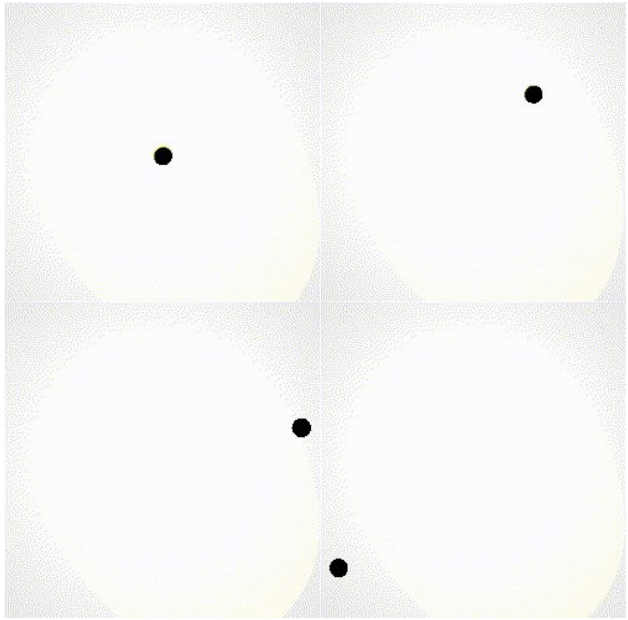
Progress Presentation

Zachary Kirby

Synthetic Data

- 2-D movement of a circle on a plane
  - Simplest model we can train
- Doesn’t necessarily need LSTM cells
  - Can be run with a RNN
- Scaleable, but starting simple
  - Also random!



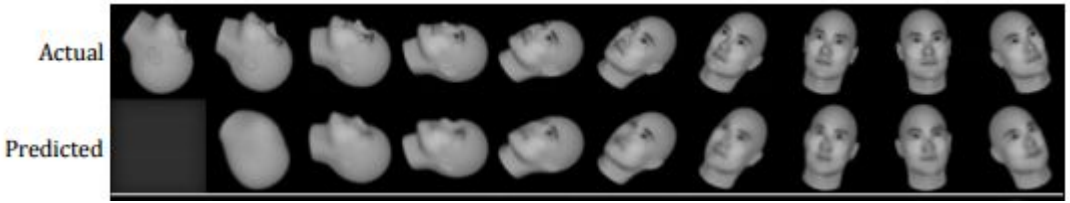
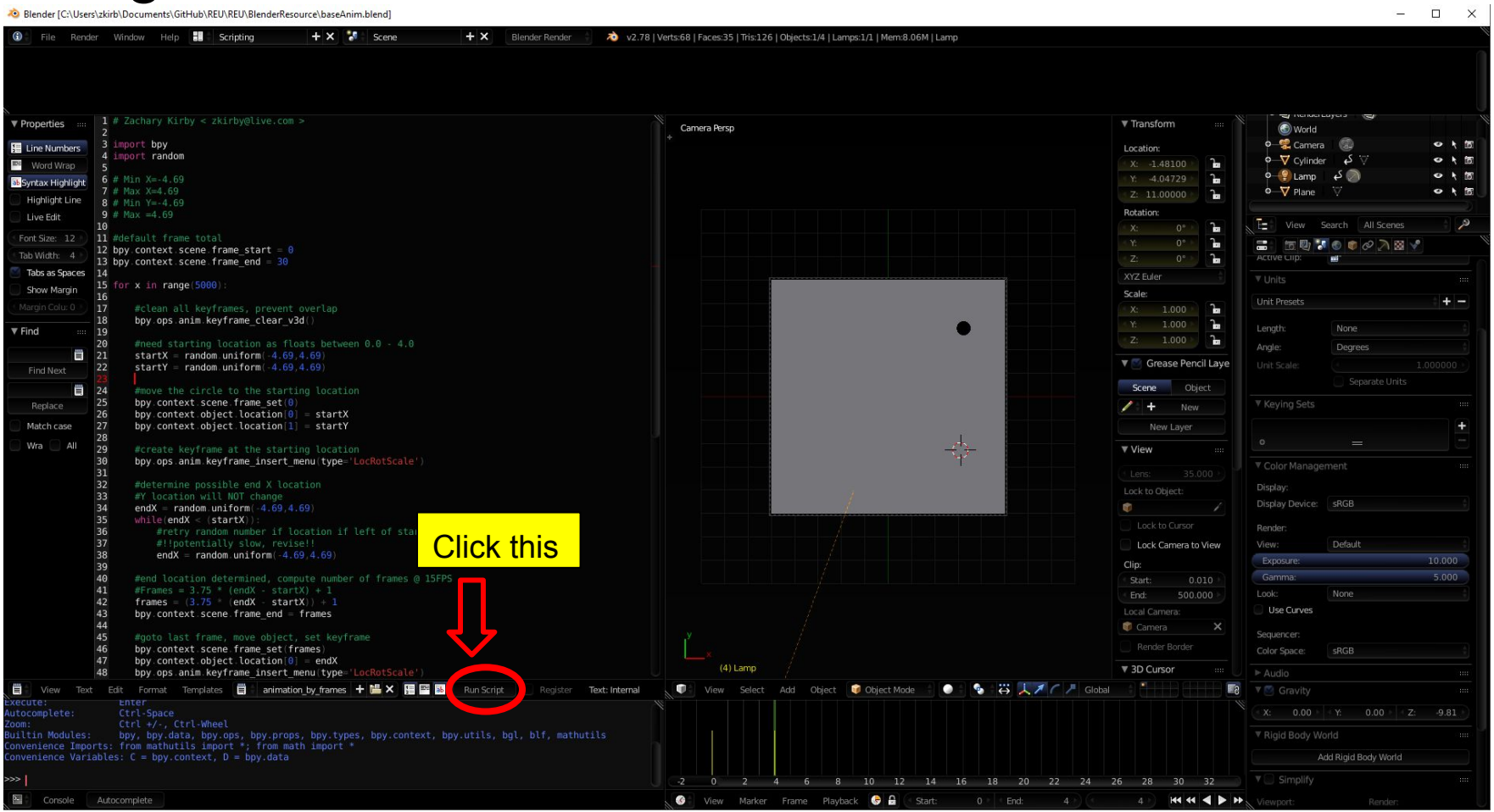


Image credit: https://coxlab.github.io/prednet/

Synthetic Data (cont)

- Animations - 256x256 px
- Output is 5000 folders of frames
  - Each folder = one animation
- Random starting X,Y in Quadrants II & III
  - End X is always 4.69 (touching the end of screen)
- Frames determined by f(x)
  - f(x) = (max_frames / 8.0) * (endX - startX) #UNDER REVISION
- Blender File - here

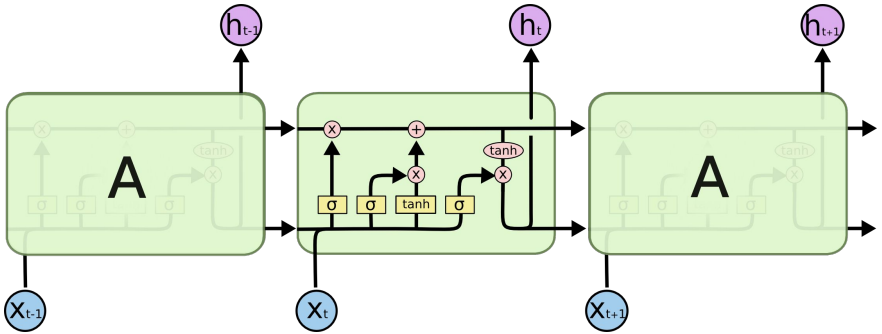
Using Blender File



Data Format

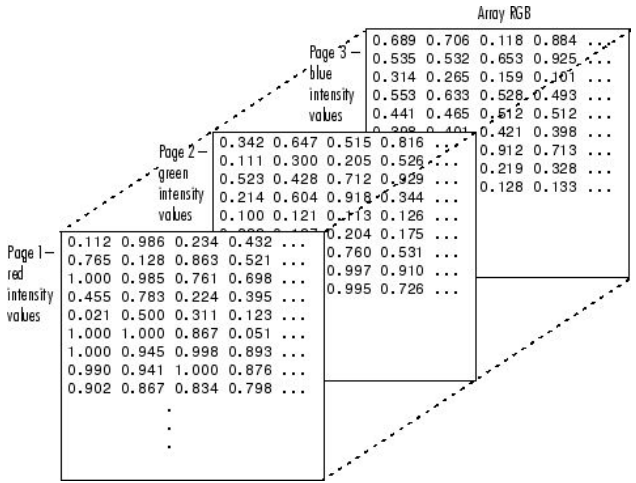
- Currently either ‘.avi’ movie renders or ‘.png’ frame renders
  - Images easier to work with than movies
- Cast frames as numpy arrays
  -
```
import Image
import numpy

pic = Image.open('frame1.png')
picArray = numpy.array(pic)
```
- Each frame will be a memory point (c_t)
- How to build the total animation?
  - Pickle serialization
  - Multidimensional arrays



numpy.ndarray()

- Create an array for each of the frames
  - pic = Image.open(‘frame.png’).convert(‘L’)
  - frame = np.array(pic)
- Then create an ndarray for all frames
  - anim = np.ndarray(shape=(total_frames,width,height),dtype=np.float32)
  - anim.append(frame) (?)
- Potentially use scikit-image
  - Single line implementation, but not entirely necessary
  - Creates an array like image shown



```
import numpy as np
from PIL import Image


file = '2.94280_4.188120000.png'
pic = Image.open(file)
arr = np.array(pic)
print(arr)
img = Image.fromarray(arr, 'L')
img.save('my.png')
img.show()
```

```
#pic = Image.open(file).convert(‘L’)
from matplotlib import pyplot as plt


plt.imshow(arr, interpolation='nearest')
plt.show()
```