Scene Editing Using Synthesis of Three-Dimensional Virtual Worlds From Monocular Images of Urban Road Traffic Scenes

Ankita Christine Victor

Jaya Sreevalsan Nair
https://www.iiitb.ac.in/GVCL/

International Institute of Information Technology Bangalore,
26/C Electronics City, Hosur Road, Bangalore,
Karnataka 560100, India.

Three dimensional (3D) modeling is an integral part of many graphics applications. 3D reconstructions of scenes from image and sensor data enable both realization and editing of a scene. The user has more control in interacting with a 3D reconstructed world than the image which can be used for various applications, such as modeling of urban spaces, simulations of activities such as traffic, and testing for autonomous driving. The interaction is critical for scene editing. However, the challenge lies in automating 3D reconstruction as the conventional methods depend heavily on crowd-sourcing to skilled 3D artists [2]. To this regard, a technology that is capable of generating the basic structure of a 3D scene using an image as a reference with minimal human input and has the provision for editing objects in the scene is an interesting area of focus. We then pose the following research question: Is it possible to design a system for automatic modeling of the basic scene structure that can later be edited to create a variety of environments and improve modeling efficiency?

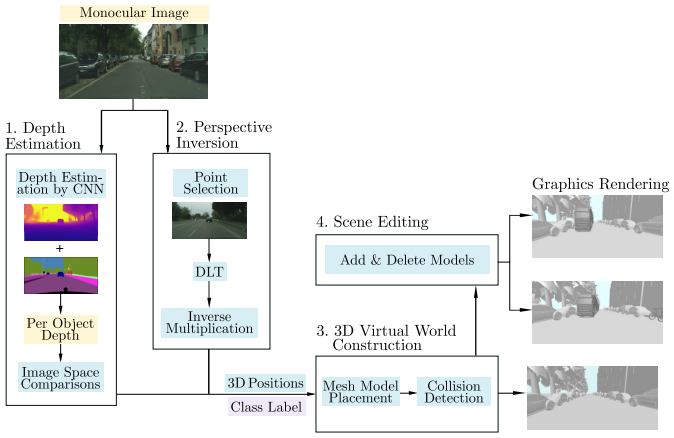

Figure 1: Overview of our workflow: Given a monocular image, we (1) estimate dense depth using a CNN and obtain per object depth using semantic segments, and (2) correct perspective distortion using DLT on guesstimated points. Using the 3D positions from (1) and (2) for our classes of interest, we (3) place 3D mesh models followed by an AABB collision detection, and lastly, (4) edit the scene. We examine the outcomes of synthesis and editing by rendering the virtual world(s) using a rendering engine, e.g., a basic one using OpenGL, shown here.

To address this question on image-to-3D world synthesis, we propose a workflow that uses a convolutional neural network (CNN) to estimate depth and computes a matrix to correct the effects of perspective projection using direct linear transform (DLT). Our workflow automates the initial steps of 3D scene modeling up to the stage where the primary objects in the scene have been placed, and the scene is rendered with ambient light. Following this, the scene can be edited manually to add or remove detail as per requirements. We have identified four main steps in the workflow (Figure 1), namely, depth estimation, perspective correction, model placement, and editing. Our contribution is in the automatic construction of 3D virtual worlds, per-frame from video footage of traffic scenes shot by a camera placed in a moving vehicle.

In step 1, we use the CNN architecture proposed by Godard *et al.* [3], which poses the problem of depth estimation from a monocular image as an image reconstruction problem. Our depth estimation module expects semantically segmented images. For an input image without prior semantic segmentation, a deep semantic segmentation network, e.g. DeepLab [1], can be used in a preprocessing step. We estimate depth from the image using the network and use ground truth semantic segments to compute per-object depth for our classes of interest.

In step 2, we approximate the projection matrix that created the image

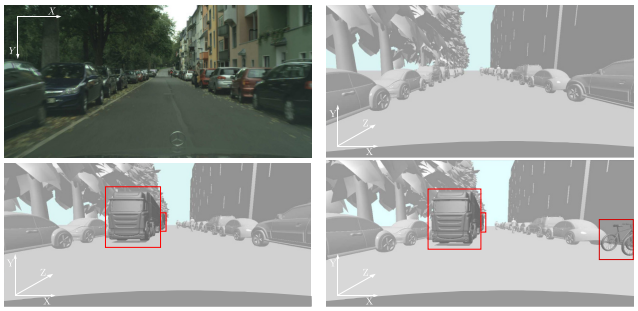

Figure 2: From top left clockwise: Input monocular image, 3D scene generated as is, scene with add and delete edits, scene with add edits only. The edits are highlighted using red boxes.

using DLT by guesstimating 4-5 world-image point pairs and apply an inverse of this to obtain the positions of objects as they would be in the real world. But, the matrix returned is highly sensitive to the choice of points and does not always yield the desired inverse projection. To obtain the position of each object along the X-axis, we multiply the extents of the bounding box around each segment with the inverse matrix obtained and averaged. The center of the object is translated along X by this value.

In step 3, for each object in the scene, we identify a mesh model (untextured here for uniformity in visualization) belonging to the same semantic class from a manually curated database. We then place the model at the 3D location computed from steps 1 and 2. Here, we have fixed the orientation of every model in 3D space as well as the relative scale of every object by comparing unit mesh model sizes. We run an axis-aligned bounding box (AABB) collision detection to eliminate any mesh intersection in the 3D world construction. The output at the end of the first three steps is a 3D virtual world composed of objects with their semantic class, X-, Y- and Z-coordinates, orientation, and scale, which together make the 3D scene. This 3D scene can either be rendered as-is, or edited to add or delete mesh models.

In step 4, the scene parameters from step 3 can be passed as input to modeling tools and rendering engines, such as Unity or Unreal. The permissible edits also include changing the type of object or position of an existing object. This allows an artist to create a variety of scenes using the same skeleton, which is a feature that can be particularly useful in autonomous vehicle simulation.

The workflow we have proposed can be operated to generate basic 3D worlds of urban traffic scenes from monocular images that can be edited and can work seamlessly with sophisticated modeling tools. Our proposed pipeline is currently intended for use on urban, outdoor scenes with straight roads and the objects of interest being cars, people, trees, buildings, and sidewalk (Figure 2). The main inspiration behind our work is to reduce the time and manual effort that goes into scene modeling. Given the vast collection of reference images on the Web, our proposed workflow constructs approximate 3D virtual worlds from these images efficiently and speeds up the process of scene modeling by building a basic 3D scene that can be further detailed and animated using scripts.

***Acknowledgements**: We thank T. K. Srikanth and Dinesh Babu Jayagopi for their invaluable feedback, and the Machine Intelligence and Robotics (MINRO) grant under the Government of Karnataka for funding our work.*

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[2] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016.

[3] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 270–279, 2017.