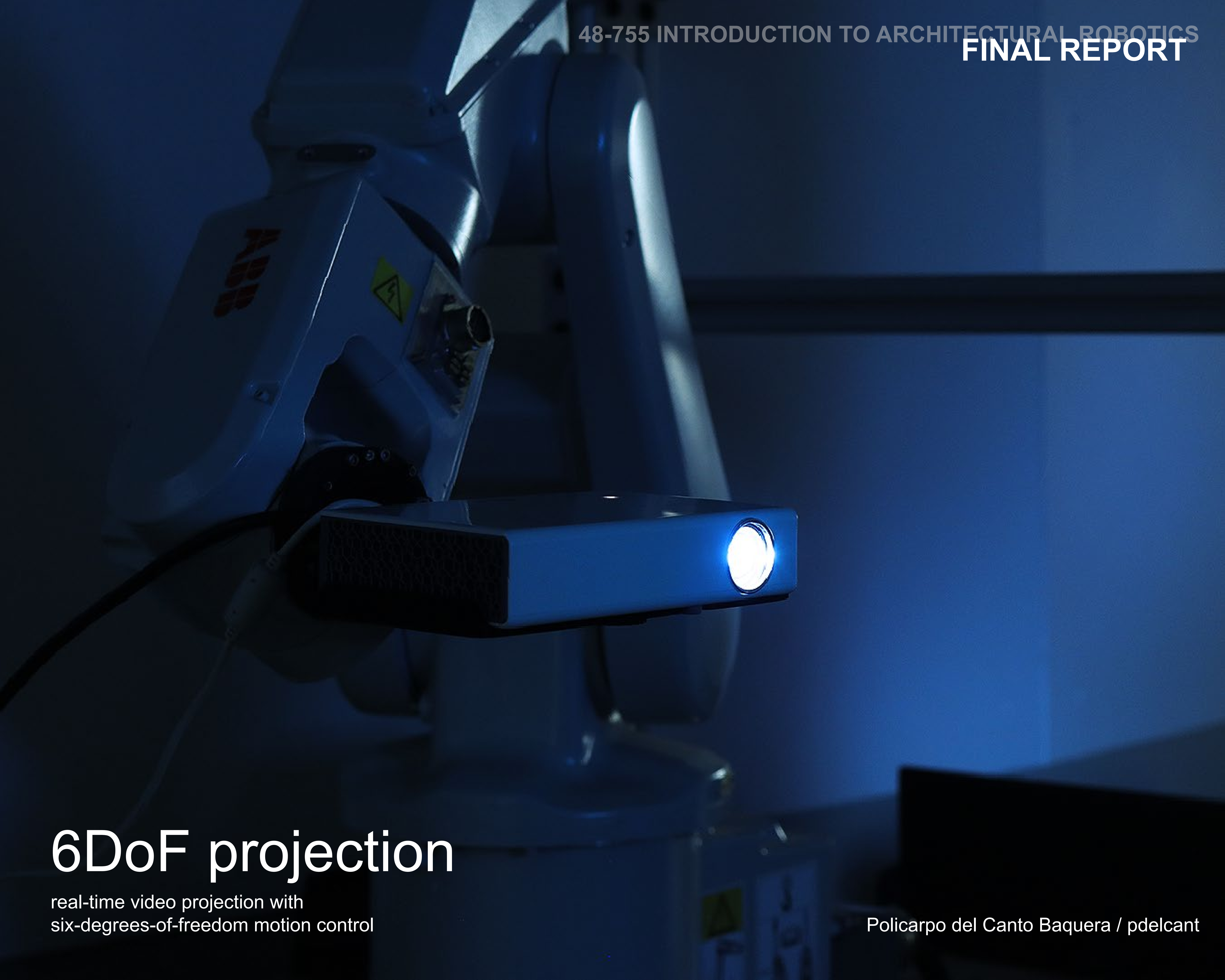
48-755 INTRODUCTION TO ARCHITECTURAL ROBOTICS

FINAL REPORT



6DoF projection

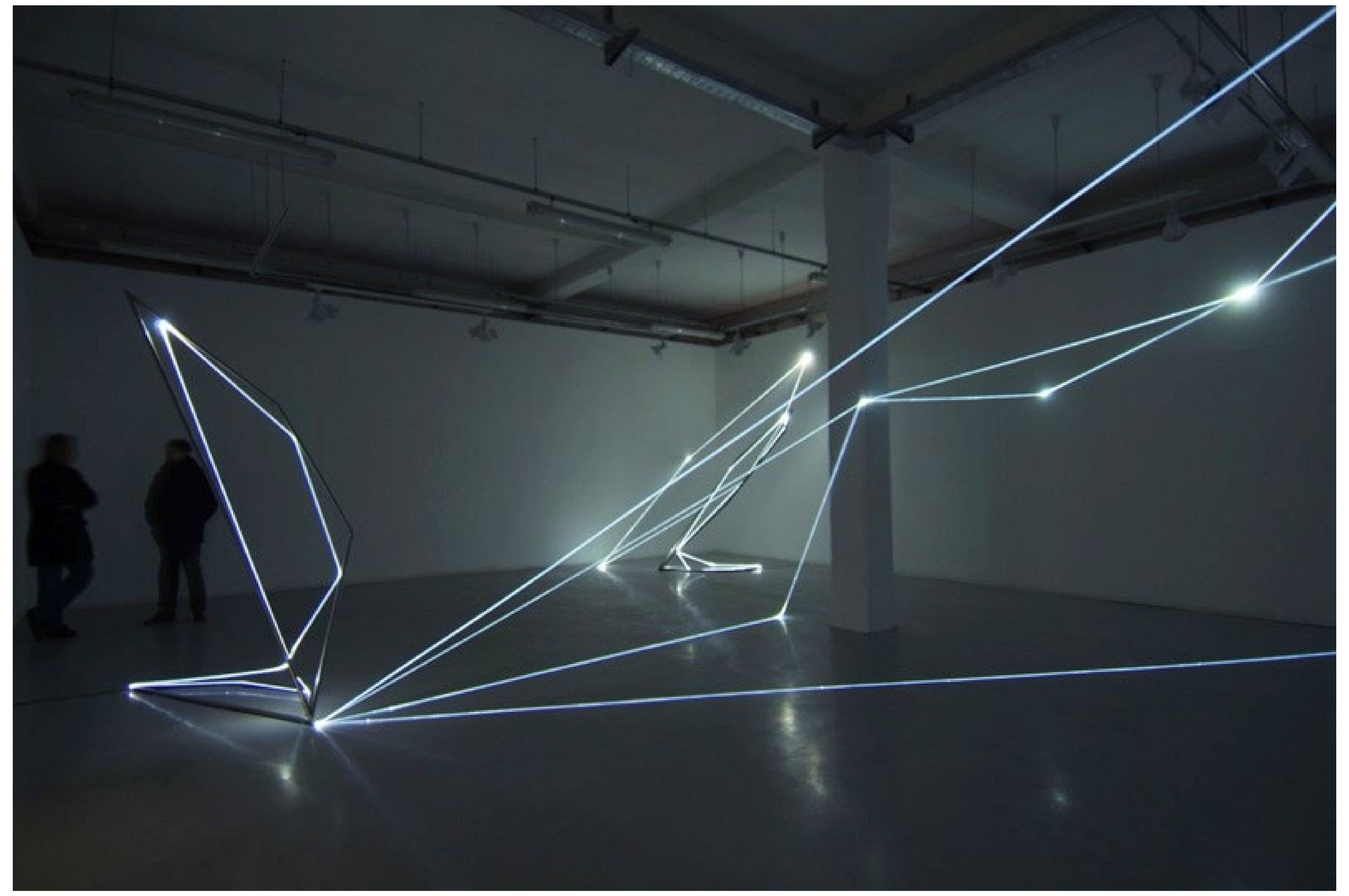
real-time video projection with
six-degrees-of-freedom motion control

Policarpo del Canto Baquera / pdelcant

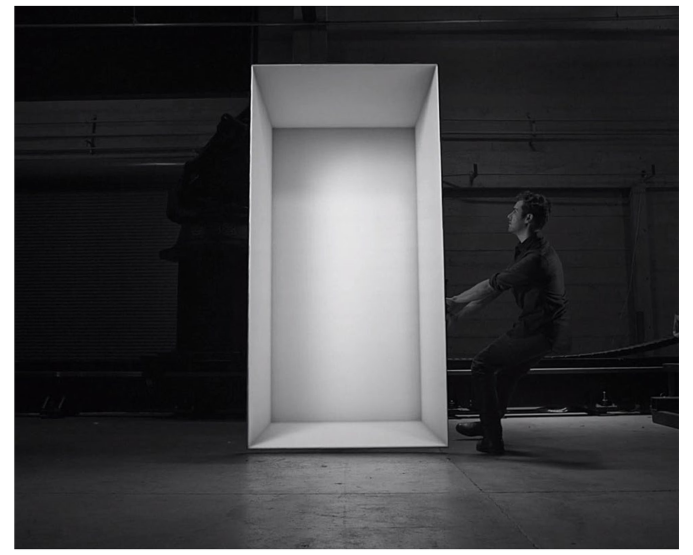
PRECEDENTS

Video mapping is a common technique in interactive and digital arts that enables the activation of architectural spaces through the projection of media content onto surfaces. It also unleashes new symbolic values and amplifies the dimensions of a digital production that no longer has to be displayed on a flat-screen. However, normally, video mapping is executed with fix projectors which are precisely located and carefully manipulated to maintain a stable and accurate projection onto the objects and spaces. Projectors already incorporate systems to warp, blend, and focus the image automatically relying on orientation and laser sensors. Yet, still, these mechanisms are used only to set the initial conditions of the projection, and consequently, a successful video mapping immersion experience relies on the precision of these initial settings.

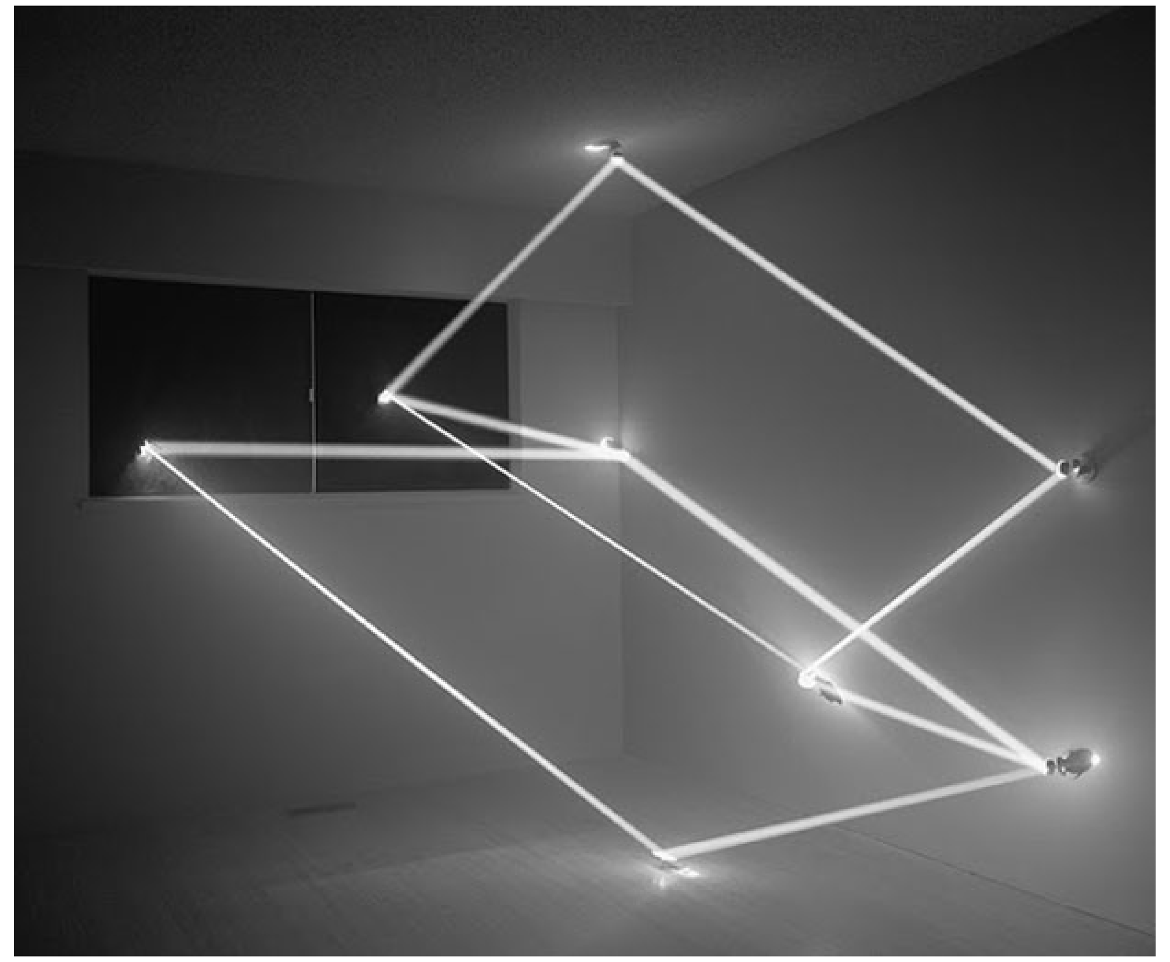
This project will explore the possibilities of mobilizing a video projection with the six degrees of freedom of a robotic arm. With this aim, I will create a set of instruments and programming procedures so the IRB 120 robot can execute a short video mapping performance to video map one of the rooms of the Design Fabrication Lab. The tool of the robot will be a small projector mobilized and oriented around a set of mirrors to reach a big surface with just one small light beam.

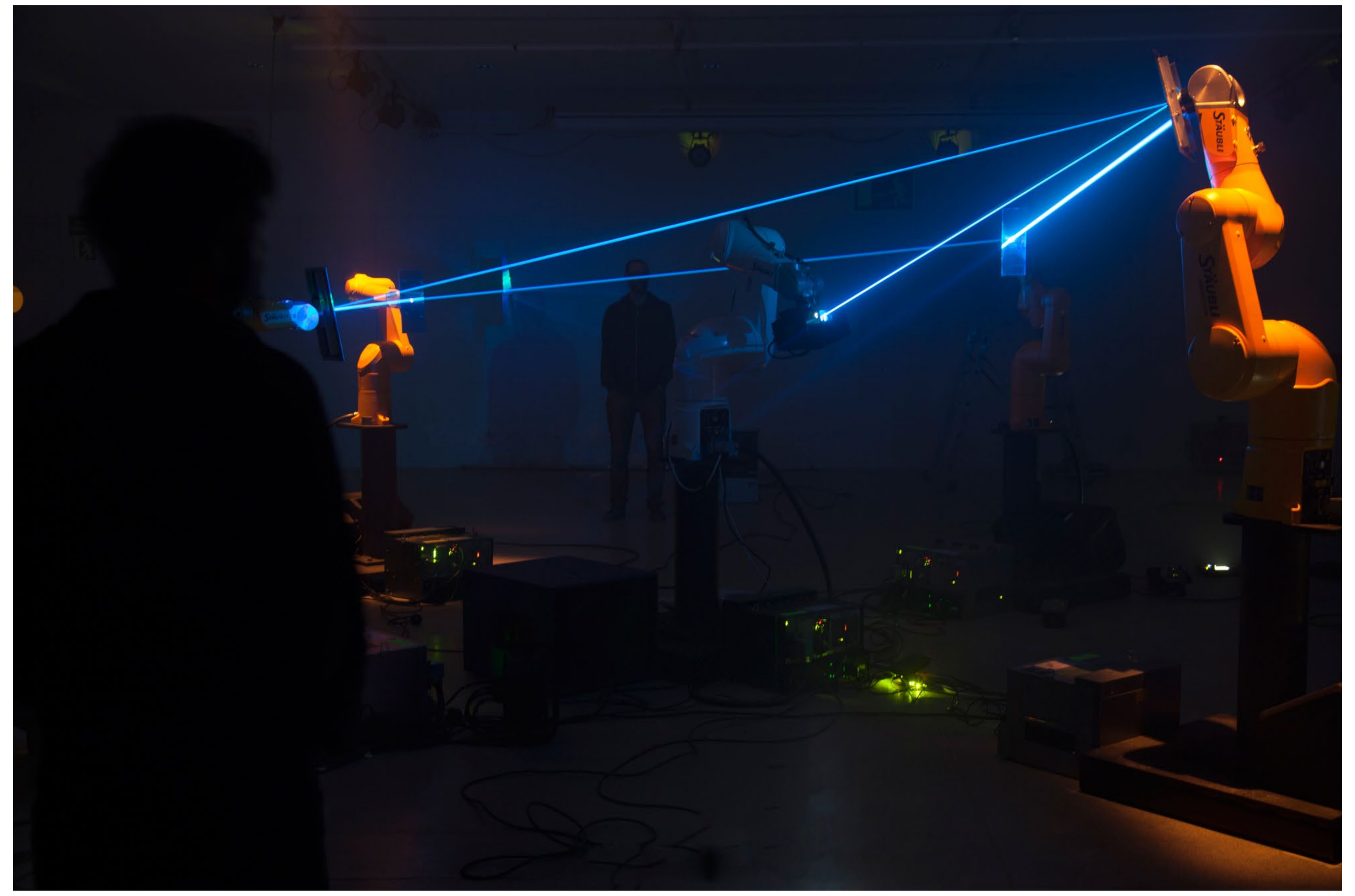

Light Catalyst (2005) Carlos Bernardino


Telestron (2017) GMUNK
vimeo.com/256512922


Box (2013) GMUNK
vimeo.com/75361102

WHY & WHAT


Trace Heavens (2011) James Nizam


AI Vectors for Mira Festival (2016) OnionLab

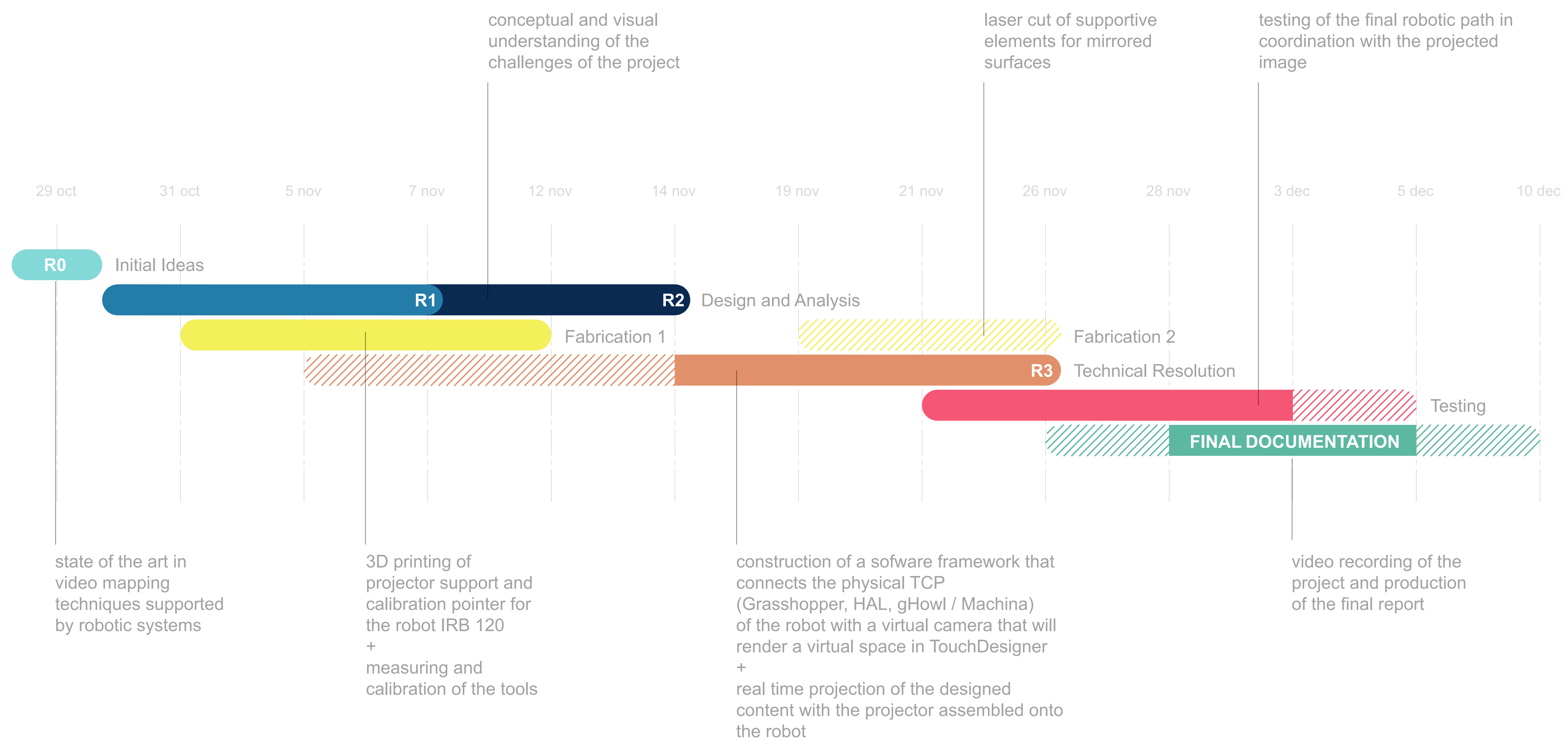
Why is the project relevant?

Aside from their efficiency to repeat tedious tasks multiple times in fabrication environments, robotic systems can have many advantages to achieve very precise routines with artistic purposes. Also, attending to their popularization with professional and academic purposes and, their increasing affordability in the market, artists could use them to explore new realms of creative practices where human maneuverability exceeds the tolerance limits required.

What is the importance of the project?

The main goal of the project is to solve the geometrical questions required to track a leveled image onto a flat surface (one or more walls of a room) from a projector assembled on a robotic arm. The programming framework will be the critical element of the project: while the robotic arm will seamlessly drive the projector to cover the mapped surface, the programmed algorithm will be able to link the physical position and orientation of the projector with the projection coordinates of the resulting throw image.

SCHEDULE



REQUIREMENTS

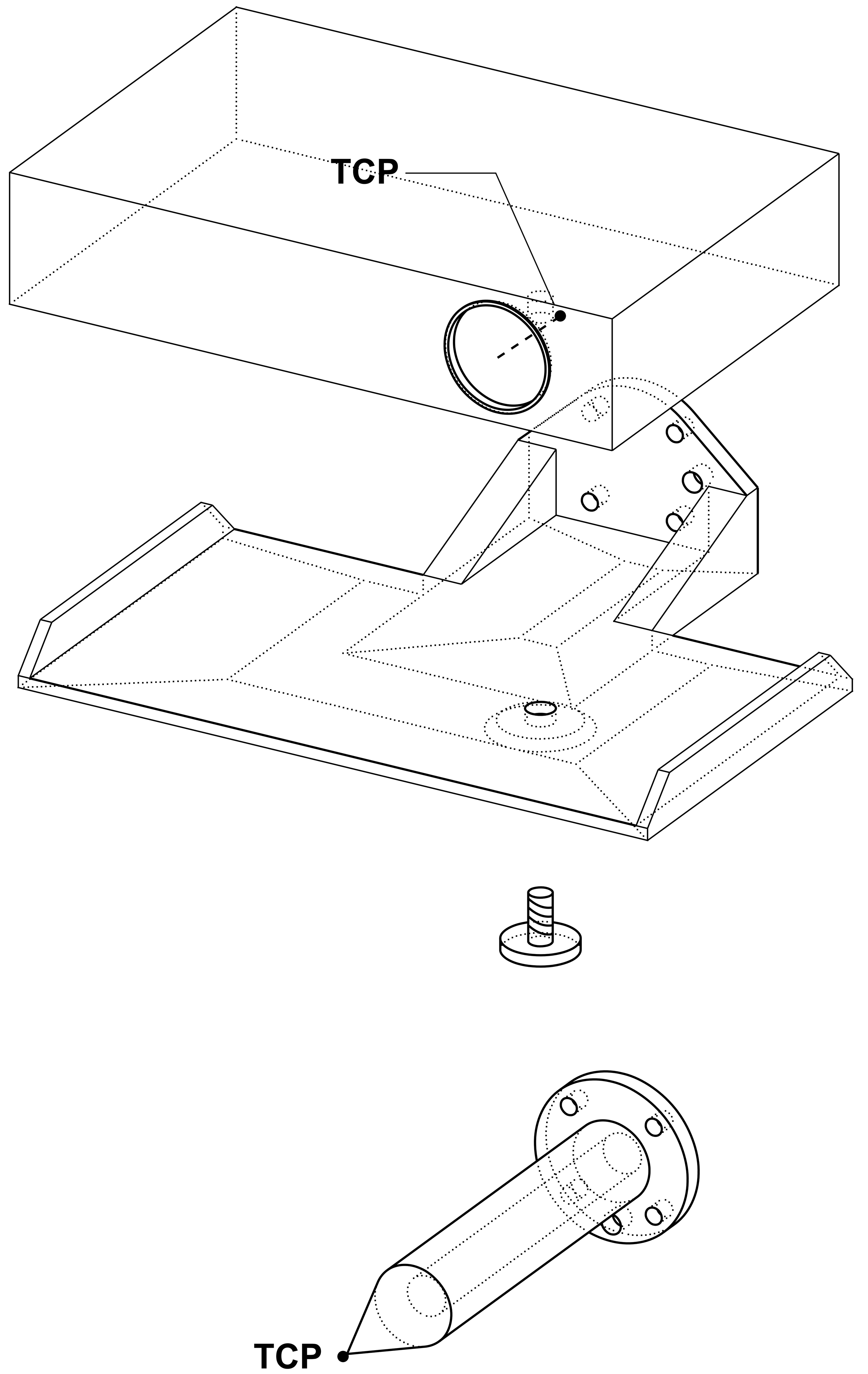
Hardware

- ☑ IRB 120 3kg
- ☑ Mini projector
- ☑ HDMI cable
- ☑ 3D printed pointer for calibration
- ☑ 3D printed project holder
- ☐ 3D printed mirror support and flat mirrored surface

Software

- ☑ Design and prototyping (Rhinoceros 5.0)
- ☑ Programming framework (Grasshopper / Python)
- ☑ Robot Control (RobotStudio / HAL / Machina)
- ☑ Video Mapping (TouchDesigner / MadMapper)
- ☑ Video Editing and Animation (Adobe After Effects)

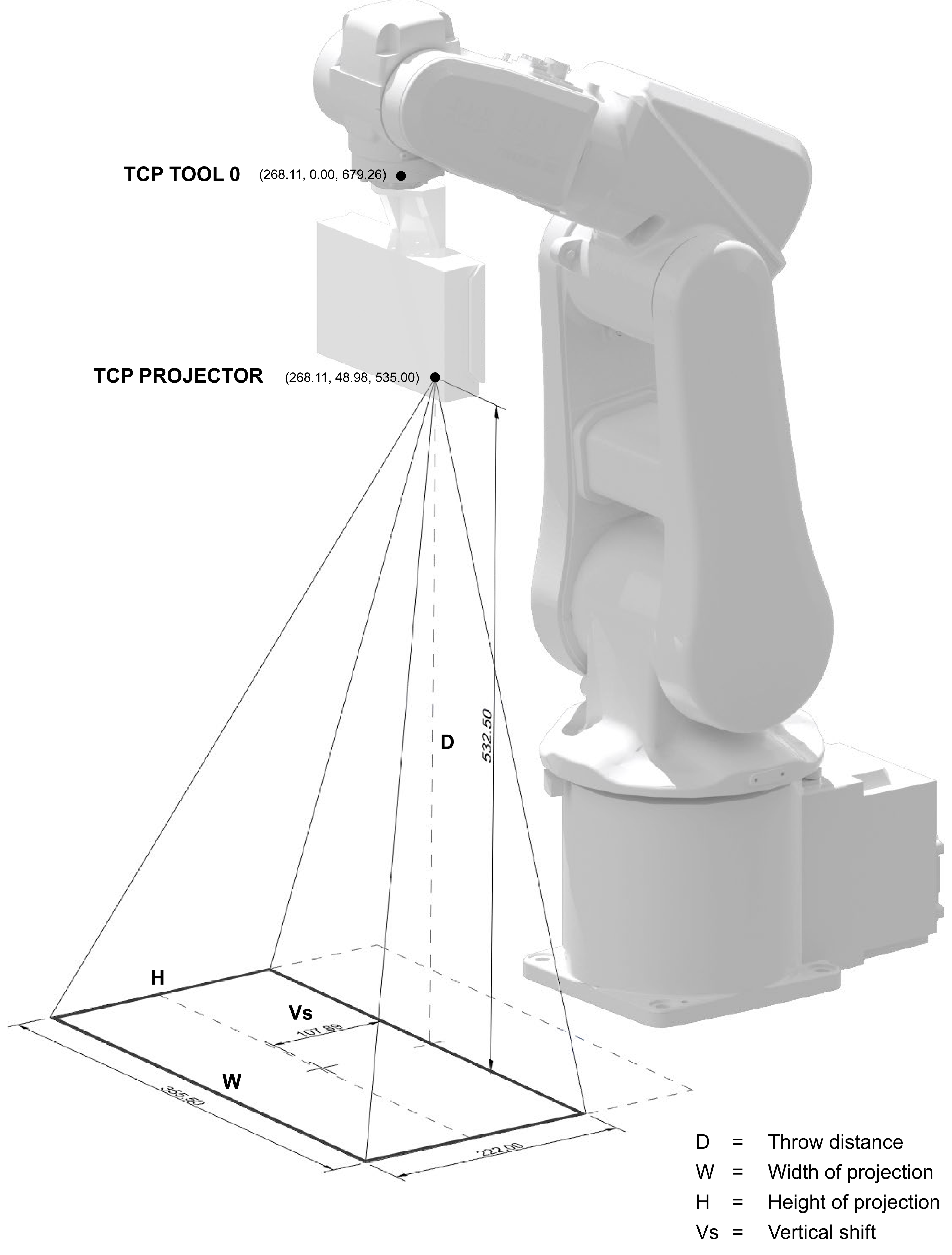
FABRICATION



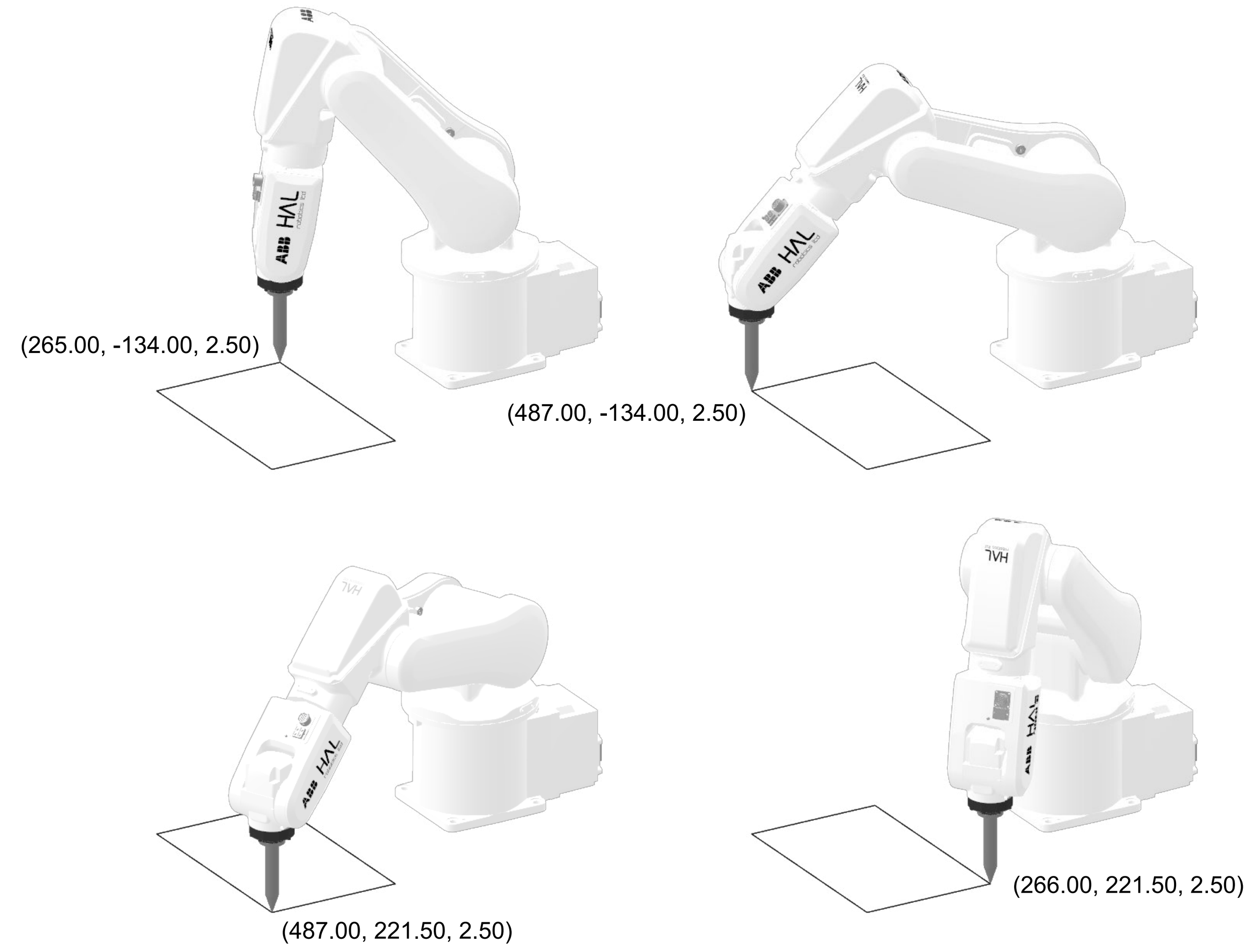
The scope of the project requires ease to maneuver with the robot and quickly define the properties that will control the final projection. For that reason, I have chosen a micro projector with a weight under 0.5kg that can be easily moved by the robotic arm ABB IRB 120. The supportive tool that will hold the projector in position is designed in 3d with Rhinoceros reinforcing the joint to the robotic arm and lighten the points with no structural stress. Parallelly, I also decide to design a short pointer for calibration purposes.

Both tools are printed in the Ultimaker 3D printer at dFab Lab using black nylon to provide extra strength to the pieces.

CALIBRATION



Calibration of the projector as Tool



Use of the pointer to obtain the projection values

First, the Tool_Pointer is defined as a tool using the teaching pendant and the robot in manual mode. This involves defining the tool with 4 points using a mark in the table.

Now, I define the Tool_Projector. For that, the tool is assembled in the robot, and while selecting the Tool_0, I positioned the robot in a comfortable position where the Y value is 0. This aligns the projection with the table. The 4 points that define the projection are transferred to the table, and then, I use the Tool_Pointer to record them. Finally, using the 4 points, the position of Tool_0 and, the Throw Ratio of the projector, I can obtain the TCP of the Tool_Projector.

COMMUNICATION DIAGRAM

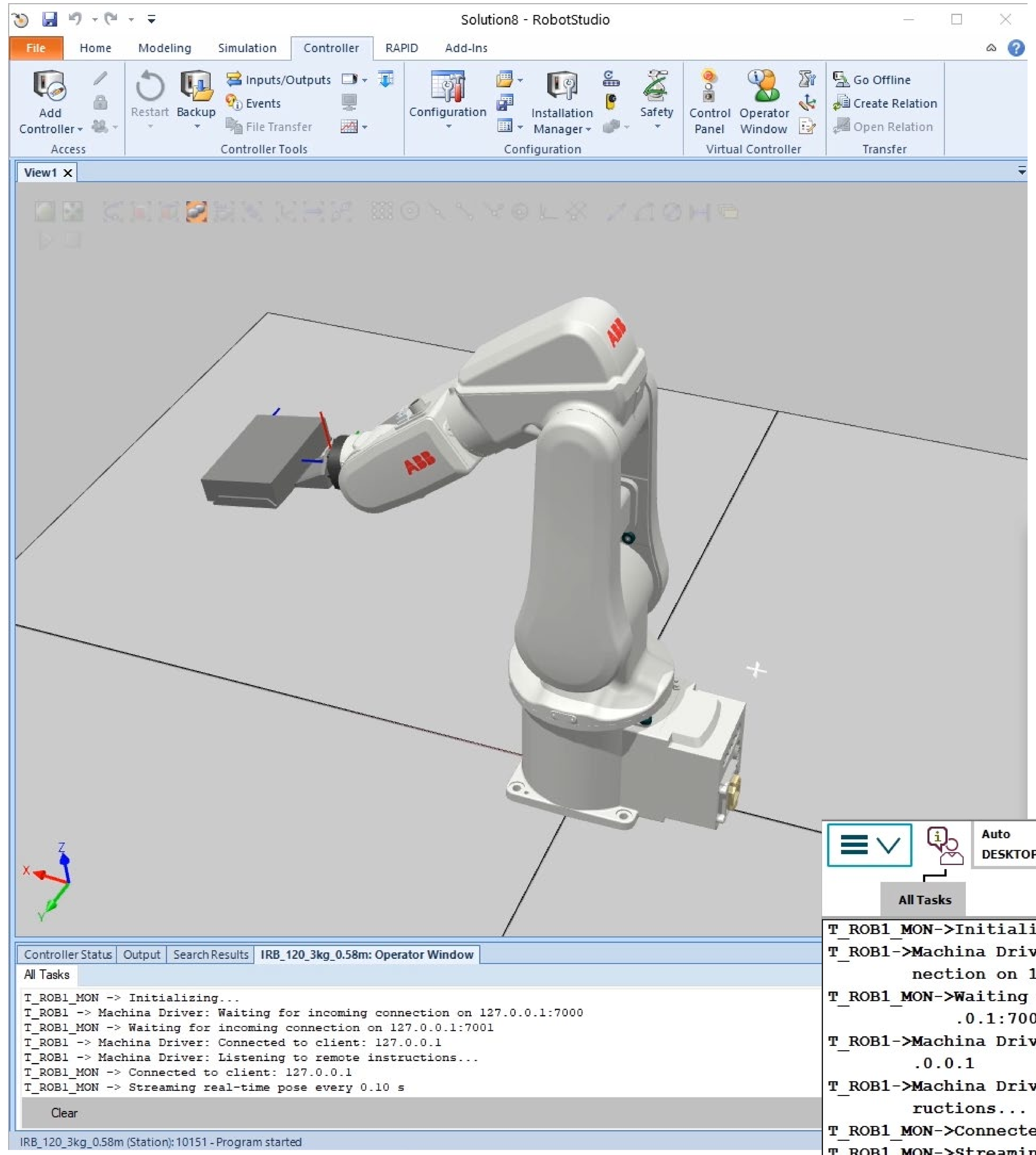
FINAL REPORT

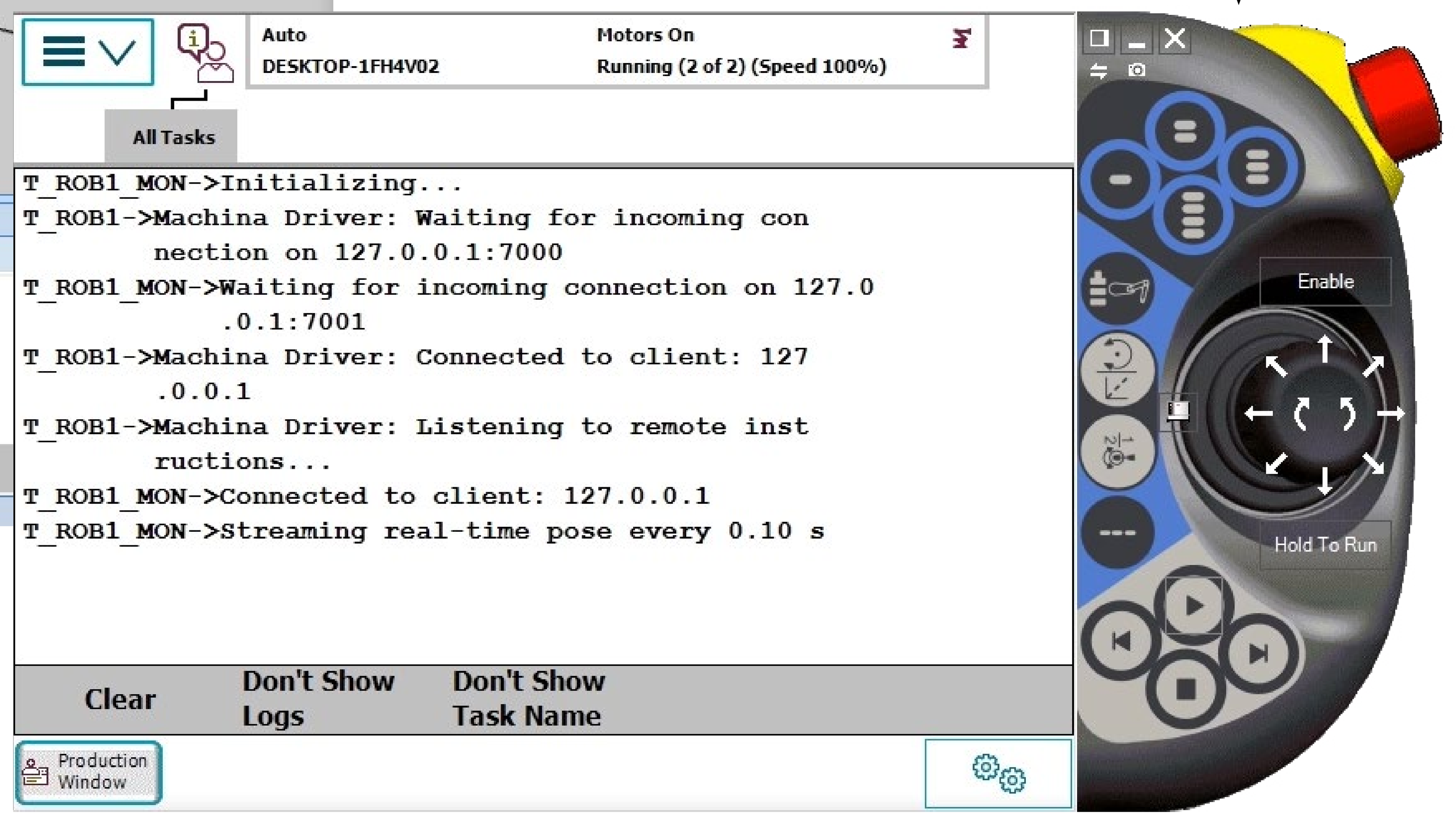
SIMULATION

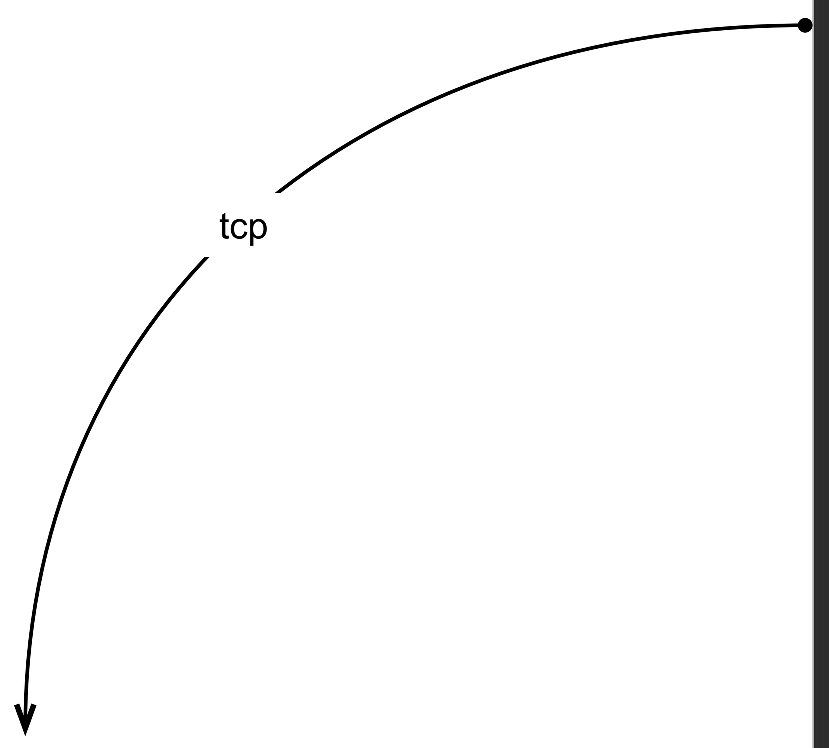
Machina Bridge is a open-source software framework designed by Jose Luis Garcia del Castillo that enables real-time interaction with a robotic arm through command written lines. I have decided to use this software not only for the ease in communicating with the robot but because it also enables the feedback in real-time of the TCP of the robot, which will correspond with the focal plane of the projection.

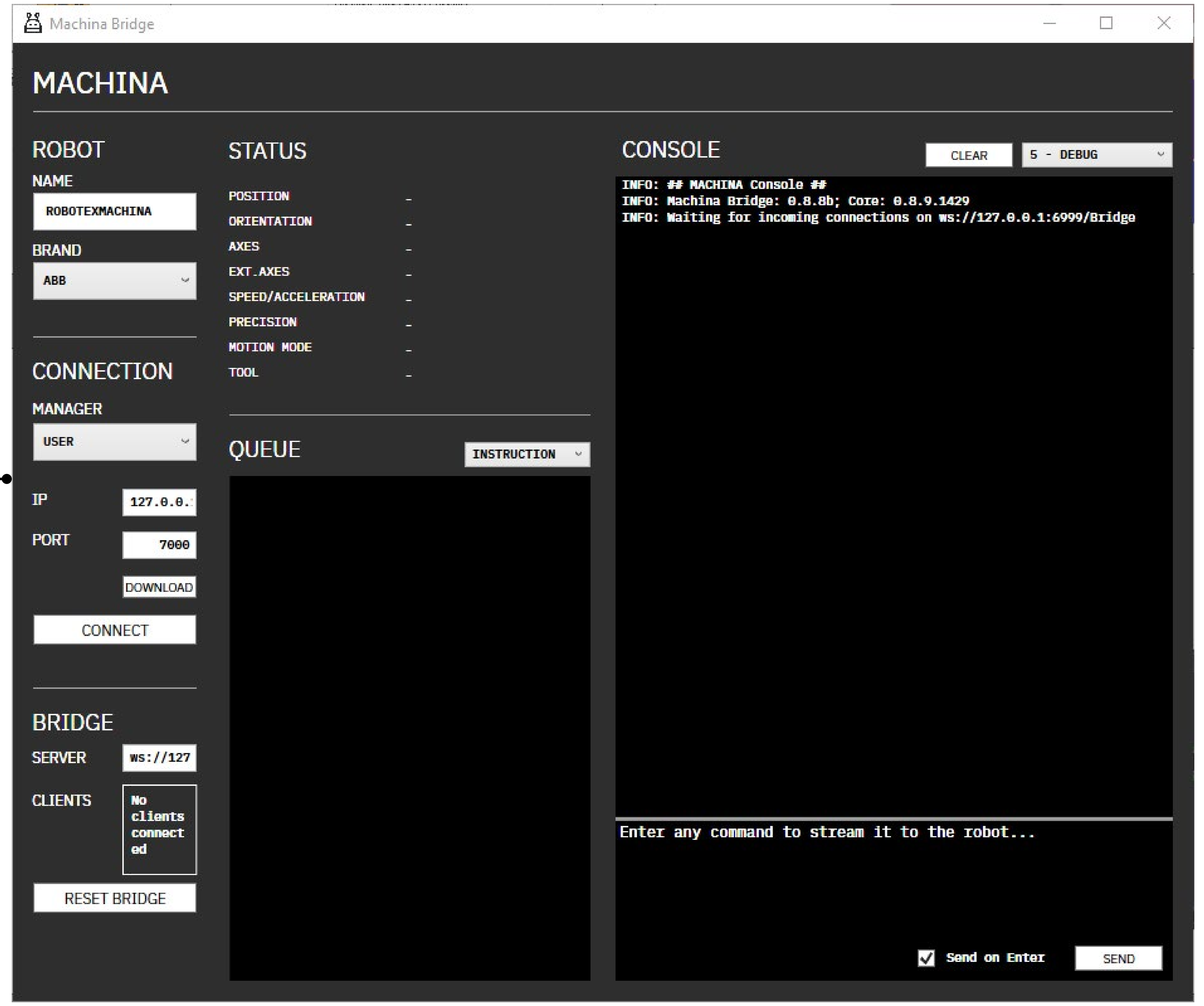
Before running Machina Bridge in the robotic arm, first, I test a path composed by four targets in RobotStudio. For that, I define a tool referred as “projector” with the data obtained after the calibration of the projector in the robot.

RobotStudio also will determine if the targets defined in Machina are reachable; although in some cases, even if the targets seem unapproachable, it is possible to reach them by changing the rotation of the joints in the previous configuration.









Tool Definition	DefineTool(“projector”, 0, 49, 144.26, -1, 0, 0, 0, -1, 0, 0.5, 0, 20, 70); AttachTool(“projector”);
Speed	SpeedTo(10);
Zone	Precision(100);
Wait(seconds)	Wait(2000);
Target 0 (Home)	TransformTo(400, 400, 350, 0, 0, -1, -1, 0, 0);
Target 1	TransformTo(500, 350,400, 0, 0, -1, -0.426095, 0.904678, 0); Wait(3000);
Target 2	TransformTo(250, 250, 350, 0, 0.42305, -0.906106, -1,0,0); Wait(1000);
Target 3	TransformTo(184, 324, 571, -0.080316, 0.0669, -0.994522, -0.64001, -0.76836,0);
Target 0 (Home)	TransformTo(400, 400, 350, 0, 0, -1, -1, 0, 0);

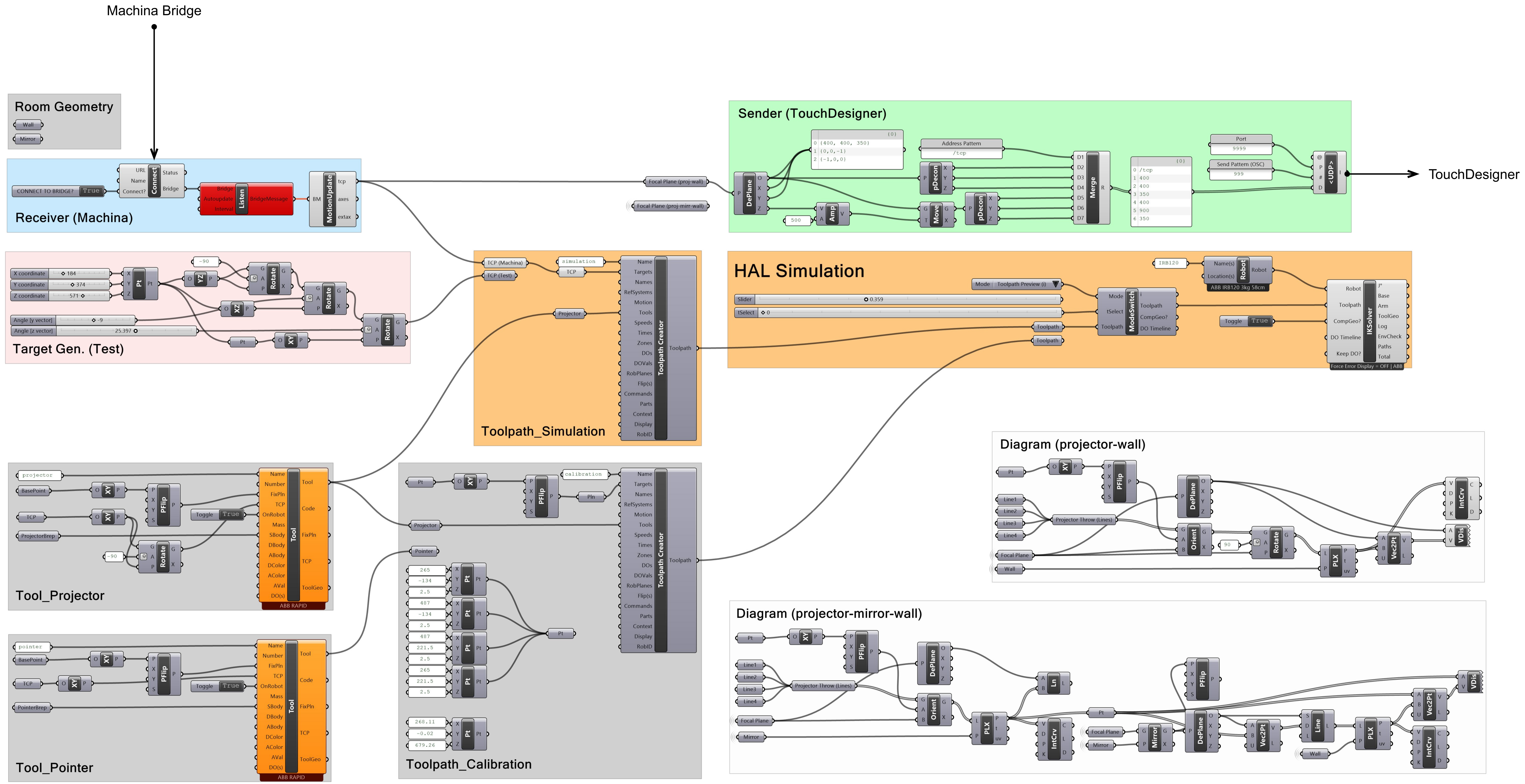
PROJECTION TEST

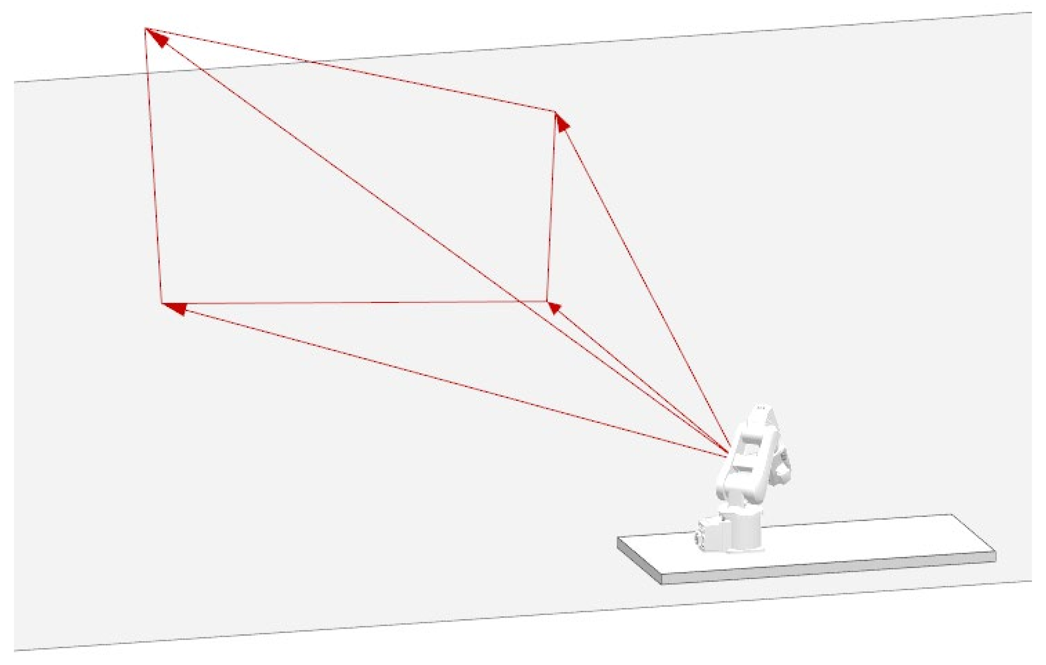
FINAL REPORT

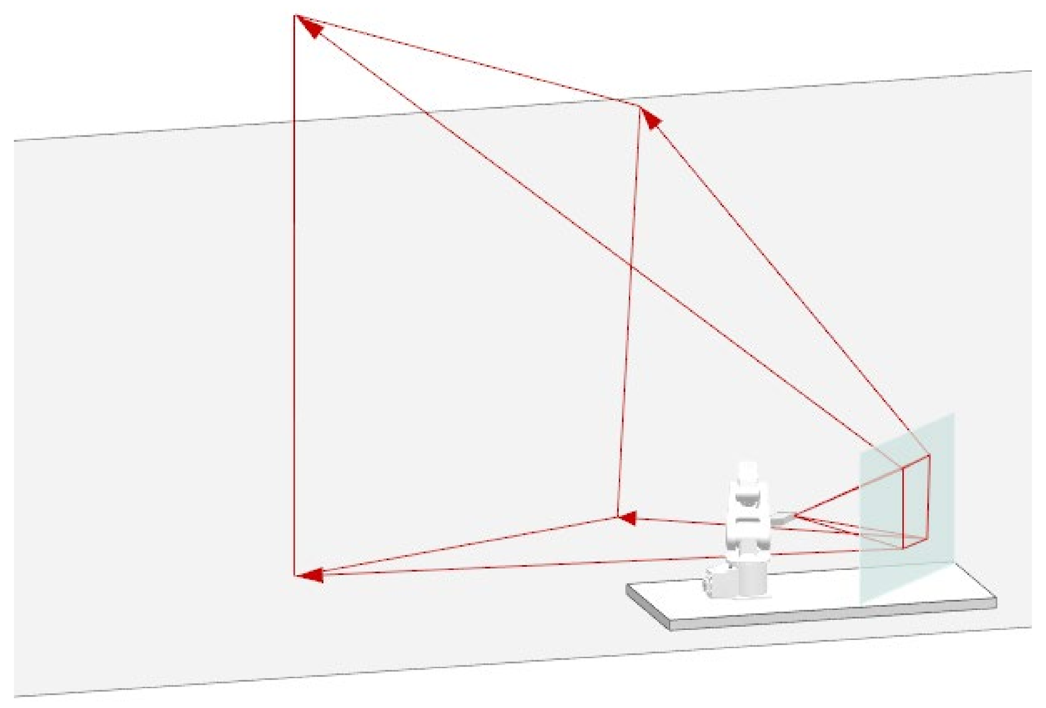
Previously, I have configured a second task named 'monitor' in the teaching pendant that runs parallelly to the main. This task is responsible for sending back to Machina the TCP of the robot in between the targets defined by our written commands. Simultaneously, Machina will send this real-time TCP to its Grasshopper component.

Grasshopper is the software that coordinates the motion data that defines the movement of the robot and its visual representation in Rhinoceros for testing purposes. Apart from receiving and visualizing the TCP received from Machina, Grasshopper also will be used to diagram the area covered by the projection in the 3d model of the dFab lab and also, future implementation of mirrored surfaces in the video mapping. HAL is used to reconstruct the robot in Rhinoceros, but it does not supplement RobotStudio as simulation software.

Finally, Grasshopper will also send the TCP to TouchDesigner through a computer port as a string that contains both the position (x,y,z) and the orientation of the plane at this point.




Projection-wall Diagram


Projection-mirror-wall Diagram

CAMERA SIMULATION

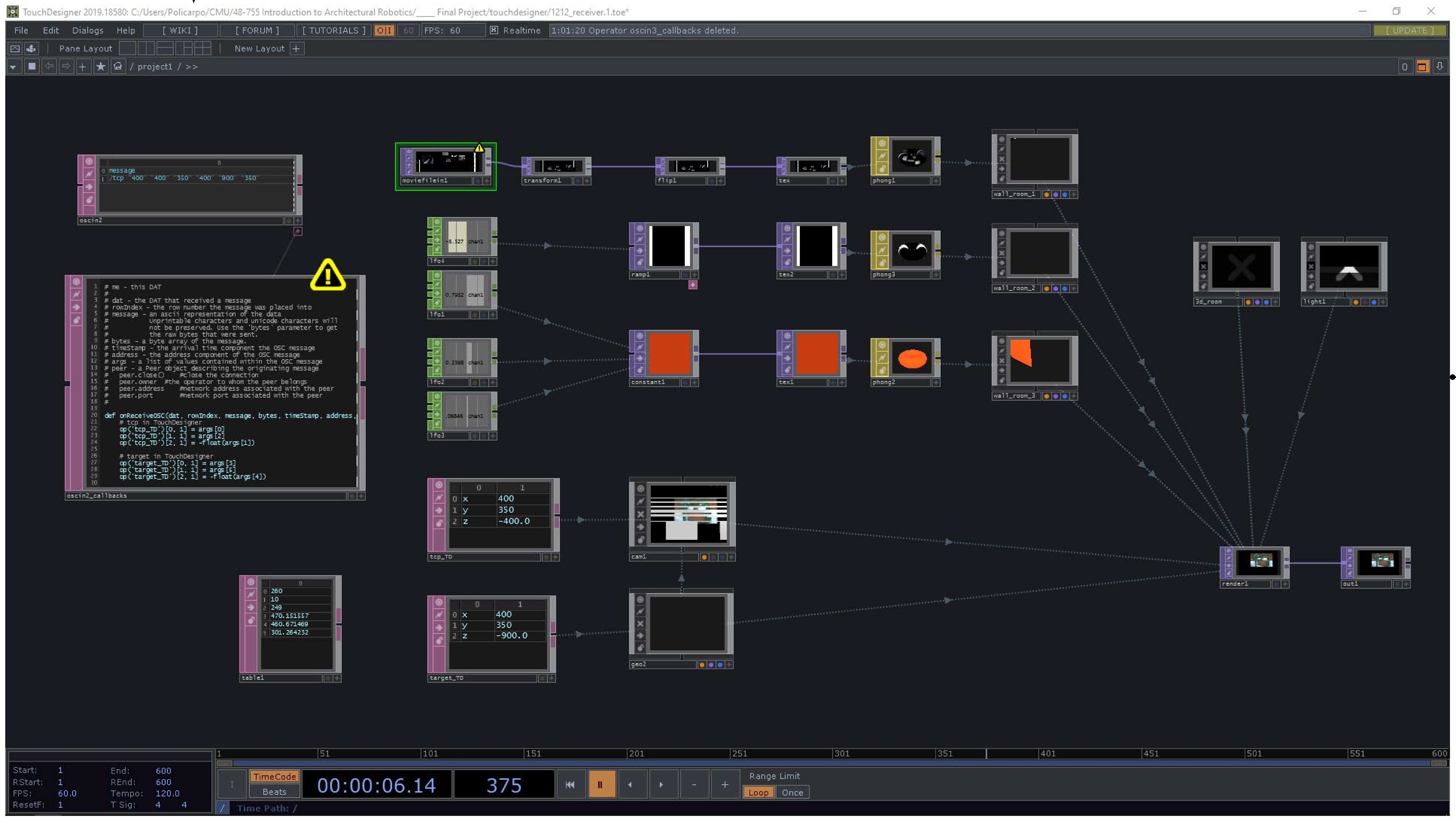
TouchDesigner is a software to design and manage interactive audiovisual content. Here, I am using it to simulate a camera placed in the exact coordinates and orientation where the projector is. Aside from the camera, I import the 3d model of the Fab lab into TouchDesigner to reconstruct the whole setting.

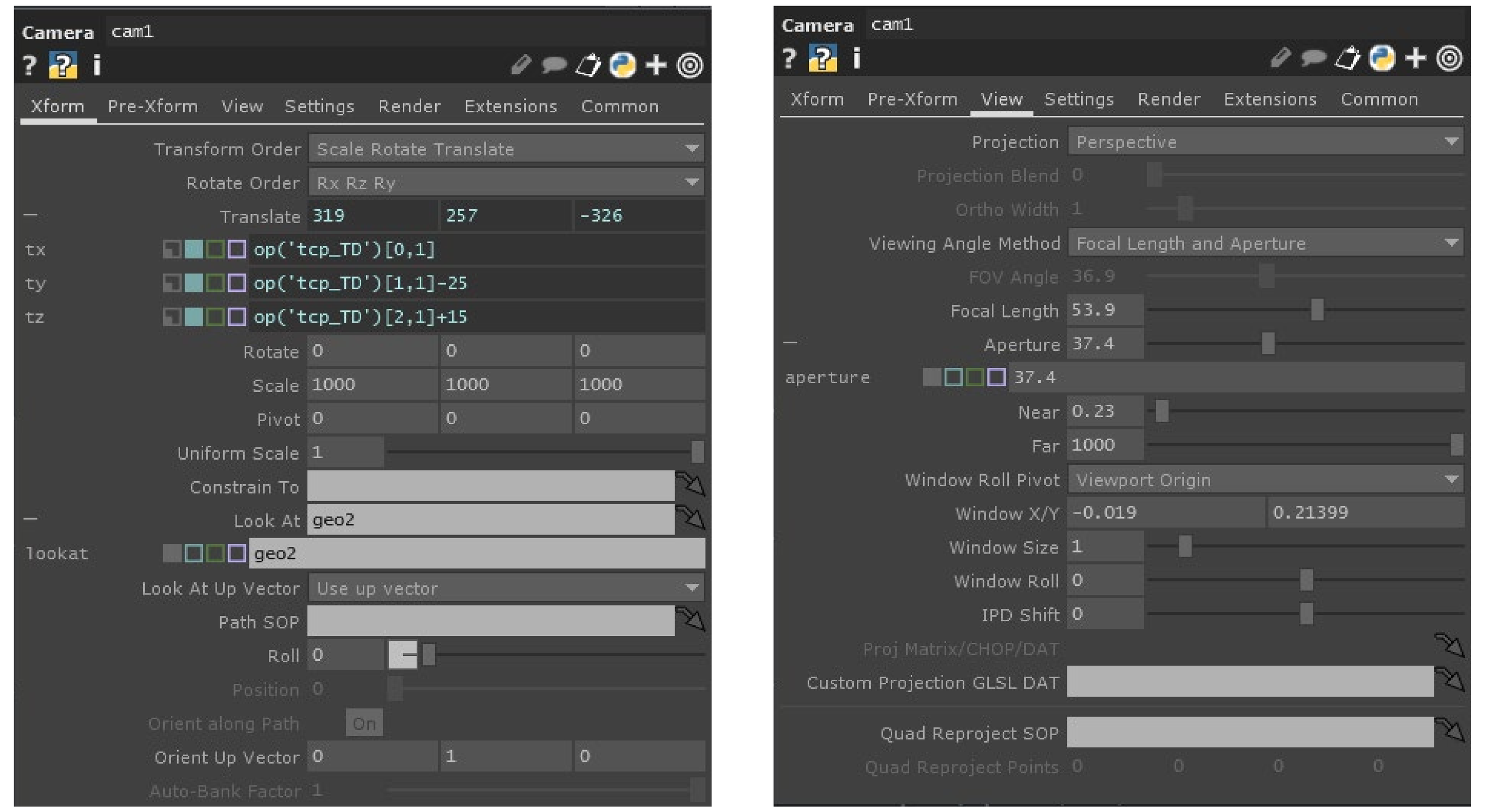
The virtual camera record the room with the same focal length and shift as the projector. Simultaneously, the recorded image is streamed to the projector attached to the robot. In this way, the projector would seem to map the space as it is. Finally, I add a video as the texture for the 3d surfaces in order to augment the existing space with a real-time projection.

```
def onReceiveOSC(dat, rowIndex, message, bytes,
        timeStamp, address, args, peer):
       #tcp in TouchDesigner
       op('tcp_TD')[0, 1] = args[0]
       op('tcp_TD')[1, 1] = args[2]
       op('tcp_TD')[2, 1] = -float(args[1])

       #target in TouchDesigner
       op('target_TD')[0, 1] = args[3]
       op('target_TD')[1, 1] = args[5]
       op('target_TD')[2, 1] = -float(args[4])
```

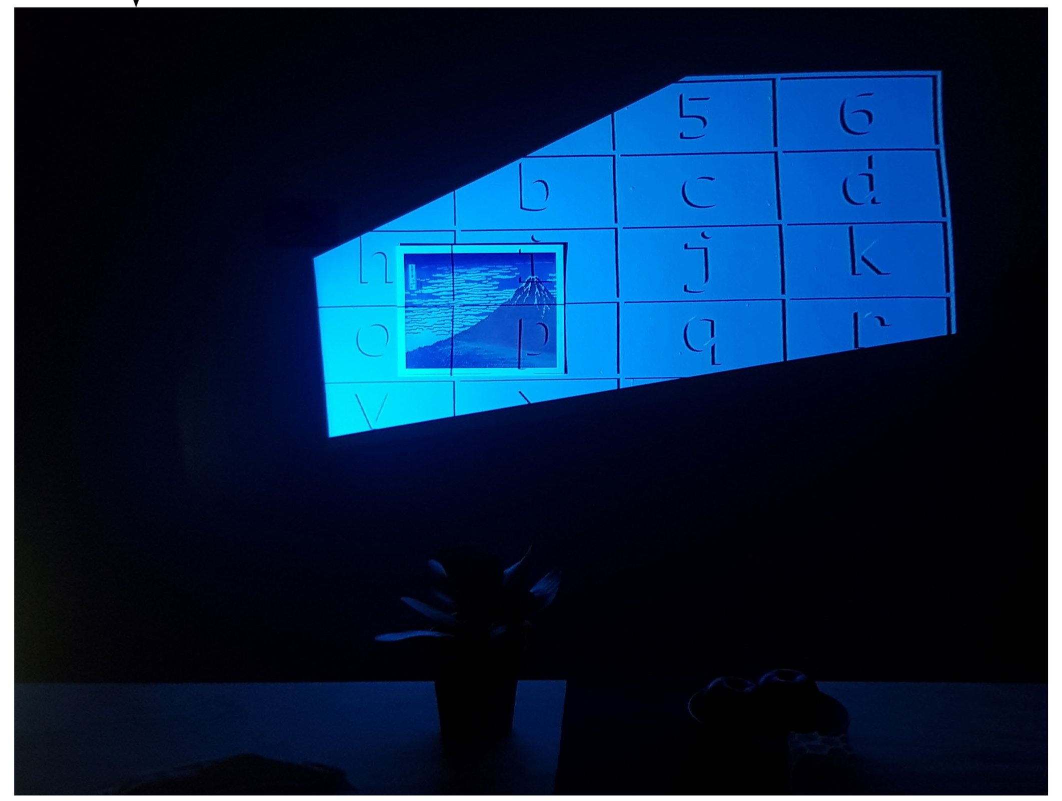
Machina Bridge





Camera settings

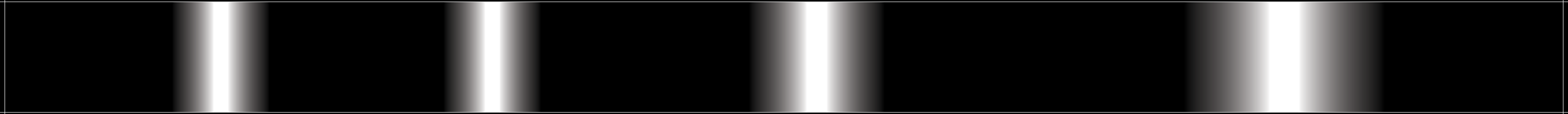
Projector
(1280 x 800px)



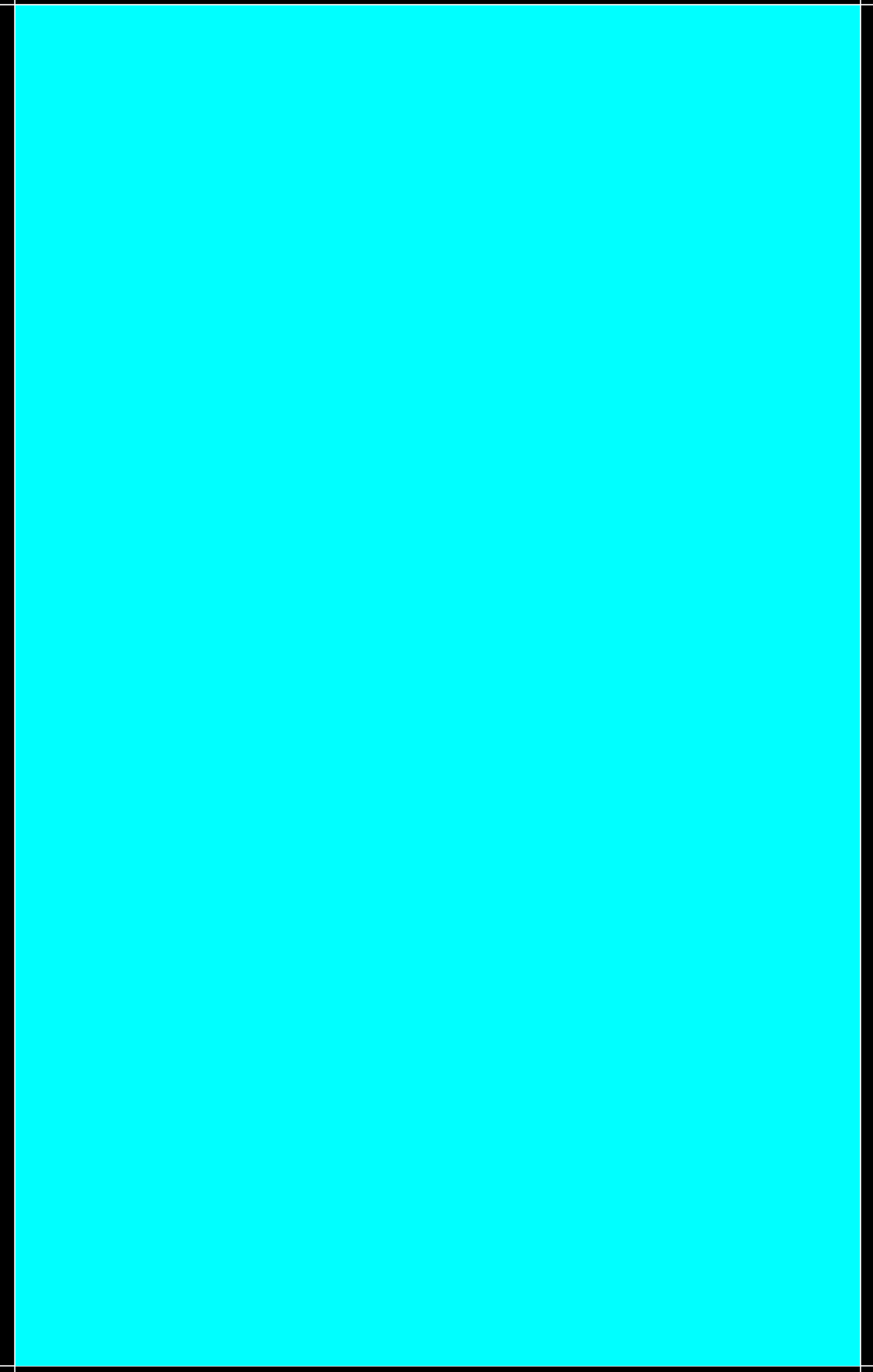
Test with numeric pattern (without robot)

VIDEO COMPOSITION

Strobe Gradient Pattern
TouchDesigner (Sine wave)
[ceiling]

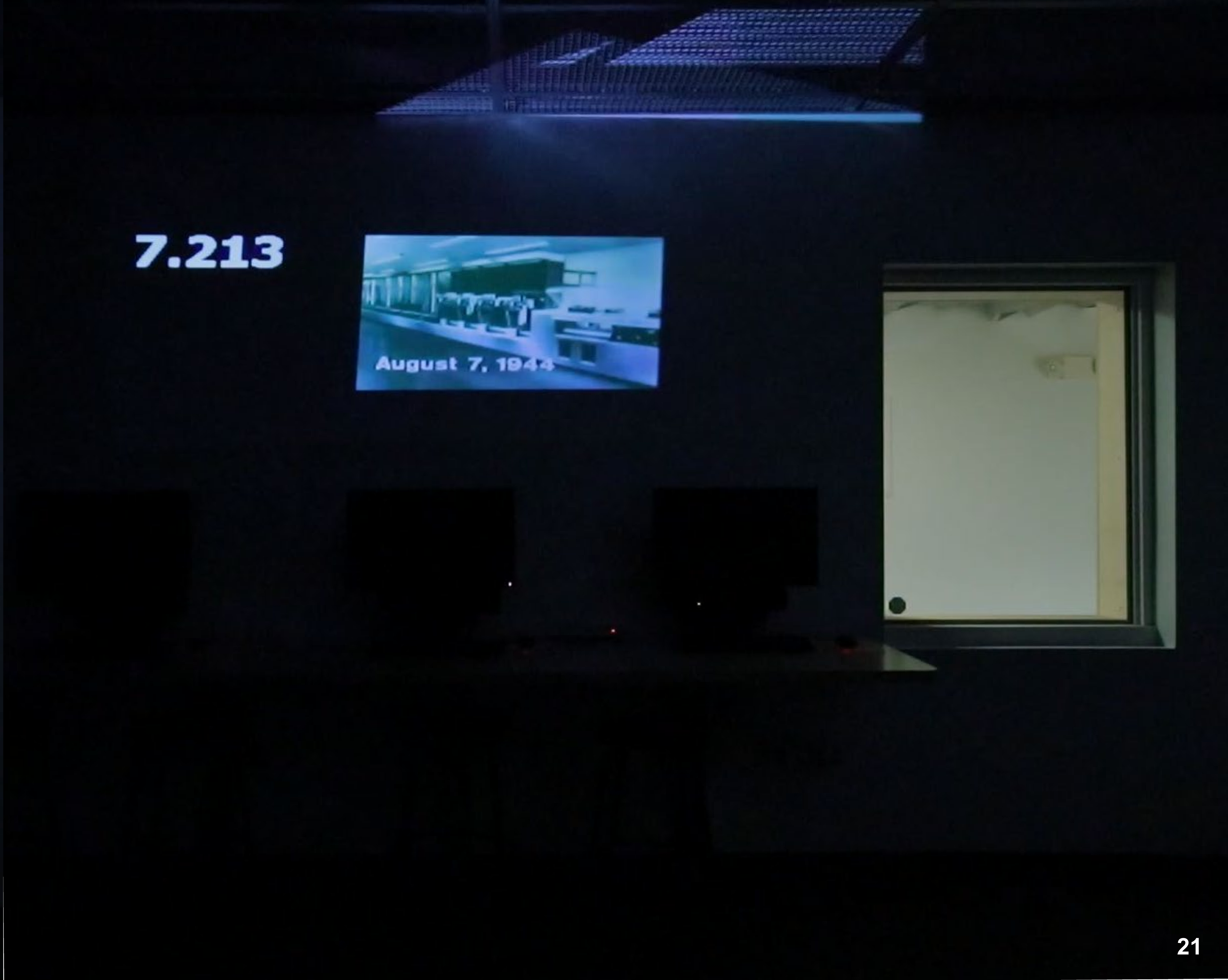


Video footage from Personal Computer
Companies advertising campaigns
Premiere (5814 x 1850 px)
[main wall]



Color Gradient
[adjacent wall]


MONSANTO
vimeo.com/378980300


7.213
August 7, 1944