Processed Oasen

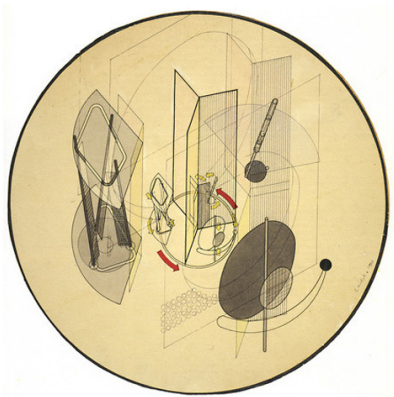
partial final presentation/

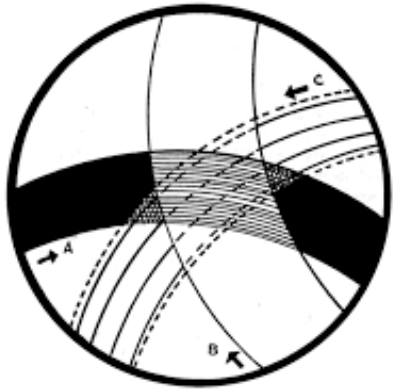
grayson daniel bailey

t e c h n i c a l
e x p e r i m e n t a t i o n

e p h e m e r a l
o u t p u t

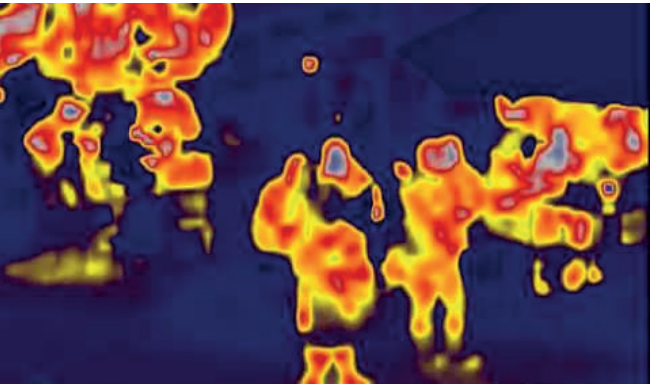
aggregatory
build of elements

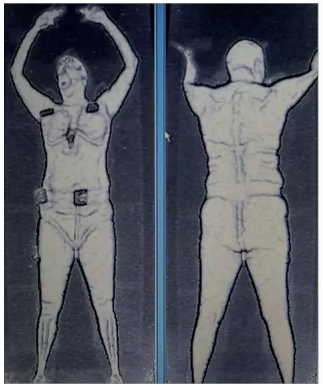


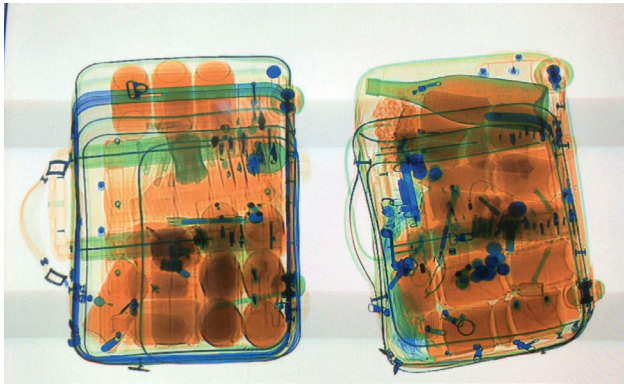


physical translation
between media

contemporary
technical vision



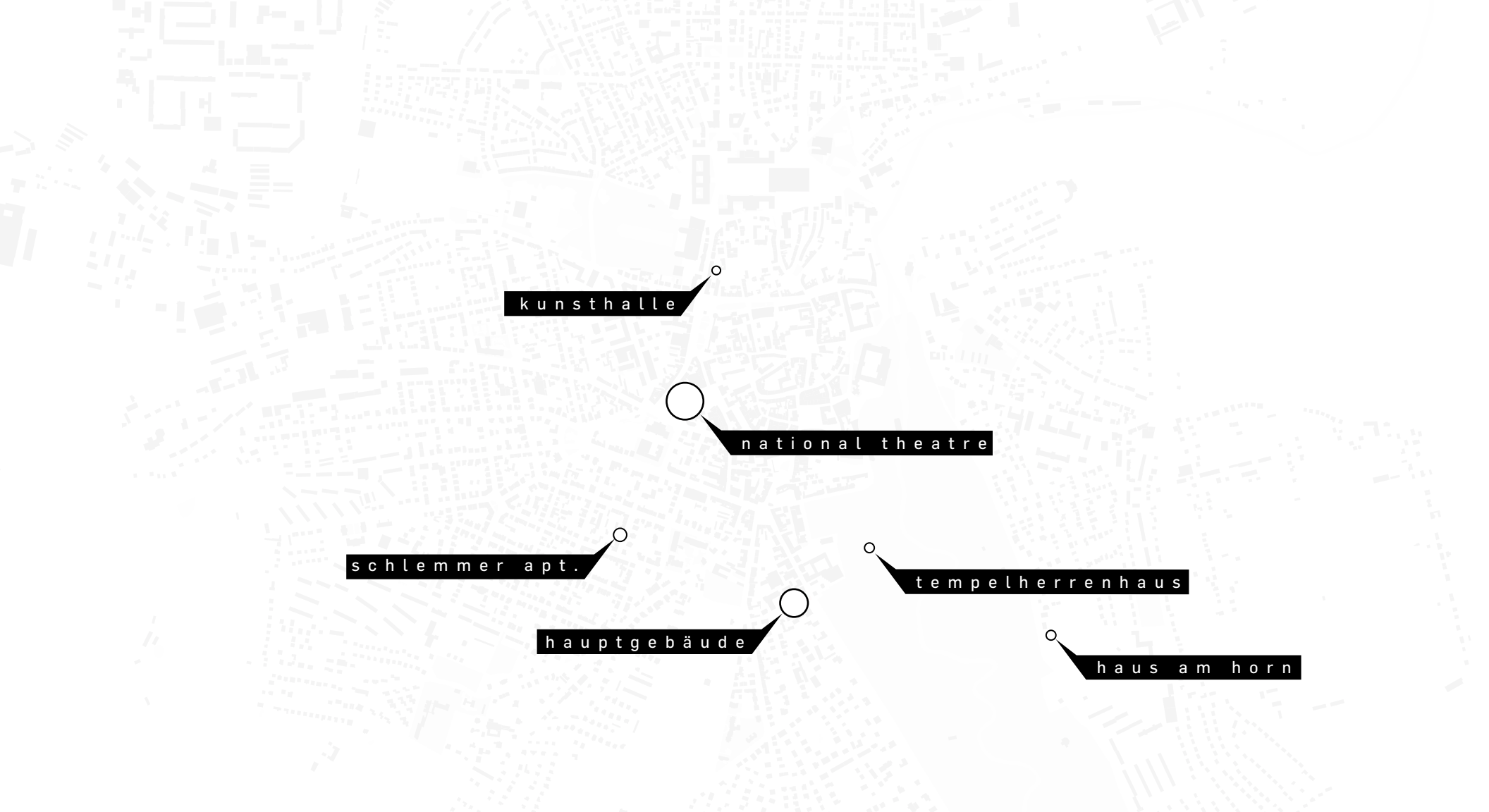


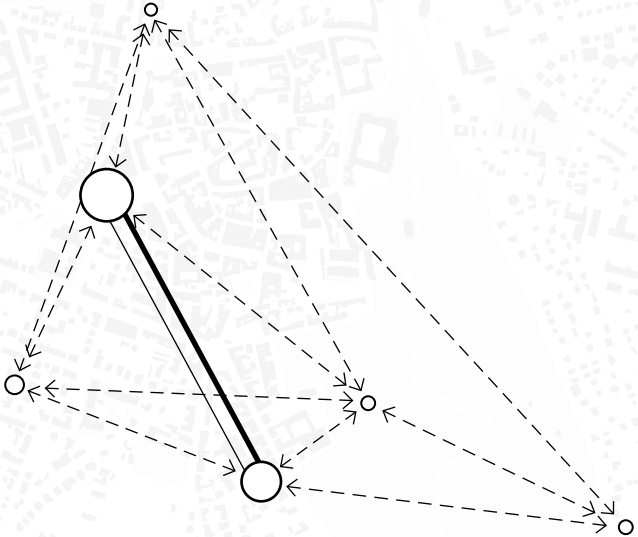


data inventory
& translation

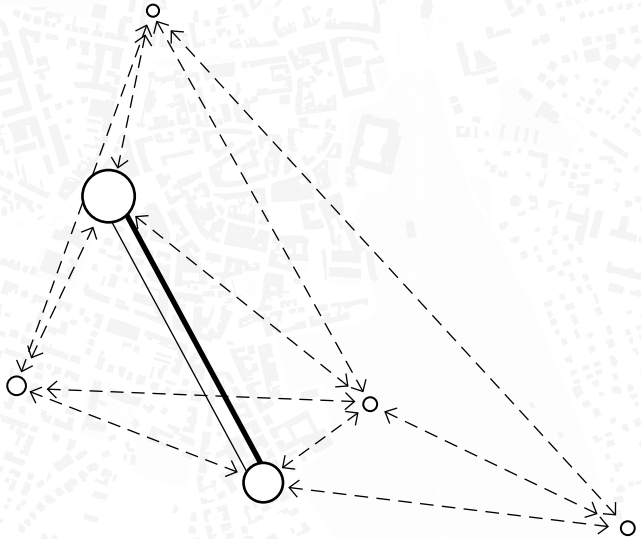
human relation
to nature & technology

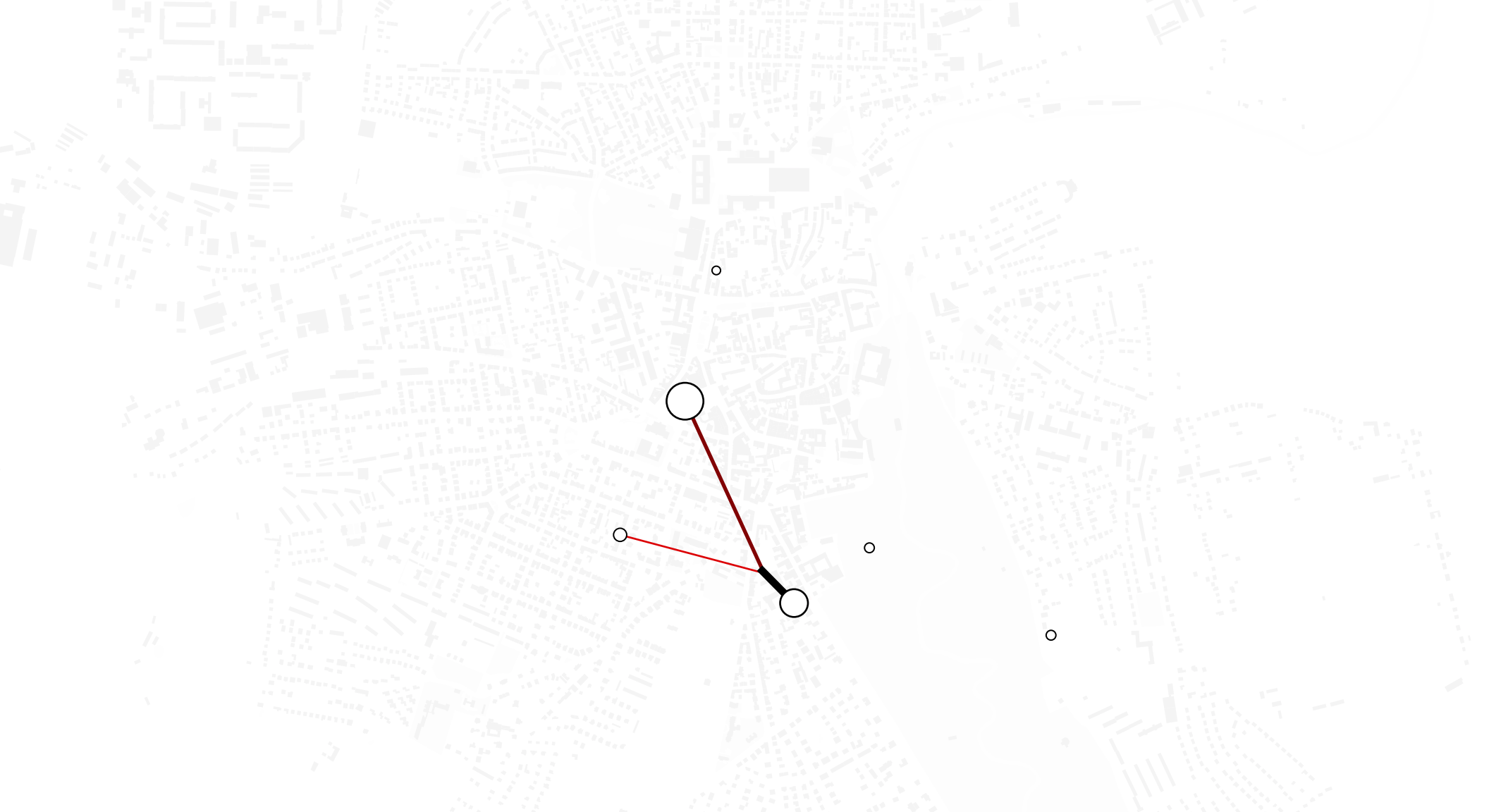
agrregatory fusion
& updated vision

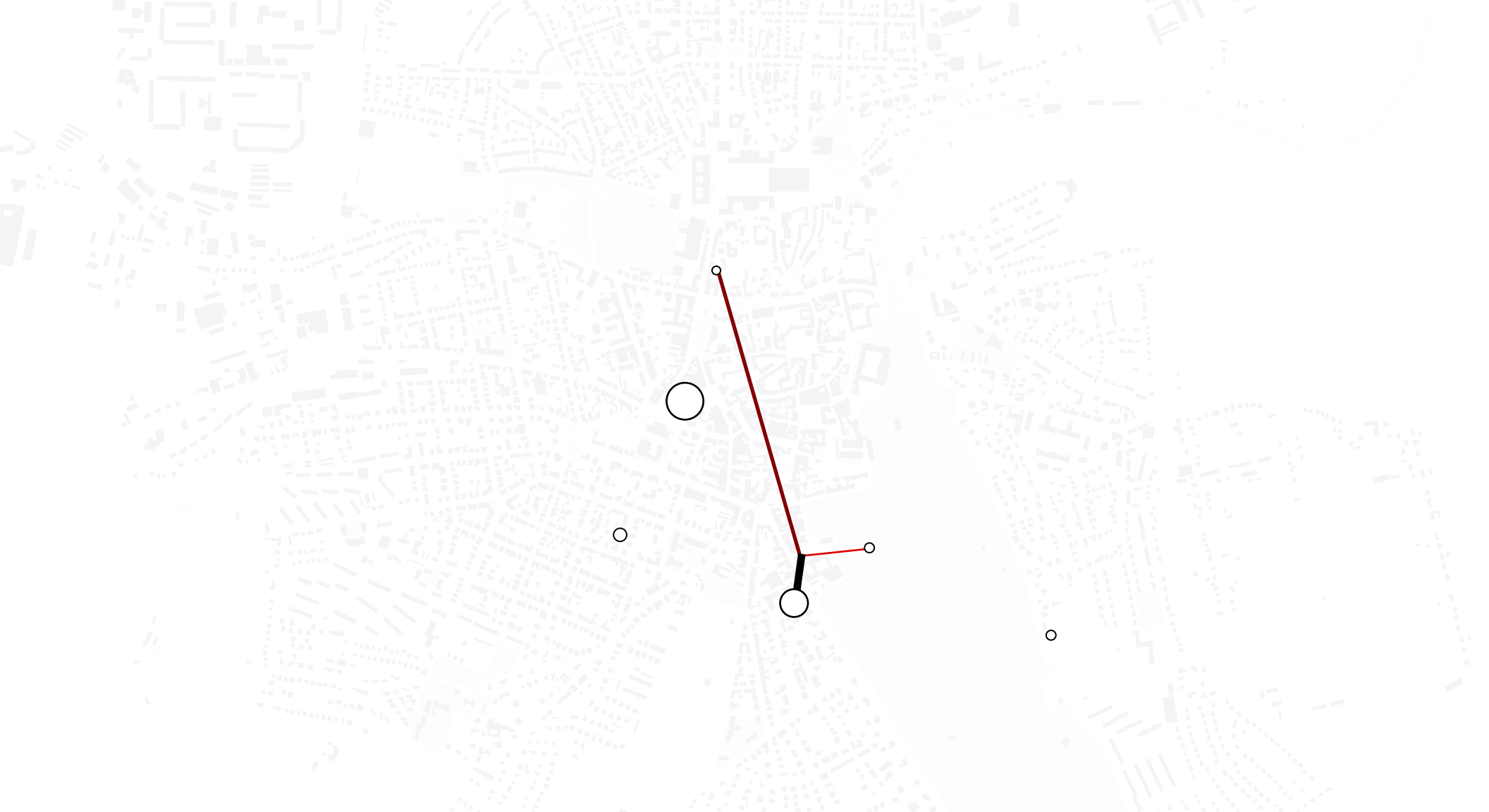

kunsthalle
national theatre
schlemmer apt.
tempelherrenhaus
hauptgebäude
haus am horn

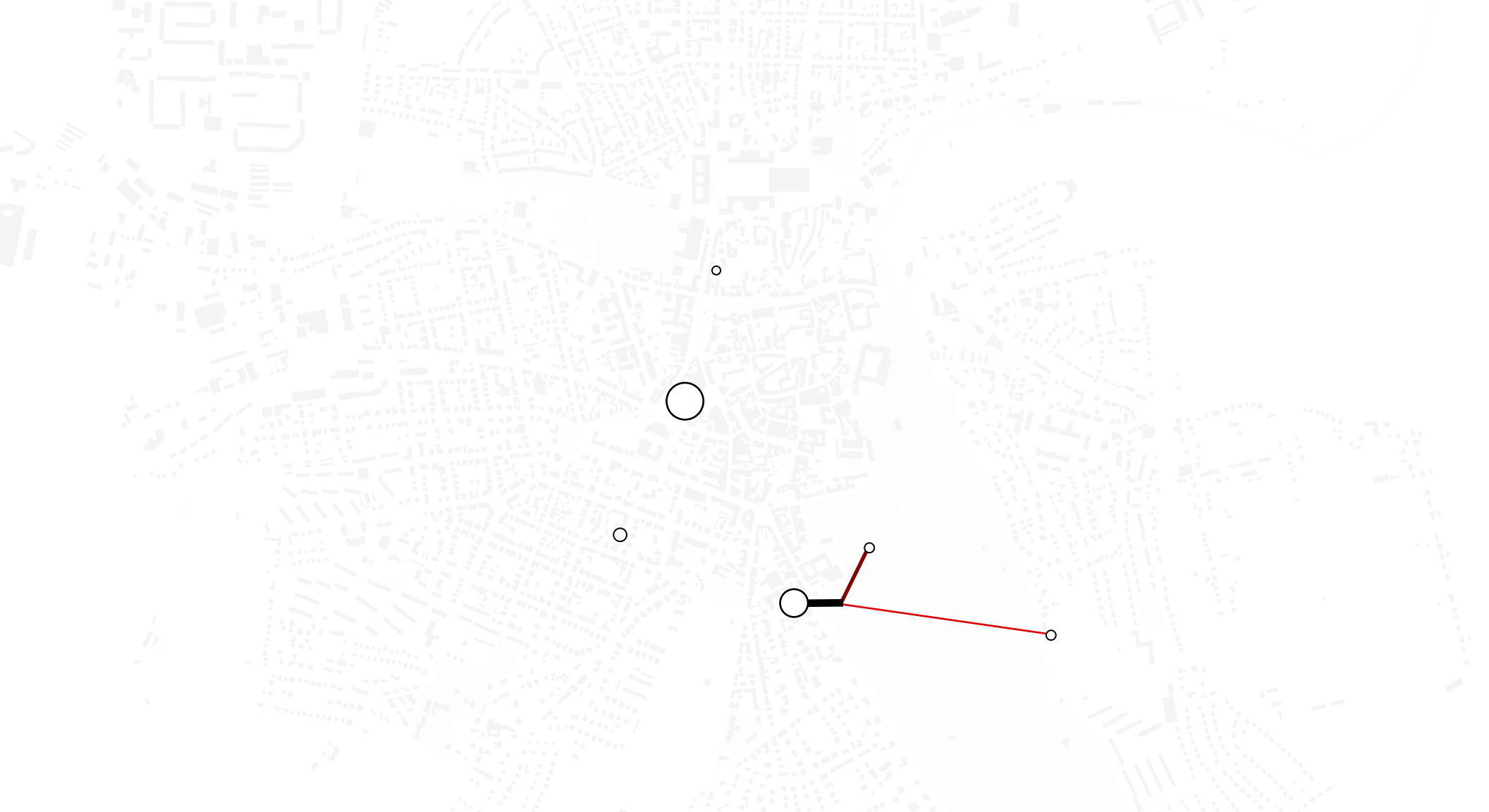


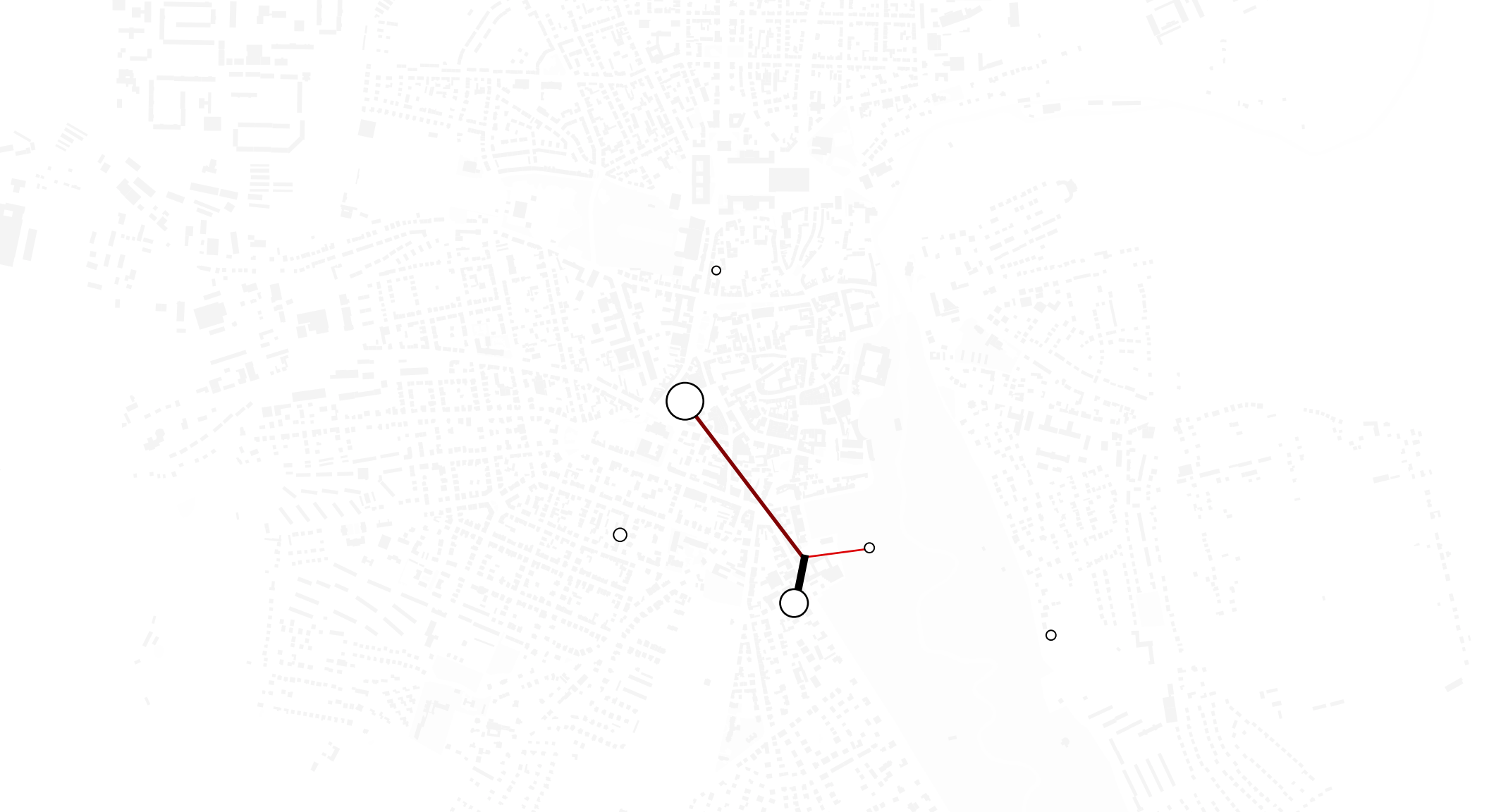
direct locational exchange //
ubiquitous

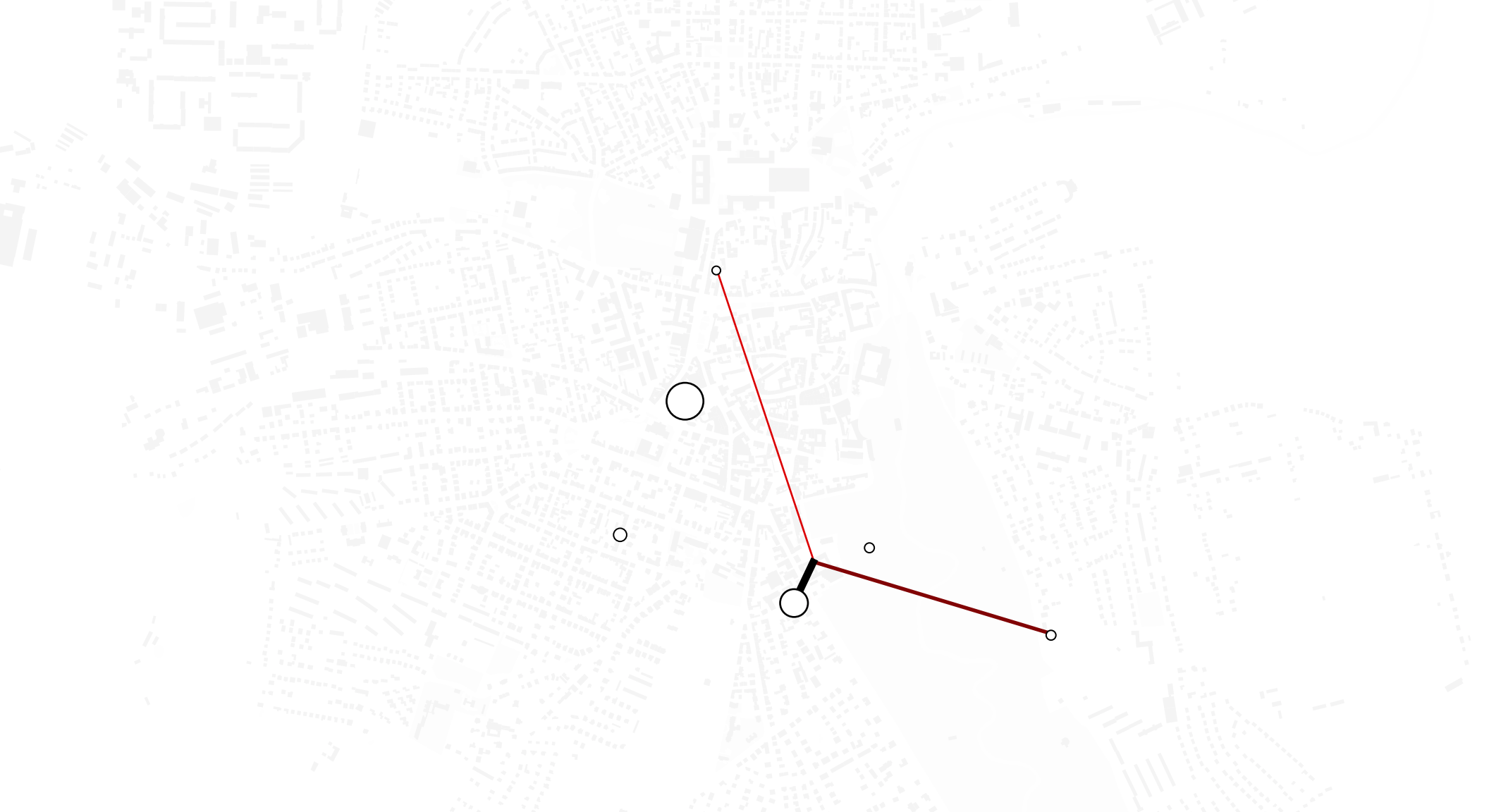






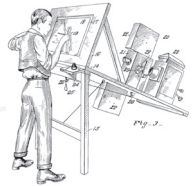


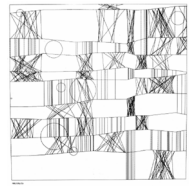


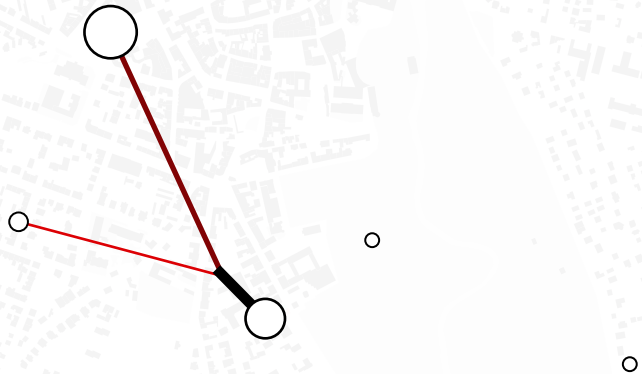


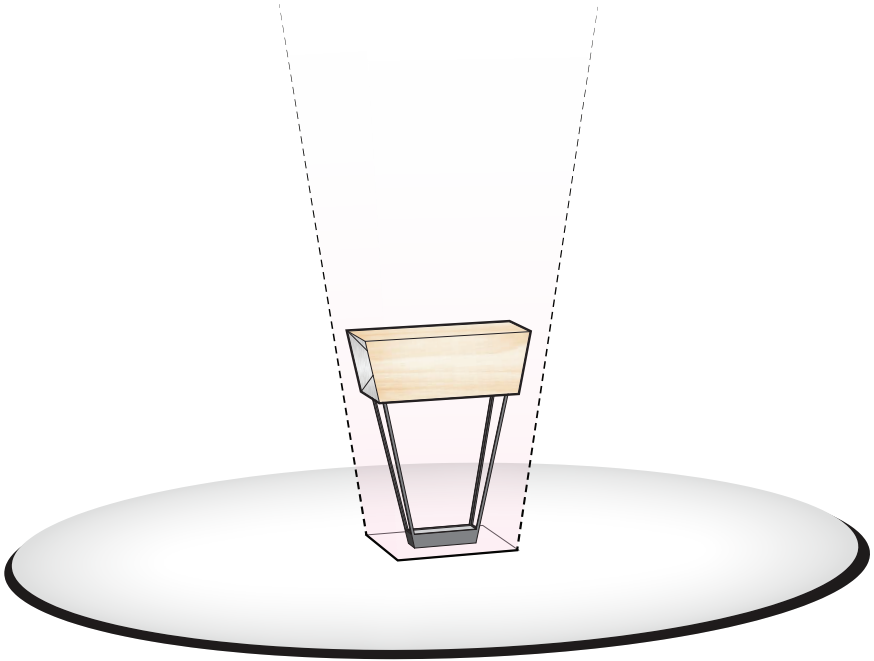
aggregate collection & synthesis output //

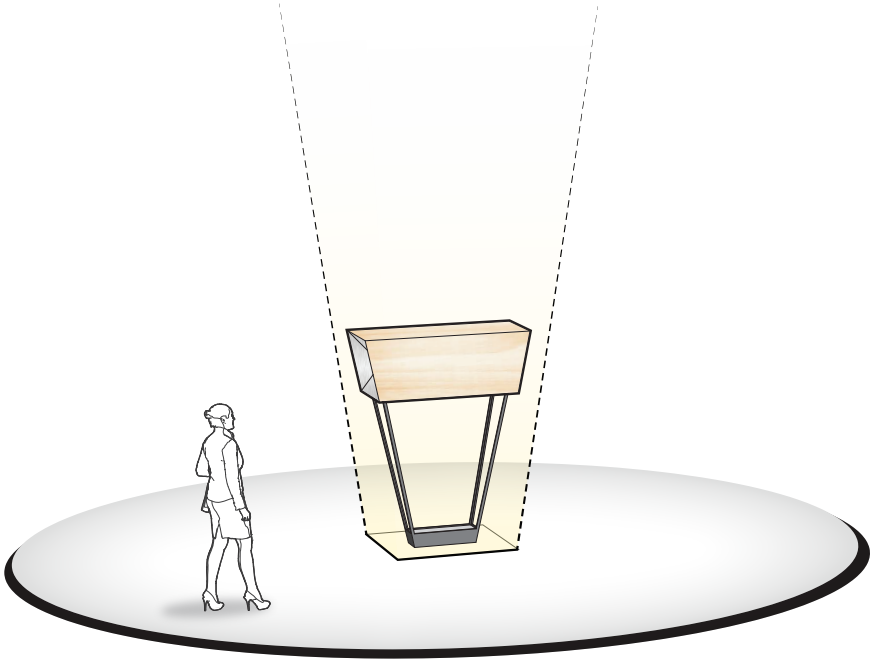
novel translation between local media

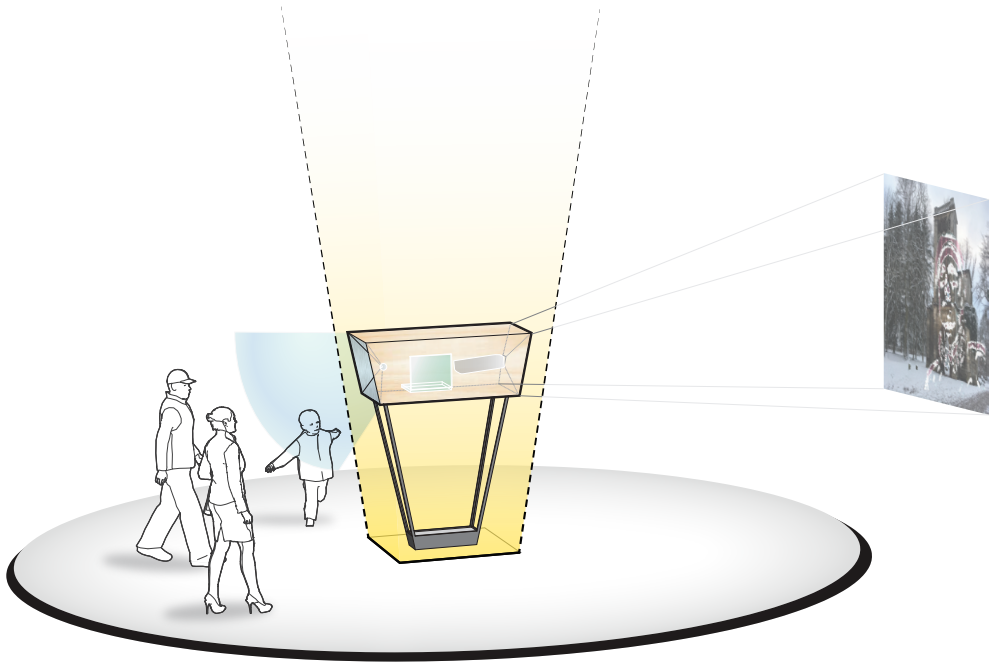


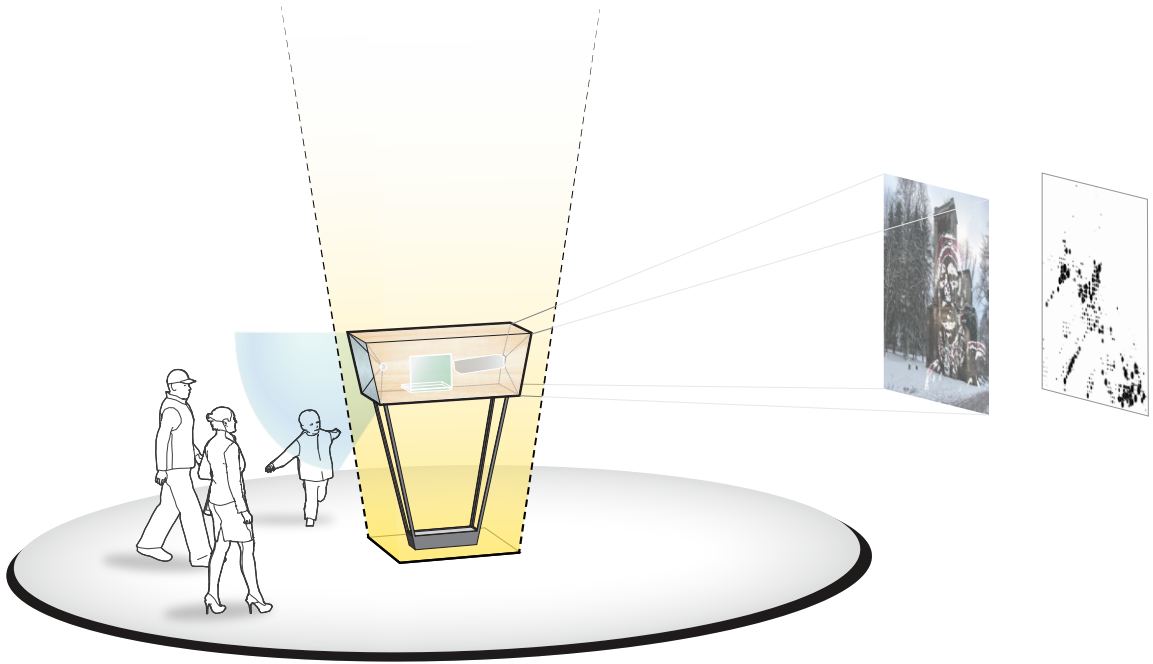


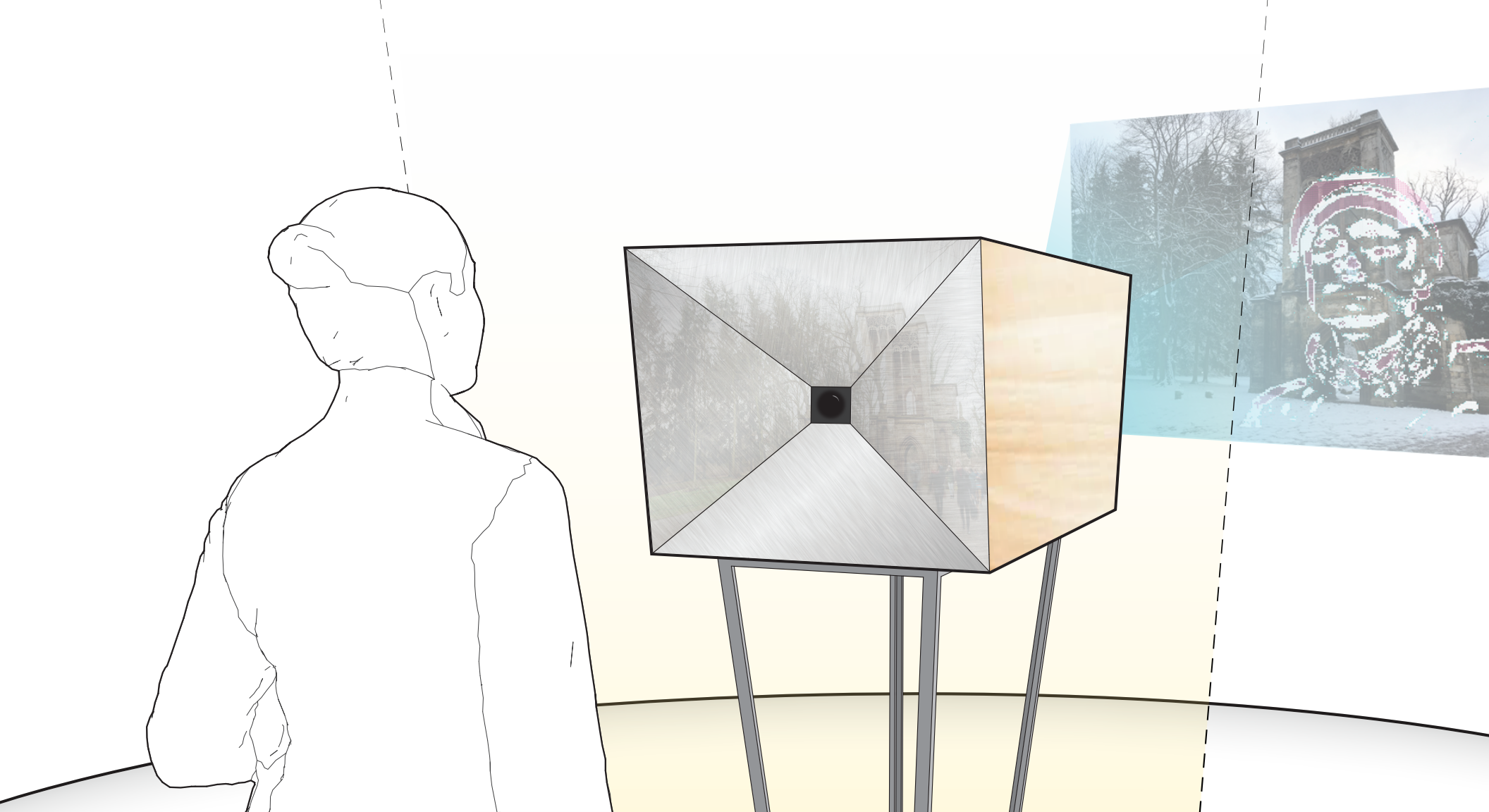


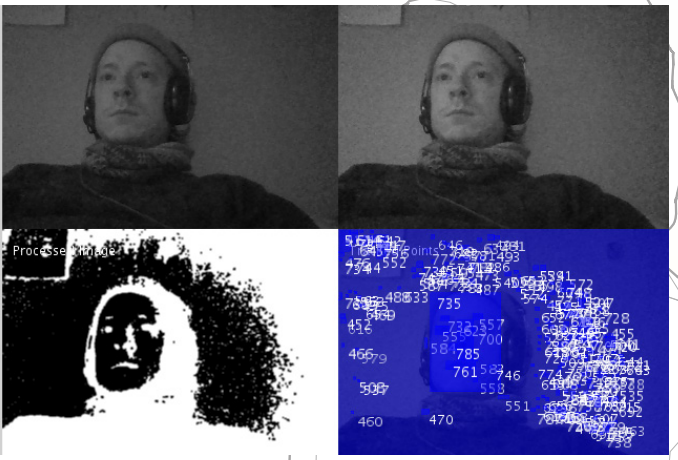


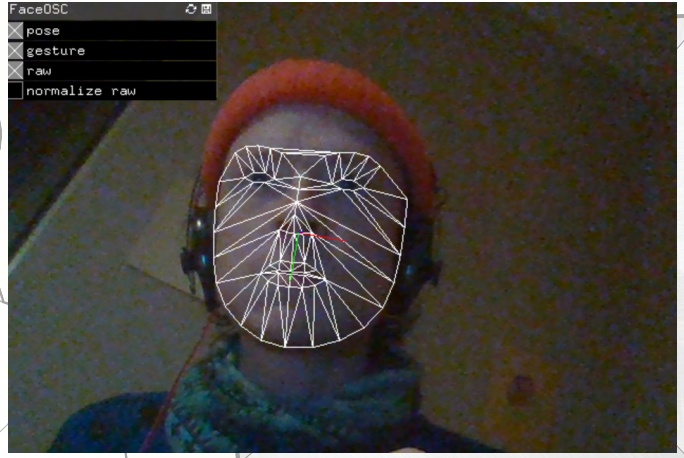


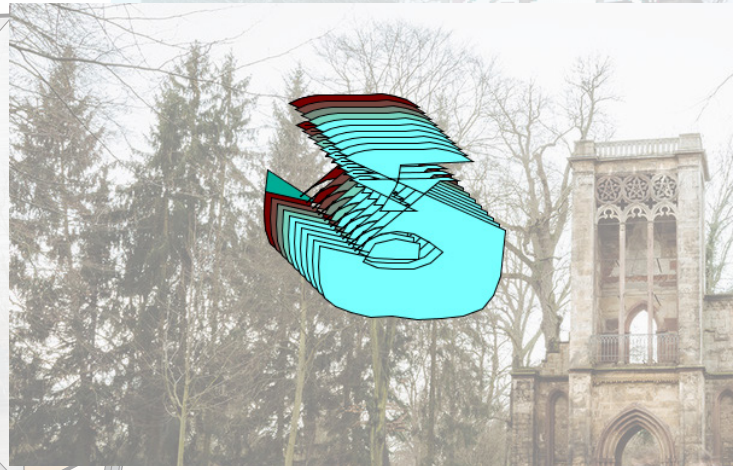


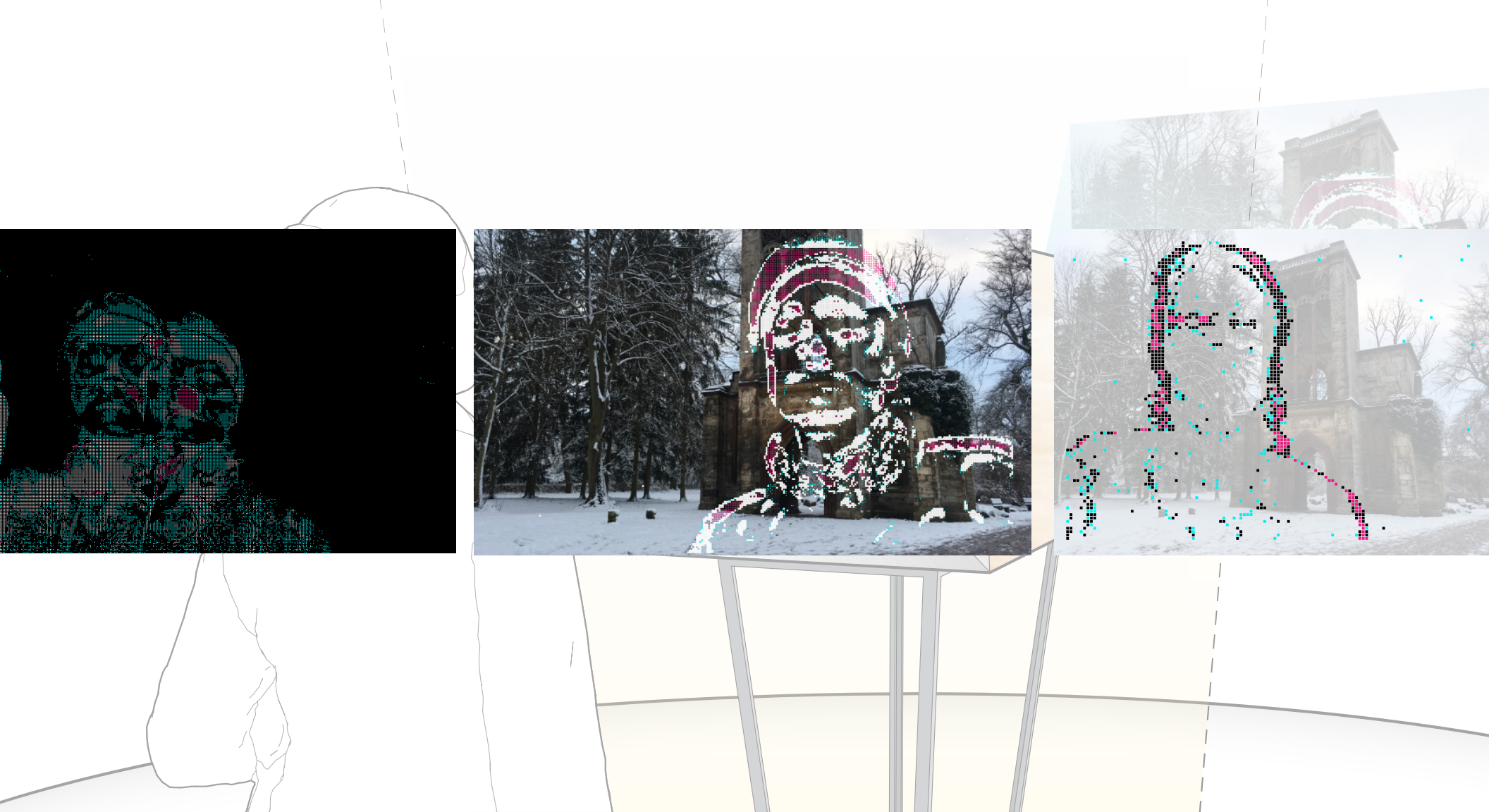


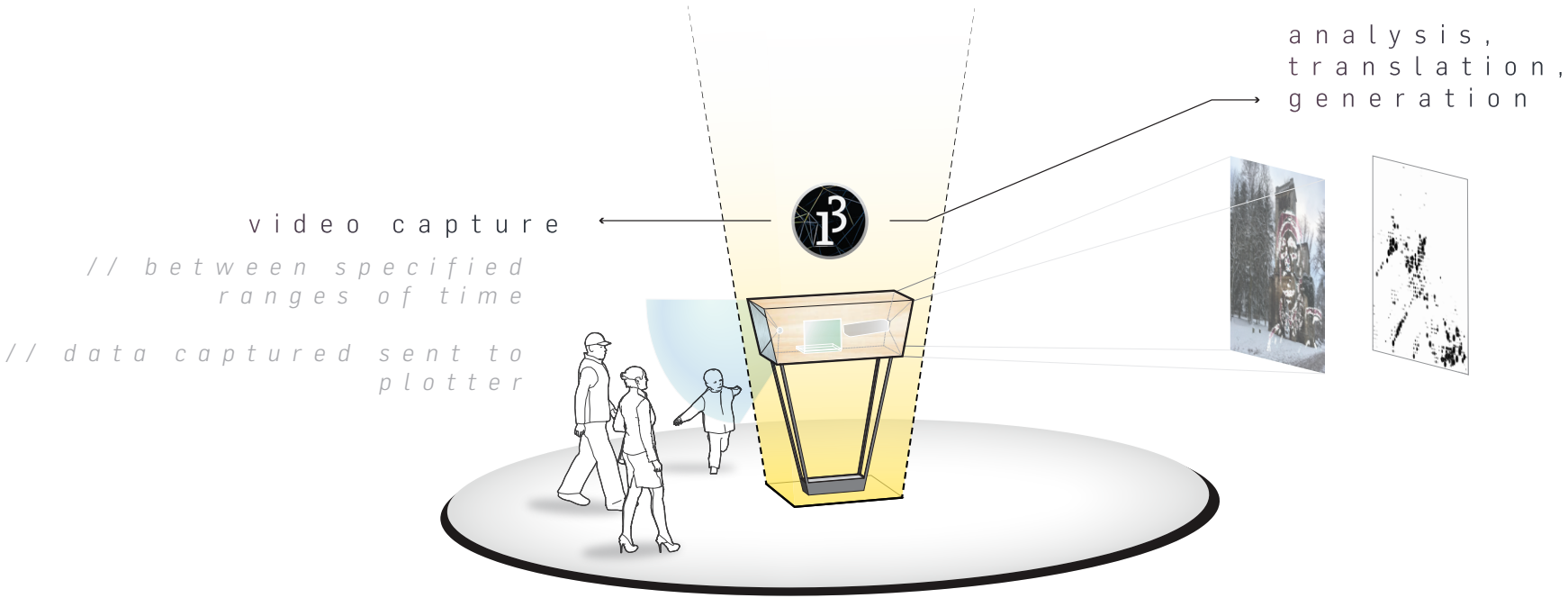


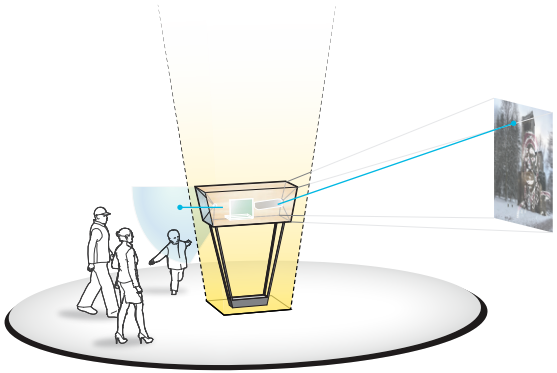

Processed Image
Points

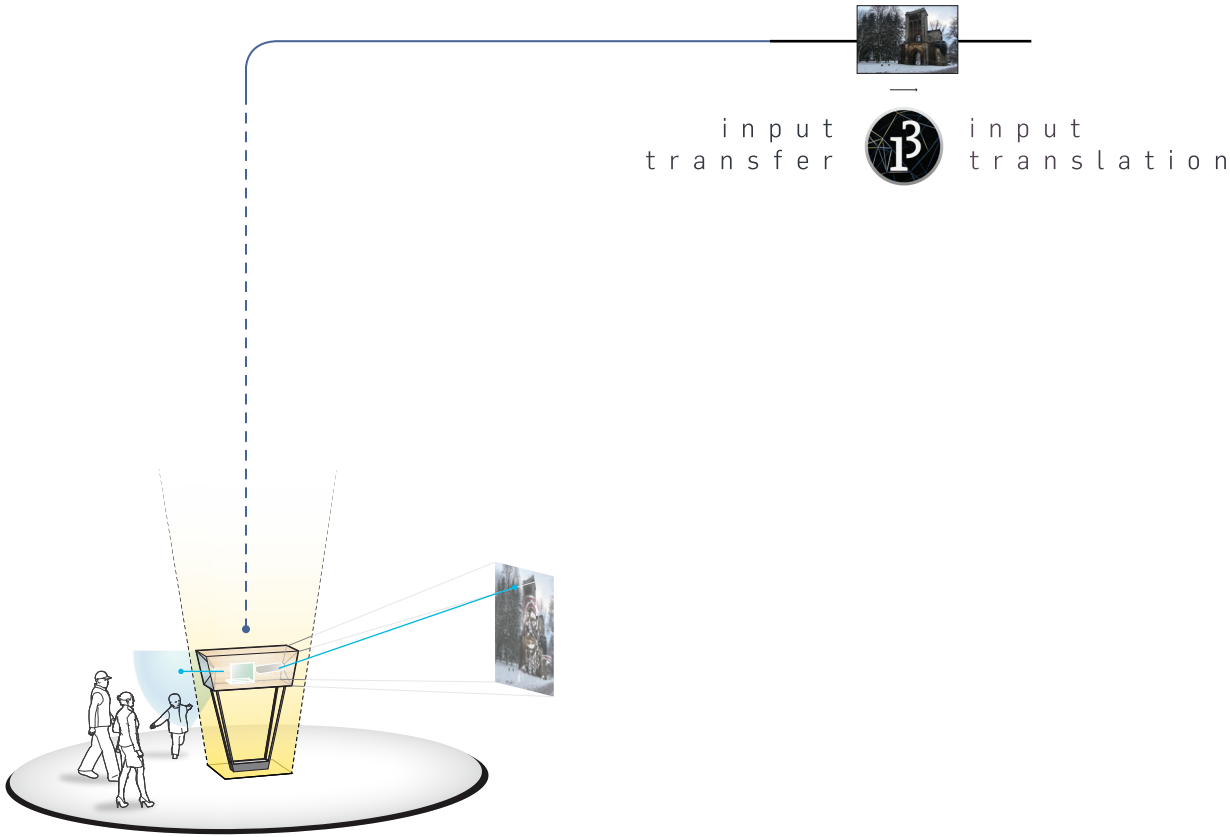

FaceOSC
pose
gesture
raw
normalize raw

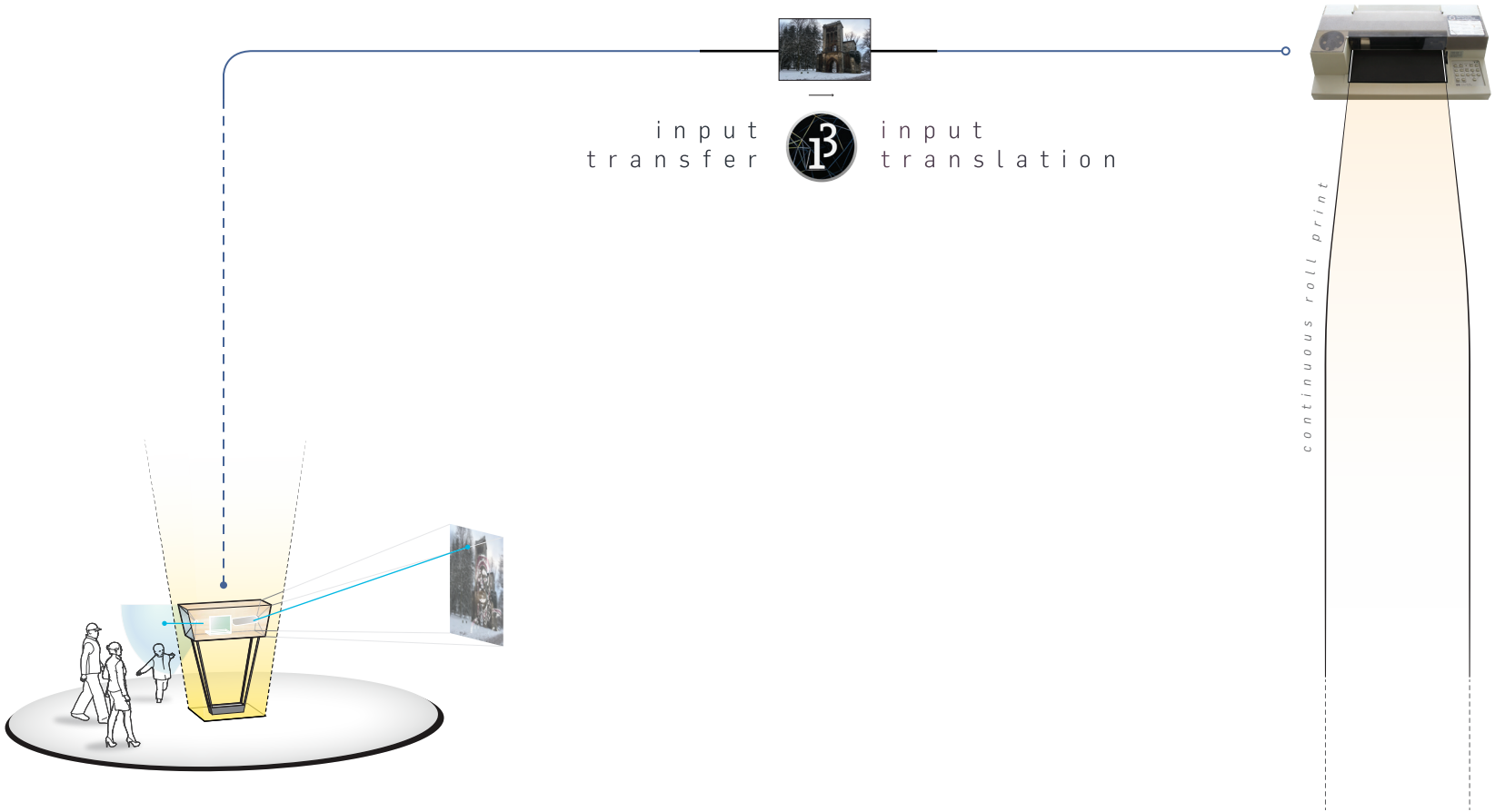


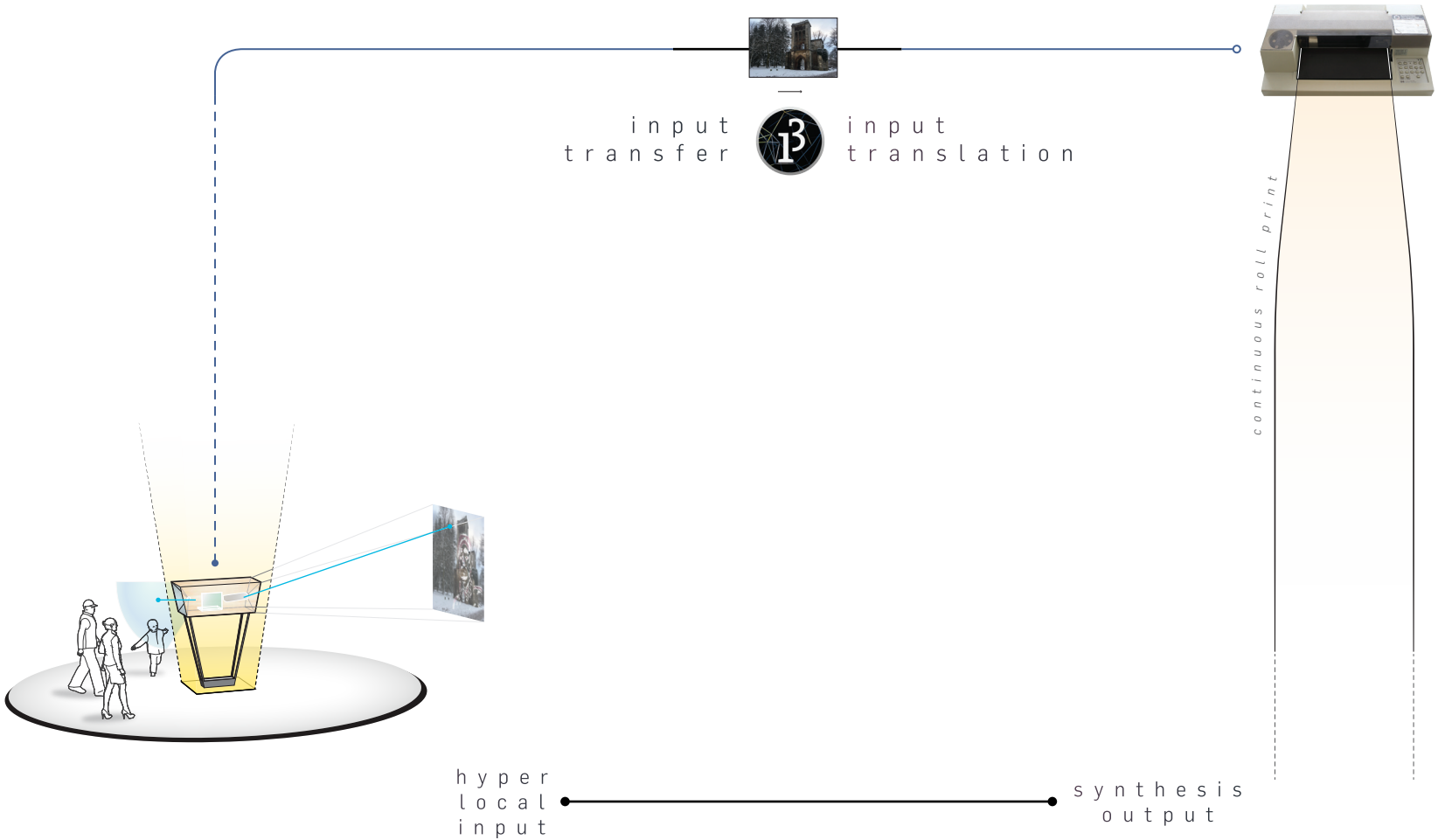


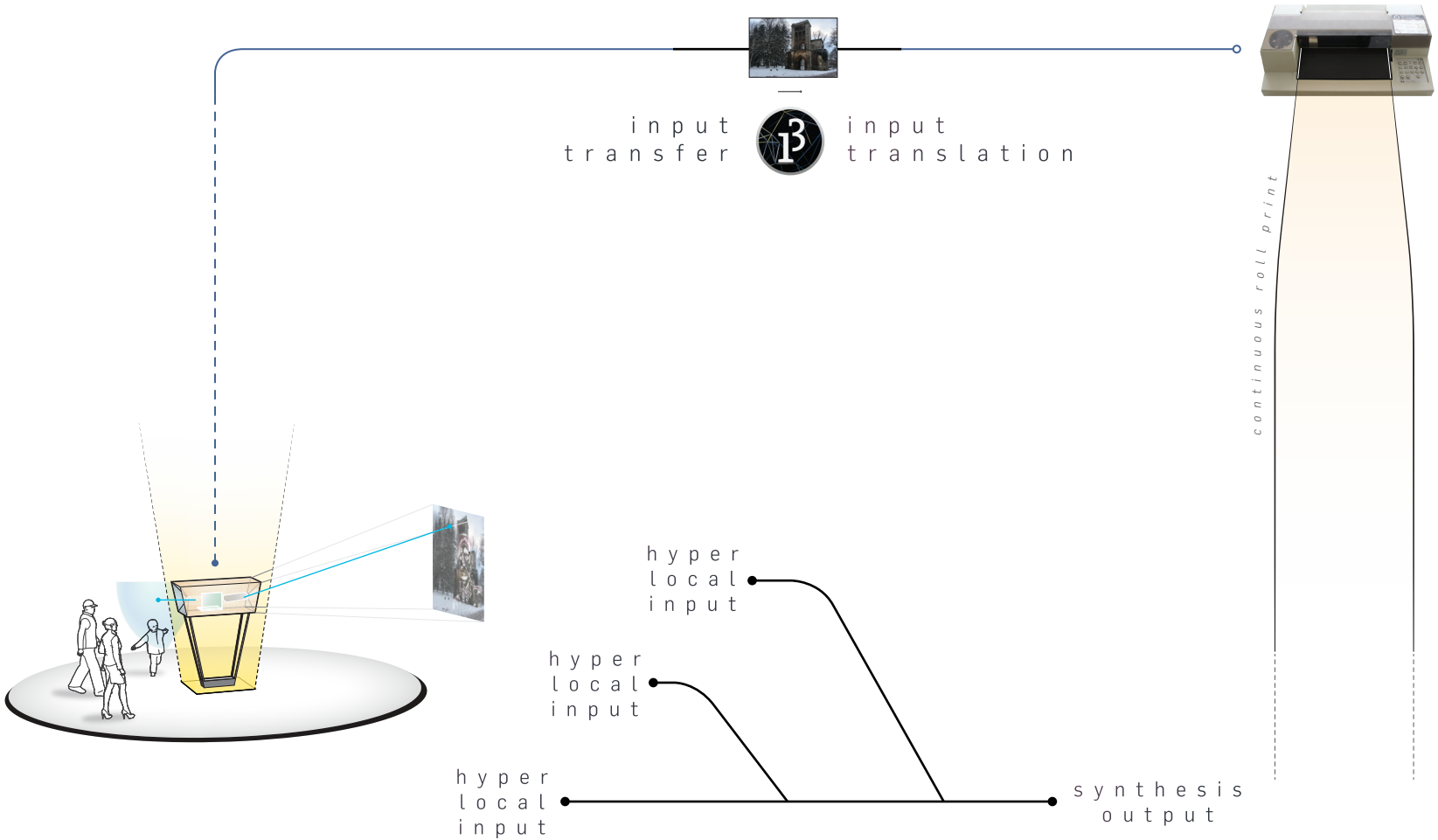

analysis,
translation,
generation
video capture
// between specified ranges of time
// data captured sent to plotter



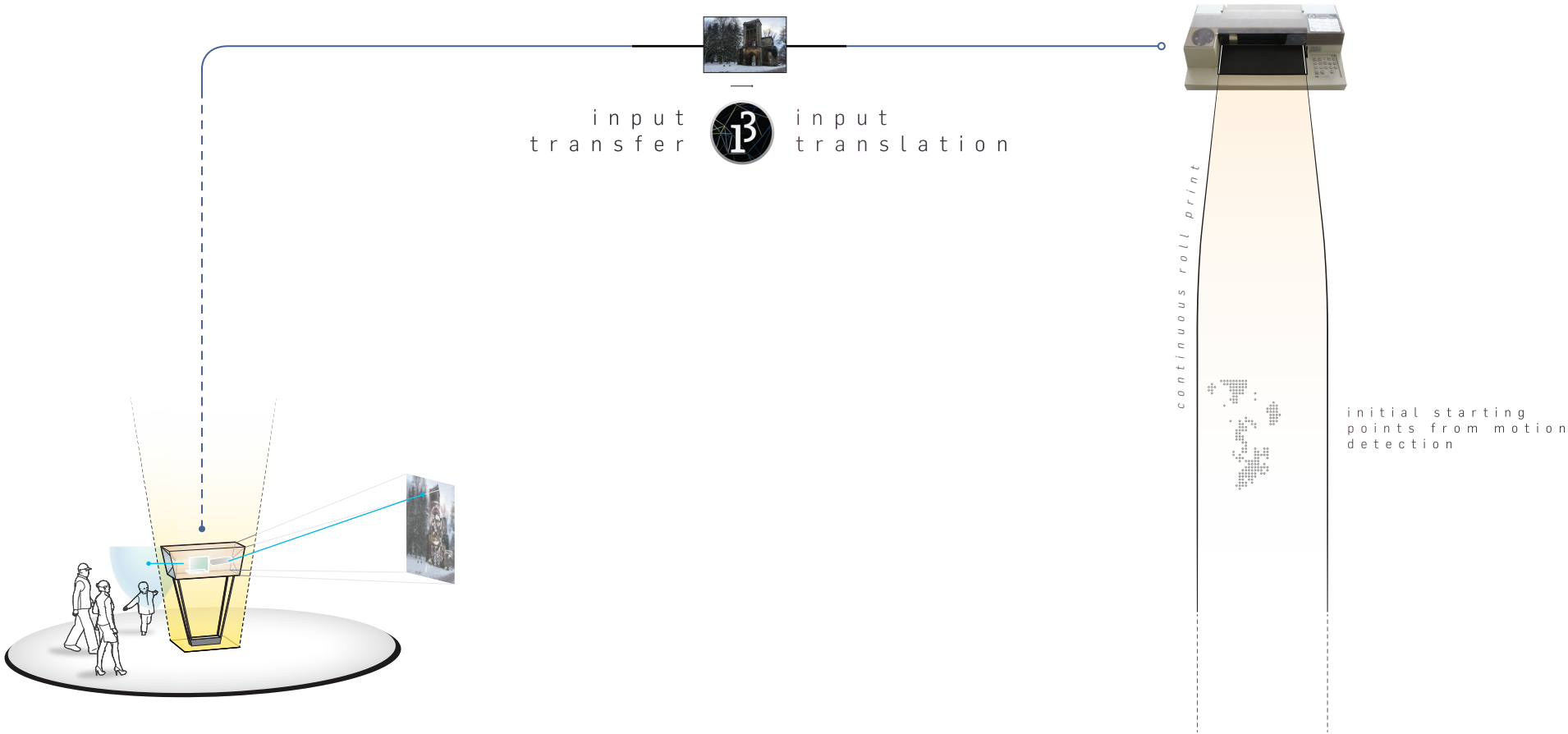

input
transfer
input
translation

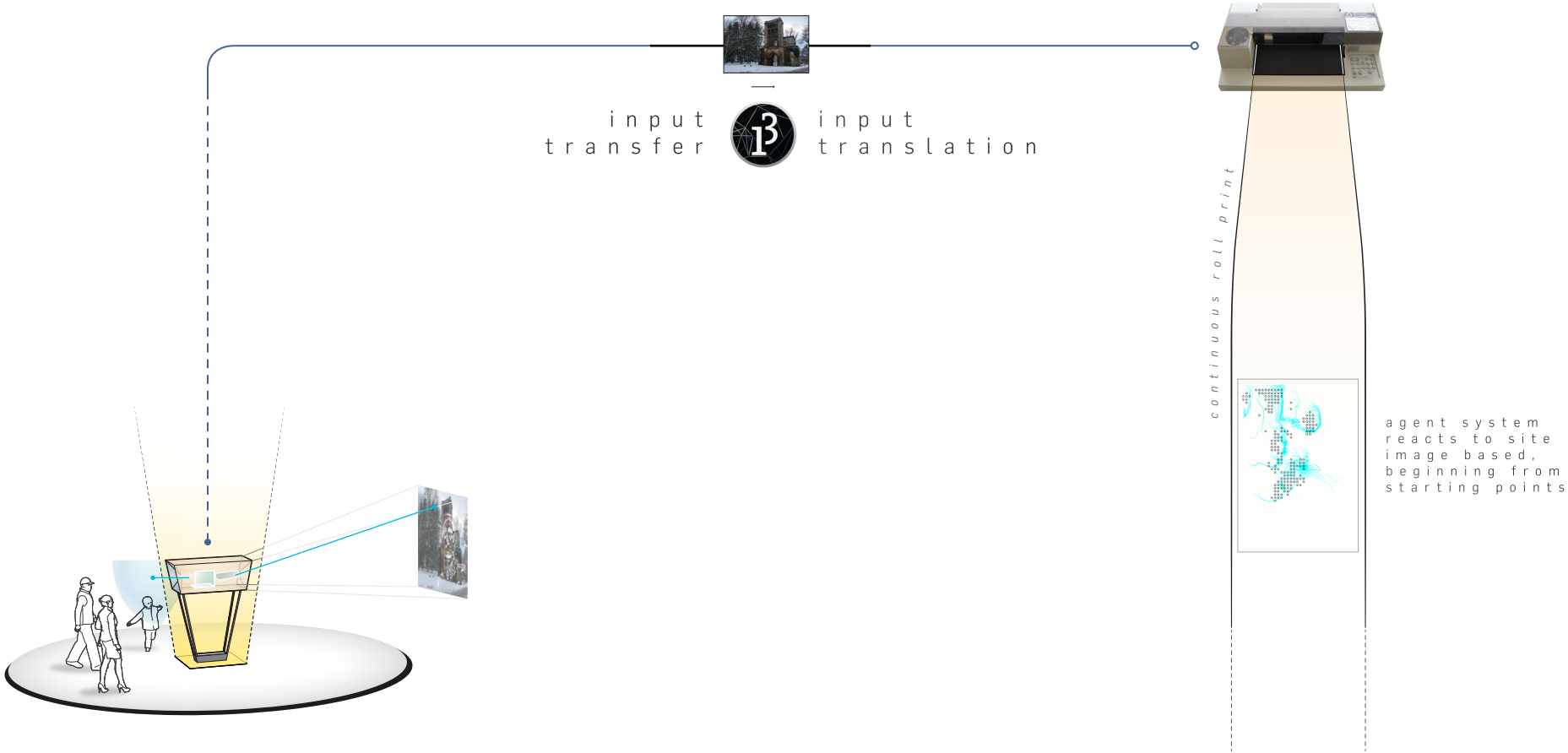

input
transfer
input
translation
continuous roll print

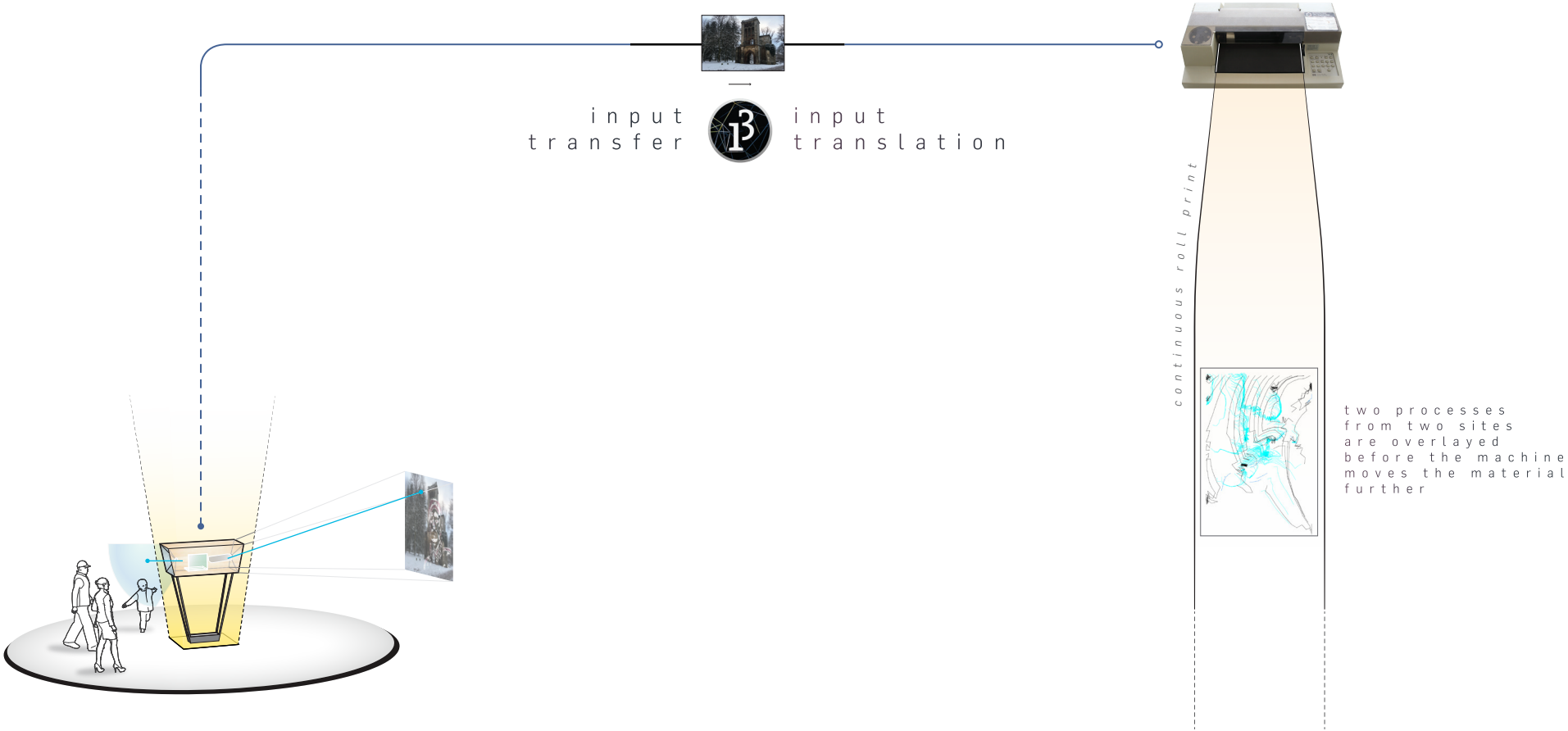

input
transfer
input
translation
continuous roll print
hyper
local
input
synthesis
output


input
transfer
input
translation
continuous roll print
hyper
local
input
hyper
local
input
hyper
local
input
synthesis
output


input
transfer
input
translation
continuous roll print


input
transfer
input
translation
continuous roll print
initial starting
points from motion
detection


input
transfer
input
translation
continuous roll print
agent system
reacts to site
image based,
beginning from
starting points

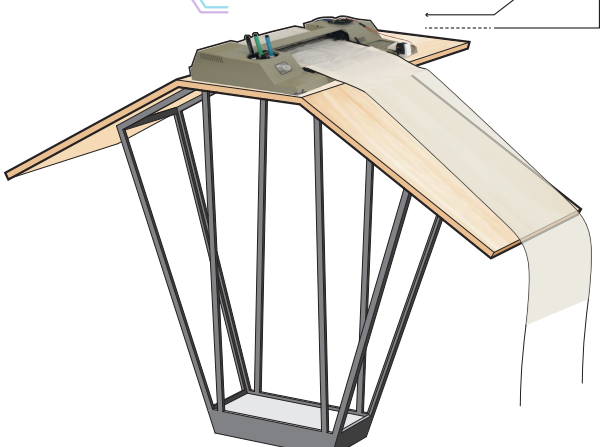

input
transfer
input
translation
continuous roll print
two processes
from two sites
are overlayed
before the machine
moves the material
further

based image:
DNT

based image:
tempelherrenhaus

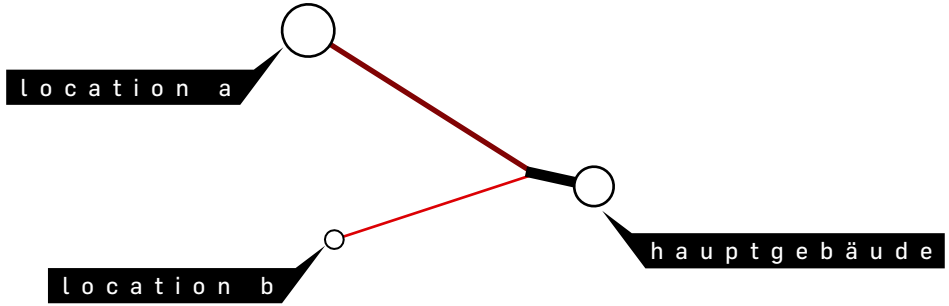
// each location provides a list of starting points via motion detection

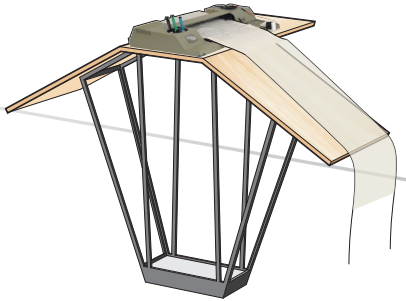
image processing, agent-based drawing operations



mechanical reset per overlay to allow continuous roll plotting

drawing operation overlays from locations, continuous visual output


location a
location b
hauptgebäude



A base image is used for analysis

// this image is either periodically taken of the the site through the day, or a static image representing the site in every drawing.

motion detection points are imported into the processed image

// these points are from the webcam motion detection operations which happen at each site

// the motion capture points are taken periodically from the sites and transferred to the drawing machine location as data files

Image / pixel values are derived from the image at the imported points

// each location has inate RGB values, as well as a brightness and saturation

// these values can be used to influence drawing operations

generative operations begin from the starting points, based on image data

// line and changed path generation based on underying image

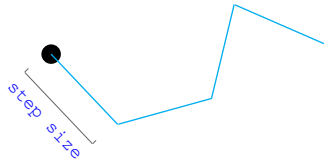
each set of connected lines has a number of characteristics

next step location, step size, amount of steps

step size of path is determined by brightness of each point:

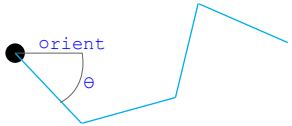
```
void uploadPoints() {

  File tempFile = new File("C:/Users/grays/Desktop/CODING/PlotterContinousRun_
        skeleton/motionX_"+saveCount+".txt");
  if (tempFile.exists()) {
    String[] tempHolderX = loadStrings("C:/Users/grays/Desktop/CODING/
        PlotterContinousRun_skeleton/motionX_"+saveCount+".txt");
    String[] tempHolderY = loadStrings("C:/Users/grays/Desktop/CODING/
        PlotterContinousRun_skeleton/motionY_"+saveCount+".txt");
    float scale;
    for (int i = 0; i < tempHolderX.length; i++) {
      scale = map(brightness(currentImage
        get(int(tempHolderX[i]),int(tempHolderY[i]))),0,255,5,90);
      swankAgents.add(new SwankAgent(new PVector(int(tempHolderX[i]),
         int(tempHolderY[i])), scale));
    }
  }
}
```



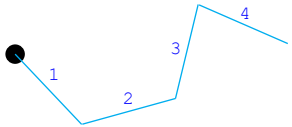
orientation of segment location determined through blue values:

```
void moveImageDir(float scale) {
   float angle = ((blue(currentImage.get((int)location.x,
         (int)location.y))/255.0))*scale;
   prevLocation = location;
   location = new PVector(location.x + sin(angle)*speed,
         location.y + cos(angle)*speed);
   checkBorder();
   stroke(0);
 }
```



number of segments determined by number of starting points

```
void processImage(int _pen) {
  uploadPoints();
  String[] postDate = loadStrings("C:/Users/grays/Desktop/CODING/
        PlotterContinousRun_skeleton/postDate_"+saveCount+".txt");
  int runCount = ceil(500/(swankAgents.size()+1));
  if (runCount > 15) runCount = 15;
  beginRecord(hpgl);
  hpgl.selectPen(_pen);
  println(swankAgents.size());
  updateSwank(runCount);
  textSize(15);
  text(postDate[0], width-400, height-20);
  text("N G Z C M | P 3 | 7 4 7 5 A", width-200, height-20);
  endRecord();
  saveCount++;
}
```



# of segments

changes in values, changes in generation:

step size: based on saturation of the img.location

orientation: based on the red of the img.location

number of segmentts: 600 / (points+1)

changes in values, changes in generation:

step size: based on red of the img.location

orientation: based on the green of the img.location

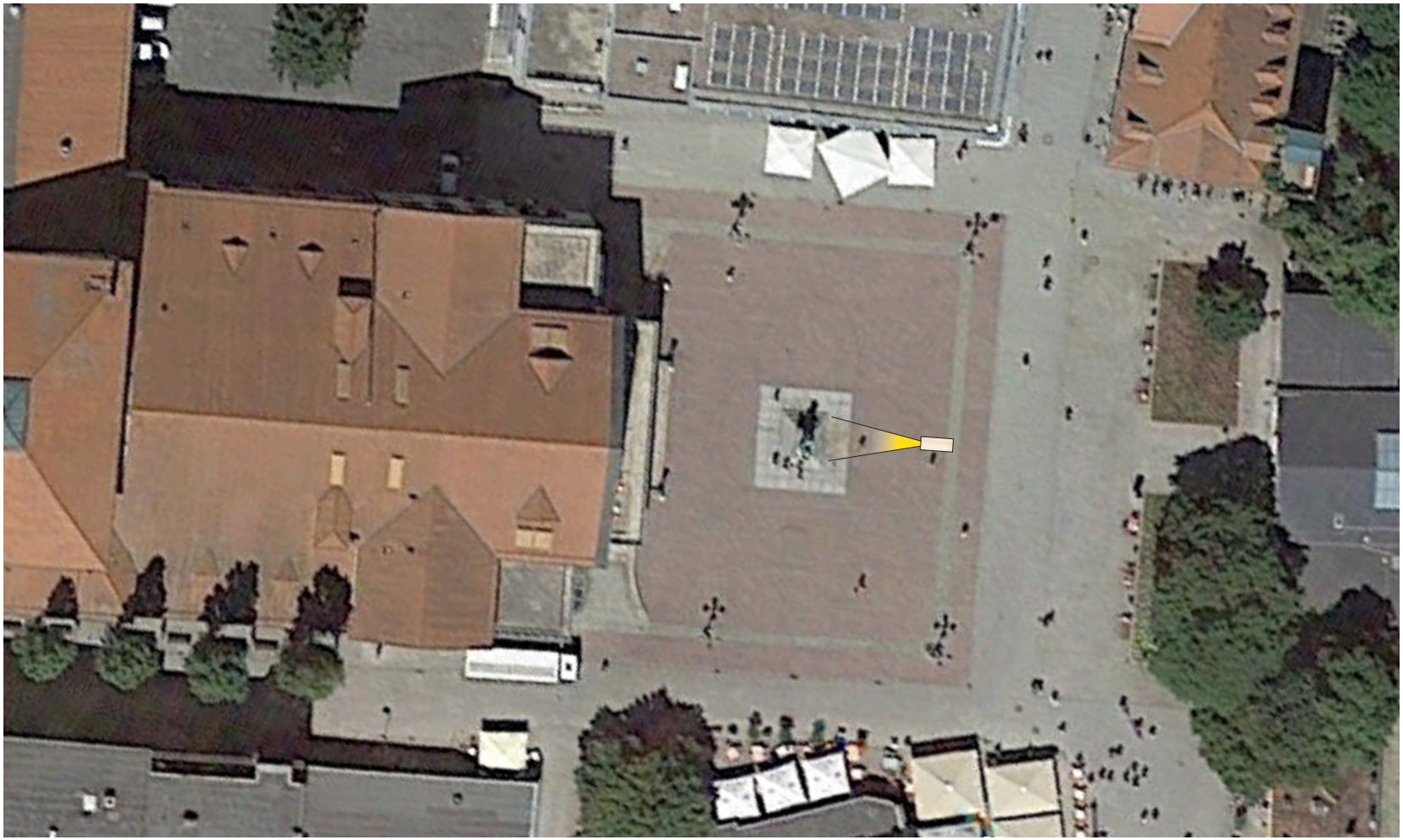
number of segmentts: 500 / (points+1)

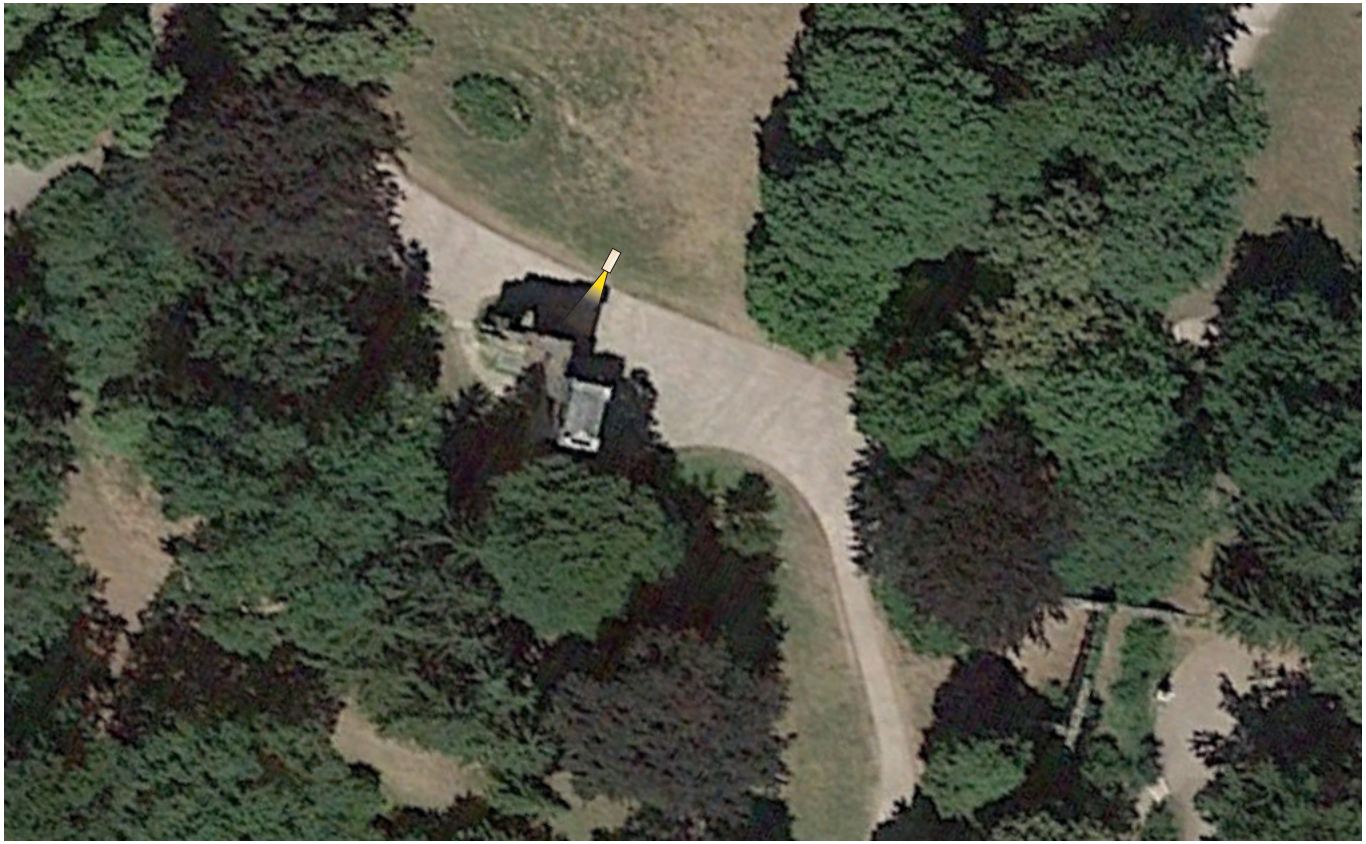
changes in values, changes in generation:

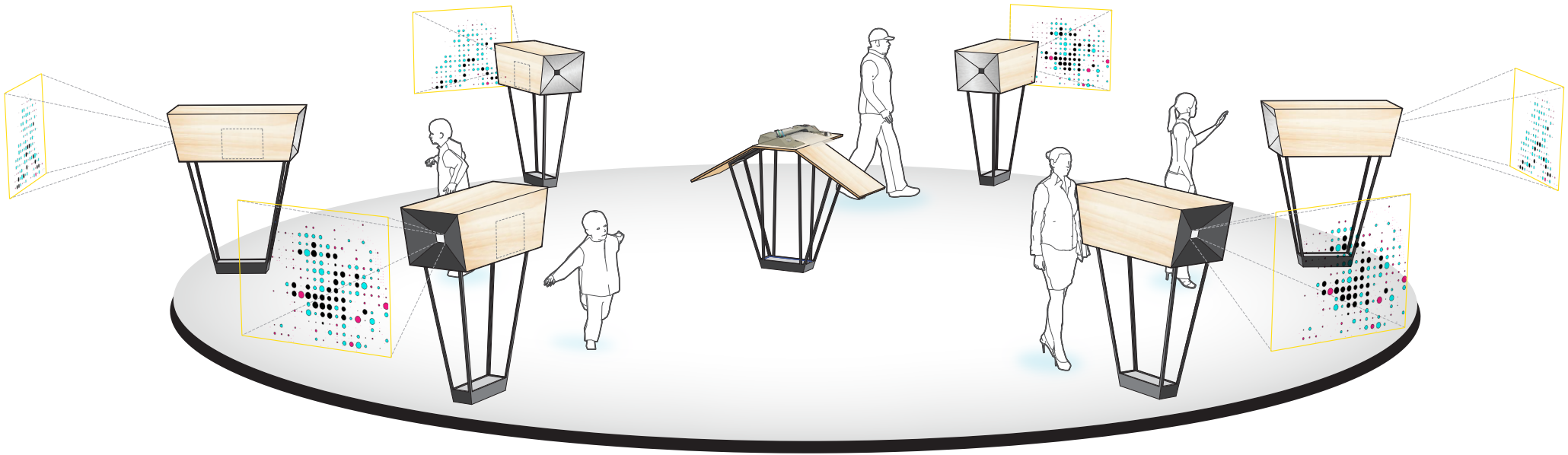
step size: based on brightness of the img.location

orientation: based on the blue of the img.location

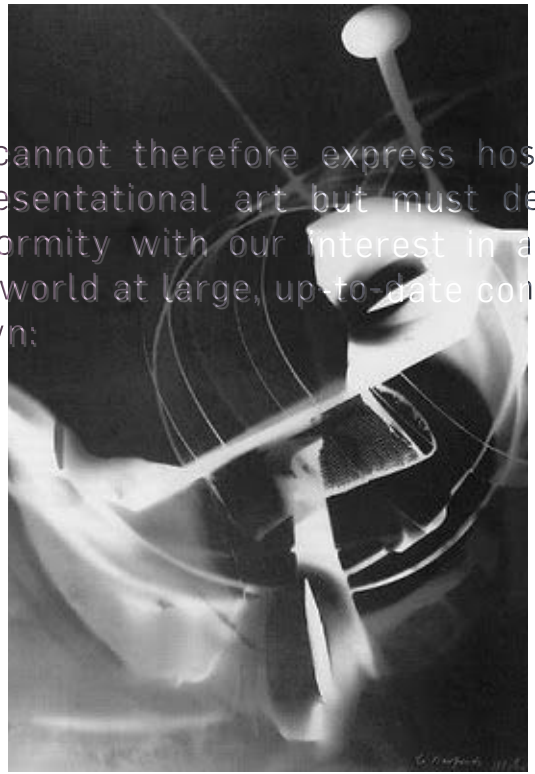
number of segmentts: step size / 10

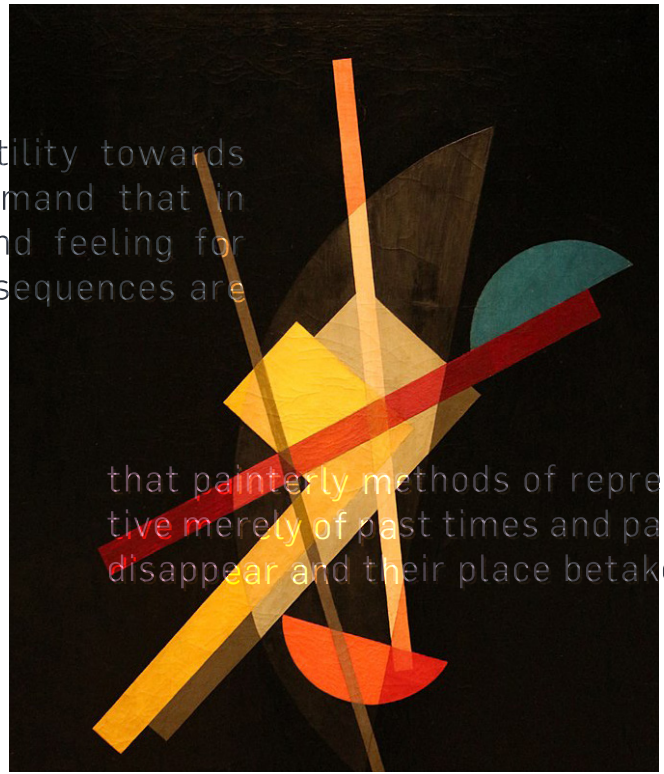






We cannot therefore express hostility towards representational art but must demand that in conformity with our interest in and feeling for the world at large, up-to-date consequences are drawn:





that painterly methods of representation sugges-
tive merely of past times and past ideologies shall
disappear and their place betaken by

mechanical means of representation and their as yet unpredictable possibilities of extension.