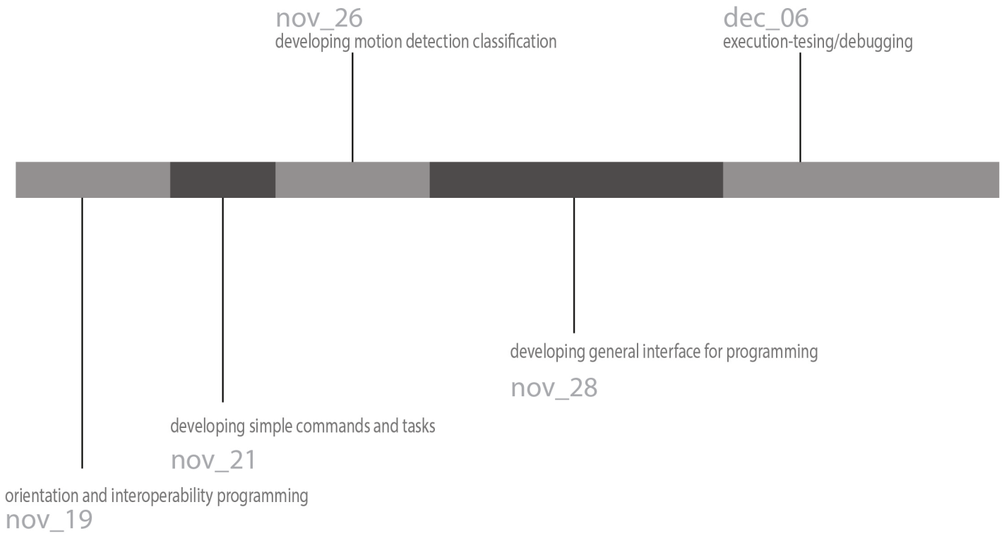
eMULATE AEC

cloud - based gestural co-bot programming
and management interface for AEC

Vishal Vaidhyanathan
Twisha Raja
Aprameya Pandit

Carnegie Mellon University

Today the AEC industries increasingly demand more flexible and agile working systems. Through this project , we present a human-robot collaborative cloud interface, which is based on task-level-programming and gesture-based teaching. This allows synchronization of tasks through the cloud interface and allows replication of tasks through gestures without the knowledge of coding or programming.



DEBRIEF

- LITERATURE REVIEW
- EQUIPMENT PRECEDENTS
  - PREFERRED EQUIPMENT
  - PREFERRED DOMAIN

  → BALL PARKED → W.R.T. A.E.C.
- REQUIREMENT ANALYSIS → S.W.O.T.
- TASK FLOW → SOFTWARE+EQUIPMENT USED AT EVERY STAGE
  - BACKEND
  - FRONT END
  - USE-CASE SCENARIO
- TRAINING
- CLOUD STORAGE
- USE-CASE SCENARIO → BALL PARKED, SIMULATION → NORMAL CASE V.S. EMULATION

IMPLEMENTATION

- SCHEDULE OF PROJECT
- INITIAL WORKFLOWS → TEACHING
  - LOGIC AND IMPLEMENTATION
  - EXECUTION
- INTERFACE DESIGN → INTERFACE
  - PRECEDENTS
- SECONDARY WORKFLOWS
  - CODE LOGIC
  - EXECUTION TESTS
- WORKFLOWS IN HAL → PATH PLANNING
  - CODE LOGIC
  - EXECUTION TESTS

1. R.M. Voyles ; P.K. Khosla _ Gesture-based programming: a preliminary demonstration
https://ieeexplore.ieee.org/abstract/document/770058

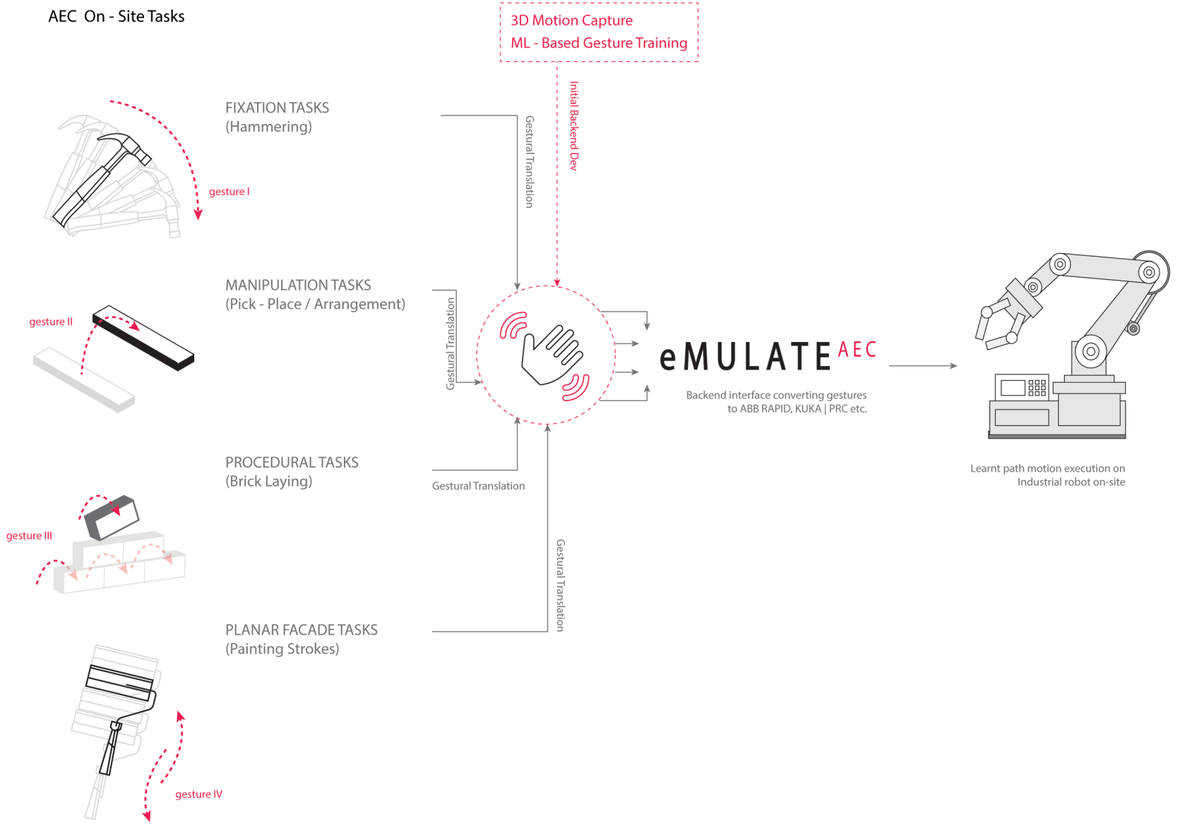
2. Joshua D. Bard; David Blackwood; Nidhi Sekhar, Brian Smith _ Decorative Robotic Plastering: A Case Study of Real-Time Human Machine-Collaboration in High-Skill Domains
http://papers.cumincad.org/data/works/att/ecaade2015_74.content.pdf

3. Joshua D. Bard; David Blackwood; Nidhi Sekhar, Brian Smith _ Reality is interface: Two motion capture case studies of human–machine collaboration in high-skill domains
http://papers.cumincad.org/cgi-bin/works/paper/ijac201614408

4. Thibault Schwartz, Sebastian Andraos, Jonathan Nelson, Chris Knapp, Bertrand Arnold _Towards On-site Collaborative Robotics
https://link.springer.com/content/pdf/10.1007%2F978-3-319-26378-6_31.pdf

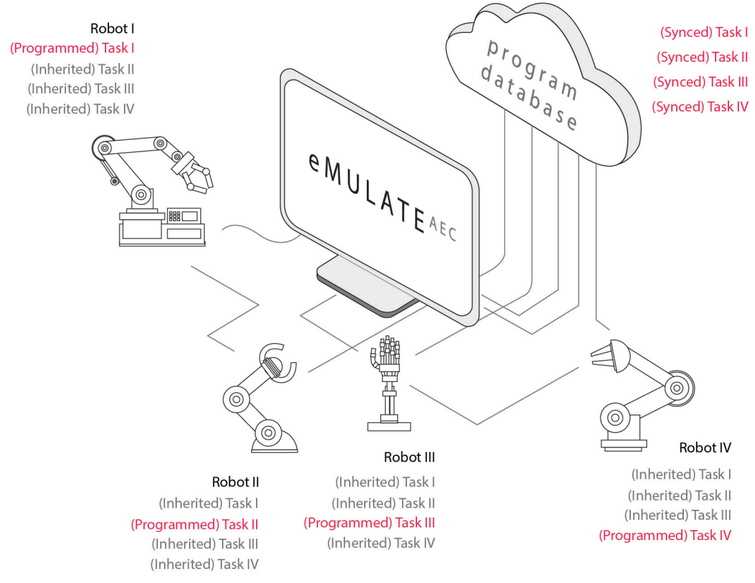
GESTURE BASED MIMICKING

PROGRAMMING INTERFACE



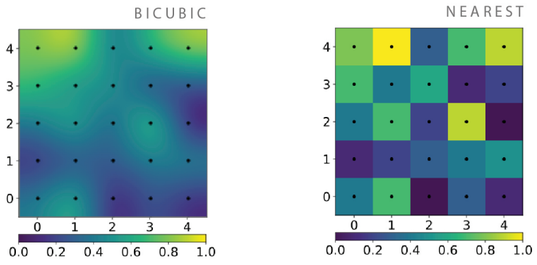
SYNCHRONOUS CLOUD INHERITANCE

COLLABORATIVE SWARM PUSH-PULL MANAGEMENT

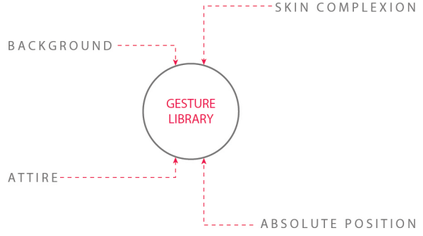


Tasks taught individually to robots are synchronized through the interface onto cloud. The tasks can then be inherited in any other robots on the network to perform these taught tasks from cloud.

Considerations For Training



Implementation of the KNN algorithm for supervised learning where relational value-based pixel clusterization for detection for gesture recognition is used.



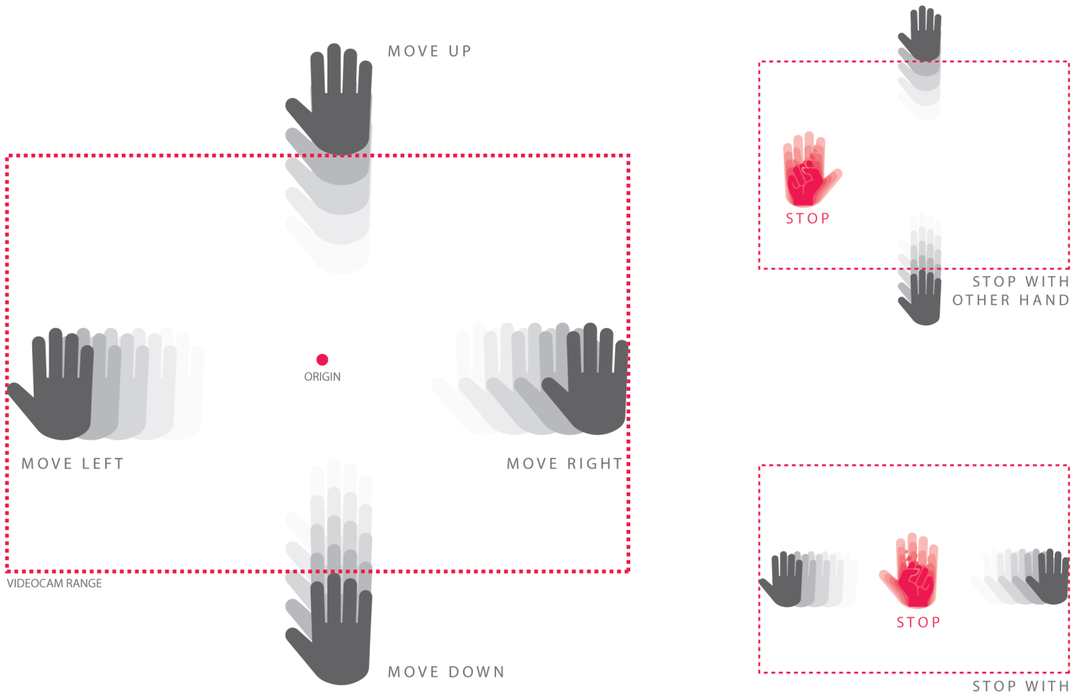
The library is a repertoire of gestures recorded by taking into consideration factors like the colour of the background, attire, skin complexion, absolute position of user wrt the camera and the position of gesture within the recording screen.

Programming on Processing

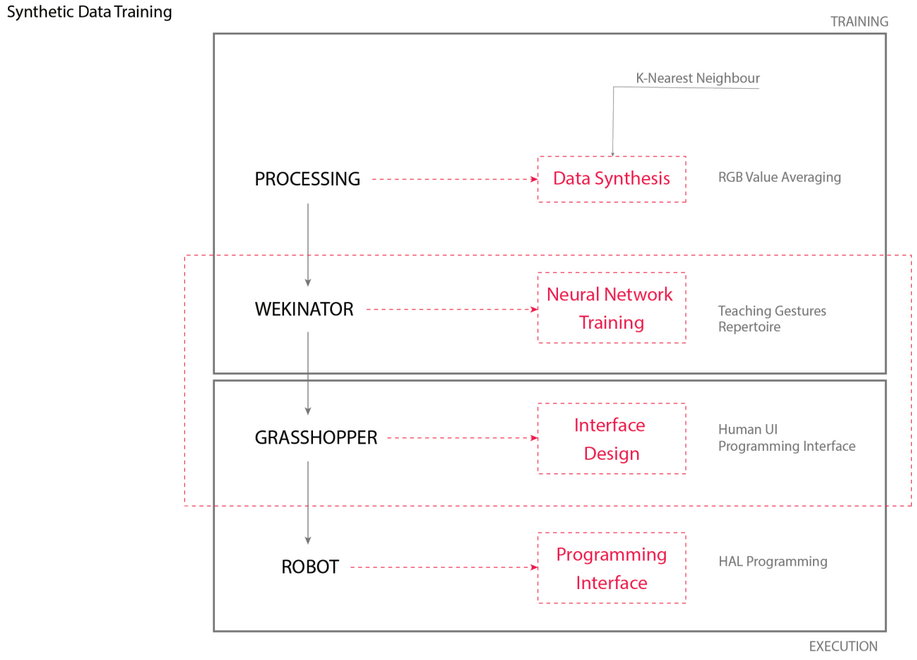
VishTwishWebcamStreamer

```
import processing.video.*;
import oscP5.*;
import netP5.*;

int numPixelsOrig;
int numPixels;
boolean first = true;

int boxWidth = 16;
int boxHeight = 12;

int numHoriz = 640/boxWidth;
int numVert = 480/boxHeight;

color[] downPix = new color[numHoriz * numVert];


Capture video;

OscP5 oscP5;
NetAddress dest;

void setup() {
size(640, 480, P2D);
video = new Capture(this, 640, 480);
video.start();

numPixelsOrig = video.width * video.height;
loadPixels();
noStroke();

oscP5 = new OscP5(this,9000);
dest = new NetAddress("127.0.0.1",6448);

}

void draw() {

if (video.available() == true) {
video.read();
video.loadPixels();
int boxNum = 0;
int tot = boxWidth*boxHeight;
for (int x = 0; x < 640; x += boxWidth) {
for (int y = 0; y < 480; y += boxHeight) {
float red = 0, green = 0, blue = 0;

for (int i = 0; i < boxWidth; i++) {
for (int j = 0; j < boxHeight; j++) {
int index = (x + i) + (y + j) * 640;
red += red(video.pixels[index]);
green += green(video.pixels[index]);
blue += blue(video.pixels[index]);
}
}
downPix[boxNum] =  color(red/tot, green/tot, blue/tot);
fill(downPix[boxNum]);


int index = x + 640*y;
red += red(video.pixels[index]);
green += green(video.pixels[index]);
blue += blue(video.pixels[index]);
rect(x, y, boxWidth, boxHeight);
boxNum++;
}
}
if(frameCount % 2 == 0)
sendOsc(downPix);
}
first = false;
fill(0);
text("Sending 100 inputs to port 6448 using message /wek/inputs", 10, 10);
}

float diff(int p, int off) {
if(p + off < 0 || p + off >= numPixels)
return 0;
return red(video.pixels[p+off]) - red(video.pixels[p]) +
green(video.pixels[p+off]) - green(video.pixels[p]) +
blue(video.pixels[p+off]) - blue(video.pixels[p]);
}

void sendOsc(int[] px) {
OscMessage msg = new OscMessage("/wek/inputs");
for (int i = 0; i < px.length; i++) {
msg.add(float(px[i]));
}
oscP5.send(msg, dest);
}
```

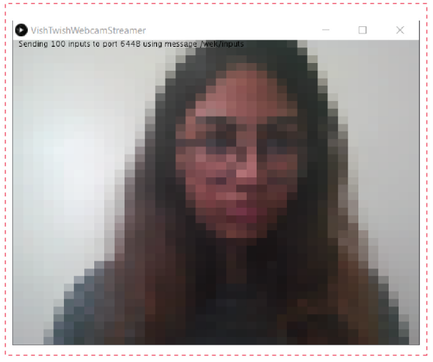
TRAINING
TEACHING GESTURES

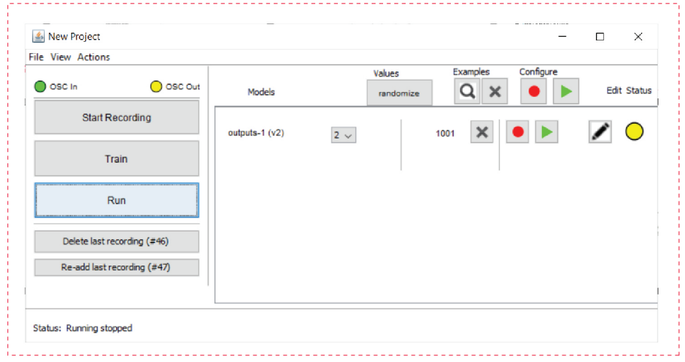


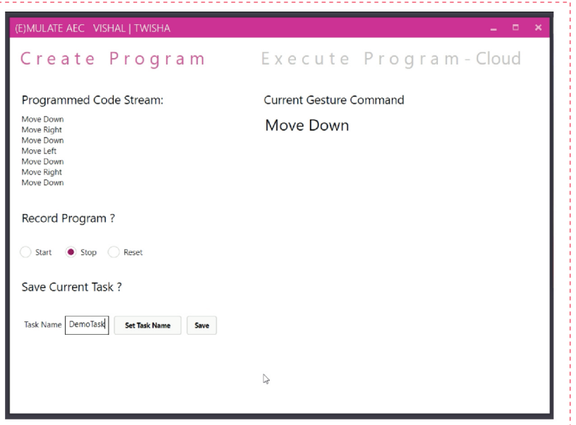
INTERFACE DESIGN
OSC PIPELINE CYCLE

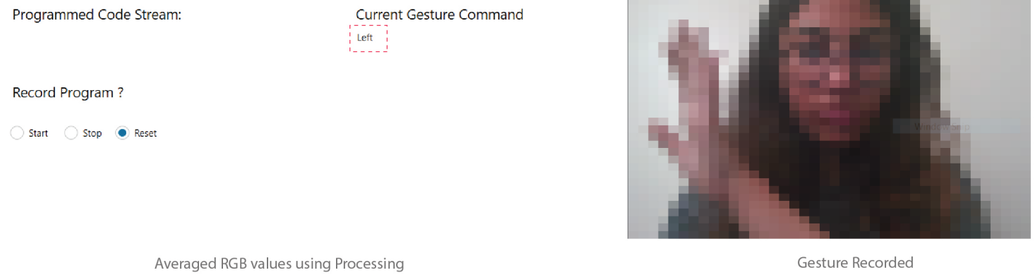


In this project, software integration takes place at 4 main stages.
Firstly, Processing converts the webcam stream into pixelated output. The neural network is trained for averaging RGB values in Wekinator using the KNN algorithm. This data is then received by grasshopper which converts received signals into robotic workflows. The interface allows the visualisation of tasks as well as the creates toolpaths for the taught tasks. Eventually, HAL produces the RAPID code which is received by the robot for task execution.



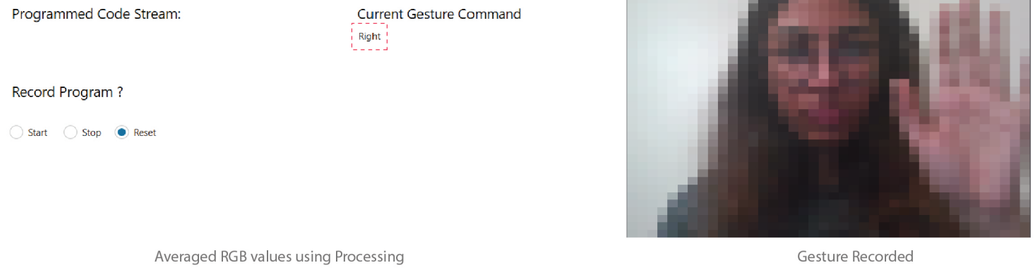






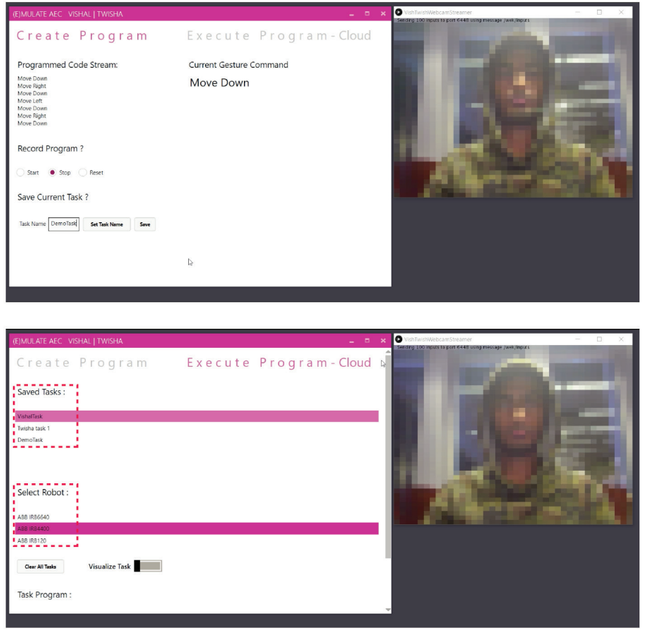
Averaged RGB values using Processing

Gesture Recorded



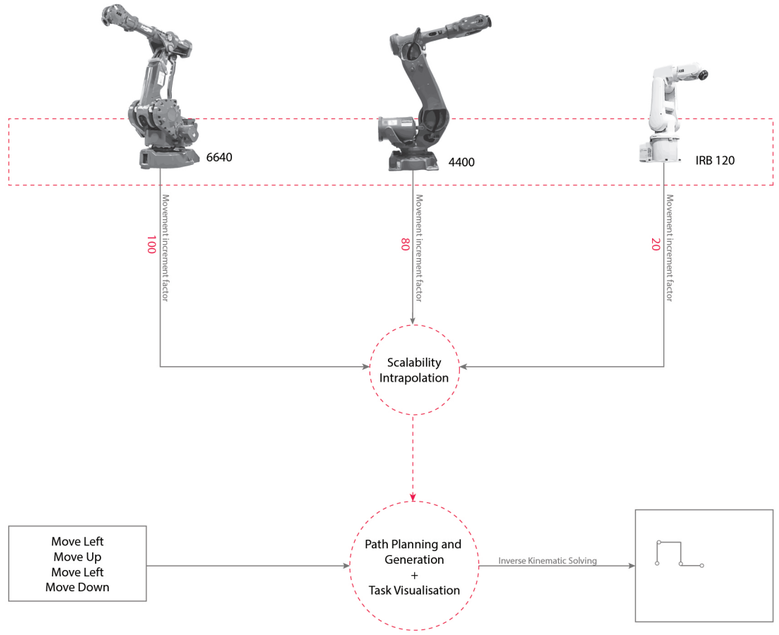
Averaged RGB values using Processing

Gesture Recorded

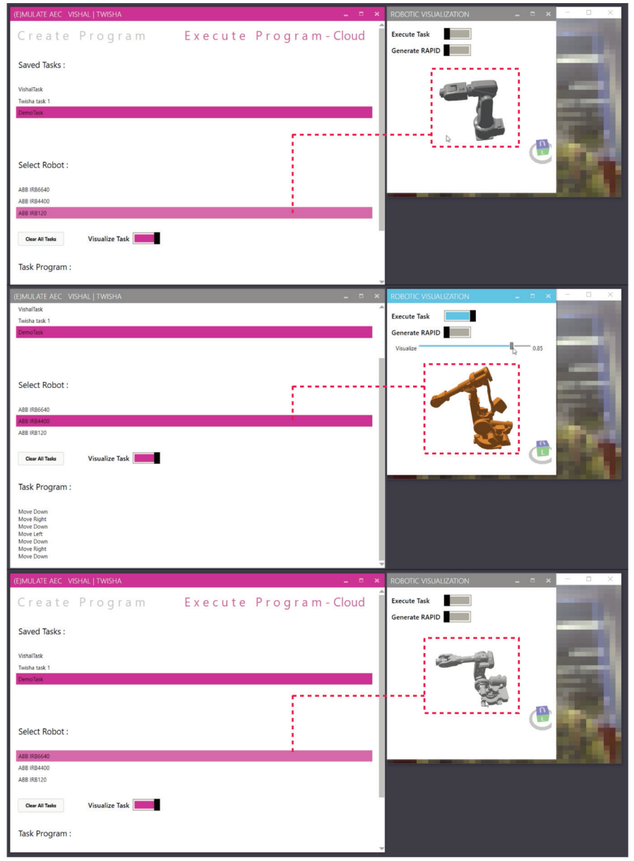


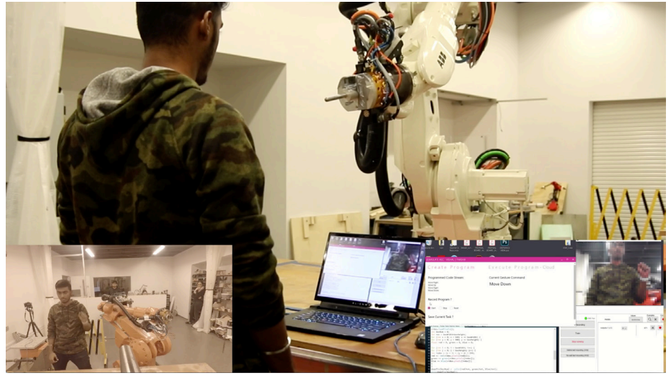
The two parts of the interface include creating a program and executing of the program. The user has the option to save the recorded program and name it. The latter includes selection of pre-saved tasks and selection of robot for task execution.Tasks can be visualised beforehand with the robot visualization option

Toolpath Porting/ Scalabilty



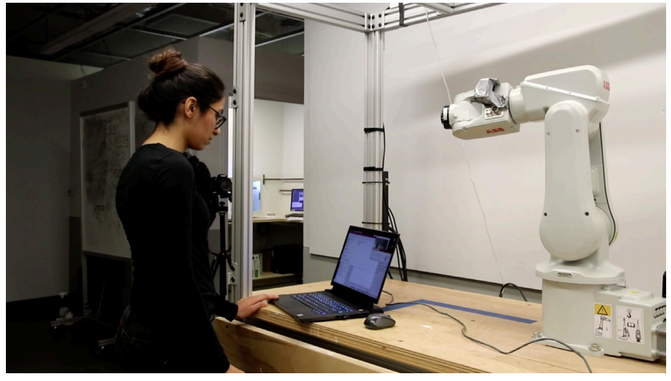
Scalability allows the same task to be replicated by different robots. Network based web sockets and OSC based data streaming allows interoperability of robots.





Task 1 taught to Robot 1

Task 1 executed by Robot 2



Task 2 taught to Robot 2

Task 2 executed by Robot 1