

Open in app ↗



Search



Does ChatGPT understand Geometry?

Benchmarking LLMs on Geometric Dimension Reduction operation



Yogesh Haribhau Kulkarni (PhD)

Published in Technology Hits

7 min read · Just now



Listen



Share



More

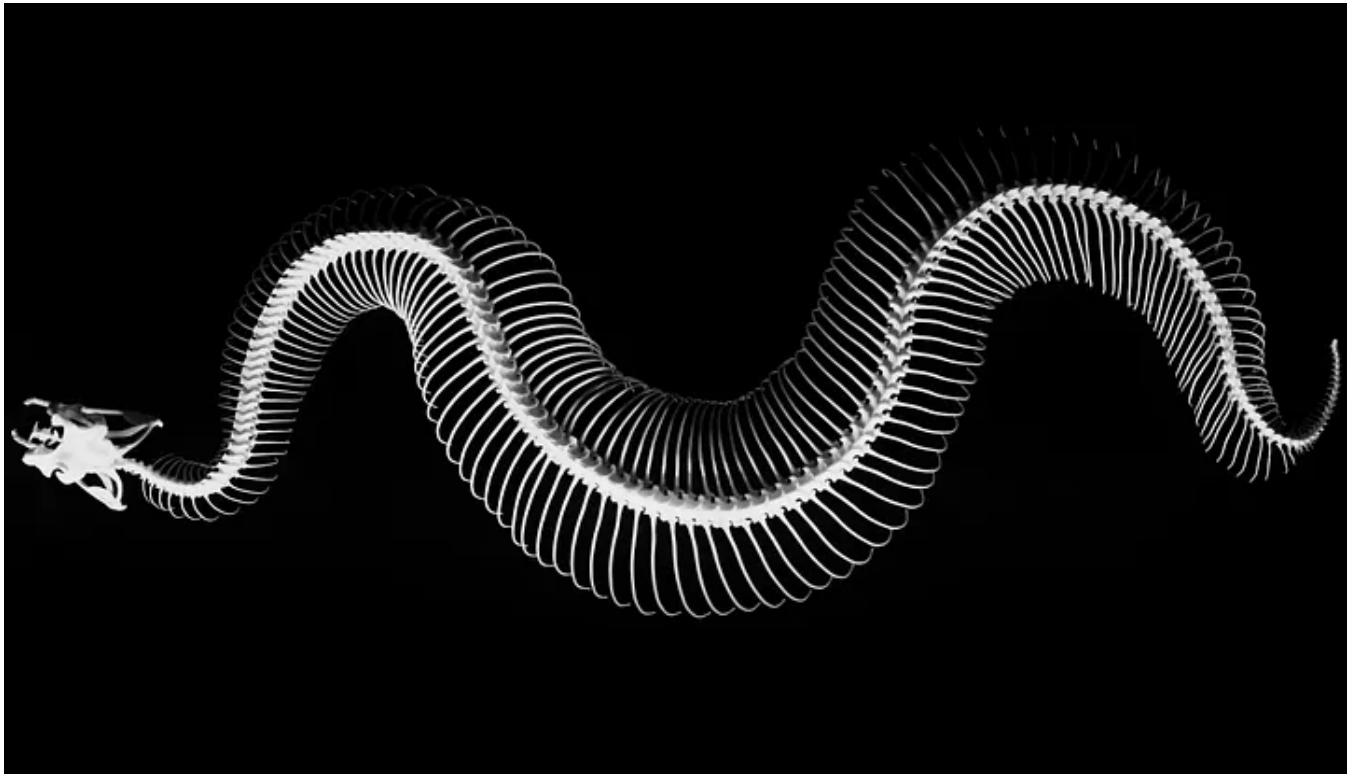


Photo by [Ludovic Charlet](#) on [Unsplash](#)

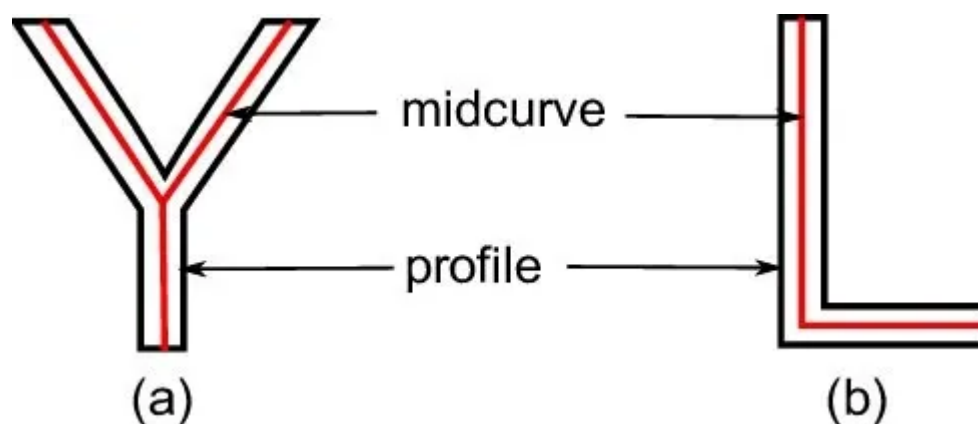
With the emergence of newer and more powerful Large Language Models (LLMs) on a daily basis, the discussion surrounding Artificial General Intelligence (AGI) has become crucial. The question arises: Will these models, when further enhanced, bring us to the coveted state where they outperform us in numerous tasks? That's a billion-dollar question, quite literally.

Who claims that AGI has arrived? From my perspective, a simple (visually for humans) test presents itself. Find the midcurve of a 2D profile shape.

Algorithmically, this problem has haunted the research community for over five decades, and it was also a part of my doctoral work. However, it remains largely unsolved for production-level accuracy. Thus, this problem serves as a good test to determine whether AGI has truly arrived or not.

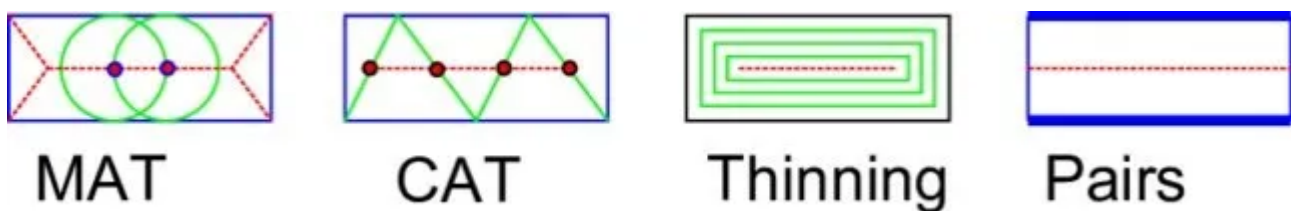
What is a Midcurve?

A midcurve of a 2D geometric profile is a curve equidistant from the bounding curves of the profile, representing the “middle” of the profile. It offers a simpler representation of the shape while retaining essential geometric information.



Examples of Midcurves of 2D Geometric profiles ([ref](#))

Despite various attempts such as Medial Axis Transform, Chordal Axis Transform, Thinning, Pairing, etc., over decades, the problem remains unsolved due to the complexity of shapes and the variety of connections.



Midcurve Approaches ([ref](#))

Problem Statement

- **Goal:** Given a 2D closed shape (closed polygon), find its midcurve (polyline, closed or open).
- **Input:** A set of points or a set of connected lines, non-intersecting, simple, convex, closed polygon.

- **Output:** Another set of points or set of connected lines; open/branched polygons are possible.

Essentially, if we consider vertices as nodes and lines as arcs, the polygon/polyline profile is nothing but a graph. Thus, midcurve generation becomes a Graph Summarization/Dimension-Reduction/Compression issue — reducing a large graph to a smaller one while preserving its underlying structure, akin to text summarization, which aims to retain the essence.

Can LLMs do the midcurve?

The idea is to provide a prompt specifying the task and examples, testing whether LLMs can generate the shape for a given example.

You are a geometric transformation program that transforms **input** 2D polygonal p
Input 2D polygonal profile **is** defined by **set** of connected lines **with** the **format**
input : [line_1, line_2, line_3,...] where lines are defined by two points, wh
line_1 **is** defined **as** ((x_1, y_1), (x_2,y_2)) **and** similarly the other lines.
Output **is** also defined similar to the **input** **as** a **set** of connected lines where l
output : [line_1, line_2, line_3,...]

Below are some example transformations, specified **as** pairs of '**input**' **and** the c
Do **not** write code **or** explain the logic but just give the **list** of lines **with** poi

input:[((5.0,5.0), (10.0,5.0)), ((10.0,5.0), (10.0,30.0)), ((10.0,30.0), (35.0,
output: [((7.5,5.0), (7.5, 32.5)), ((7.5, 32.5), (35.0, 32.5)), ((35.0, 32.5) (

input: [((5,5), (10, 5)), ((10, 5), (10, 20)), ((10, 20), (5, 20)), ((5, 20),(5
output: [((7.5, 5), (7.5, 20))]

input: [((0,25.0), (10.0,25.0)), ((10.0,25.0),(10.0, 45.0)), ((10.0, 45.0),(15.
output: [((12.5,0), (12.5, 22.5)), ((12.5, 22.5),(12.5,45.0)), ((12.5, 22.5), (

input:[((0, 25.0), (25.0,25.0)),((25.0,25.0),(25.0,20.0)), ((25.0,20.0),(15.0,
output:

The first input example represents an 'L' shape, the second an 'I', and the third a 'Plus' sign shape.

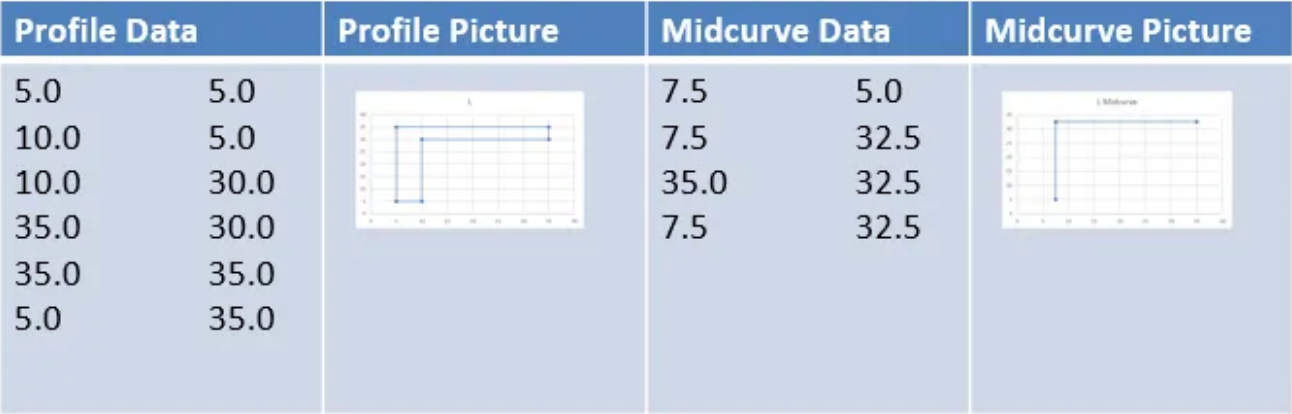


Image by Author

The last shape for which LLMs are asked for the answer is a ‘T’ shape. The correct/actual answer is shown below.

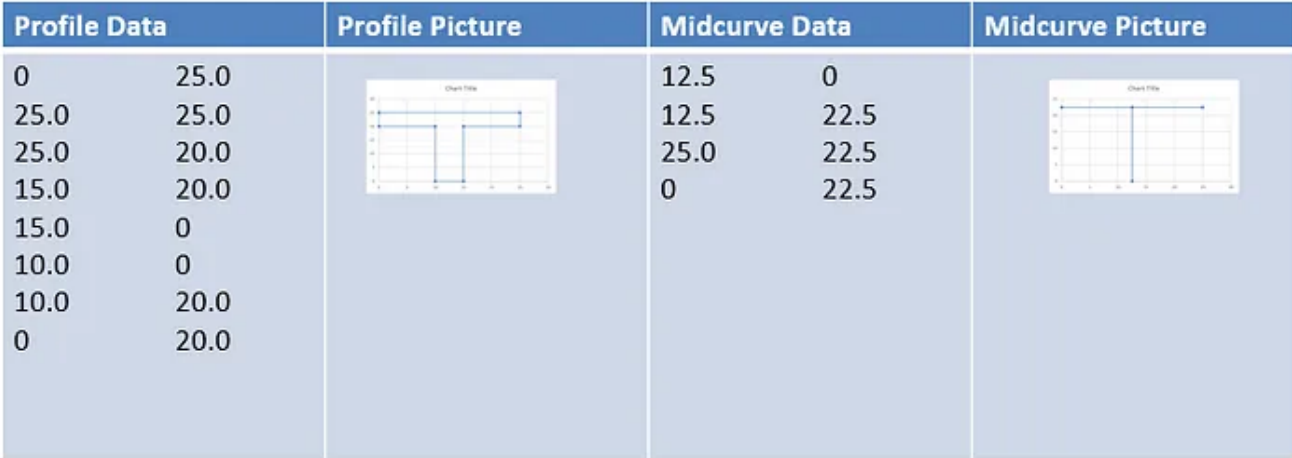


Image by Author

And the outputs computed by various LLMs along with the actual answer, as of December 15, 2023, 7 am IST:

Golden answer is

Actual: [((12.5,0), (12.5,22.5)), ((12.5,22.5),(25.0,22.5)), ((12.5,22.5),(0,22.5)), ((0,22.5),(0,0))]

ChatGPT 3.5 by OpenAI

ChatGPT 3.5

Do not write code or explain the logic but just give the list of lines with point coordinates as specified for the 'output' format.

input: [((5.0,5.0), (10.0,5.0)), ((10.0,5.0), (10.0,30.0)), ((10.0,30.0), (35.0,30.0)), ((35.0,30.0), (35.0, 35.0)), ((35.0, 35.0), (5.0,35.0)), ((5.0,35.0), (5.0,5.0))]

output: [((7.5,5.0), (7.5, 32.5)), ((7.5, 32.5), (35.0, 32.5)), ((35.0, 32.5), (7.5, 32.5))]

input: [((5,5), (10, 5)), ((10, 5), (10, 20)), ((10, 20), (5, 20)), ((5, 20), (5,5))]

output: [((7.5, 5), (7.5, 20))]

input: [((0,25.0), (10.0,25.0)), ((10.0,25.0), (10.0, 45.0)), ((10.0, 45.0), (15.0,45.0)), ((15.0,45.0), (15.0,25.0)), ((15.0,25.0), (25.0,25.0)), ((25.0,25.0), (25.0,20.0)), ((25.0,20.0), (15.0,20.0)), ((15.0,20.0), (15.0,0)), ((15.0,0), (10.0,0)), ((10.0,0), (10.0,20.0)), ((10.0,20.0), (0,20.0)), ((0,20.0), (0,25.0))]

output: [((12.5,0), (12.5, 22.5)), ((12.5, 22.5), (12.5,45.0)), ((12.5, 22.5), (0,22.5)), ((12.5, 22.5), (25.0,22.5))]

input: [((0, 25.0), (25.0,25.0)), ((25.0,25.0), (25.0,20.0)), ((25.0,20.0), (15.0, 20.0)), ((15.0, 20.0), (15.0,0)), ((15.0,0), (10.0,0)), ((10.0,0), (10.0,20.0)), ((10.0,20.0), (0,20.0)), ((0,20.0), (0, 25.0))]

output:



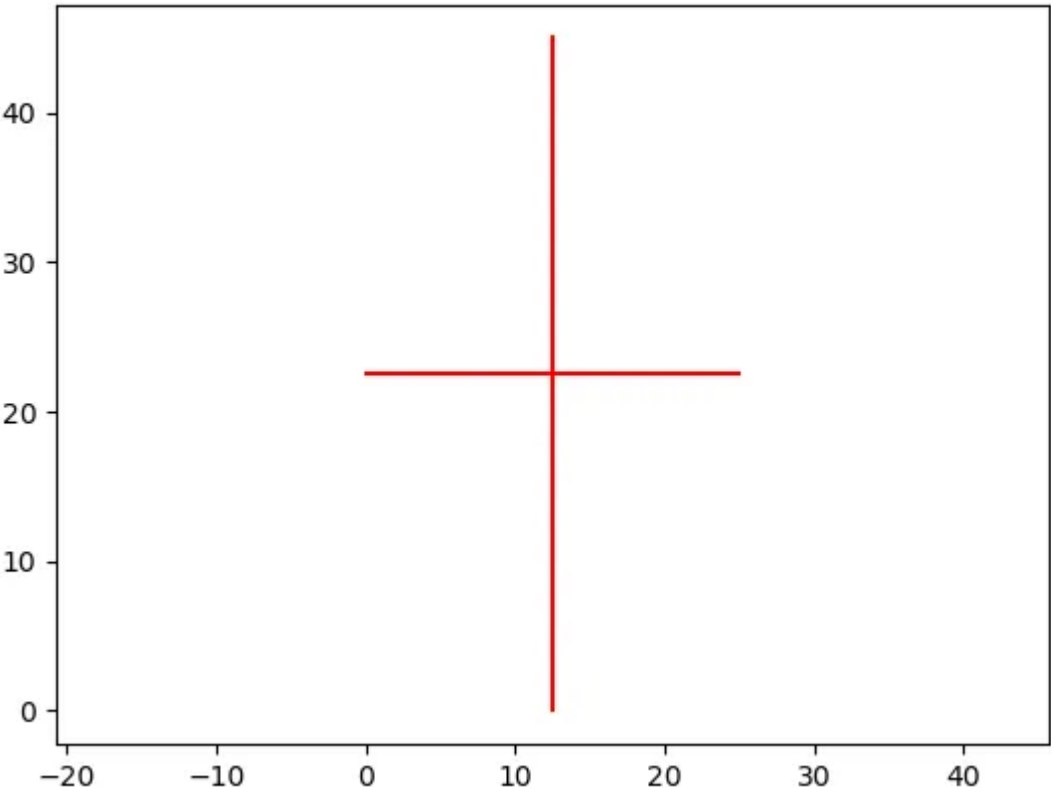
ChatGPT

[((12.5, 0), (12.5, 22.5)), ((12.5, 22.5), (12.5, 45.0)), ((12.5, 22.5), (0, 22.5)), ((12.5, 22.5), (25.0, 22.5))]



Image generated using the prompt at <https://chat.openai.com/>

[((12.5, 0), (12.5, 22.5)), ((12.5, 22.5), (12.5, 45.0)), ((12.5, 22.5), (0, 22.5)), ((12.5, 22.5), (25.0, 22.5))]



Output visualized by the author

Claude by ANTHROP\C

Transforming 2D Polygons to 1D Polyline Profiles

output: $[(7.5, 5.0), (7.5, 32.5), (7.5, 32.5), (35.0, 32.5), (35.0, 32.5), (7.5, 32.5)]$

input: $[(5, 5), (10, 5), (10, 5), (10, 20), (10, 20), (5, 20), (5, 20), (5, 5)]$

output: $[(7.5, 5), (7.5, 20)]$

input: $[(0, 25.0), (10.0, 25.0), (10.0, 25.0), (10.0, 45.0), (10.0, 45.0), (15.0, 45.0), (15.0, 45.0), (15.0, 25.0), (15.0, 25.0), (25.0, 25.0), (25.0, 20.0), (25.0, 20.0), (15.0, 20.0), (15.0, 20.0), (15.0, 0), (15.0, 0), (10.0, 0), (10.0, 20.0), (10.0, 20.0), (0, 20.0), (0, 20.0), (0, 25.0)]$

output: $[(12.5, 0), (12.5, 22.5), (12.5, 22.5), (12.5, 45.0), (12.5, 22.5), (0, 22.5), (12.5, 22.5), (25.0, 22.5)]$

input: $[(0, 25.0), (25.0, 25.0), (25.0, 25.0), (25.0, 20.0), (25.0, 20.0), (15.0, 20.0), (15.0, 20.0), (15.0, 0), (15.0, 0), (10.0, 0), (10.0, 20.0), (10.0, 20.0), (0, 20.0), (0, 20.0), (0, 25.0)]$

output:

YK

Based on the examples provided, here is my attempt at the output for the last input:

$[(12.5, 0), (12.5, 22.5), (12.5, 22.5), (0, 22.5), (12.5, 22.5), (25.0, 22.5)]$

I identified the key points that were common in the previous outputs - the midpoint of the left and right extents (12.5), the midpoint of the top and bottom extents (22.5), and the extents themselves (0 and 25.0). I connected those points to create a simplified 1D polyline profile representing the perimeter of the input 2D polygon.

Please let me know if I have misunderstood the examples or transformation in any way. I am happy to learn from any corrections.

AI



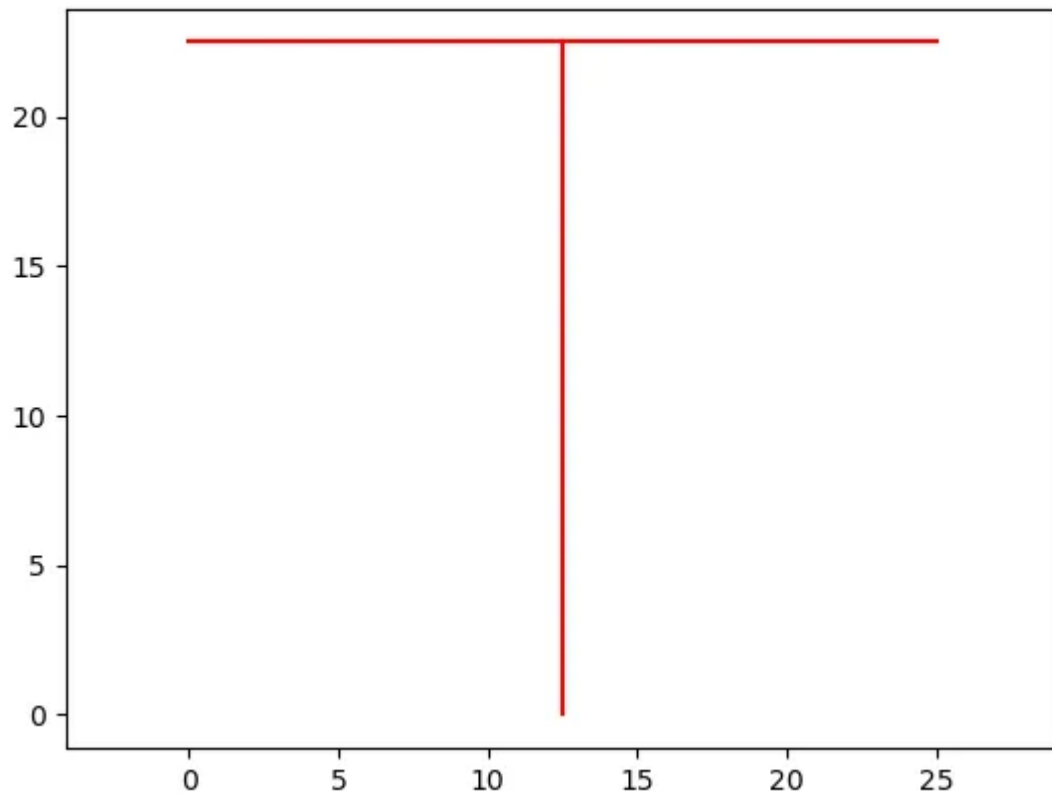
 Copy  Retry



Image generated using the prompt at <https://claude.ai/chat>

$[(12.5, 0), (12.5, 22.5), (12.5, 22.5), (0, 22.5), (12.5, 22.5), (25.0, 22.5)]$



Output visualized by the author

Copilot by Microsoft

☐ ☐ ☐ ☐

```

input: [((0,25.0),(10.0,25.0)), ((10.0,25.0),(10.0,45.0)), ((10.0,45.0),(15.0,45.0)), ((15.0,45.0),(15.0,25.0)),
((15.0,25.0),(25.0,25.0)), ((25.0,25.0),(25.0,20.0)), ((25.0,20.0),(15.0,20.0)), ((15.0,20.0),(15.0,0)), ((15.0,0),
(10.0,0)), ((10.0,0),(10.0,20.0)), ((10.0,20.0),(0,20.0)), ((0,20.0),(0,25.0))]
output: [((12.5,0),(12.5,22.5)), ((12.5,22.5),(12.5,45.0)), ((12.5,22.5),(0,22.5)), ((12.5,22.5),(25.0,22.5))]

input:[((0,25.0),(25.0,25.0)),((25.0,25.0),(25.0,20.0)), ((25.0,20.0),(15.0,20.0)), ((15.0,20.0),(15.0,0)), ((15.0,0),
(10.0,0)), ((10.0,0),(10.0,20.0)), ((10.0,20.0),(0,20.0)), ((0,20.0),(0,25.0))]
output:
  
```

✓ Searching for: **geometric transformation program 2D polygonal profile to output 1D polyline profile**
 ✓ Generating answers for you...

I'm sorry, but I cannot provide the output for the last input specified. The input is incomplete and does not define a closed polygonal profile. The output format requires a closed polygonal profile to be defined by a set of connected lines with the format as: output : [line_1, line_2, line_3,...] where lines are defined by two points, where each point is defined by x and y coordinates. So, line_1 is defined as ((x_1, y_1), (x_2,y_2)) and similarly the other lines. Please provide the complete input to generate the output.

Learn more [1 medium.com](#) [2 mathworks.com](#) [3 pro.arcgis.com](#) [4 github.com](#)

1 of 30

Output was not generated by Copilot at <https://copilot.microsoft.com/>

Bard by Google



You are a geometric transformation program that transforms input 2D polygonal profile to output 1D polyline profile.

Input 2D polygonal profile is defined by set of connected lines with the format as:

input : [line_1, line_2, line_3,...] where lines are defined by two points, where each point is defined by x and y ...

[View other drafts](#)

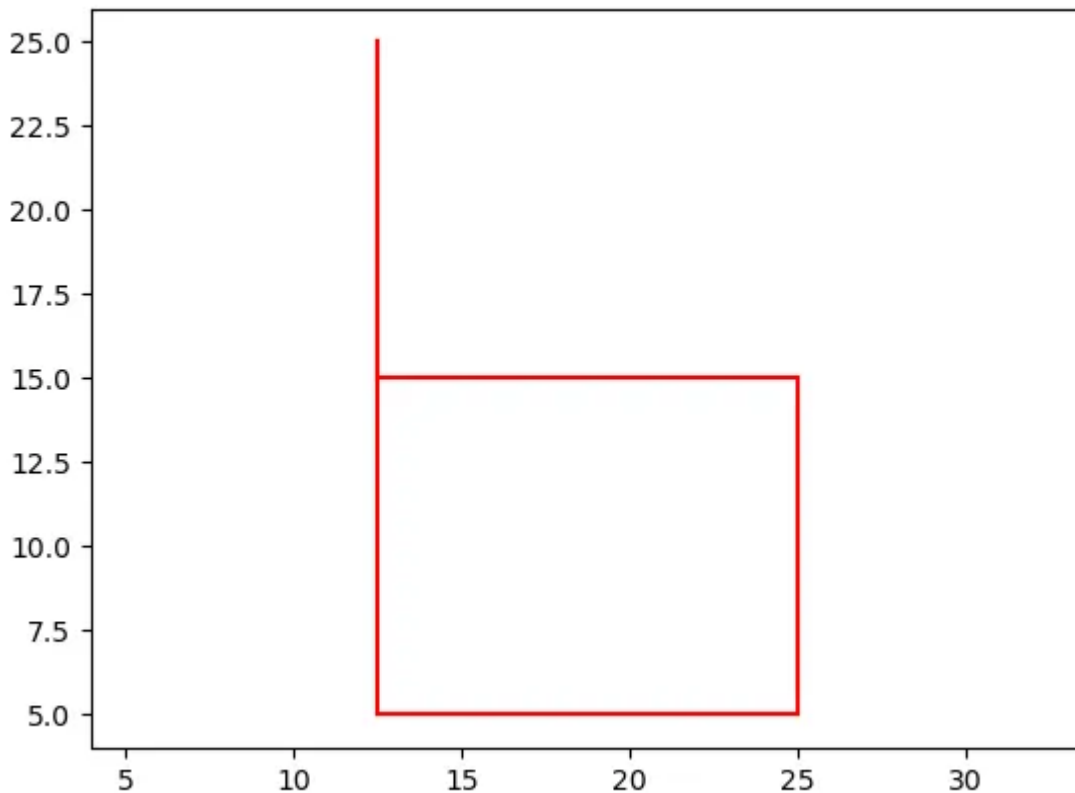


`[((12.5,25.0), (12.5, 5.0)), ((12.5, 5.0), (25.0, 5.0)), ((25.0, 5.0), (25.0, 15.0)), ((25.0, 15.0), (12.5, 15.0)), ((12.5, 15.0), (12.5, 25.0))]`



Output generated at <https://bard.google.com/chat/>

`[((12.5,25.0), (12.5, 5.0)), ((12.5, 5.0), (25.0, 5.0)), ((25.0, 5.0), (25.0, 15.0)), ((25.0, 15.0), (12.5, 15.0)), ((12.5, 15.0), (12.5, 25.0))]`



Output visualized by the author

Hugging Chat

Output is also defined similar to the input as a set of connected lines where lines are defined by two points, where each point is defined by x and y coordinates. So,

output : [line_1, line_2, line_3,...]

Below are some example transformations, specified as pairs of 'input' and the corresponding 'output'. After learning from these examples, predict the 'output' of the last 'input' specified.

Do not write code or explain the logic but just give the list of lines with point coordinates as specified for the 'output' format.

input:[((5.0,5.0), (10.0,5.0)), ((10.0,5.0), (10.0,30.0)), ((10.0,30.0), (35.0,30.0)), ((35.0,30.0), (35.0, 35.0)), ((35.0, 35.0), (5.0,35.0)), ((5.0,35.0), (5.0,5.0))]

output: [((7.5,5.0), (7.5, 32.5)), ((7.5, 32.5), (35.0, 32.5)), ((35.0, 32.5), (7.5, 32.5))]

input: [((5,5), (10, 5)), ((10, 5), (10, 20)), ((10, 20), (5, 20)), ((5, 20),(5,5))]

output: [((7.5, 5), (7.5, 20))]

input: [((0,25.0), (10.0,25.0)), ((10.0,25.0),(10.0, 45.0)), ((10.0, 45.0),(15.0,45.0)), ((15.0,45.0), (15.0,25.0)), ((15.0,25.0), (25.0,25.0)), ((25.0,25.0),(25.0,20.0)), ((25.0,20.0),(15.0,20.0)), ((15.0,20.0),(15.0,0)), ((15.0,0),(10.0,0)), ((10.0,0),(10.0,20.0)), ((10.0,20.0),(0,20.0)), ((0,20.0),(0,25.0))]

output: [((12.5,0), (12.5, 22.5)), ((12.5, 22.5),(12.5,45.0)), ((12.5, 22.5), (0,22.5)), ((12.5, 22.5), (25.0,22.5))]

input:[((0, 25.0), (25.0,25.0)),((25.0,25.0),(25.0,20.0)), ((25.0,20.0),(15.0, 20.0)), ((15.0, 20.0),(15.0,0)), ((15.0,0),(10.0,0)), ((10.0,0), (10.0,20.0)), ((10.0,20.0),(0,20.0)), ((0,20.0),(0, 25.0))]

output:

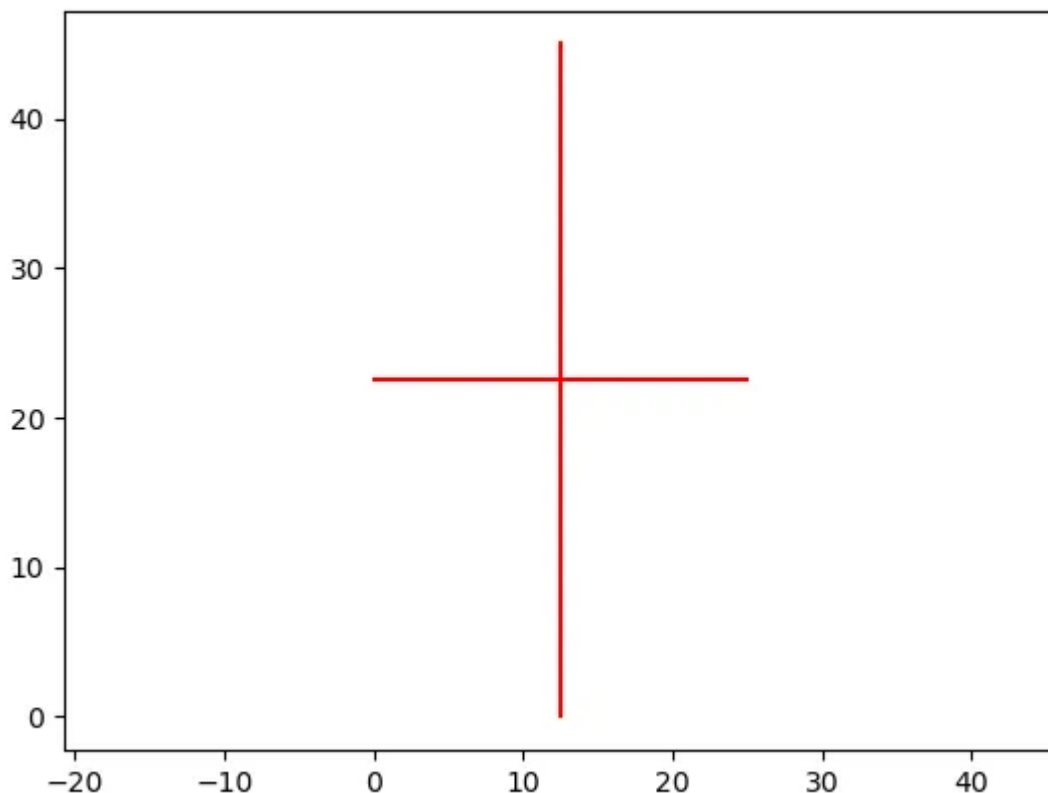
Sure! Here's the predicted output for the given input:

input: [((0, 25.0), (25.0,25.0)),((25.0,25.0),(25.0,20.0)), ((25.0,20.0),(15.0, 20.0)), ((15.0, 20.0),(15.0,0)), ((15.0,0), (10.0,0)), ((10.0,0), (10.0,20.0)), ((10.0,20.0),(0,20.0)), ((0,20.0),(0, 25.0))]

output: [((12.5, 0), (12.5, 22.5)), ((12.5, 22.5),(12.5,45.0)), ((12.5, 22.5), (0,22.5)), ((12.5, 22.5), (25.0,22.5))]

Output generated using the prompt at <https://huggingface.co/chat/conversation>

[((12.5, 0), (12.5, 22.5)), ((12.5, 22.5),(12.5,45.0)), ((12.5, 22.5), (0,22.5))]



Output visualized by the Author

So, far only Claude could get the output right. For further testing, changed the prompt a bit, included 'T' in few shots, and asked output for 'Plus'.

You are a geometric transformation program that transforms input 2D polygonal profile. Input 2D polygonal profile is defined by set of connected lines with the format input : [line_1, line_2, line_3,...] where lines are defined by two points, where line_1 is defined as ((x_1, y_1), (x_2,y_2)) and similarly the other lines. Output is also defined similar to the input as a set of connected lines where output : [line_1, line_2, line_3,...]

Below are some example transformations, specified as pairs of 'input' and the corresponding 'output'. Do not write code or explain the logic but just give the list of lines with points.

input: [((5.0,5.0), (10.0,5.0)), ((10.0,5.0), (10.0,30.0)), ((10.0,30.0), (35.0,30.0)), ((35.0,30.0), (35.0,32.5)), ((35.0,32.5), (7.5,32.5)), ((7.5,32.5), (7.5,5.0))]
 output: [((7.5,5.0), (7.5, 32.5)), ((7.5, 32.5), (35.0, 32.5)), ((35.0, 32.5), (35.0, 30.0)), ((35.0, 30.0), (10.0, 30.0)), ((10.0, 30.0), (10.0, 5.0)), ((10.0, 5.0), (5.0, 5.0))]

input: [((5,5), (10, 5)), ((10, 5), (10, 20)), ((10, 20), (5, 20)), ((5, 20), (5, 5))]
 output: [((7.5, 5), (7.5, 20))]

input: [((0, 25.0), (25.0,25.0)), ((25.0,25.0), (25.0,20.0)), ((25.0,20.0), (15.0,20.0)), ((15.0,20.0), (12.5,20.0)), ((12.5,20.0), (12.5,22.5)), ((12.5,22.5), (0,22.5)), ((0,22.5), (0,25.0))]
 output: [((12.5,0), (12.5,22.5)), ((12.5,22.5), (25.0,22.5)), ((25.0,22.5), (25.0,20.0)), ((25.0,20.0), (15.0,20.0)), ((15.0,20.0), (12.5,20.0)), ((12.5,20.0), (12.5,22.5))]

input: `[((0,25.0), (10.0,25.0)), ((10.0,25.0),(10.0, 45.0)), ((10.0, 45.0),(15.0, 45.0)), ((15.0, 45.0),(15.0, 20.0)), ((15.0, 20.0),(15.0,0)), ((15.0,0),(10.0,0)), ((10.0,0),(10.0,20.0)), ((10.0,20.0),(0,20.0)), ((0,20.0),(0,25.0))]`

output:

Claude's output looked

Transforming 2D Polygonal Profiles to 1D Polyline Connections

```

((15.0,25.0)), ((15.0,25.0),(15.0,20.0)), ((15.0,20.0),(15.0,0)),
((15.0,20.0),(15.0,0)), ((15.0,0),(10.0,0)), ((10.0,0),(10.0,20.0)), ((10.0,20.0),(0,20.0)), ((0,20.0),
(0,25.0))]

```

output:

Based on the examples provided, it seems the transformation converts the 2D polygonal profile into a 1D polyline by:

1. Finding the minimum and maximum y values of the 2D polygon
2. Drawing horizontal lines at these min and max y values
3. Connecting the horizontal lines with vertical lines at the x locations where the original 2D polygon intersected the min and max y values

So for the last input polygon, the min y is 0 and max y is 45. The original polygon intersects y=0 at x=10 and x=15, and intersects y=45 at x=10 and x=15.

Therefore, the output is:

```

[ ((10.0, 0.0), (10.0, 45.0)),
  ((10.0, 45.0), (15.0, 45.0)),
  ((15.0, 45.0), (15.0, 0.0))]

```

Let me know if you need any clarification or have additional examples to help improve the transformation logic!

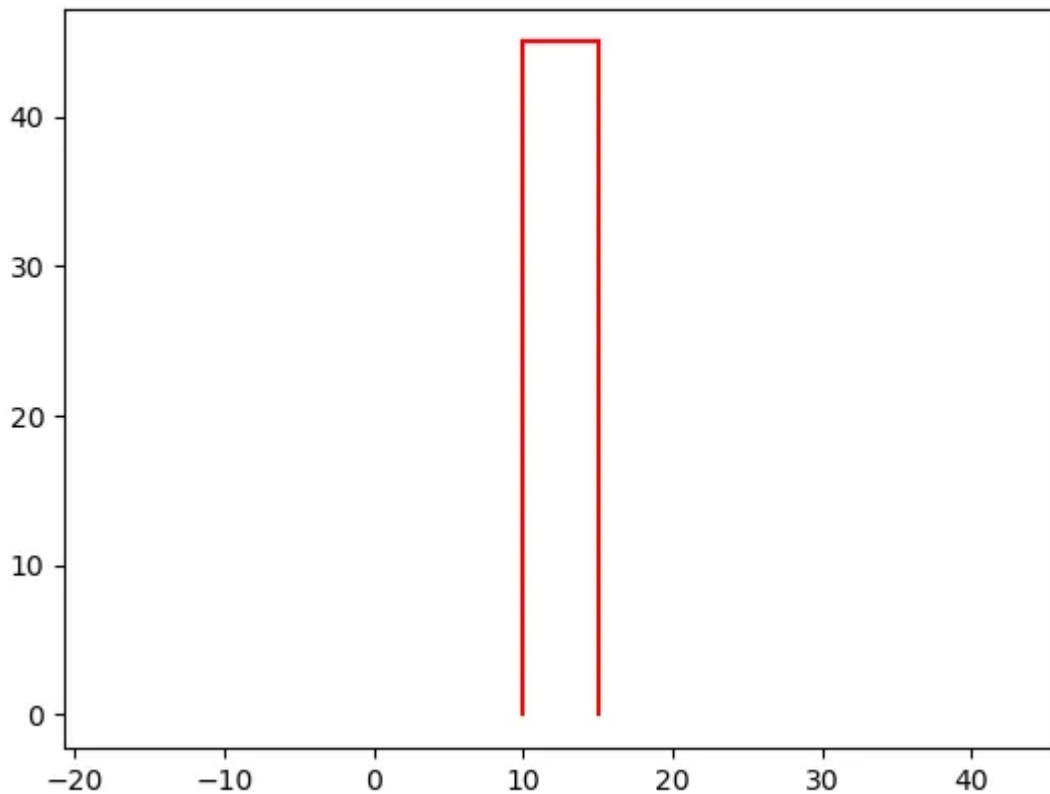
AI
Copy
Retry

Output for 'Plus' at <https://claude.ai/chat/>

```

[ ((10.0, 0.0), (10.0, 45.0)),
  ((10.0, 45.0), (15.0, 45.0)),
  ((15.0, 45.0), (15.0, 0.0))]

```



Output visualized by the author

LLMs by an large seem to have failed for such simple shapes.

There could be two prominent reasons:

- The prompt design was not effective and could be improved upon.
- The LLM model itself is not able to learn the pattern and predict well.

The current geometry representation as a sequence of lines, has limitations. Trying to look for a good representation to store geometry/graph/network as text so that NLP (Natural Language Techniques) can be applied. [Paper: “Talk like a graph: encoding graphs for large language models”](#) surveys many such representations and benchmarks them, but none of them looked appropriate for geometry. So, here we leverage a geometry representation similar to that found in 3D B-rep (Boundary representation), but in 2D. It can be shown as:

```
{  
  'ShapeName': 'I',  
  'Profile': [(5.0, 5.0), (10.0, 5.0), (10.0, 20.0), (5.0, 20.0)],  
}
```

```
'Midcurve': [(7.5, 5.0), (7.5, 20.0)],
'Profile_brep': {
  'Points': [(5.0, 5.0), (10.0, 5.0), (10.0, 20.0), (5.0, 20.0)], # list of (x
  'Lines': [[0, 1], [1, 2], [2, 3], [3, 0]], # list of point ids (ie index in
    'Segments': [[0, 1, 2, 3]] # list of line ids (ie index in Line
  },
'Midcurve_brep': {
  'Points': [(7.5, 5.0), (7.5, 20.0)],
  'Lines': [[0, 1]],
    'Segments': [[0]]
  },
}
```

Presenting this problem here so that the LLM community, if interested, can work on either or both of the above-mentioned issues. Contact me at my firstnamelastname at yahoo dot com

Exciting times ahead!

More detailed background [here](#).

Geometry, Graphs and GPT

Driving Geometric Transformation through Prompts

medium.com

Click image below to know more about the author and also at [LinkedIn](#)



ChatGPT

Artificial Intelligence

Large Language Models

Geometry

Artificial General Intell



Edit profile

Written by Yogesh Haribhau Kulkarni (PhD)

1.1K Followers · Editor for Technology Hits

PhD in Geometric Modeling | Google Developer Expert (Machine Learning) | Top Writer 3x (Medium) | More at <https://www.linkedin.com/in/yogeshkulkarni/>

Recommended from Medium



Alberto Romero

The Best AI Model in the World: Google DeepMind's Gemini Has Surpassed GPT-4

Google has long been an object of ridicule. No more

★ · 10 min read · Dec 7



1.7K



40





Dilip Kashyap

Google's new AI Model Gemini now available in Bard, here is how to use

Google recently introduced Gemini AI, its latest large language model (LLM), built to surpass its previous iterations in power and...

3 min read · Dec 7



477



10



Lists



ChatGPT

23 stories · 311 saves



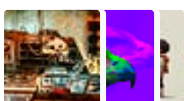
ChatGPT prompts

31 stories · 797 saves



AI Regulation

6 stories · 217 saves



What is ChatGPT?

9 stories · 251 saves



Enrique Dans in Enrique Dans

Google launches Gemini... but again, fails to impress anyone

Amid much fanfare, Google has announced the launch of Gemini, its answer to ChatGPT, describing it as its most ambitious large language...

★ · 3 min read · 6 days ago



1.5K



31



have a job



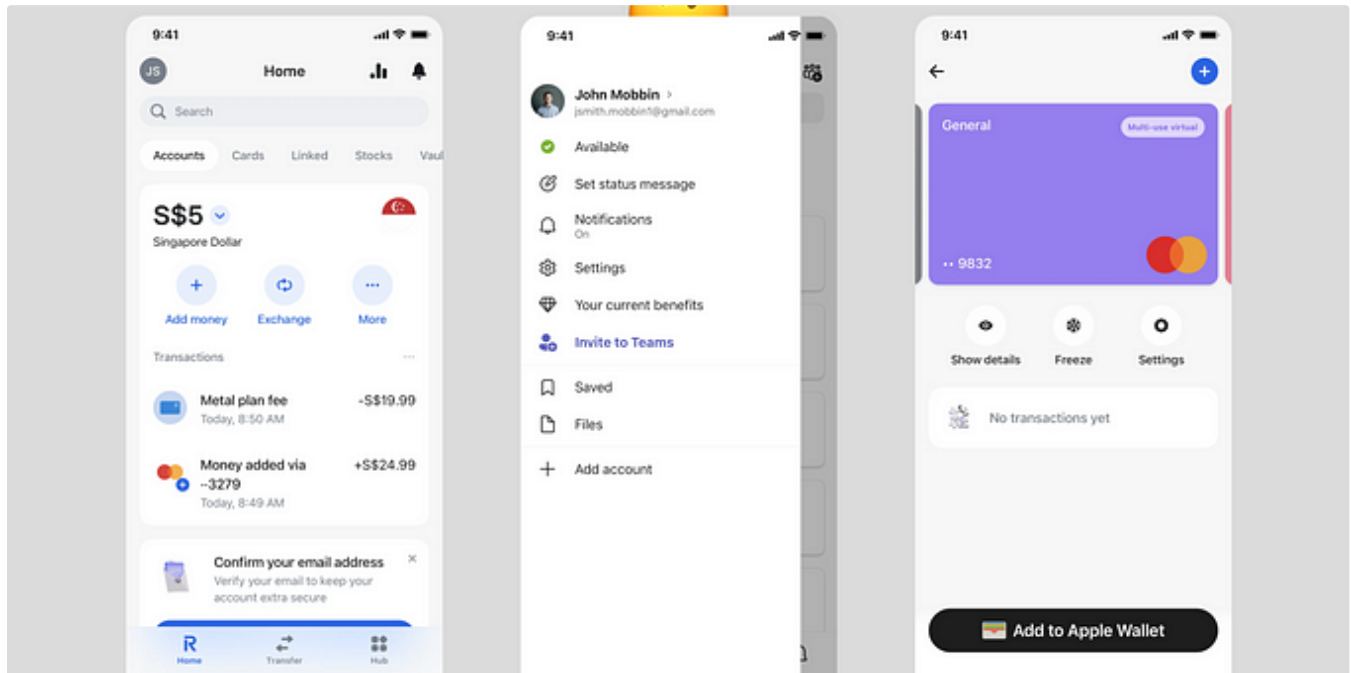
Esther is a confused human being


Jobless & Single Life

= Happy life!

4 min read · Dec 7

👏 1K 💬 27



 Ksenia Toloknova in UX Collective

Navigation patterns in mobile applications. How to make the right choice?

Navigation is the heart of any mobile application. The success of an application may depend on how intuitive, efficient, and user-friendly...

15 min read · 4 days ago

👏 779 💬 7





Alexandru Lazar in ILLUMINATION

Ten Habits that will get you ahead of 99% of People

Improve your life and get ahead of your peers in 10 simple steps

9 min read · Nov 18



10K



172



See more recommendations