Group 97: Road Segmentation

Laurent Lejeune, Tatiana Fountoukidou, Guillaume de Montauzon

I. Related works

A. Road Segmentation in Aerial Images by Exploiting Road Vector Data [1]

B. Morphological road segmentation in urban areas from high resolution satellite images [2]

C. Connected Component-Based Technique for Automatic Extraction of Road Centerline in High Resolution Satellite Images [3]

D. Machine Learning Based Road Detection from High Resolution Imagery

E. Road Extraction Using K-Means Clustering and Morphological Operations [4]

References

[1] J. Yuan and A. M. Cheriyadat, "Road segmentation in aerial images by exploiting road vector data," in *2013 Fourth International Conference on Computing for Geospatial Research and Application*, pp. 16–23, July 2013.

[2] R. Gaetano, J. Zerubia, G. Scarpa, and G. Poggi, "Morphological road segmentation in urban areas from high resolution satellite images," in *International Conference on Digital Signal Processing*, (Corfu, Greece), 2011.

[3] C. Sujatha and D. Selvathi, "Connected component-based technique for automatic extraction of road centerline in high resolution satellite images," *J Image Video Proc*, vol. 2015, no. 1, p. 8, 2015.

[4] R. Maurya, P. Gupta, and A. S. Shukla, "Road extraction using k-means clustering and morphological operations," in *Image Information Processing (ICIIP), 2011 International Conference on*, pp. 1–6, IEEE, 2011.

[5] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[6] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157 vol.2, 1999.

[7] Y. Lv, G. Wang, and X. Hu, "Machine Learning Based Road Detection from High Resolution Imagery," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 891–898, June 2016.

[8] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 670–677, Sept 2009.

II. Data exploration

The provided training set contains 100 images of size 400x400 along with their ground-truth. A total of 6 images are discarded because they either show a too small quantity of positive class pixels. Most images are made of grid-like roads, sometimes occluded by trees.

III. Mid-level segmentations

The image pixels are first grouped in two different manners:

1) Square patches: The image is divided in non-overlapping patches of size 16x16.
2) SLIC Superpixels (Simple Linear Iterative Clustering) [5]: Pixels are grouped in mid-level regions in an iterative manner. The algorithm starts from a regular grid of cluster centers and iteratively updates the labels of their neighboring centers based on a distance measure. This method improves over the square patches method because the pixels are already pre-segmented. Their feature vector will therefore be easier to discriminate.

IV. Feature extraction

Following an exploration of the related litterature, we select a set of features that will be extracted.

- SIFT (Scale-Invariant Feature Transform) [6]: This descriptor is used extensively in computer-vision applications. It computes a histogram of oriented gradients on 16x16 windows centered at a keypoint and gives a descriptor of 128 scalar values. The keypoint detection step is not performed, instead we extract the descriptors on a dense grid and encode them in a "bag-of-features" manner through the following steps:
  1) Based on a sufficiently large number of SIFT descriptors computed on 10 images, we start by fitting a PCA model. We have checked that the explained variance at 60 components is above 99%.
  2) A codebook is generated on the aforementioned training samples. A codebook is merely a set of K-means clusters that is used to encode the input (compressed) descriptors to integer values.
  3) We then compute a normalized histogram of codes (bag-of-features) in each segment. This gives us a single texture feature vector for mid-level regions.
- Hough line transform: This transformation has already been used in a state-of-the-art method [7]. First, the edge map is computed using a canny edge detector. Given some parameters, a set of lines are extracted on the edge maps and sorted based on

their RGB variance, i.e. we want to keep the lines along which the color variations is minimal.
- Euclidean distance transform. This straightforward transform is used to compute, at each pixel location, the shortest "taxicab" distance to an edge pixel. Again, a canny edge map is used as input.

V. METHODS

Two (several) methods have been implemented and tested. A Conditional Random Field approach will provide a baseline. It will be compared to a Convolution Neural Network approach.

A. Refinement using Conditional Random Field

Using a Conditional Random Field model, one can exploit the spatial relations between mid-level regions. Indeed, a segment considered as road gives a strong prior on the "roadness" of its neighboring segment. This is formalized as an undirected graph on which the node features, also called unary potentials, contain regression probabilities given by a "segment-wise" estimator. Inspired by [8], the edge costs are made off of two features: The difference in mean LUV color, and the length of the frontier between two superpixels. This last feature allows to penalize superpixels that are "weakly" connected.

Formally, structured models aim at maximizing an energy functions of the form:

$$\begin{aligned} E_w(X,Y) &= \sum_{i \in \mathcal{V}} E_{data}(y_i; x_i) + \sum_{i,j \in \mathcal{E}} E_{smooth}(y_i; y_j) \\ &= \mathbf{w}^T \psi(X,Y) \end{aligned} \tag{1}$$

Where $\mathcal{V}$ is the set of vertices representing a segment, $\mathcal{E}$ are the edges. The data and smoothness term are combined in the joint-features vector ψ. A logistic regression model is used for the data term. Following [8], the pairwise edges potentials are given by:

$$\phi(c_i, c_j | s_i, s_j) = \frac{L(s_i, s_j)}{1 + \|s_i - s_j\|} \tag{2}$$

Where c and s are the mean LUV-space colors. The function L expresses the length of the shared boundaries between two segments.