Qeios

Peer Review

Review of: "LiLMaps: Learnable Implicit Language Maps"

Muzhi Han[1]

1. Independent researcher

The paper presents a framework to incrementally reconstruct a language-aware 3D scene representation from RGB-D streams. The scene representation is an octo-tree that stores implicit coarse-to-fine features, which can be decoded into language features for downstream tasks. It is updated by optimizing both the language decoder and octo-tree features as new observations come in, by minimizing the cosine distance between decoded language features and vision-language features encoded from observed images. In experiments, the authors show that the method outperforms other language-aware mapping methods and that the strategy to adaptively optimize the language decoder is better than using a pretrained decoder.

The paper is, in general, easy to follow, and the proposed method sounds solid to me. The octo-tree-based representation that stores coarse-to-fine features does save memory and is expected to operate in relatively large-scale environments. However, I do have some questions about the detailed method, and also suggestions to improve the clarity of the presentation.

1. Introduction and overall approach:
    1. It is worth making clear that the 3D octo-tree (and point cloud) is explicit, and only the language-aware features stored are implicit. This is fundamentally different from implicit scene representations where the 3D geometry is decoded.
    2. It's good to add a teaser image to clearly explain how the proposed method and representation are different from the existing ones, and what the benefits are (e.g., saving memory). I don't quite understand how the method is special until I read the method section.
    3. In the third and fourth paragraphs in the Introduction, it's good to provide references accordingly to help readers get familiar with the topic and existing methods.

doi.org/10.32388/49EFKR

2. Method
    1. One thing that draws confusion is the usage of “encoding,” “features,” “F-vectors,” and “language features.” It's good to clarify these wordings and give better names. For example, I would say the octo-tree stores “coarse implicit features” (F-vectors) and “fine implicit features,” which can be decoded to “language features.”
    2. Before equation 1, it may also be necessary to have an equation that describes how the map is implicit, i.e., how the stored features can be decoded into language features.
    3. Algorithm 1 is also a bit confusing. Please make sure the variables and functions used are clear and easy to understand.
    4. What is stored in the scene representation? Other than implicit features, are 2D features from history observations also saved for providing supervision?
3. Experiments
    1. It's good to make clear that the color visualization of scenes is based on the most similar semantic class, and what categories are considered.
    2. It's also good to briefly describe all the baseline methods compared in Table 2. What is the semantic image used in LiLMaps_SEM? And what is the setup for LiLMaps_LSeg?
    3. Table 3 is missing subscripts in decoders.
    4. How is the memory usage of the proposed representation compared to baselines?

Declarations

Potential competing interests: No potential competing interests to declare.