Editable Scene Simulation for Autonomous Driving via Collaborative LLM-Agents

Yuxi Wei[1*] Zi Wang[3*] Yifan Lu[1*] Chenxin Xu[1*] Changxing Liu[1] Hao Zhao[4] Siheng Chen[1,2] Yanfeng Wang[1,2]

[1] Shanghai Jiao Tong University [2] Shanghai AI Laboratory [3] Carnegie Mellon University [4] Tsinghua University

* Equal Contribution

CVPR SEATTLE, WA JUNE 17-21, 2024

video here

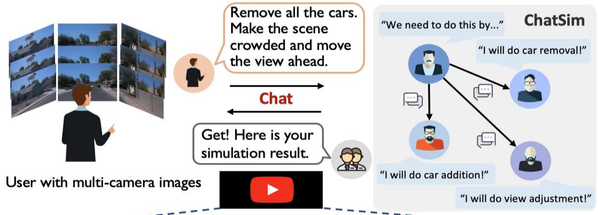
yifanlu0227.github.io/ChatSim

Overview

ChatSim is the first system achieves **language-controlled photorealistic** driving scene simulation. **Easy to use**

Control the simulation with language

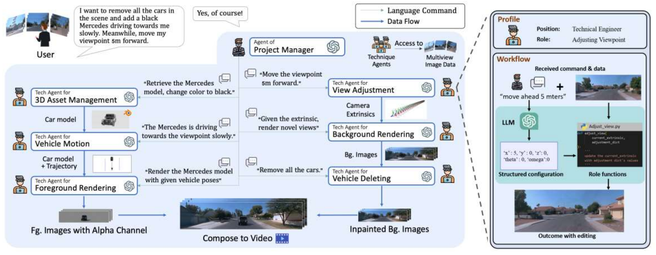
1. **Multi-view consistent photorealistic rendering**
   McNeRF for background scene reconstruction
2. **Flexible simulation with external 3D assets**
   McLight for lighting estimation for virtual object insertion



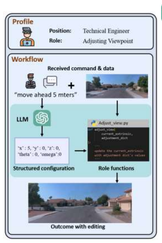
Simulation result with editing

Framework

ChatSim adopts a large language model (LLM)-based multi-agent collaboration framework. The key idea is to exploit multiple LLM agents, each with a specialized role, .



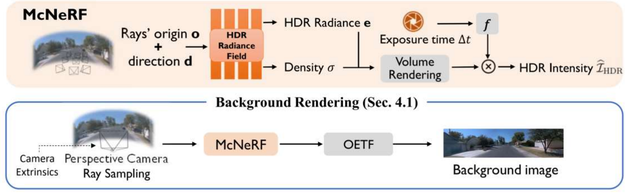
Single-Agent LLM Prompting




```
[Task Prompt]: "I will give you a transformation operation for my viewpoint, which
may include translation in 'x', 'y', 'z' or a rotation 'theta' around z-axis. "

[Detail Prompt]: "For translation, positive 'x' represents forward, positive 'y'
represents left, and 'z' represents up. It follows a left-hand coordinate system."
"For rotation, positive 'theta' is counterclockwise. So from own perspective, my
viewpoint turns to the left. 'theta' is in degree."

[Format Prompt]: "Given my operation, return a dictionary in JSON format, with
keys 'x', 'y', 'z', 'theta'."

[Example Prompt]: "I will give you some examples:
<user>: rotate the viewpoint 30 degrees to the left;
<assistant>: {\n  'x': 0,\n  'y': 0,\n  'z': 0,\n  'theta': 30,\n } \
<user>: move the viewpoint to the right by 1;
<assistant>: {\n  'x': 0,\n  'y': -1,\n  'z': 0,\n  'theta': 0,\n} "
```

Background: An HDR radiance field

Exposure-aware design for consistent brightness rendering



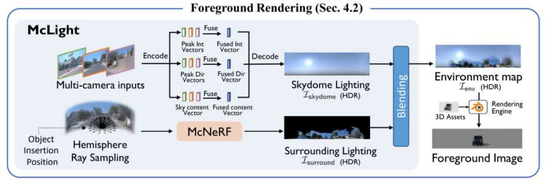
Predict radiance in HDR space (linear)

$$\hat{\mathcal{I}}_{\mathrm{HDR}}(\mathbf{r}) = f(\Delta t) \cdot \sum_{k=1}^{K} T_k \alpha_k \mathbf{e}_k$$

Calculate loss in LDR space (gamma-corrected)

$$\mathcal{L} = \frac{1}{|R|} \sum_{\mathbf{r} \in R} \left(\mathrm{OETF}\left(\hat{\mathcal{I}}_{\mathrm{HDR}}(\mathbf{r})\right) - \mathcal{I}(\mathbf{r}) \right)^2$$

Foreground: Lighting Estimation + Blender Rendering



1. We predict peak intensity vectors, peak direction vectors and sky content vectors from multi-view image.
2. We leverage extrinsic-aware self-attention to fuse latent vectors to one, which will decode a skydome HDRI.
3. We sample the rays directed at the hemispherical surface in McNeRF to obtain a surrounding HDRI.
4. We blend two HDRIs with alpha-blending and use Blender for virtual object rendering.

Results

Simulation with language command

Case 1 (highly abstract command)

Language command: *"Create a traffic jam."*

Case 2 (complex command)

Language command: *"Remove all cars in the scene and add a Porsche driving the wrong way toward me fast. Additionally, add a police car also driving the wrong way and chasing behind the Porsche. The view should be moved 5 meters ahead and 0.5 meters above."*

Virtual Object Insertion Comparision (foreground rendering)

(a) Hold-Geoffroy et al. (b) Wang et al. (c) McLight (Ours) (d) Spatial-Varying Effect