Efficient Photorealistic Avatars using ML/AI

Milestone 2

Group 1
Minh Khoa Đoàn
Cyrine Boudaya
Belinda Myteberi
Rebecca Charlotte Barth

Agenda

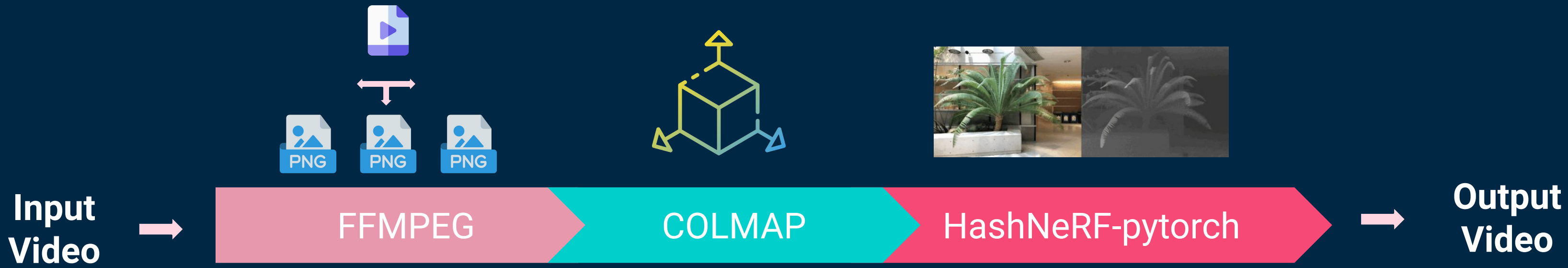
- Problem Statement
- Preliminary Solution
- Roadblocks
- First Demo
- Updated Schedule

Problem Statement

Goal: Rendering a photorealistic avatar with

- Monocular camera input ✅
- Using an optimized neural radiance fields with state of the art input encoding ✅
- Displaying the fourth dimension in terms of facial expressions and emotions ⛔

Preliminary Solution



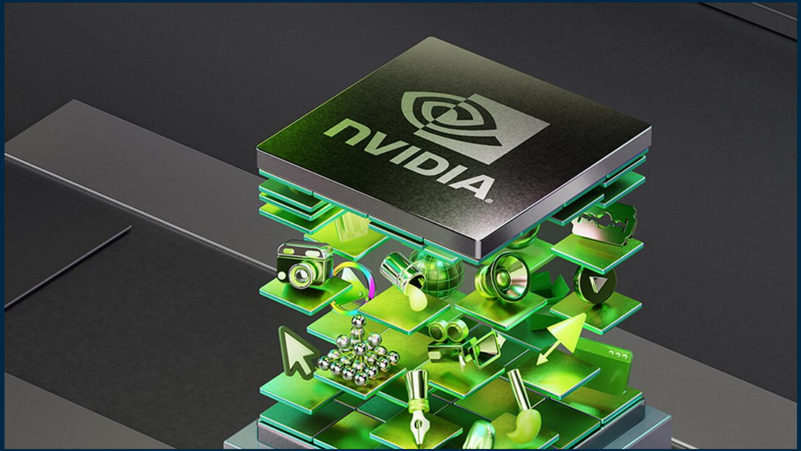
FFMPEG enables automatic **frame extraction** from video input.

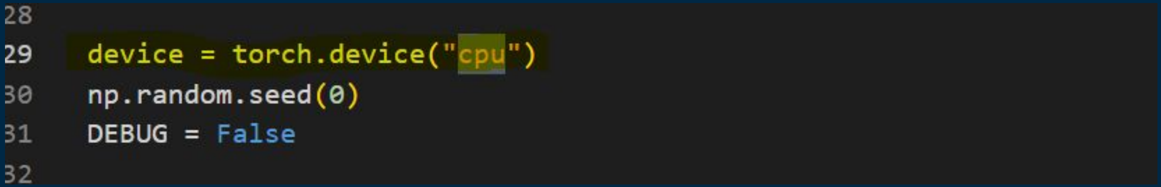
Colmap runs structure from motion to get **camera poses** and **near/far depth bounds** for the scene.

Use **NeRF-Pytorch** in combination with **hash-encoding** to synthesize **novel views** of complex scenes more efficient.

Roadblocks

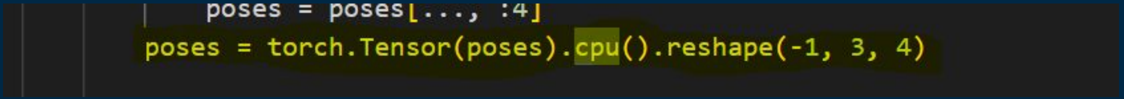
Need of Nvidia GPU



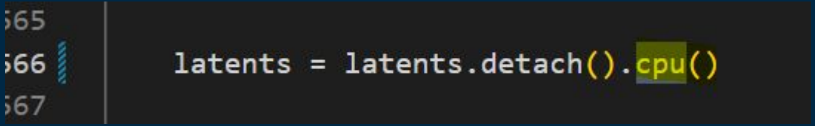


```python
device = torch.device("cpu")
np.random.seed(0)
DEBUG = False
```

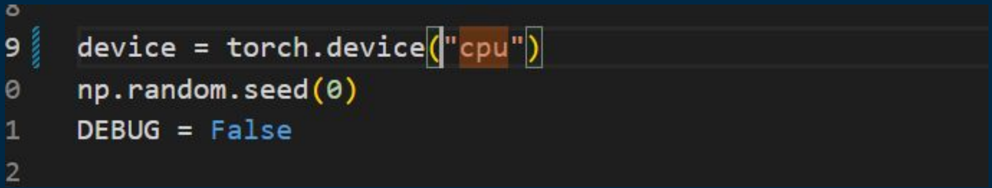
```python
device = torch.device("cpu")
```



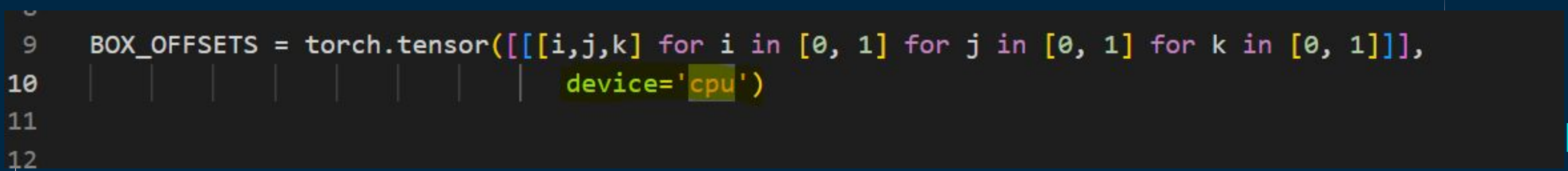
```python
        poses = poses[..., :4]
    poses = torch.Tensor(poses).cpu().reshape(-1, 3, 4)
```



```python
    latents = latents.detach().cpu()
```



```python
device = torch.device("cpu")
np.random.seed(0)
DEBUG = False
```

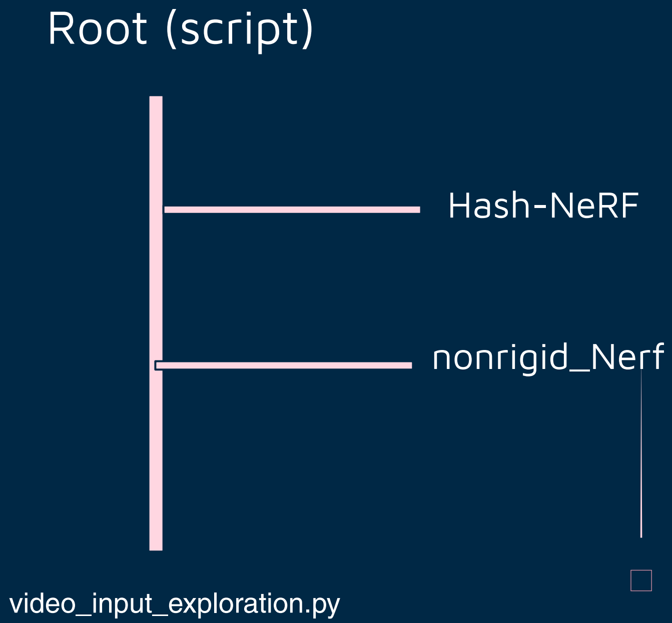


```python
BOX_OFFSETS = torch.tensor([[[i,j,k] for i in [0, 1] for j in [0, 1] for k in [0, 1]]],
                                device='cpu')
```

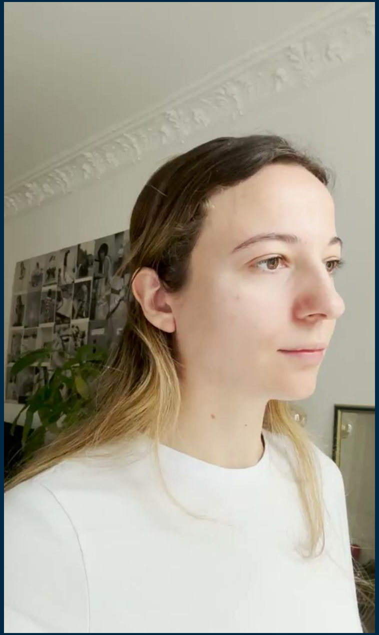
```python
device = torch.device("cpu")
```

Roadblocks

- Use of submodules messed up import structure
- Different operating systems

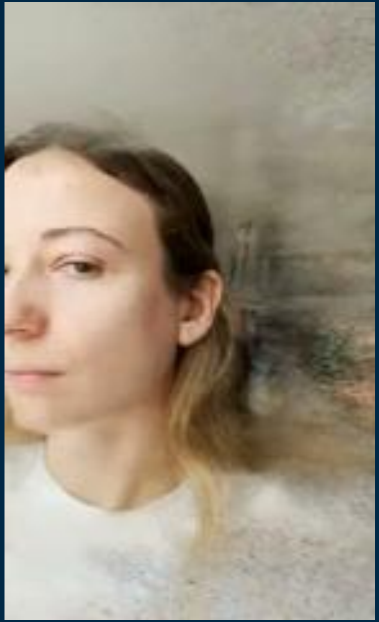


First Demo



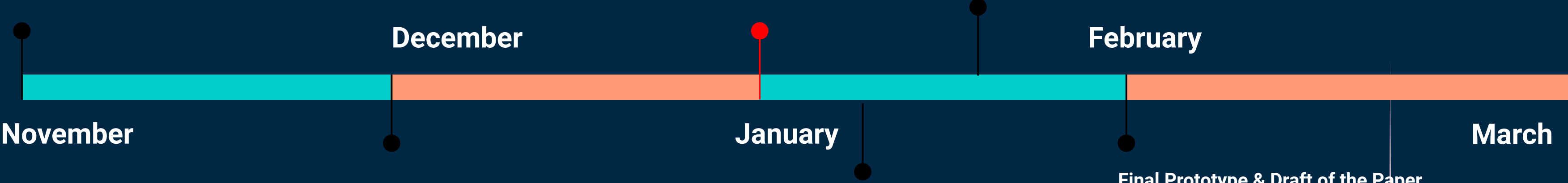
Input

Git: Source Code



Output

Road Map



First prototype
Error free render of a static photorealistic Avatar.

Optimization of performance

Flame Mesh for face expression

Final Prototype & Draft of the Paper

Additional dynamic face expression and emotions.

Create a draft for the Paper

Final Product & Paper

Fully efficiently rendered 4D photorealistic avatar with monocular video input integrated in web platform.

Next Steps

- Add camera calibration to preprocessing to undistort input sequences recorded
- Compare performance with D-NeRF
- Rendering of 3D avatar (Blender)
- Add facial expression with flame mesh

Thank
You!