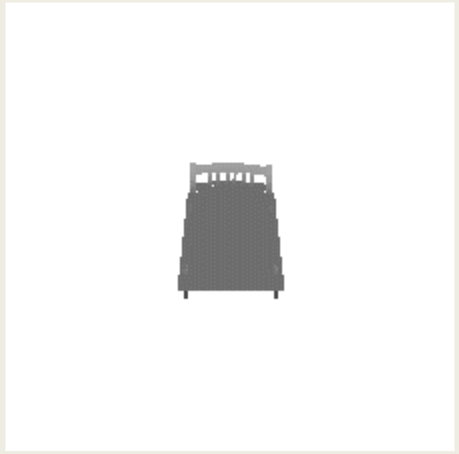


Projections – Good ones



Projections – Bad ones





References

- https://www.cs.sfu.ca/~haoz/pubs/zhu_sig17_scsr_small.pdf
- https://github.com/activatedgeek/LeNet-5