Duckify

Week 6 Review

Milestone

For the week 6

Using the **complete pipeline**, starting from an **orchestration interface**, draw **AI-generated contours**, on a duck using **multiple colors with automatical changes**.

Have a **promoting website** for the project

Interface GenAI	☑	Video	☑
Interface Tracing	☑	Montage	☒
Interface Robot	☒	Website	☒

GenAI – Milestones & TODO

- **T2T Improving Quality of results :**
  - *Investigate FreeU* ☑
  - *Finetuning of T2T weights with specific data* ☒

GenAI – Finetuning of T2T

- Exploration of new angle on prompt engineering
  - Secondary objective : Provide data to train model MV-Adapter (aborted)
  - Add external context :
    - "A duck disguised as …"
  - Try difference between short and more descriptive prompt
    - Not significative enough (same quantity of weird texture)
  - Pop-culture (video game)
    - Inconclusive
- Used for tracing stress test (explained later)

GenAI – Prompt examples

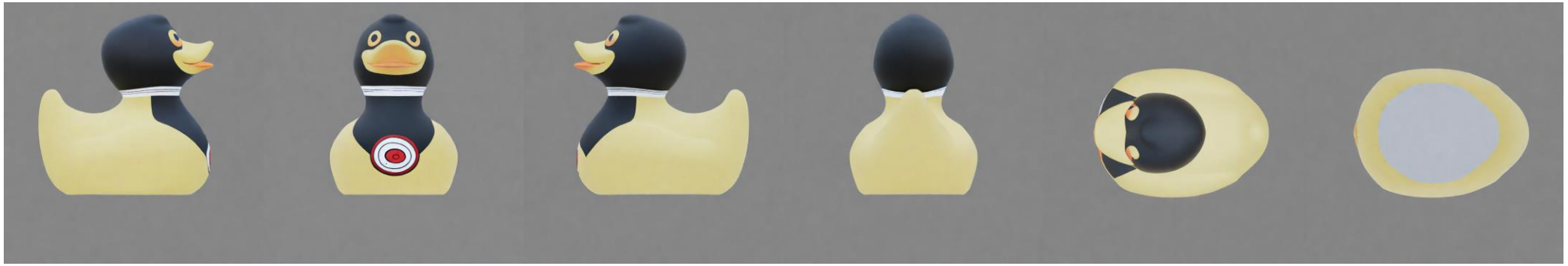
A rubber duck disguised as a police officer with only red yellow blue white and a small

A rubber duck disguised as Mario

A rubber duck disguised as Link from the video game serie The Legend of Zelda

A rubber duck disguised as a cow with white skin and black stain using only black and white

GenAI – Prompt examples



A rubber duck disguised as a ninja of Naruto anime with **sharigan** on his eyes

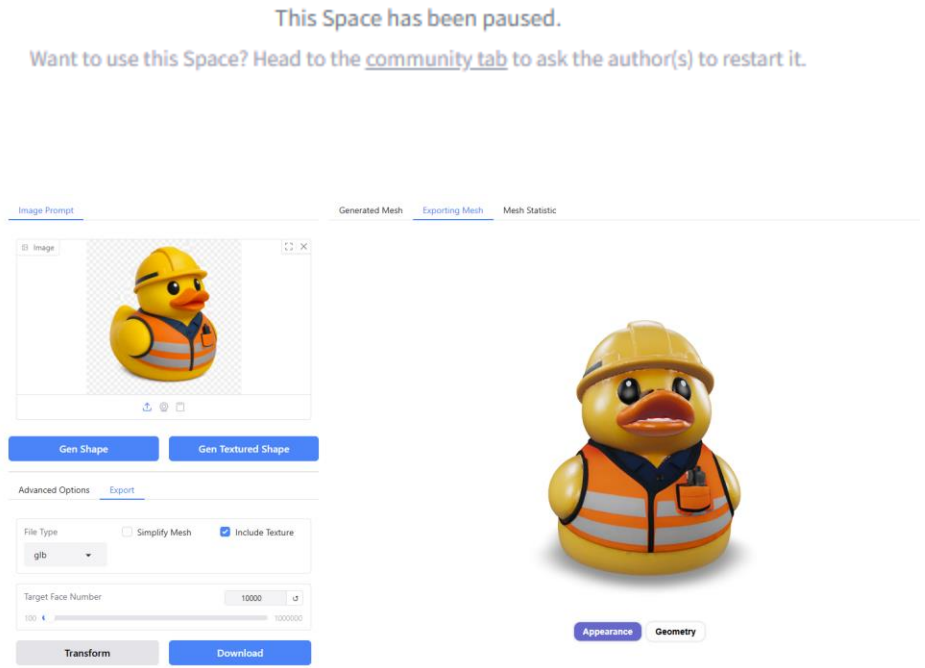
GenAI – Prompt examples

A rubber duck disguised as a pharaon

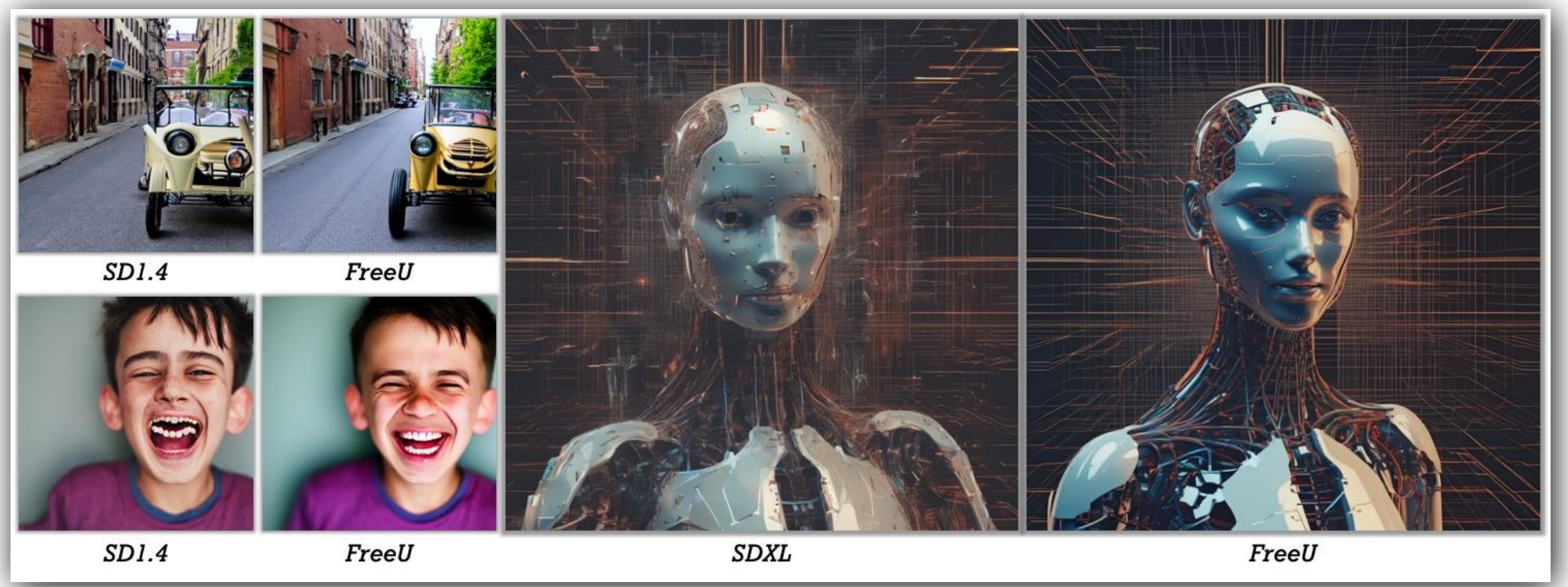
A rubber duck disguised as a geisha

GenAI – Finetuning

- Objective of finetuning
- Debug of training architecture
- Choice and creation of the dataset



FreeU : Free Lunch in Diffusion U-Net



FreeU : Free Lunch in Diffusion U-Net

How does it works ?

- `pipe.enable_freeu(b1=1.3, b2=1.4, s1=0.9, s2=0.2)`

Factors $\mathbf{b_i}$: Texture Smoothing

- **Texture Impact:** Tend to crush micro-details & respect more of the global geometry

Factors $\mathbf{s_i}$: Texture Restoration

- **Texture Impact:** Counteract the smoothing induced by bi by reinjecting micro-details.

Factors limits :

- **b1**: $1 \leq b1 \leq 1.2$
- **b2**: $1.2 \leq b2 \leq 1.6$
- **s1**: $s1 \leq 1$
- **s2**: $s2 \leq 1$

FreeU : Free Lunch in Diffusion U-Net

Results…

- Onedrive – results

FreeU : Free Lunch in Diffusion U-Net

Sources

- github repository
- Paper

Tracing

Work axis this week

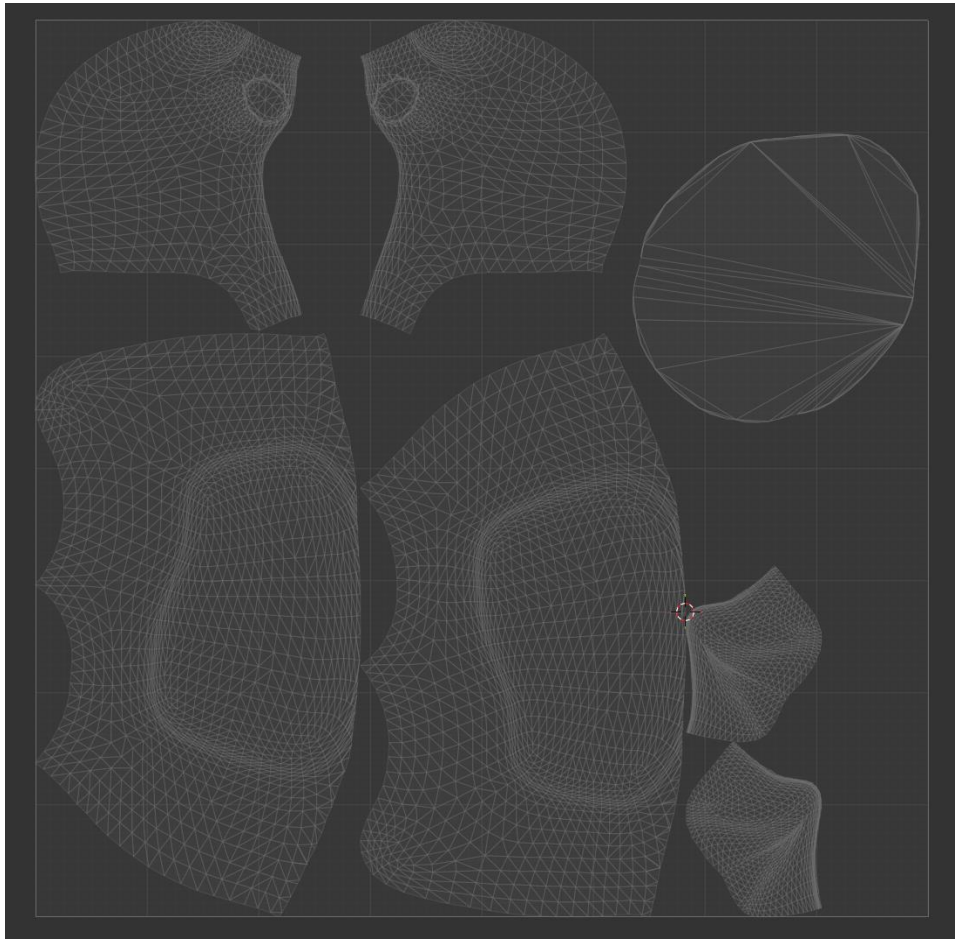
- New UV map disposition — *DONE*
- Reduce points amount per traces — *DONE*
- New tests for team robot — *DONE*
- Fill slicing debug 2 — *DONE*
- Tracing pipeline stress-test — *WIP*

  Work started but more time is needed to complete

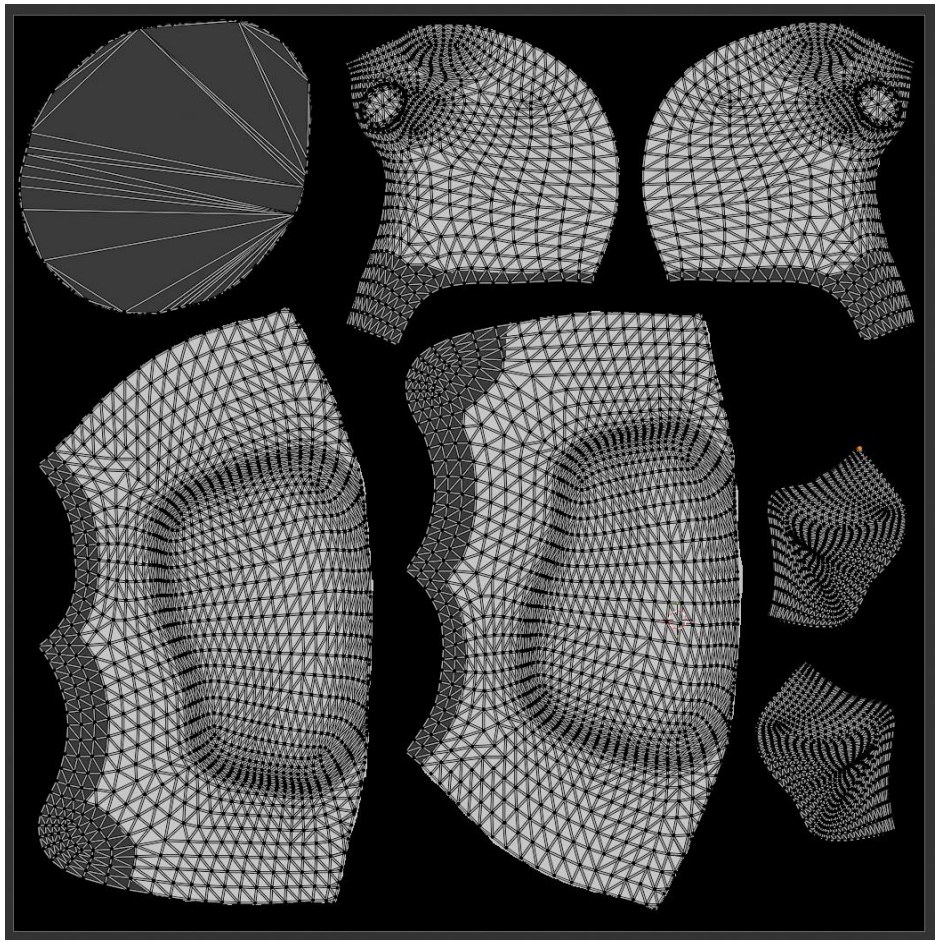
Tracing

New UV map disposition (+ mask update)

Why : *to ensure no multiple points/vertex in same pixel*



before



after

Tracing

Reduce number of points per trace

Why : *(demand from TR) to reduce potential error*

***How* :** *a two steps process*

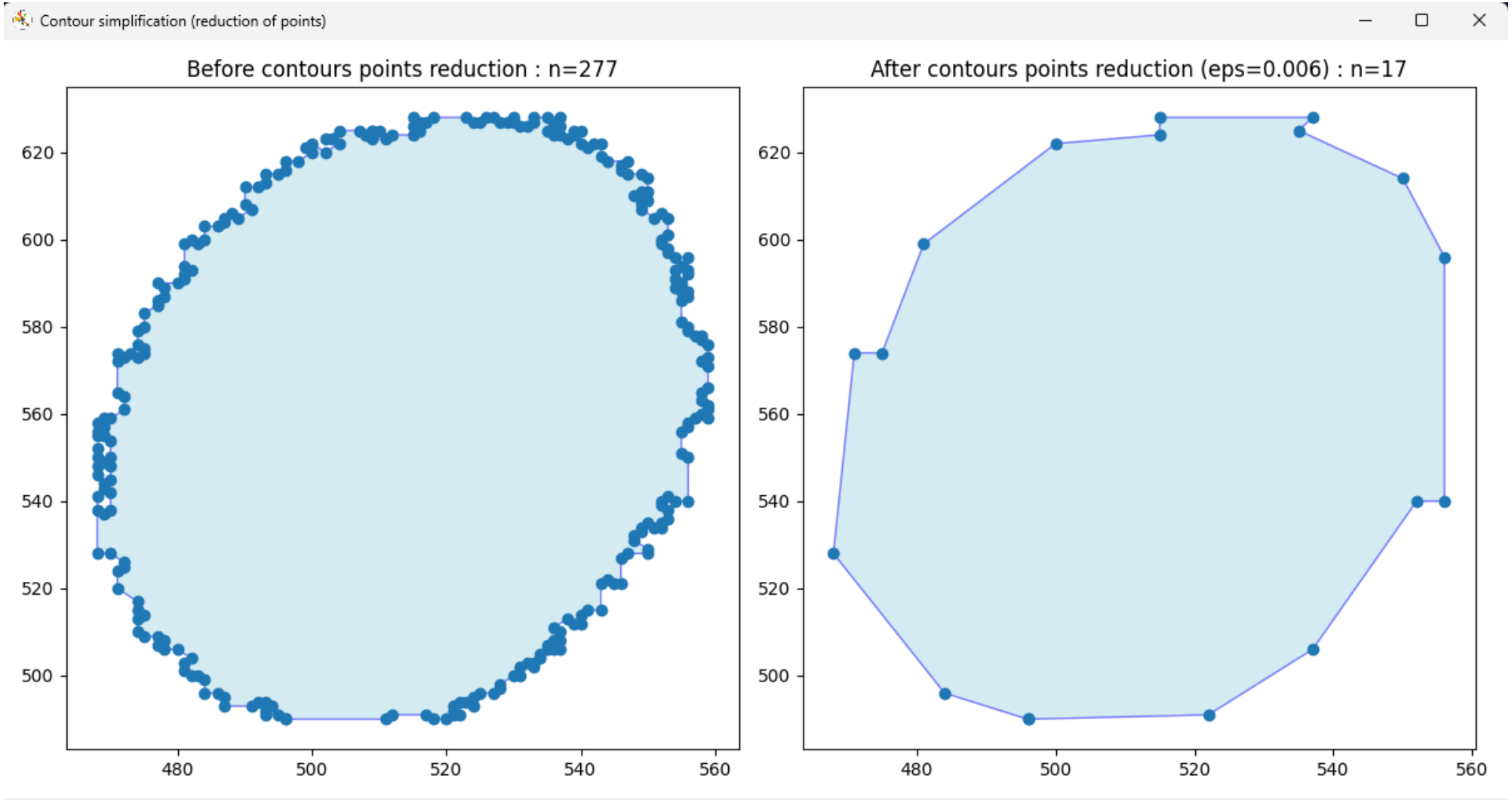
- *Polishing the detected contours*
- *In the trace projection, use an angular threshold (normals between consecutive face) to tolerate faces successions without always adding a point between them*

before

after

Tracing

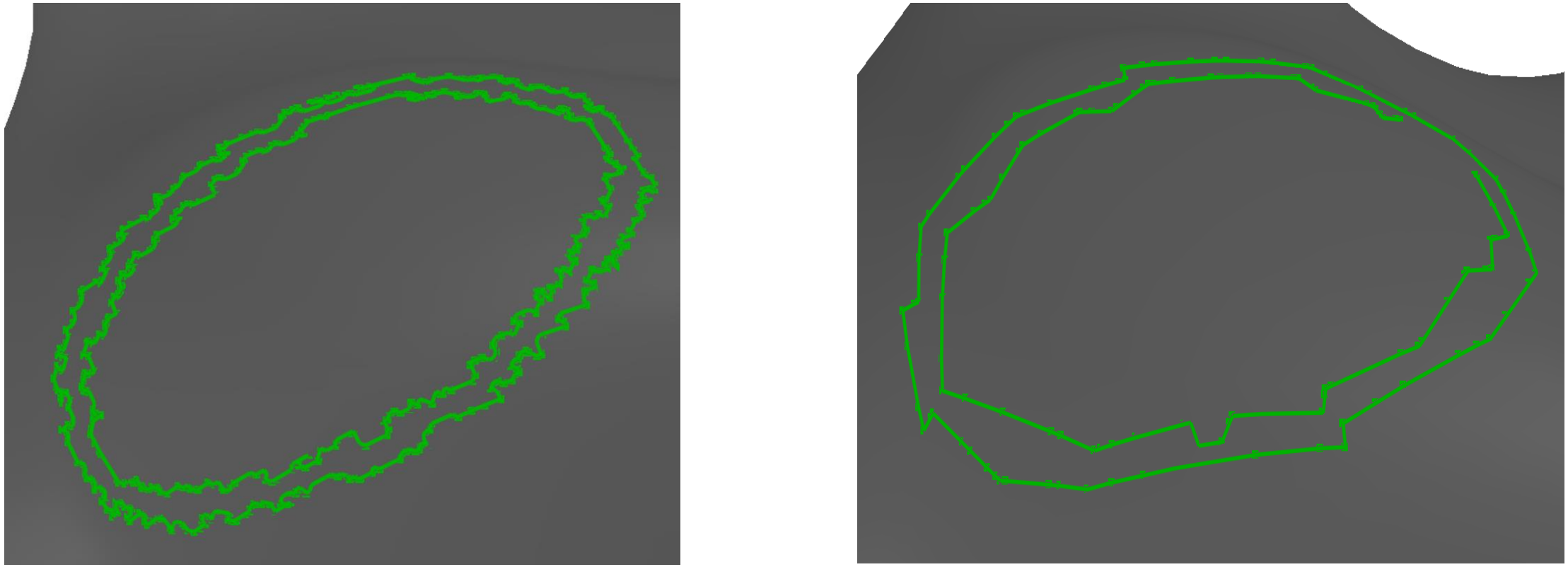
Reduce number of points per trace : step 1 example



Views of the before/after polished contour

Tracing

Reduce number of points per trace : step 2 example



Views of the before/after traces reduced

Tracing

Create new tests for team robot

Why : *(demand from TR) to have new specific test, with the new masks and UV*

***What*:**

- *Test n.11 : a small line on the right wing of the duck*
- *Test n.12 : a circled line on the right wing of the duck*
- *Test n.13 : 3 forms contours (on the left wing, head side and bill) made from cutted lines*
- *Test n.14 : a line wandering around all the duck*
- *Test n.15 : three simple colored lines (R,G,B) on the upper right wing of the duck*
- *Test n.16 : three simple colored lines (R,G,B) on the upper part of both wing of the duck*

Tracing

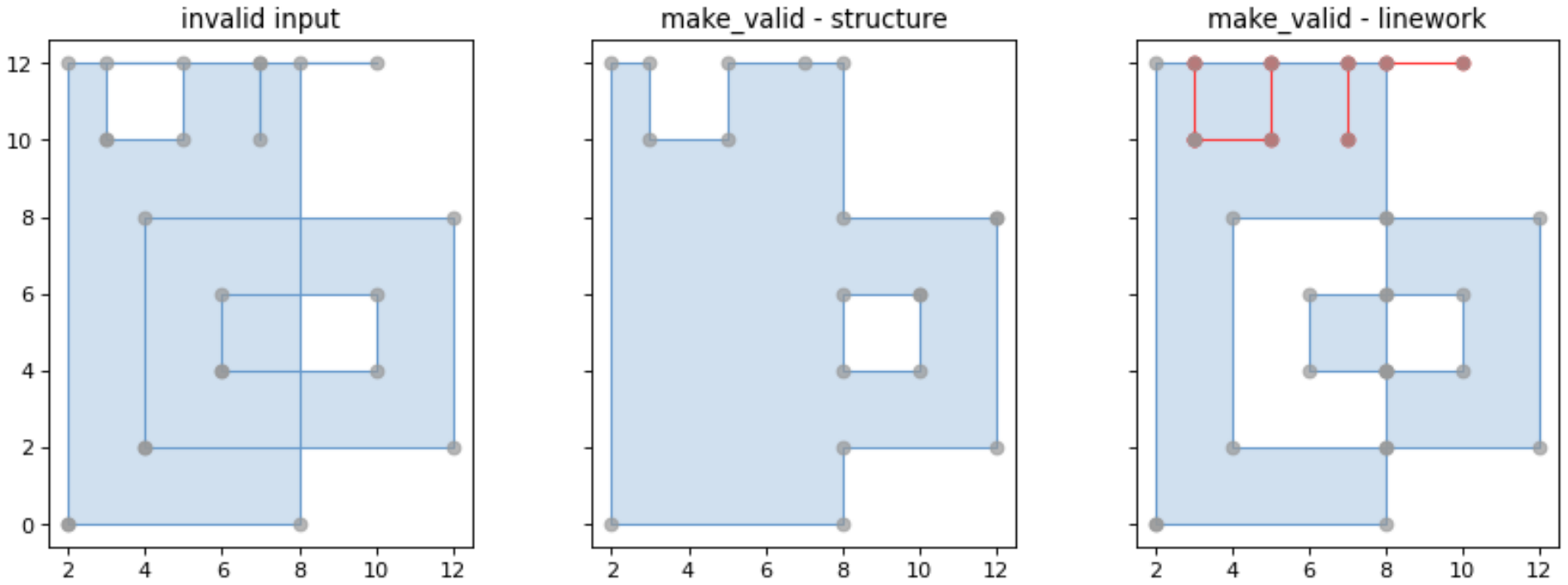
Fill slicing debug 2

Why : issue appeared with the reduction update, the geometry became sometimes invalid

How:

- *use shapely polygon.make_valid()*

Conclusion Although this specific problem was solved, the current implementation/tracing pipeline should be further tested to ensure robustness



Example from the shapely doc

Tracing

Tracing pipeline stress-testing

Why : *During the project, the tracing pipeline had multiples updates and debug phases since it's first implementation. At this point, the pipeline is for sure effective when used with the textures tested during our implementation work. To further tests and discover possible flaws, it should be further tested with new textures.*

How: *using a textures set (40 textures) generated by Alexandre to cover more cases*

It's WIP

3D Printing

No milestone planned

New set of wood support available

- Possibility to increase the pen panel to **8**
  - Technically 7 (yellow considered as done by default)
- If new duck support printed : could do parallel tests

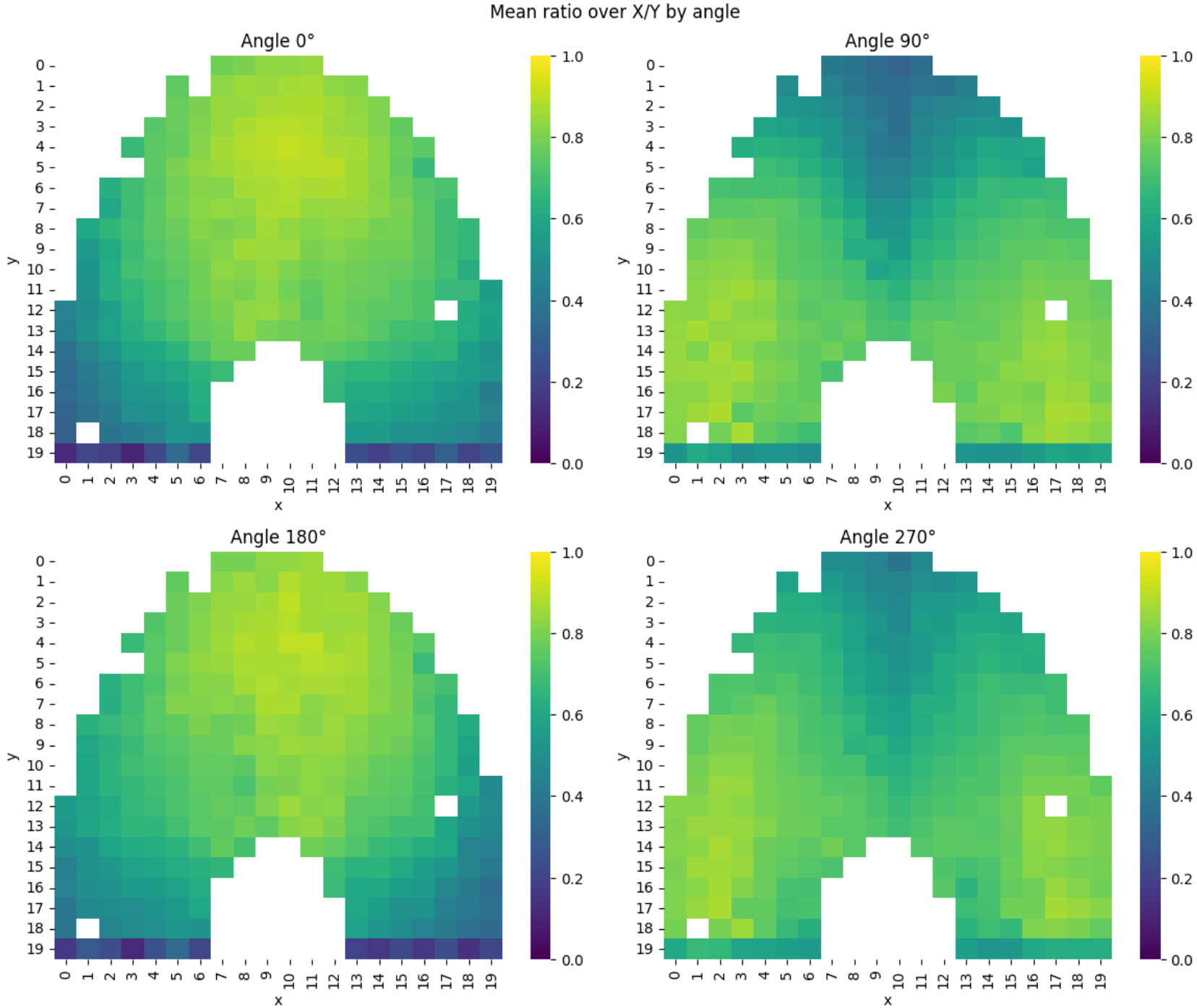
Duck can be displaced by robot but for now, no critical error

Robot – Support placement

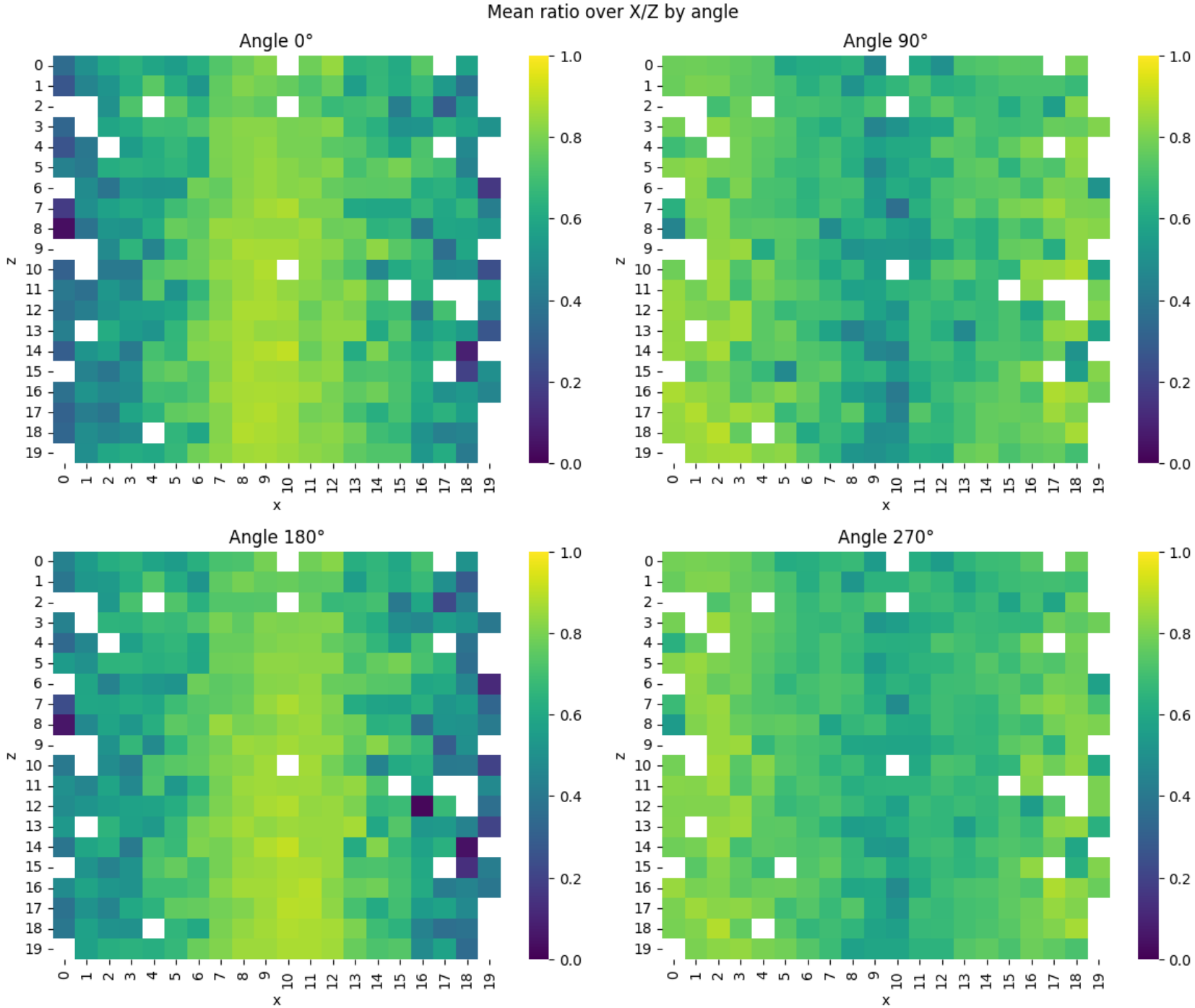
Goal
Determine an optimal support/duck placement, maximizing the number of accessible points

Method
1. Generate random points on duck
2. Generate random support placements (position and rotation)
3. Test each point for each placement: with collisions and cone search

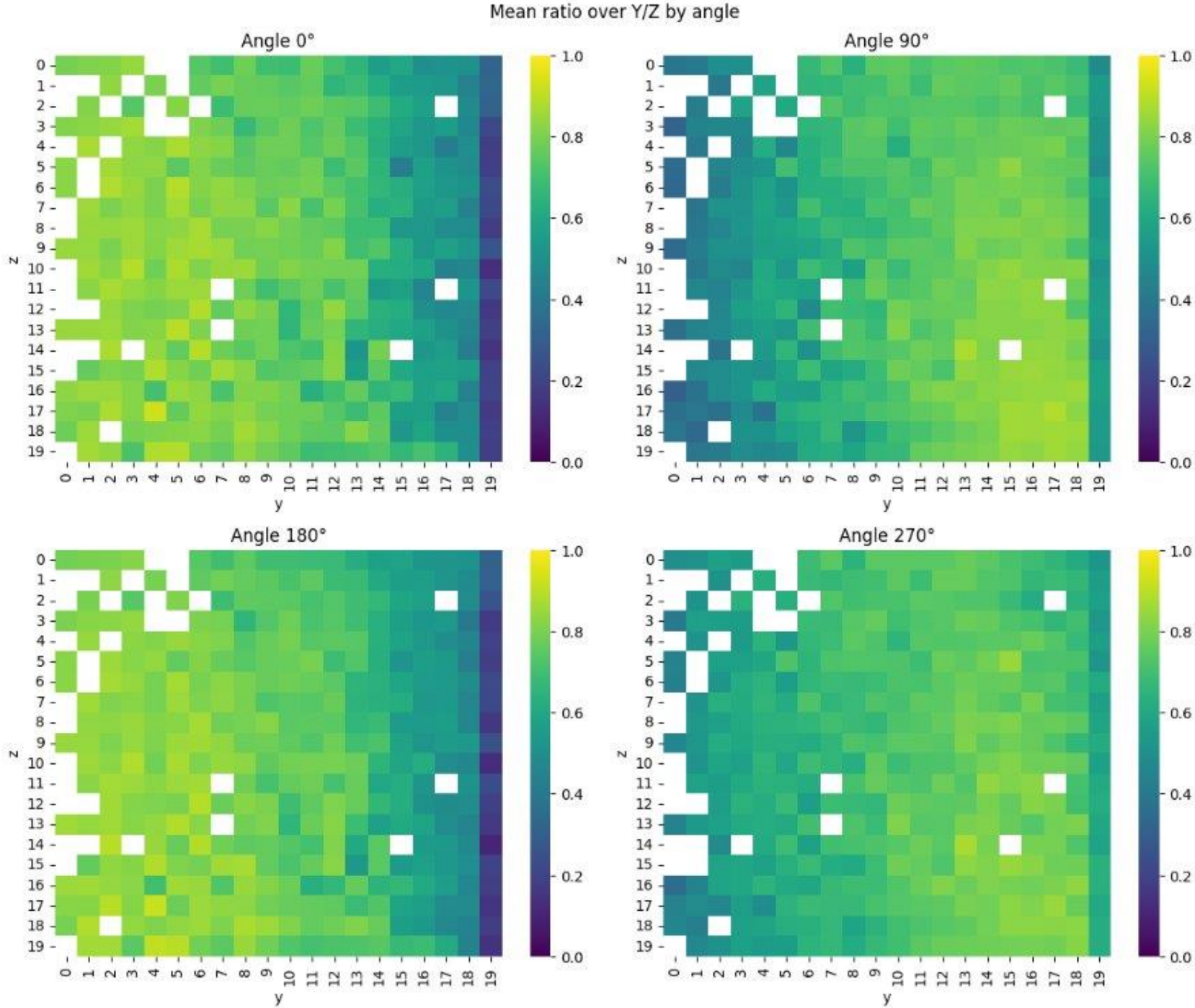
Robo


Mean ratio over X/Y by angle
Angle 0°
Angle 90°
Angle 180°
Angle 270°
x
y
0.0
0.2
0.4
0.6
0.8
1.0

Robo


Mean ratio over X/Z by angle
Angle 0°
Angle 90°
Angle 180°
Angle 270°
z
x
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
0.0 0.2 0.4 0.6 0.8 1.0

Robo


Mean ratio over Y/Z by angle
Angle 0°
Angle 90°
Angle 180°
Angle 270°
z
y
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
0.0
0.2
0.4
0.6
0.8
1.0

Robot – Support placement

Results: not really surprising, matches our expectations

Robot - Intégration


```
[stderr]
C:\Users\alexa\AppData\Local\uv\cache\builds-v0\.tmp8dcnyW\Lib\site-packages\setuptools\dist.py:765: SetuptoolsDeprecationWarning: License classifiers are deprecated.
!!

        ********************************************************************************
        Please consider removing the following classifiers in favor of a SPDX license expression:

        License :: OSI Approved :: zlib/libpng License

        See https://packaging.python.org/en/latest/guides/writing-pyproject-toml/#license for details.
        ********************************************************************************

!!
  self._finalize_license_expression()
error: Microsoft Visual C++ 14.0 or greater is required. Get it with "Microsoft C++ Build Tools": https://visualstudio.microsoft.com/visual-cpp-build-tools/

hint: This usually indicates a problem with the package or the build environment.
p: If you want to add the package regardless of the failed resolution, provide the `--frozen` flag to skip locking and syncing.
```

Robot

Robot

Robot

Temps de calcul:

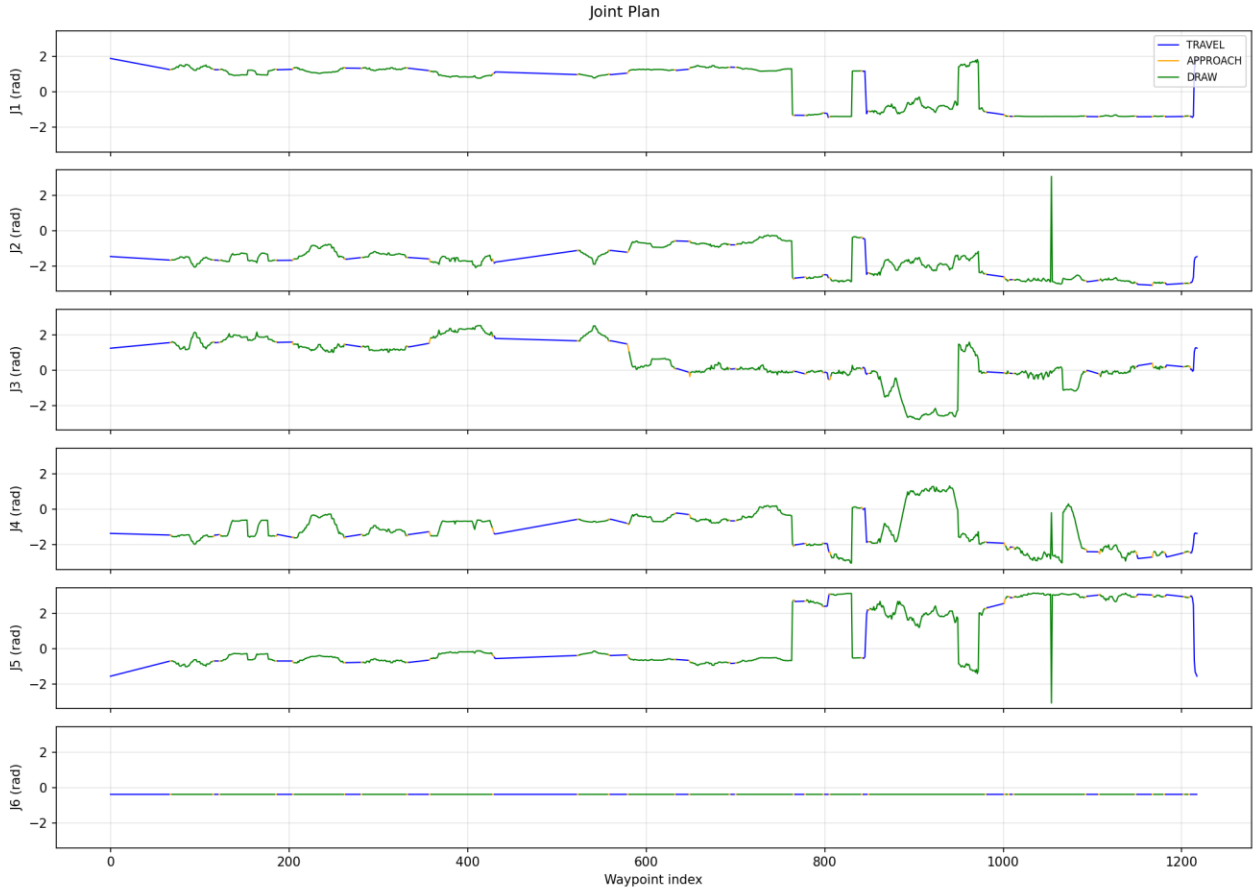
- Triangle inverted: 30s
- 3 traits multipen : 2 m10s
- Tatouage tribal multipen: 25min

~ collisions

~ manual

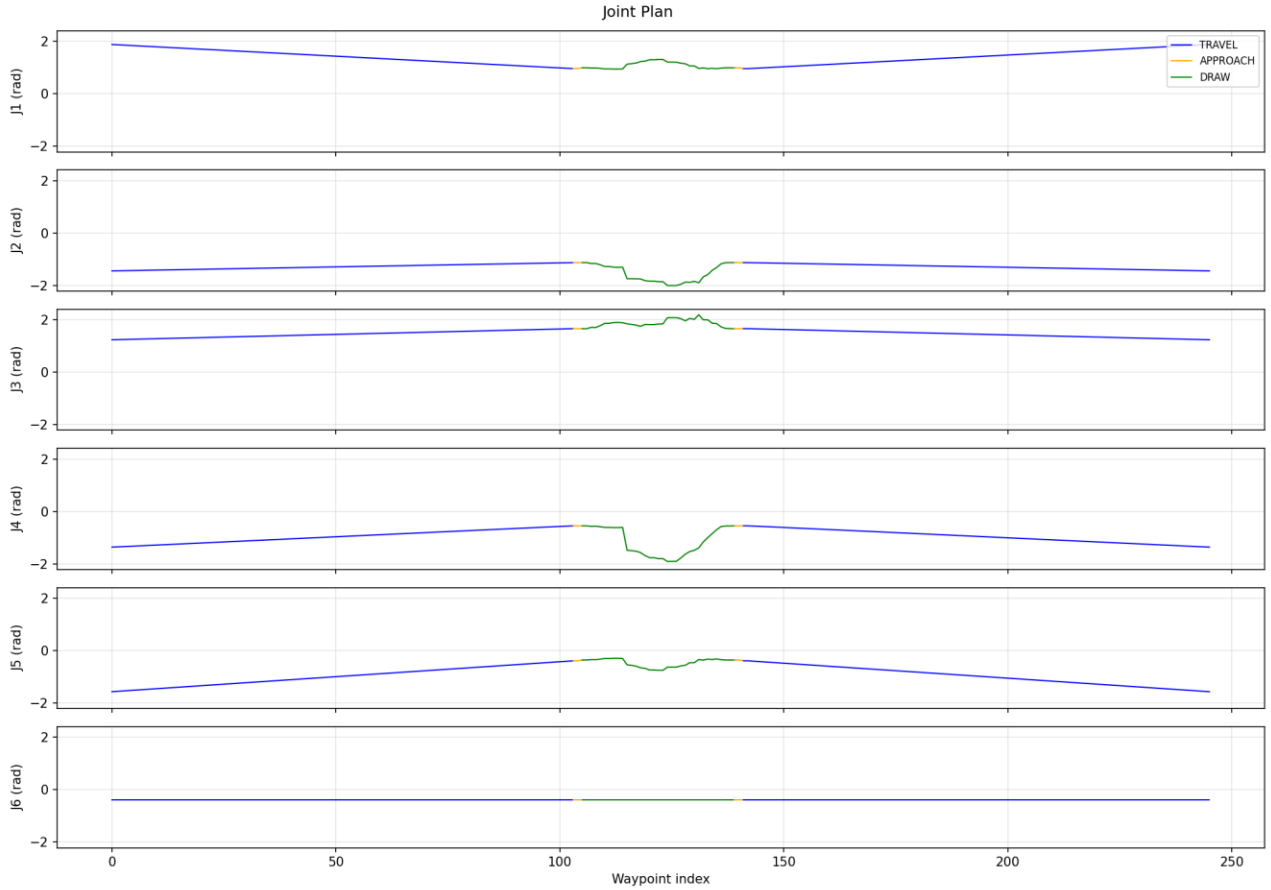
Robot

Switch non bloquant



Robot

Switch non bloquant 2



Integration

- Tracing → fully integrated
- GenAI → integrated but pending review
- Robot → not integrated, was blocked by integration of `ur3e-control`

Promoting website

- Description of the project's steps
- Duck model (glb) examples or images
- Videos (robots, eventually other parts)

Milestone

For the small week 7

Using the **complete pipeline**, starting from an **orchestration interface**, draw **AI-generated contours**, on a duck using **multiple colors with automatical changes**.

Have a **promoting website** for the project.

Make it work