ODSC India 2019

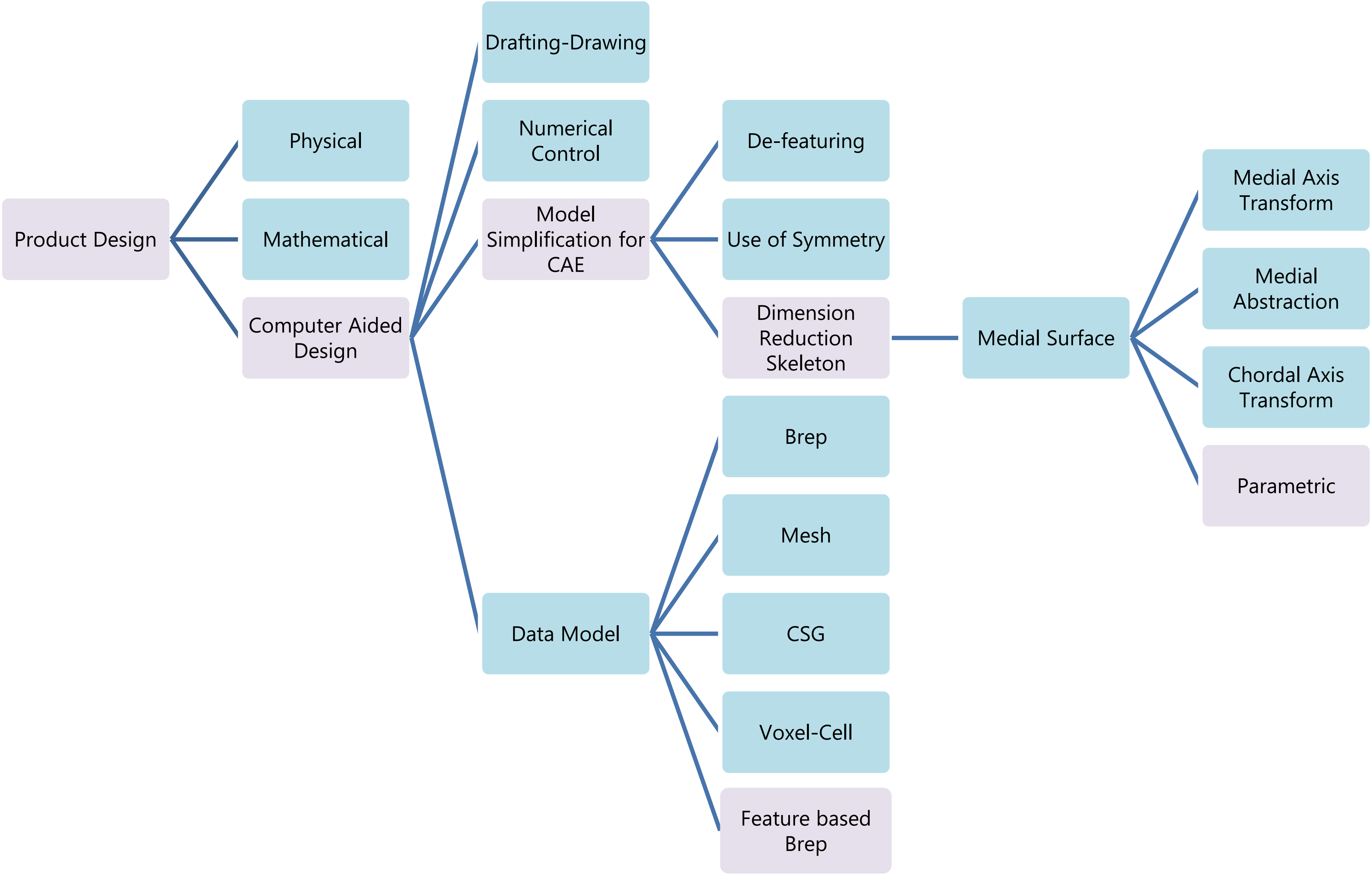
Midcurve by Neural Networks

Yogesh Kulkarni

MidcurveNN : Encoder-Decoder Neural Network for Computing Midcurve of a Thin Polygon

INTRODUCTION

The Context



Product Design

Physical

Mathematical

Computer Aided Design

Drafting-Drawing

Numerical Control

Model Simplification for CAE

Data Model

De-featuring

Use of Symmetry

Dimension Reduction Skeleton

Brep

Mesh

CSG

Voxel-Cell

Feature based Brep

Medial Surface

Medial Axis Transform

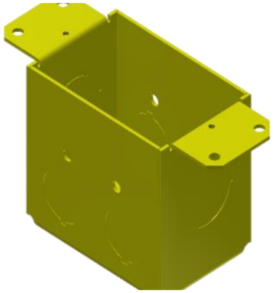
Medial Abstraction

Chordal Axis Transform

Parametric

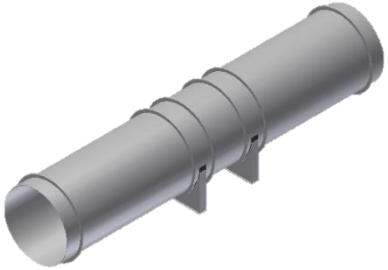
Main Applications

Aerospace



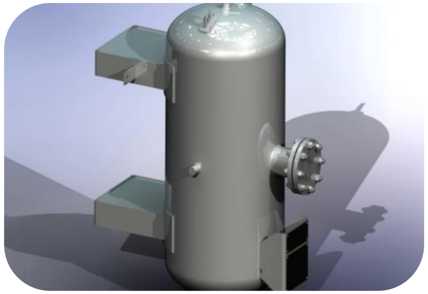
Machinery

Consumer



Energy

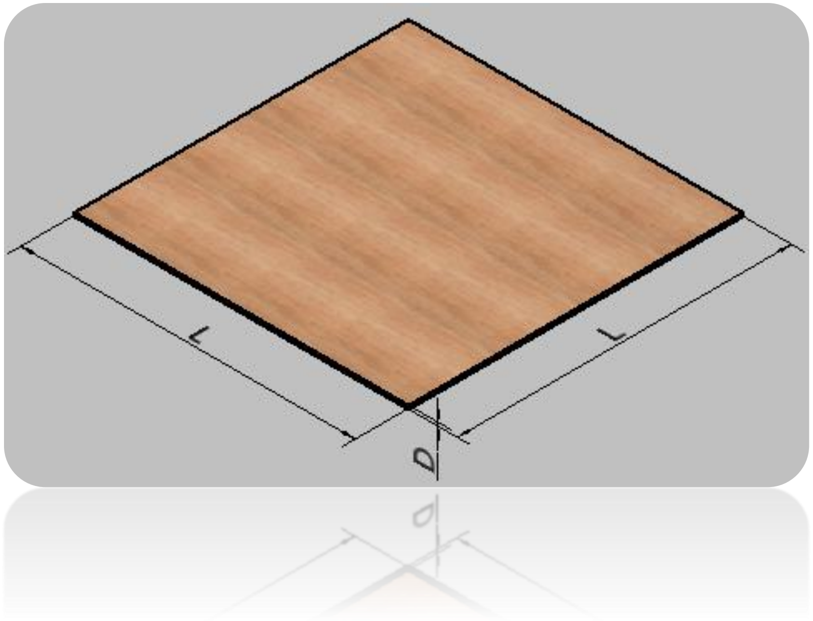
Construction



Industrial

What is Thin

It is defined as a part or body with large effective span to thickness ratio (L/D)



✓ Well-common FEA practice

$L/D \geq 100$

✓ Generally, pressure vessel

$R \geq 5xT$

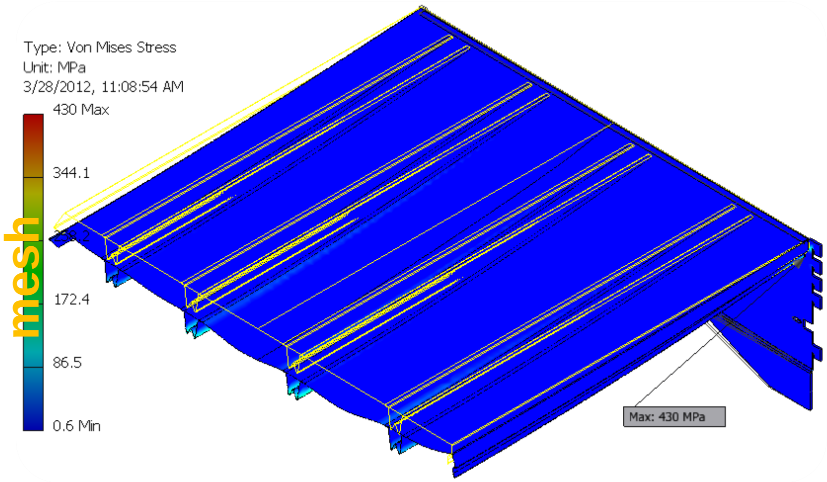
Shell vs. Solid mesh

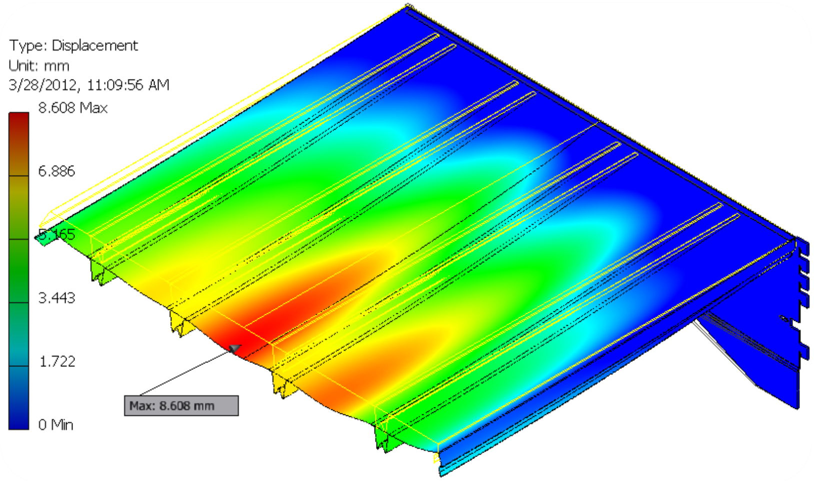
	Solid mesh	Shell+Solid mesh	Difference (%)
Element number	344,330	143,063	-58%
Node Number	694,516	75,941	-89%
Total Degrees of freedom	2,083,548	455,646	-78%
Maximum Von. Mises Stress	418.4 MPa	430 MPa	+3%
Meshing + Solving time	Out of memory	22 mins	N/A (4G RAM)
Meshing + Solving time	30 mins	17 mins	-43% (12G RAM)

Dramatic performance gain without sacrificing accuracy!

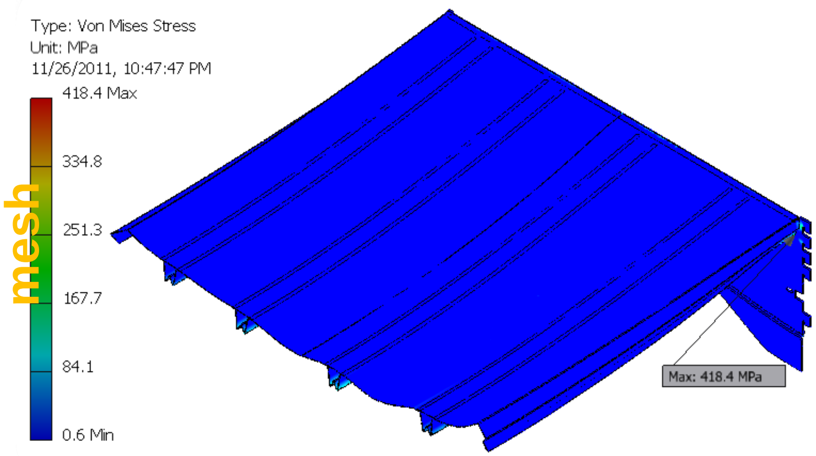
Shell vs. Solid mesh

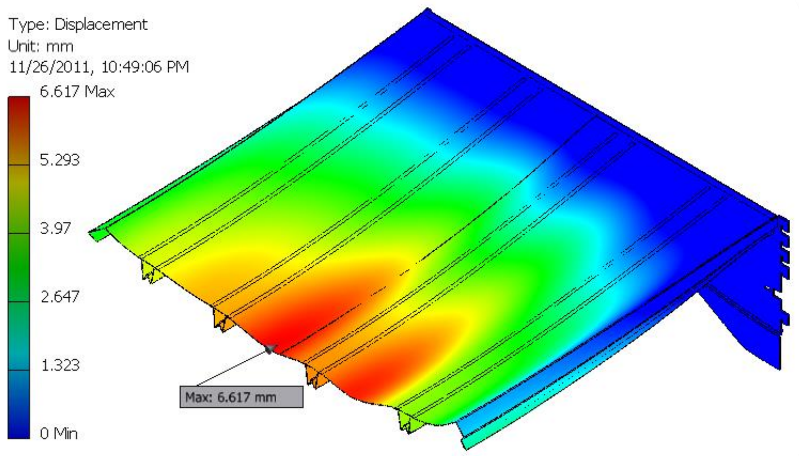
Shell mesh





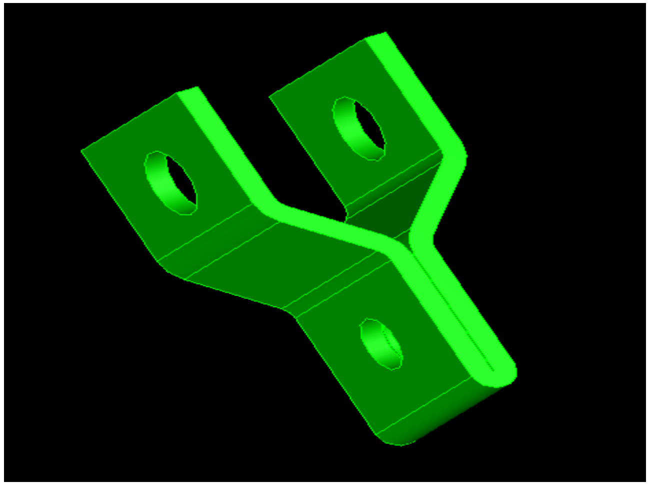
Solid mesh



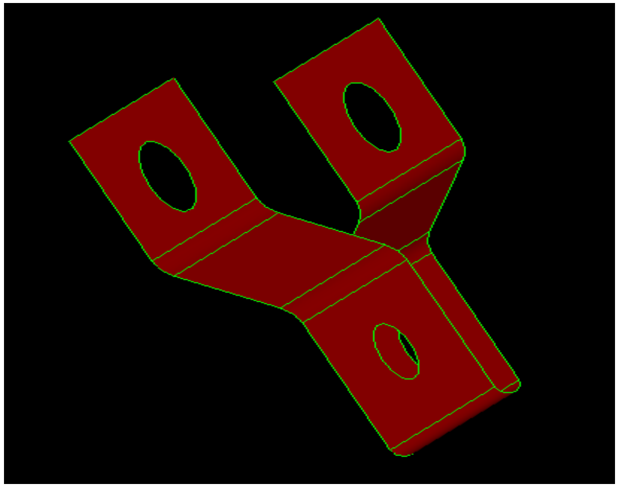


Midsurface is?

- Surface approximation for the thin-walled model
- Used to create shell element model for shell analysis
- Not expected to work for thick models



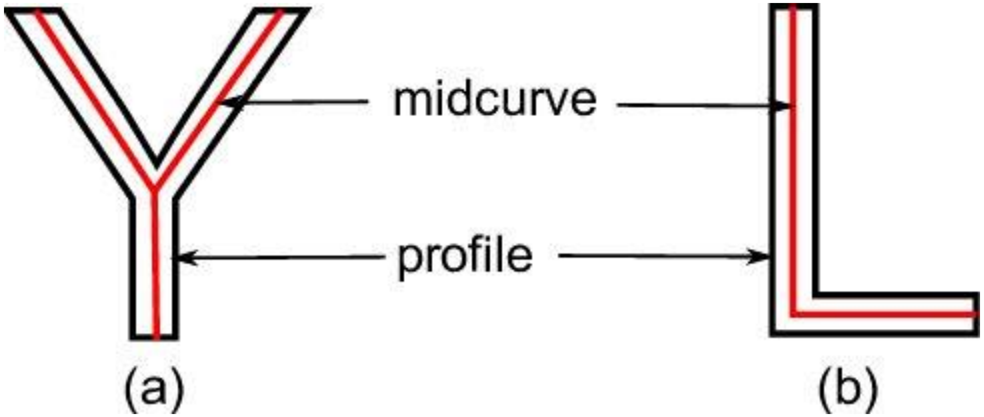
Input: Solid



Output: Midsurface

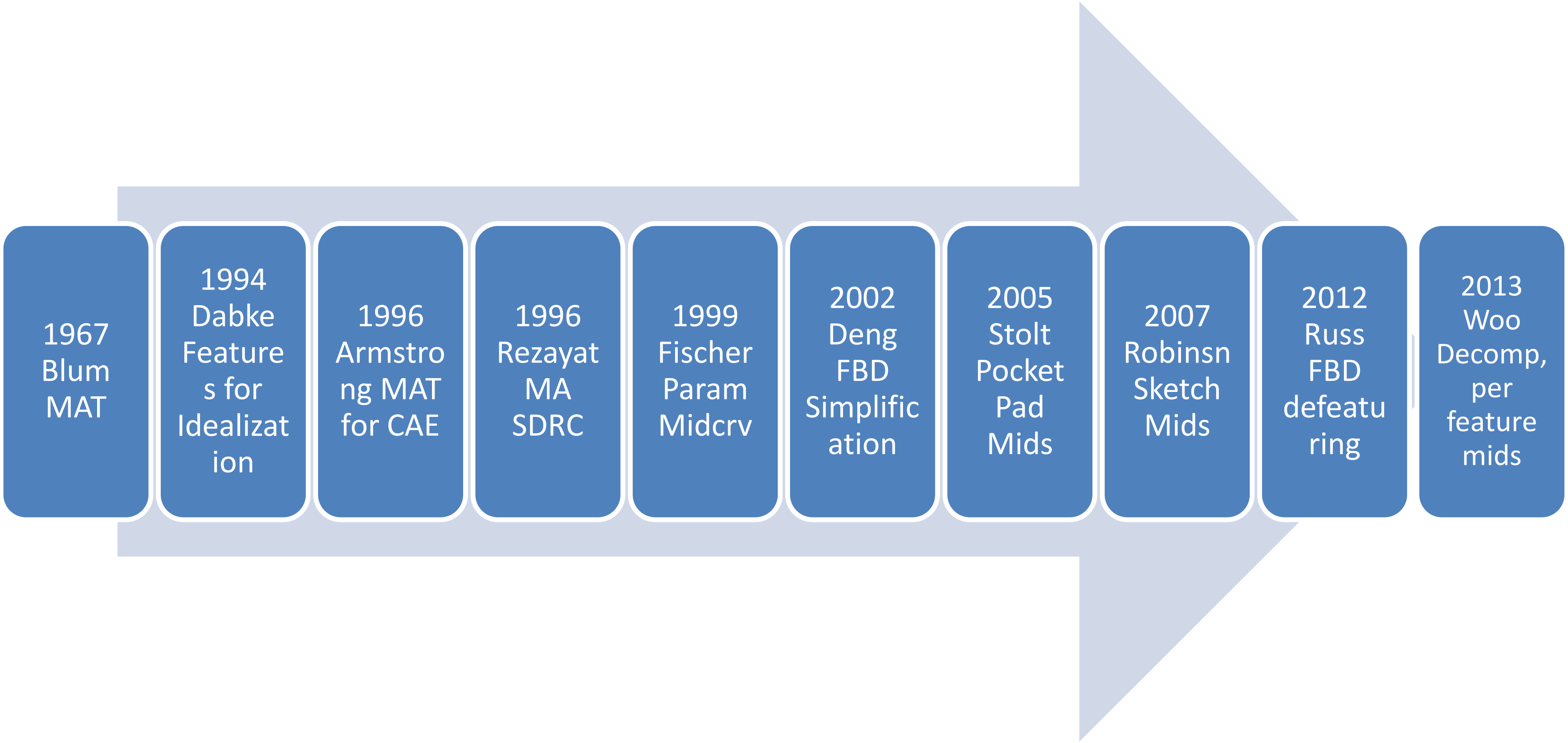
What is a Midcurve?

- **Midsurface : From 3D thin Solid to 2D Surface**
- **Midcurve : From 2D Profile to 1D Curve**



STATE OF THE ART

When-What



1967 Blum MAT

1994 Dabke Features for Idealization

1996 Armstrong MAT for CAE

1996 Rezayat MA SDRC

1999 Fischer Param Midcrv

2002 Deng FBD Simplification

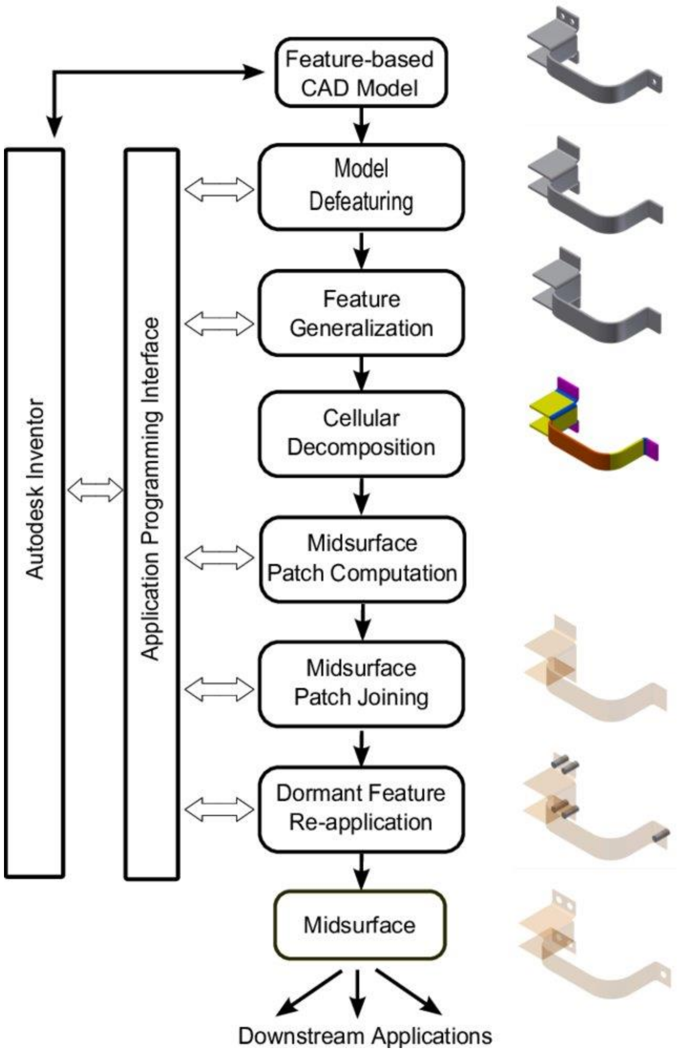
2005 Stolt Pocket Pad Mids

2007 Robinsn Sketch Mids

2012 Russ FBD defeaturing

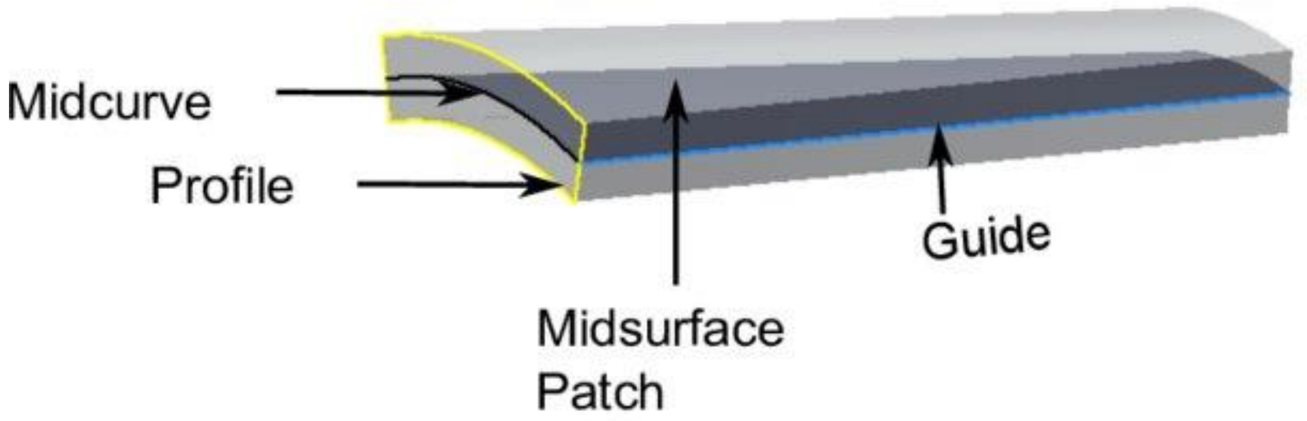
2013 Woo Decomp, per feature mids

2017 My Doctoral Work



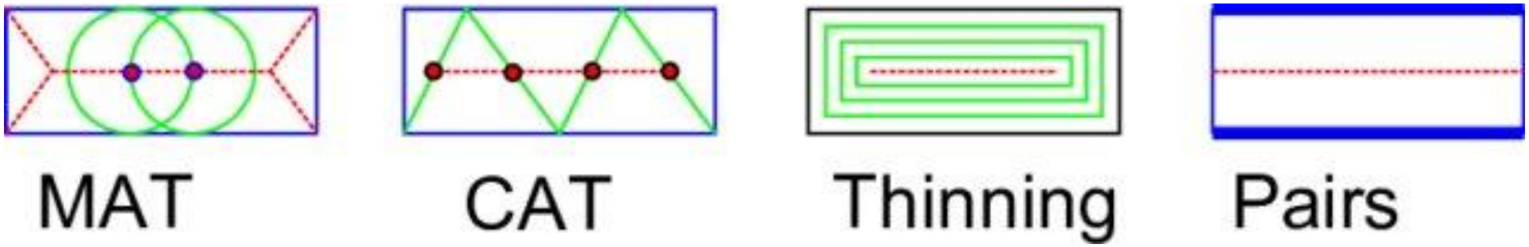
Midcurve Computation

- Midsurface of a Patch is Midcurve of its profile extruded.
- So, it boils down to computing 1D midcurve of a 2D profile



Midcurve Approaches

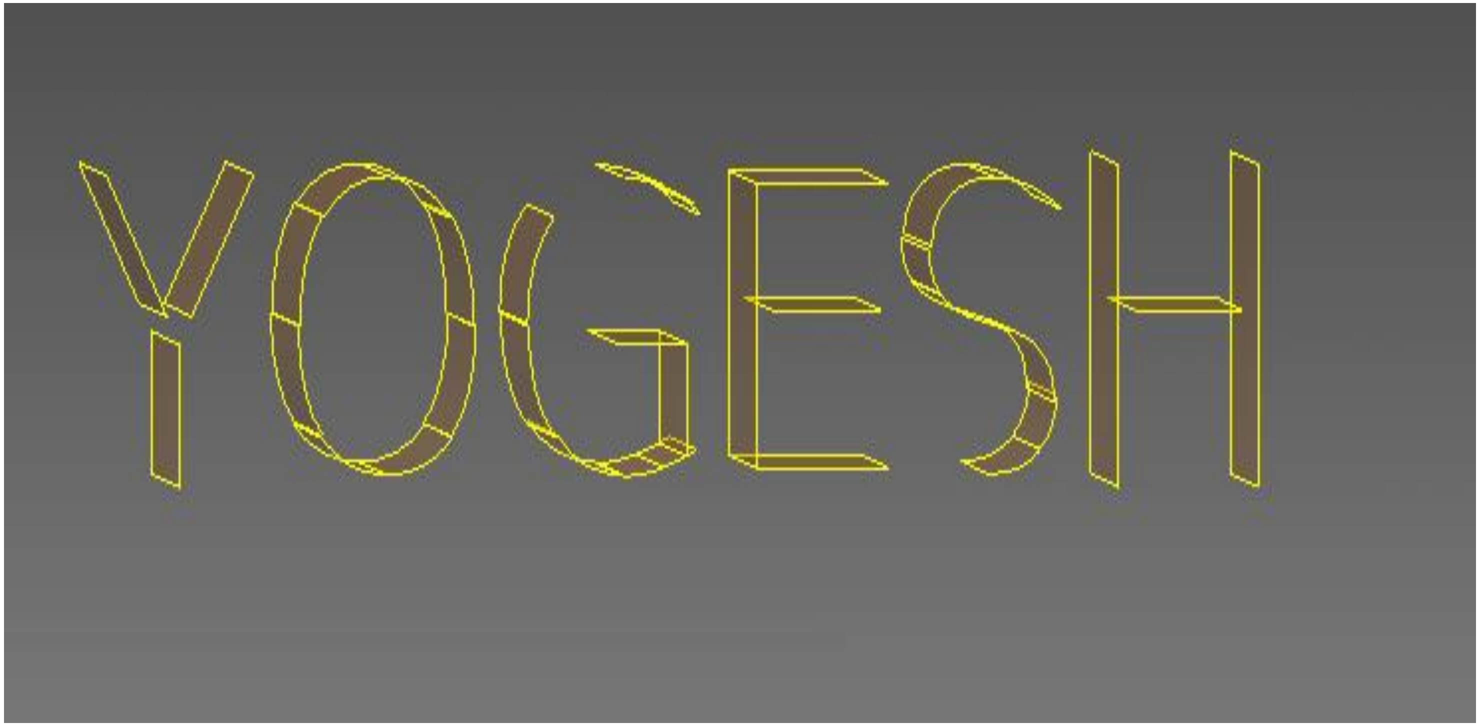
- Midsurface of a Patch is Midcurve of its profile extruded.
- So, it boils down to computing 1D midcurve of a 2D profile



Literature Survey - Conclusions

- *“There is a definite need for a dimensional reduction capability that is more powerful and easier to use than those currently available in the market. Such a capability should deliver an automated scheme for handling cases that have traditionally caused problems for algorithms in this field*” - Stanley2010
- “*Much of research is yet to be done, use of symmetry, various features, various abstractions are not yet handled.”*- Smit2011

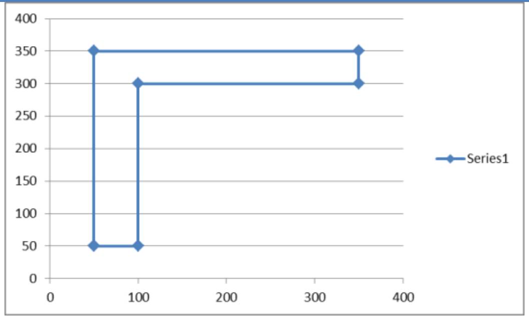
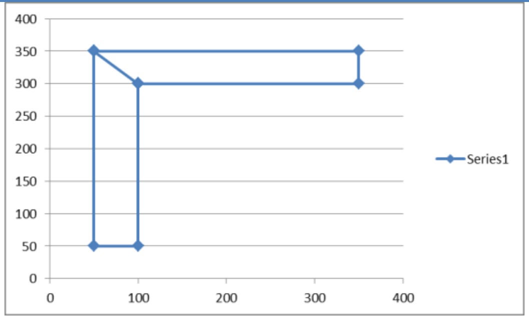
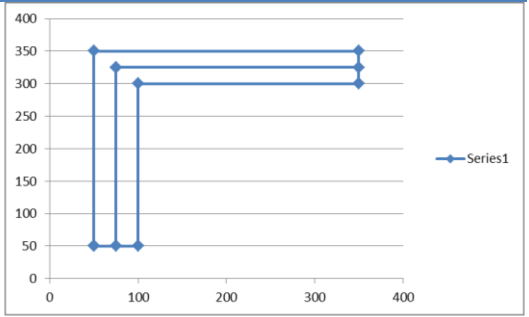
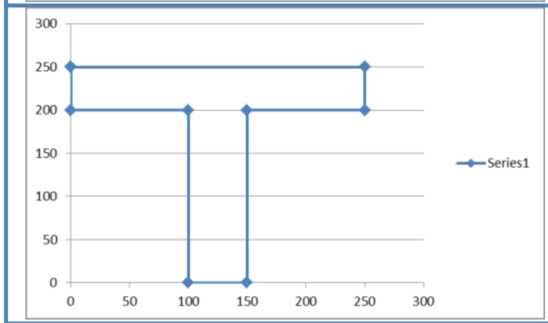
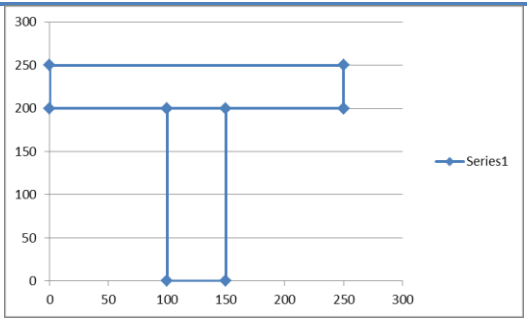
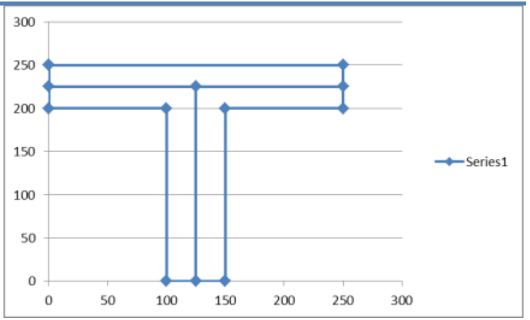
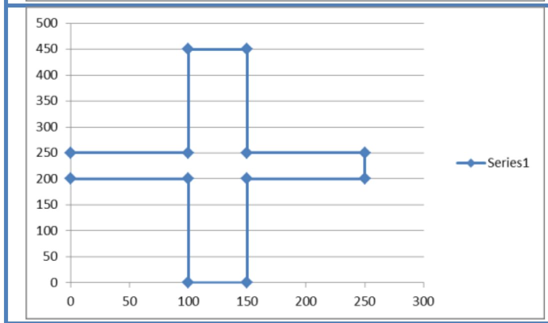
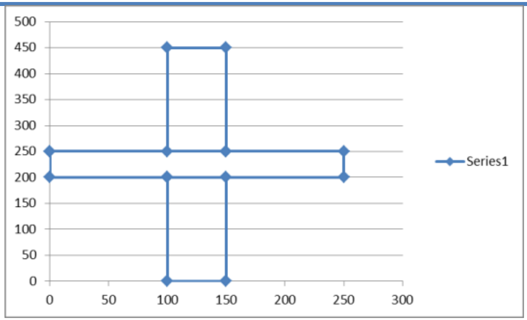
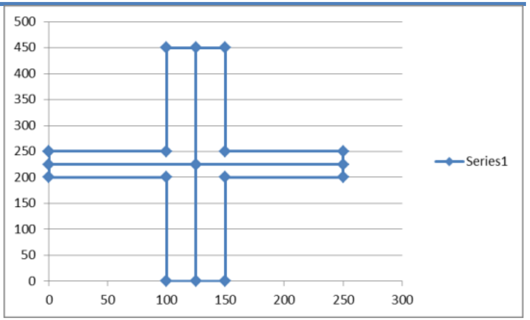
Problems Identified so far



2017: Proposed Approach

- Given a 2D closed profile, get connected medial curves, no extra branches
- Decompose 2D to find its sub regions, like features in 3D. Sub regions being simpler, it would be easy to get Midcurves, than skeleton of whole Profile.
- Generate individual midcurves. Extend & Join

2017: Results 2D Midcurves

Shape	My Impl Partitioning	My Impl Midcurves
		
		
		

Limitations

- Fully rule-based
- Need to add/adjust rules if new type of shape comes
- So, not scalable

Idea

- Can Neural Networks “learn” the dimension reduction transformation?
- Supply lots of training data of profiles and their corresponding midcurves and train.
- Then given an unseen profile, can Neural Network compute a midcurve, mimicking the original profile shape?

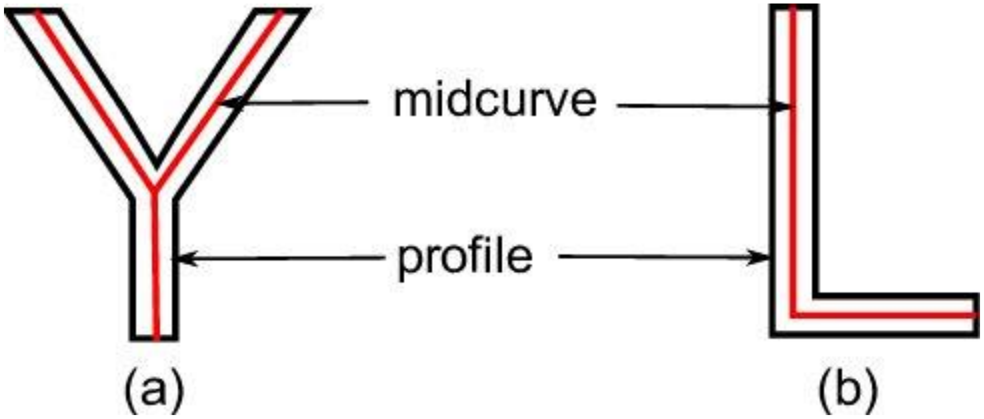
MIDCURVE BY NEURAL NETWORK

Midcurve : The Problem

- **Goal**: Given a 2D closed shape (closed polygon) find its midcurve (polyline, closed or open)
- **Input**: set of points or set of connected lines, non-intersecting, simple, convex, closed polygon
- **Output**: another set of points or set of connected lines, open/branched polygons possible

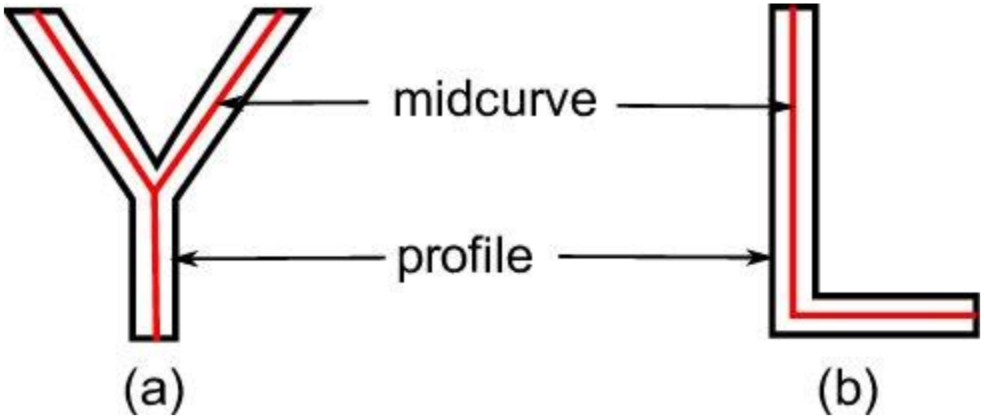
Midcurve == Dimension Reduction

- Like PCA (Principal Component Analysis), wish to find Principal curve
- That ‘represents’ the original profile shape



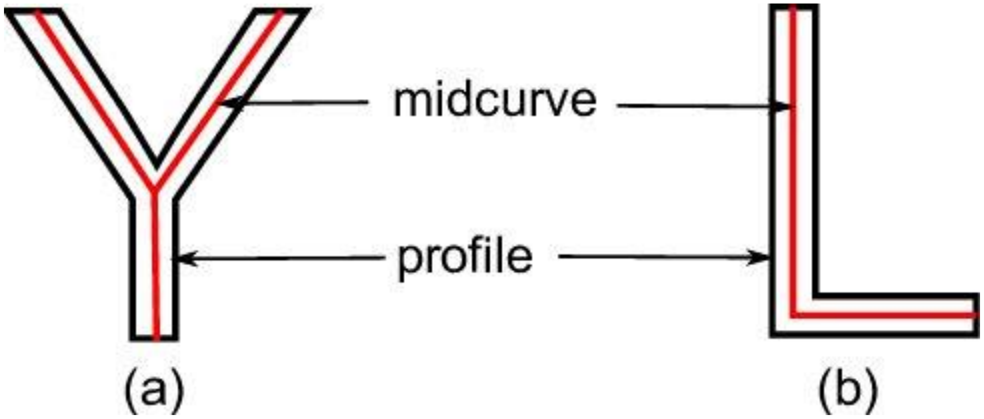
Midcurve == Translation

- Left side (input): 2D Sketch Profile
- Right Side (output): 1D Midcurve
- Sequence 2 Sequence problem



Midcurve using Encoder Decoder

- Its not Auto-Encoder as Input and Output are different
- Its not fixed size i/o as Input and Output sizes are different



Variable Size Encoder Decoder

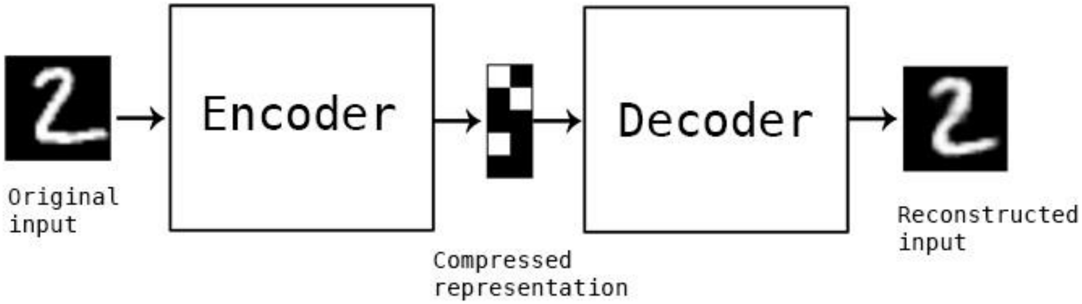
- Typically, variable size gets converted to fixed size by Padding.
- That’s OK for Machine Translations where padding values like “-1” can be added along with other words (vectors or indices)
- But in Geometry, its not OK.
- Because any value can represent a Valid Input, even though we don’t want it to be the input.

In the mean time

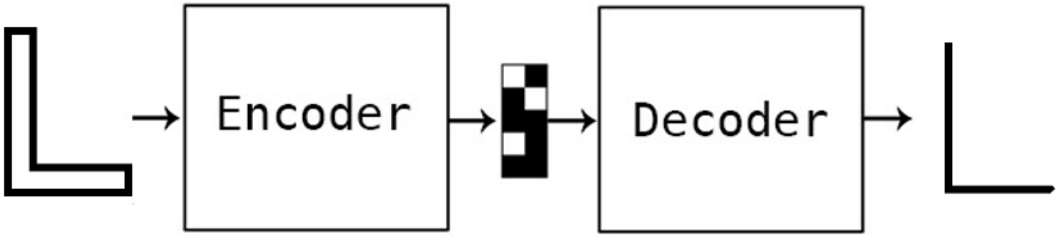
- Till we get good variable size encoder decoder network…
- Decided to convert this Sequence 2 Sequence problem as Image 2 Image problem.
- Input: Black & White Image of 2D profile
- Output: Black & White Image of 1D midcurve

Updated Problem

- Image Encoder Decoder



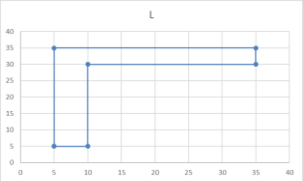
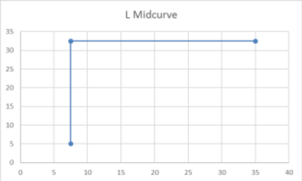
- Can be used for



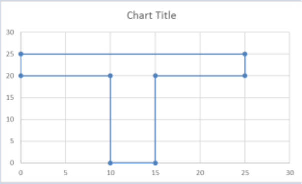
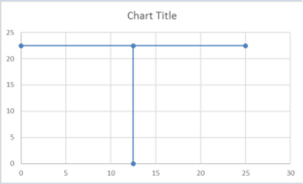
DATA PREPARATION

Data

- Original input and output are in the form of polylines, meaning a list of points, each having x,y coordinates

Profile Data	Profile Picture	Midcurve Data	Midcurve Picture
5.0 5.0 10.0 5.0 10.0 30.0 35.0 30.0 35.0 35.0 5.0 35.0		7.5 5.0 7.5 32.5 35.0 32.5 7.5 32.5	

Data

Profile Data	Profile Picture	Midcurve Data	Midcurve Picture
0 25.0 25.0 25.0 25.0 20.0 15.0 20.0 15.0 0 10.0 0 10.0 20.0 0 20.0		12.5 0 12.5 22.5 25.0 22.5 0 22.5	

- For each shape, we have this pair of input and output. That’s it.
- We need to start with these few samples only

Augmentation

- Such few profile shapes, are just not enough for Neural Networks to train.
- Need more with as much diversity as possible.
- Will need to artificially augment data with transformations, like pan, rotate, mirror, etc.
- All needs to be automatically, programmatically

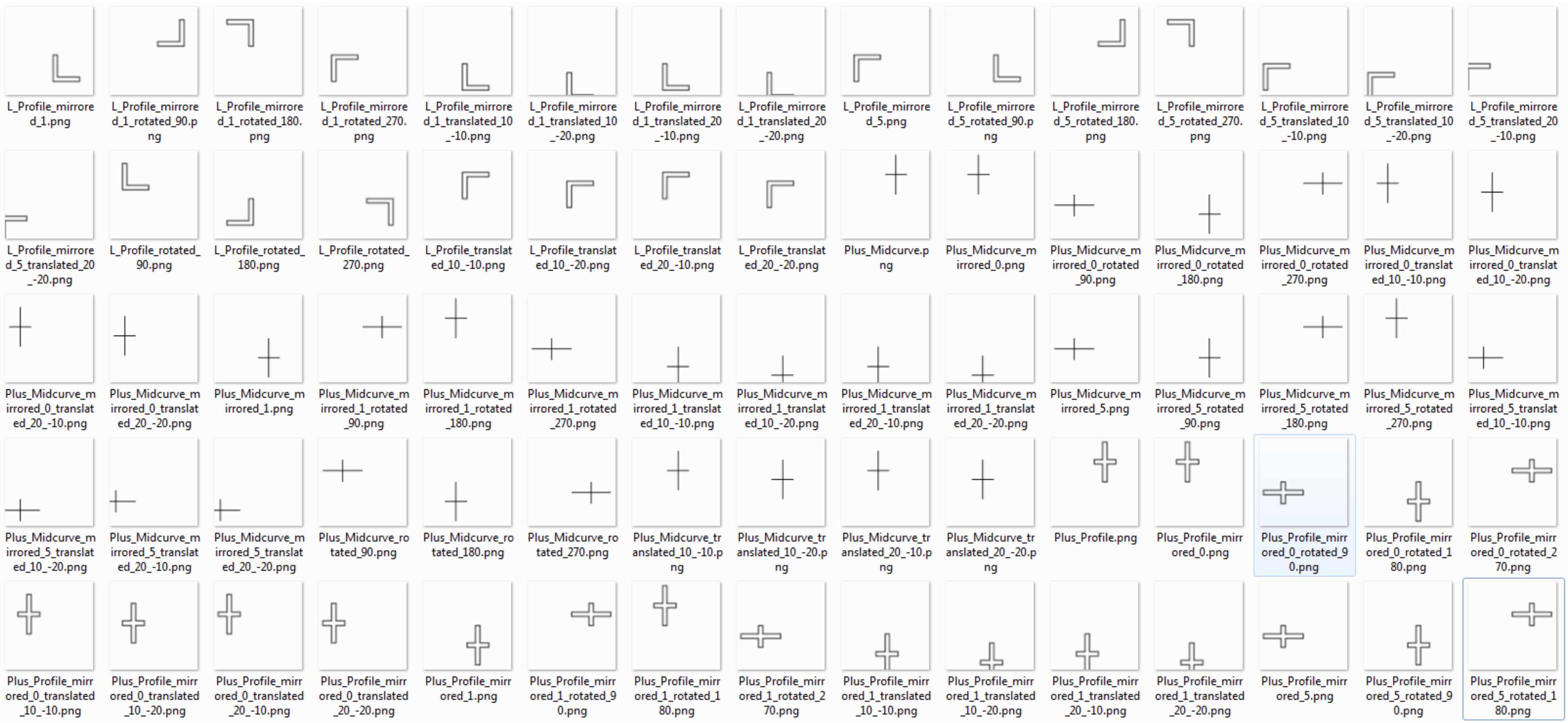
Geometry to Image

- Raw input data is in the Vector format
- Converted it to fixed size (100x100) image by rasterization of drawSVG library.
- This image was then transformed using various operations and saved.

Data Range

- Input shapes: I, L, Plus, T
- Operations:
  - Translated
  - Rotated
  - Mirrored
  - Mirrored Translated
  - Mirrored Rotated
- Total: 896 images (still less, but not bad)

Training Data Samples



L_Profile_mirrored_1.png
L_Profile_mirrored_1_rotated_90.png
L_Profile_mirrored_1_rotated_180.png
L_Profile_mirrored_1_rotated_270.png
L_Profile_mirrored_1_translated_10_-10.png
L_Profile_mirrored_1_translated_10_-20.png
L_Profile_mirrored_1_translated_20_-10.png
L_Profile_mirrored_1_translated_20_-20.png
L_Profile_mirrored_5.png
L_Profile_mirrored_5_rotated_90.png
L_Profile_mirrored_5_rotated_180.png
L_Profile_mirrored_5_rotated_270.png
L_Profile_mirrored_5_translated_10_-10.png
L_Profile_mirrored_5_translated_10_-20.png
L_Profile_mirrored_5_translated_20_-10.png
L_Profile_mirrored_5_translated_20_-20.png
L_Profile_rotated_90.png
L_Profile_rotated_180.png
L_Profile_rotated_270.png
L_Profile_translated_10_-10.png
L_Profile_translated_10_-20.png
L_Profile_translated_20_-10.png
L_Profile_translated_20_-20.png
Plus_Midcurve.png
Plus_Midcurve_mirrored_0.png
Plus_Midcurve_mirrored_0_rotated_90.png
Plus_Midcurve_mirrored_0_rotated_180.png
Plus_Midcurve_mirrored_0_rotated_270.png
Plus_Midcurve_mirrored_0_translated_10_-10.png
Plus_Midcurve_mirrored_0_translated_10_-20.png
Plus_Midcurve_mirrored_0_translated_20_-10.png
Plus_Midcurve_mirrored_0_translated_20_-20.png
Plus_Midcurve_mirrored_1.png
Plus_Midcurve_mirrored_1_rotated_90.png
Plus_Midcurve_mirrored_1_rotated_180.png
Plus_Midcurve_mirrored_1_rotated_270.png
Plus_Midcurve_mirrored_1_translated_10_-10.png
Plus_Midcurve_mirrored_1_translated_10_-20.png
Plus_Midcurve_mirrored_1_translated_20_-10.png
Plus_Midcurve_mirrored_1_translated_20_-20.png
Plus_Midcurve_mirrored_5.png
Plus_Midcurve_mirrored_5_rotated_90.png
Plus_Midcurve_mirrored_5_rotated_180.png
Plus_Midcurve_mirrored_5_rotated_270.png
Plus_Midcurve_mirrored_5_translated_10_-10.png
Plus_Midcurve_mirrored_5_translated_10_-20.png
Plus_Midcurve_mirrored_5_translated_20_-10.png
Plus_Midcurve_mirrored_5_translated_20_-20.png
Plus_Midcurve_rotated_90.png
Plus_Midcurve_rotated_180.png
Plus_Midcurve_rotated_270.png
Plus_Midcurve_translated_10_-10.png
Plus_Midcurve_translated_10_-20.png
Plus_Midcurve_translated_20_-10.png
Plus_Midcurve_translated_20_-20.png
Plus_Profile.png
Plus_Profile_mirrored_0.png
Plus_Profile_mirrored_0_rotated_90.png
Plus_Profile_mirrored_0_rotated_180.png
Plus_Profile_mirrored_0_rotated_270.png
Plus_Profile_mirrored_0_translated_10_-10.png
Plus_Profile_mirrored_0_translated_10_-20.png
Plus_Profile_mirrored_0_translated_20_-10.png
Plus_Profile_mirrored_0_translated_20_-20.png
Plus_Profile_mirrored_1.png
Plus_Profile_mirrored_1_rotated_90.png
Plus_Profile_mirrored_1_rotated_180.png
Plus_Profile_mirrored_1_rotated_270.png
Plus_Profile_mirrored_1_translated_10_-10.png
Plus_Profile_mirrored_1_translated_10_-20.png
Plus_Profile_mirrored_1_translated_20_-10.png
Plus_Profile_mirrored_1_translated_20_-20.png
Plus_Profile_mirrored_5.png
Plus_Profile_mirrored_5_rotated_90.png
Plus_Profile_mirrored_5_rotated_180.png

MIDCURVE NEURAL NETWORK

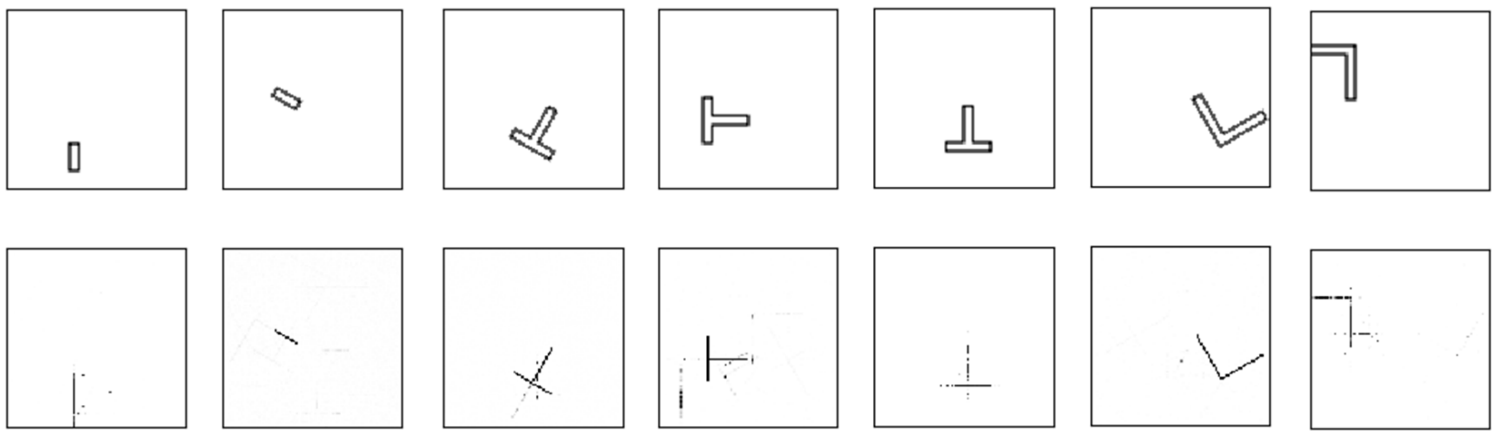
Various Architectures

- Tried various architectures like:
  - Simple Encoder Decoder (one layer each)
  - Dense Encoder Decoder
  - Convolutional Encoder Decoder
  - Pix2Pix
- With the given (small) data, Simple Encoder Decoder worked best amongst all the options.

Simple Encoder Decoder

```
input_img = Input(shape=(input_dim,))

encoded = Dense(encoding_dim,
activation='relu',activity_regularizer=regularizers.l1(10e-5))(input_img)
decoded = Dense(input_dim, activation='sigmoid')(encoded)

autoencoder = Model(input_img, decoded)

encoder = Model(input_img, encoded)
encoded_input = Input(shape=(encoding_dim,))
decoder_layer = autoencoder.layers[-1]
decoder = Model(encoded_input, decoder_layer(encoded_input))

autoencoder.compile(optimizer='adadelta', loss='binary_crossentropy')
```

Results



- Not very perfect but encouraging
- NN is correct with
  - The location (bounding box)
  - Dimension Reduction is seen
- But, still some stray points and misses

What Next?

- More Network Architectures
- Sequence-to-Sequence based approaches, taking closed thin polygon as input and polyline as output
- Extending to 3D, ie Midsurface

END NOTES

Summary

- Various applications need lower dimensional representation of shapes.
- Midcurve is one- dimensional(1D) representation of a two-dimensional (2D) planar shape.
- Used in animation, shape matching, retrieval, finite element analysis, etc.

Summary

- Approaches: Thinning, Medial Axis Transform (MAT), Chordal Axis Transform (CAT), Straight Skeletons, etc., all of which are rule-based.
- Proposing a novel method called MidcurveNN which uses Encoder-Decoder neural network for computing midcurve from images of 2D thin polygons in supervised learning manner.

Summary

- This dimension reduction transformation from input 2D thin polygon image to output 1D midcurve image is learnt by the neural network,
- Which can then be used to compute midcurve of an unseen 2D thin polygonal shape.

Thank you

yogeshkulkarni@yahoo.com