

It's about Road-side Green and Feature Mapping Using Google Street View (GSV) & ResNet-50 Convolutional Neural Network (CNN)

by **Kenny YIU** (for the mini-project of course programme URBA6402)

Reasons for Adopting Computer Vision for On-street Mapping

Many research papers discussed the way of acquiring the desired features from GSV for on-street urban planning and design study using Computer Vision technology. With use of the hidden depth map and other attributes of each 360-pano view downloadable from GSV, the green vegetation and other tiny features such as traffic sign boards and lightboxes, etc., which are not available in the current basemap (i.e. iB1000 dataset) can also be captured. Other advantages are:

1. **Free and Convenient:** GSV services are free for use in Google Map. In general, Google tries to take new street-view photos along major trunk roads of major cities once every year. 360-pano images can be downloaded using online tools or programming scripts. Data captured in Feb 2022 in Tiu Keng Leng (調景嶺) have been released.
2. **Abundant GSV Data:** Every 360-pano view covers a series of data and attributes, including 13 (V) x 26 (H) tiles, i.e. 338 pano-tile images, alignment parameters for cameras and sensors, and physical characteristics of most road-side features. Both geographic locations and dimensions can be estimated through a series of computation processes.
3. **Mature Artificial Intelligence (AI) Technologies:** Deep learning technologies for computer vision are mature. Open pretrained models (e.g. Common Objects In Context (COCO) dataset riding on ResNet-50 CNN) can be transferred and modified for segmentation and recognition of objects in local context. Computational costs on data training is relatively low.

Data Acquisition - Getting GSV Tiles, 360-pano Images and Other Attributes

1. Use a 3 x 3 meters grid along trunk roads and footpaths to get a distinct GSV 360-pano IDs following this URL structure ("https://maps.google.com/cbk?output=xml&ll=" + latitude (in deg.deg) + longitude (in deg.deg) + "&dm=1").
2. Get 360-pano attributes, such as image dates, geographic coordinates (latitudes and longitudes), camera tiles, pitches and yaw angles and depths, etc., which are essential for rectification and geolocation in later processes.
3. Use the distinct 360-pano IDs to download all GSV tiles following this URL structure ("http://maps.google.com/cbk?output=tile&panoid=" + Panoid + "&zoom=0&x=0&y=0", where x and y are the tile numbers of the Panoid).
4. Convert colour for each tile from BGR2RGB using CV2 package.
5. Concatenate all 13(V) x 26(H) tiles in a single 360-pano image.
6. Decrypt GSV depth map and convert it as 256 x 512 pixel image following this [post](#).

Data Analytics - Object Training, Segmentation, Recognition and Transformation

1. Download the matrix results of COCO dataset riding on ResNet-50 CNN architecture for localised training.
2. Use "Labelme" to outline the trees and plants (and other concerned on-street features) and label them.
3. Divide the local dataset into training and validation ones (of ratio 7 to 3).
4. Import the COCO matrix to the ResNet-50 CNN model for localised training (and tune the hyper-parameters if required).
5. Perform the 1st complete pass of the training dataset and further passes (if required) to ensure a higher model accuracy of not less than 0.7 for validation dataset of the intersection over union (IOU) at 0.5.
6. Label further trees and plants to enhance the model accuracy if needed.
7. Predict trees, plants and other on-street features using the model of the confident score (reflecting how likely the box contains an object and how accurate is the boundary box) not less than 0.7.
8. Export the image coordinates of boundary boxes and on-ground trees, plants and on-street features from the 360-pano images.
9. Transfer them to the 3D world coordinates and reduce their dimensions, e.g. heights, spreads and on-ground positions, etc.

Data Visualization - Importing 3D Objects to 3D Web Map View

1. Export the results to "model.js" file, including geographic coordinates (latitudes and longitudes), heights of objects (for scaling in the 3D map), orientations, types of objects/features, and the relative directories of the 3D models.
2. All 3D models should be reduced to the Y-scale at 1 meter in the "Blender" software, trimmed with minimal textual sizes, vertexes, edges and faces, exported to ".glb" format (other formats such as ".fbx" or ".dae" are also supported but requiring further configuration and additional parameters in "main.js" file), and stored in the ".model" folder. Scaling of the object in the model will be applied automatically based on the heights provided to.
3. Basic optimization for data streaming to local computer have been configured to allow massive repeated 3D objects being loaded and viewed smoothly (note: the web demonstration loads over 1,700 3D objects). To allow the best user experience of enjoying the virtual tour, further optimization (e.g. clear 'out-of-screen' memory from RAM of local computer) is required in coming future. Please **AVOID loading heavy 3D models or over 3,000 tiny objects/models** in this web demonstration prototype.

Data Visualization - Configuration of Flight Line/ Walk Through Pathes

1. Open "flightpath.js" file and copy the flightline attributes from Mapbox (including latitudes, longitudes, zoom levels, bearings, pitches, moving speeds, minimum zoom levels).
2. No limitation for the number of stopping/staying point using this file.

Virtual Tour at West Kowloon Cultural District (WKCD)

1. Standard navigation using mouse and three mouse buttons.
2. Drag any 3D objects by pressing "Shift" + left mouse click.
3. Rotate any 3D object by pressing "Alt" + left mouse click.
4. Move any 3D objects up and down (along z-axis) by pressing "Ctrl" + left mouse click.
5. [Enjoy it through this link](#).

Last update: 11 Apr 2022