Automatic Discovery and Geotagging of Objects from Street View Imagery

Vladimir A. Krylov Eamonn Kenny Rozenn Dahyot
ADAPT Centre, School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland

Abstract

Many applications such as autonomous navigation, urban planning and asset monitoring, rely on the availability of accurate information about objects and their geolocations. In this paper we propose to automatically detect and compute the GPS coordinates of recurring stationary objects of interest using street view imagery. Our processing pipeline relies on two fully convolutional neural networks: the first segments objects in the images while the second estimates their distances from the camera. To geolocate all the detected objects coherently and merge together instances observed from multiple images we propose a novel Markov Random Field model based on triangulation. We show experimentally the effectiveness of our approach on two object classes: traffic lights and telegraph poles. The experiments report high object recall rates and GPS accuracy within 2 meters, which is comparable with the precision of single-frequency GPS receivers.

1. Introduction

The rapid development of computer vision and machine learning techniques in recent decades has excited the ever-growing interest in automatic analysis of huge image datasets accumulated by companies and individual users all around the world. Image databases with GPS information, such as Google Street View (GSV) and images posted on social networks such as Twitter, are now widely available online and can be queried seamlessly using APIs, conveniently set up and regularly updated with new image data by the providing companies. Street view imagery (e.g., from GSV, Bing Streetside, Mapillary) represents a collection of billions of geotagged images covering millions of kilometers of roads and depicting street view scenes collected at regular intervals. This incredible amount of visual data allows to address a multitude of detection and mapping problems by exploring areas remotely through imagery.

A lot of research has been dedicated to leveraging street view imagery in combination with other data-sources such as remotely sensed imagery [14, 22] or crowd-sourced information [7] to discover particular types of objects or areas. Here we address the problem of automated discovery and geotagging of recurring objects using street view images as a sole source of input data, see Fig. 1. We consider any class of stationary objects sufficiently compact to have a single geotag that are typically located along the roads, like street furniture (post boxes, various poles and street-lamps, traffic lights and signs, transport stops, benches, etc.), small facade elements (cameras, antennas, security alarm boxes, etc.) and minor landmarks. Inventory and precise GPS mapping of such objects is a highly relevant task and, indeed, OpenStreetMap and Mapillary are currently encouraging their users to contribute such information to their databases manually. Nevertheless, the vast majority of these objects can be mapped automatically by efficiently exploring the publicly available street view imagery. To the best of our knowledge, no solution to this problem relying on street view imagery alone has been proposed in the literature.

The considered recurrent objects are not required to be visible from everywhere and may be occluded from some camera positions. In this work we do not resort to any explicit geometrical modeling of object shapes, nor do we rely on any object location patterns. In order to enable automatic geolocation we enforce a mild assumption of object sparsity: we expect the objects to be located at least 1m apart.

We propose an image processing pipeline with the following main components: (i) a state-of-the-art fully convo-
Figure 2. Flowchart of the proposed geotagging pipeline.

A convolutional neural network (FCNN) to locate objects in images, (ii) another FCNN to perform monocular depth evaluation, and (iii) a novel geotagging model that identifies object positions by combining depth information and geometric triangulation through a Markov Random Field (MRF) formulation. The flowchart of the propose pipeline is presented in Fig. 2. We evaluate the performance of this vision pipeline on two object classes: traffic lights and telegraph poles. We perform extensive experimental analysis to confirm the accuracy of both object discovery and geotagging.

The central contribution of this work is the design of a complete pipeline for geotagging recurring objects from street view imagery based on two state-of-the-art FCNNs and a novel triangulation-driven geotagging procedure.

The paper is organized as follows. We first review some relevant state-of-the-art approaches in Sec. 2. Our complete geotagging pipeline is presented in Sec. 3 and then validated experimentally in Sec. 4. Sec. 5 concludes this study.

2. Related work

In the last decade a considerable effort has been directed towards intelligent use of street view imagery in multiple applied areas: mapping [7, 22], image referencing [8], navigation [14], rendering and visualization [3, 5], etc. Methods designed in [14, 22] use street view combined with aerial imagery to achieve fine-grained road segmentation, and object detection (trees), respectively. These methods rely on object discovery through street view imagery and recover position from aerial data. In [8] street view images are used as reference to geolocate query images by resorting to scene matching. The GSV imagery is employed in [3, 5] in conjunction with social media, like Twitter, to perform visualization and 3D rendering. In [19] geotagging of road signs is performed based on explicit geometrical shape modeling, on images obtained by a custom street view camera.

Image segmentation is one of the central tasks in object extraction. A multitude of approaches have been proposed to address this problem: starting from elementary pattern analysis techniques, such as Hough transform and local gradients, through feature extraction-based tools that rely on cascades of weak classifiers [21], to more advanced machine learning methods such as random forests, support vector machines and convolutional neural networks (CNNs) [9, 18]. The latter have recently pushed the machine vision techniques to new heights by allowing automatic feature selection and unprecedented capacity to efficiently learn from huge volumes of data. FCNNs are a natural form of CNNs when addressing dense segmentation since they allow location information to be retained through the decision process by relying solely on convolutional layers [17]. Their output can be in the form of bounding boxes [15, 13] or complete segmentation maps obtained via deconvolutions only [17] or by resorting to conditional random fields [23].

Estimation of camera-to-object distances and, more generally, 3D scene analysis from RGB images can be addressed in several ways. The most explored are the stereo-vision approaches [16] that estimate camera-to-object distances from multi-view stereo image disparity analysis, or perform scene reconstruction via Structure-from-Motion methods [20]. These rely on various assumptions about camera positions and trajectory, and typically require a rich set of input RGB images and a certain form of knowledge about the analyzed scene. These classes of methods rely heavily on feature extraction and matching. Another way to address scene depth evaluation has been explored in several recent studies [6, 11, 12]. The central idea in these is to recover depth information from monocular images based on the extensive scene interpretation capabilities of FCNNs by training them on RGB and depth or disparity images. These methods achieve consistent yet approximate results relying solely on the information available in the color bands.

3. Object geotagging using street view images

Street view images are harvested by defining the area of interest with the corresponding GPS coordinates and querying the API to download the relevant geotagged imagery. Each image is processed independently allowing efficient parallelization of this most time-consuming step. All im-
Thus, we choose to rely on approximate estimates reported

dates. Furthermore, no a priori information is available
artifacts and, occasionally, distinct image acquisition
such sequences are typically characterized by substantial
result in a false positive object discovery, whereas missed
weight) that penalizes only the false positives, thus effec-
tification problem we reshape all layers in the upsampling part
by the depth evaluation FCNN, are processed by the depth evaluation FCNN.
we employ the FCN-8s all-at-once model that reports the finest
by performing deconvolutions in the form of bilinear interpolations in three consecutive steps.
Note that it is also possible to use a single
since we consider an object vs. background segmentation problem we reshape all layers in the upsampling part
as well as the last convolution in the subsampling part to have two outputs - one for background and another for the objects of interest. Note that it is also possible to use a single
employed projection system (e.g., WGS-84). Our operational assumption is that the objects are sufficiently sparse to be successfully identified as separate instances by the segmentation pipeline. Hence, whenever multiple objects are tightly clustered (less than 1 meter apart) they are recovered as a single object instance with a unique geotag.
There are two limitations associated with this strategy: first the depth estimates are not very accurate, and second, the list of all objects detected in all input images is largely redundant due to objects observed from multiple camera positions. Indeed, each of such objects spawns multiple detected object instances with distinct GPS coordinates due to depth evaluation inaccuracies. Thus, as an alternative to direct depth-based geotagging, we consider strategies where each object is observable from several camera positions.

3.2.2 Object locations from multiple view images

When an object is observed multiple times the image segmentation map needs to be matched to group together observations of the same object-instances. Such matching could also help estimate a more accurate GPS location using standard strategies from multiple view geometry. Matching could be addressed in two ways: either by relying on general image features, such as SIFT, HOG, etc., or custom features extracted by the employed segmentation CNN.
We have explored both ways but neither provides satisfactory quality of matching due to the strong degree of similarity between the objects in the considered object-classes.
3.2.3 Markov Random Field formulation

**Space of intersections** We consider the space $X$ of all pairwise intersections of view-rays from camera locations (c.f. Fig. 3). Any location $x \in X$ is generated by the intersection of two rays $r_1$ and $r_2$ from camera view pair. The binary label $z \in \{0, 1\}$ is associated to $x$ to indicate the presence ($z = 1$, referred to as positive intersection depicted as blue dots in Fig. 3) or absence ($z = 0$, empty intersection, yellow in Fig. 3) of the object of interest at the corresponding intersection (see side panel in Fig. 3). The space $Z$ of all intersections’ labels is then a binary Markov Random field (MRF) [10]. We construct this MRF as follows:

- To each site $x$ we associate two Euclidean distances $d_1$ and $d_2$ from cameras: $d_j = ||C_j - x||$, where $C_j$ are locations of two cameras ($j = \{1, 2\}$) from which intersection $x$ is observed along the rays $r_1$ and $r_2$, respectively. Any intersection $x$ is considered in $X$, only if $d_1, d_2 < 25m$. In Fig. 3, red intersection in the upper part of the scene is rejected as too distant from Camera3.
- The neighborhood of node $x$ is defined as the set of all other locations $x_k$ in the field occurring on the rays $r_1$ and $r_2$. Indeed, the state of each intersection depends only on its neighbors on the rays. Note that the number of neighbors (i.e. neighborhood size) depends on the location $x$ and its generating rays: each node $x$ on the grid $X$ has its own unique number.
- Each ray can have a positive intersection with at most one ray from any single camera, but could have positive intersections with rays generated from multiple cameras.

**MRF energy** The MRF configuration is defined by $\{(x_i, z_i)\}_{i=1,\ldots,N}$. For each site $x$ with state $z$ the associated MRF energy [10] is defined through the following energy terms:

- A unary energy term enforces consistency with the depth estimation. Specifically, the deep learning pipeline for depth estimation provides estimates $\Delta_1$ and $\Delta_2$ of distances between camera positions and the detected object at location $x$. We formulate the term as a penalty for mismatch between triangulated distances and depth estimates:

$$ u_1(z) = z \sum_{j=1,2} ||\Delta_j - d_j|| $$  \hspace{1cm} (1) 

- Pairwise energy term is introduced to penalize: (i) multiple objects of interest occluding each other, and (ii) excessive spread in case an object is characterized as several intersections. In other words, we tolerate several positive intersections on the same ray only when they are in close proximity. This may occur in multi-view scenario due to segmentation inaccuracies and noise in camera GPS coordinates. For example, in Fig. 3 Object1 is detected as a triangle of positive intersections (blue dots) - two on each of the three rays.

Two distant positive intersections on the same ray correspond to a scenario when an object closer to the camera occludes the second more distant object. Since we consider

![Figure 3. Object positions are identified based on triangulation of view-rays from camera positions. An example of irregular neighborhood of nodes is demonstrated in the side panel.](image-url)
compact objects with negligible volume we can assume this type of occlusion due to perfect object alignment with the camera position unlikely.

This term depends on the current state $z$ and those of its neighbors $z_k$. It penalizes proportionally to the distance to any other positive intersections $x_k$ on rays $r_1$ and $r_2$:

$$u_2(z) = z \sum_k z_k \|x - x_k\|$$  \hspace{1cm} (2)

- A final energy term penalizes rays that have no positive intersections: false positives or objects discovered from a single camera position (see Fig. 3). This can be written as:

$$u_3(z) = (1 - z) \prod_k (1 - z_k)$$  \hspace{1cm} (3)

The full energy of configuration $z$ over $X$ is then defined as sum of energy contributions over all $N$ sites in $Z$:

$$U(z) = \sum_{i=1}^N \left[ \alpha u_1(z_i) + \beta u_2(z_i) + (1 - \alpha - \beta) u_3(z_i) \right]$$  \hspace{1cm} (4)

subject to $\alpha, \beta \geq 0$, $\alpha + \beta \leq 1$.

The optimal configuration is characterized by the global minimum of the energy $U(z)$. The terms $u_1$ and $u_2$ penalize too many objects by increasing the total energy (for any positive intersection with $z = 1$ both $u_1 \geq 0$ and $u_2 \geq 0$), whereas $u_3$ penalizes too few objects ($u_3 = 1$ for any ray with no positive intersections). If required, the employed MRF formulation allows us to explicitly accommodate object location pattern assumptions through additional higher-order penalty terms.

**MRF optimization** We perform energy minimization with Iterative Conditional Modes algorithm [10] starting from an empty configuration: $z_i^0 = 0, \forall z_i \in Z$. This (local) optimization is run according to a random node-revisiting schedule until local minimum is reached and no further changes are accepted. Experimentally, we have observed stable performance of the optimizer over multiple reruns. Therefore, we there is no need to explore any more accurate global optimizers, see in [10].

**Clustering** The final object configuration is obtained by resorting to clustering in order to identify groups of positive intersections that describe the same object$^2$. Indeed this is required since we consider the space $X$ of all pairwise intersections, whereas some objects are observed from three or more camera positions and result in multiple detected object instances. For example, in Fig. 3 Object1 is identified as three distinct positive intersections tightly scattered around the object. In this work we employ hierarchical clustering with a maximum intra-cluster distance of $1 \text{m}$ which corresponds to our object sparsity assumption. Intersection coordinates for locations in the same cluster are averaged to obtain the final object’s geotag.

### 4. Experimental study

We test our pipeline on GSV imagery for detection of two object types: traffic lights and telegraph poles. Both kinds of objects are compact enough to be attributed a single geotag and are predominantly visible from roads. Traffic lights are visible by design and telegraph poles are also typically erected in the close vicinity of the road network.

In order to evaluate the accuracy of the estimated GPS coordinates we deploy our detector in several areas covered by GSV imagery. To extract the latter we first create a dense grid of GPS coordinates along road-centers with 5m step, query the API to retrieve closest available GSV panoramas in 4 parts each with 90 degrees field of view at $640 \times 640$ pixels resolution. Note that GSV imagery often demonstrates strong stitching artifacts, in particular in the top and bottom parts of the images, see examples in Fig. 7.

For the segmentation FCNN we use a Caffe implementation and perform training on a single NVIDIA TITAN X GPU using stochastic gradient descent with fixed learning rate of $5e^{-11}$ and momentum $m = 0.99$, with batches of 2 images. This choice is in line with recommendations in [17] and empirically demonstrated the best performance. We resort to data augmentation by performing random horizontal flipping, rotations $[-5^\circ, 5^\circ]$, small enhancements of input image’s brightness, sharpness and color. The imbalance between classes - object and background - in the datasets is handled well by the FCNN provided we maintain a fraction of object-containing images in the training set above 25% at all times. The inference speed is at $8 \text{fps}$. The depth estimation pipeline [11] is used in the authors’ implementation in MatConvNet with no modifications. The energy weights in Eq. (4) are set to $\alpha = 0.2$ and $\beta = 0.6$, both segmentation FCNN loss functions have the same weight.

#### 4.1. Traffic lights

The image segmentation pipeline on traffic lights is trained on data from two publicly-available datasets with pixel-level annotations: Mapillary Vistas [1] and Cityscapes [4]. We crop/resize the images to match the standard GSV image size of $640 \times 640$. This provides us with approx. 18.5K training images containing large traffic lights (at least $10 \times 10$ pixels each instance). We start with a
PASCAL VOC-pretrained model and carry on training with learning rate of 5e-11 for another 200 epochs.

**Object detection** The test pixel-level precision plateaus at 72.3% and recall at 73.5%. If over 25% of an object's pixels are labeled correctly we consider such object instance to be recovered accurately. In the vicinity of camera (within 25m from camera position) the instance-level precision is 95.1% and recall is 98.1% on the test set. In this experiment the introduction of the second FCNN loss function has improved the instance-level precision by 2.2% and decreased the recall by 0.2%. In general, instance-level recall is a more important characteristic for geotagging of objects, whereas a lower precision (false positives) is partially compensated at the later stage by the employed tagging procedure which requires objects to be observed *multiple* times. In this study we adopt a conservative strategy of ignoring objects located farther than 25m from camera positions due to the pronounced performance drop of image processing reliability on distant objects: substantial decrease of recall rate in semantic segmentation and high variance in depth estimation. Note that farther objects may be discovered through semantic segmentation but are rejected by the geotagging procedure described in Sec. 3.2 through the maximal distance restriction.

In Fig. 7 we show examples of traffic light segmentation in busy urban scenes. The performance is overall very good with minor issues of occasional small false positives and on some traffic lights facing away from the camera (see third and fourth row images). Neither of the two poses particular problems for geotagging since the processing is done on series of images, which allows us to observe most traffic lights both from front and rear and discard false positives due to their inconsistent and / or non-stationary nature. As far as the pixel-level performance is concerned a high precision rate is not necessary since we require pixel-level labels solely to access the relevant depth estimates obtained through the depth evaluation pipeline.

**Geotagging performance** To evaluate the geotagging performance we consider a 0.8 km stretch of Regent street in London, UK, covered by 87 GSV panoramas, see Fig. 4. This dense urban area has five clusters of traffic lights, totaling 50 individual objects. To avoid ambiguity in the count we assume that traffic lights with multiple sections (e.g., separate sections for cars and pedestrians) attached to a single pole are counted as a single object. The configuration of objects in this area satisfies the sparsity assumption. Note that we do not have the precise GPS coordinates of the traffic lights for this dataset and perform our analysis based on a human interpretation of the GSV imagery.

The segmentation pipeline reports 179 traffic lights instances, the geotagging results in 70 positive intersections and, finally, the number of clustered objects is 51. We consider objects to be recovered accurately if they are located within 2 meters from the manually identified position. This choice is in line with the official reports on the accuracy of GPS measurements that establish a 1.89m 95%-confidence interval for horizontal error for high quality single-frequency GPS receivers [2]. The object discovery results are summarized in Table 1. Of the 4 false positives 2 correspond to actual objects that are approx. 3m from reference locations. 6 detected objects were not matched with objects from other images (rays with no positive intersections): 4 are segmentation false positives, 2 - objects identified from only one image.

### 4.2. Telegraph poles

Segmentation of the telegraph poles relies on a custom training dataset. Indeed, even though multiple existing datasets incorporate class ‘poles’, none of them provides sufficient distinction between different types of poles: lampposts, poles carrying traffic lights and / or road signs, bollards (sidewalk poles), and utility poles (electricity, telegraph). We consider here a particular class of telegraph poles that are actively used in the Republic of Ireland. The corresponding network is very large with over a million of poles spread throughout the country. Overall, the main chal-

| Object | #Actual | #Detected | TP | FP | FN | Recall | Precision |
|--------|---------|-----------|----|----|----|--------|-----------|
| Traffic light | 50 | 51 | 47 | 4 | 3 | 94.0% | 92.2% |
| Telegraph pole | 77 | 75 | 72 | 3 | 5 | 93.5% | 96.0% |

Table 1. Instance-level object discovery results for traffic lights and telegraph poles in the test areas. (TP: True Positives, FP: False Positives, FN: False Negatives)
The challenge in detection of such poles is their strong visual similarity to other tall poles, in particular, electricity poles.

To prepare a segmentation pipeline for this object class we adopt the following procedure: we first train our FCNN to detect all tall poles: utilities and lampposts, and then fine-tune it to extract only telegraph poles. We do the former by combining existing datasets, Mapillary Vistas [1] and Cityscapes [4], with GSV imagery extracted at locations provided through a training GPS database of telegraph poles made available to us for this study. Note that this database cannot be used directly for segmentation training for the following reasons: First, because of the inherent GPS inaccuracy in both pole and GSV coordinates, and frequent occlusions, one cannot expect to observe poles in the center of images extracted based on geotags alone. Second, many images depict geometry rich scenes, with mixtures of different pole-types and occasional other strong vertical features, such as trees, facade elements, antennas, etc. To train the FCNN to distinguish tall poles from any other objects in the scenes we put together a dataset with 20K annotated images. We run this first step of training for 100 epochs to achieve satisfactory discriminative power. As a second step we fine-tune this network on our custom pixel-level annotated set of 500 telegraph pole images. To further boost the discriminative power of the FCNN we add 15K GSV scenes collected in areas with no telegraph poles but in presence of lampposts and electricity poles. The training is run for another 200 epochs at the learning rate of $2\times10^{-11}$.

**Object detection** Sample pole segmentation maps are presented in Fig. 7. The final FCNN reports pixel-level recall of 93.6% and precision 82.2%. As above, we consider relevant only the objects within 25m of the camera. This ensures accurate distinction from other tall poles since the decisive subtle differences are not visible at greater distances. The object-level test recall is 97.9% and precision is 92.7%.

We perform experimental analysis in the Rathmines area of Dublin, Ireland, see Fig. 5. The area under study has approx. 8 km of roads covered by 945 GSV panoramas. 77 telegraph poles are visible through GSV imagery. 273 instances are discovered via the image segmentation pipeline, and the geotagging results are summarized in Table 1. Out of the 3 false positives - 2 are wrong objects and 1 is a telegraph pole 3m away from its true location.

**Geotagging performance** We now evaluate the accuracy of geotags compared to the ground truth manually collected with a single-frequency GPS device: each pair of coordinates has been recorded three times on three different days and averaged to produce a single reference geotag for each object. This ground truth contains 41 poles in the considered area. These poles have been automatically detected in 89 GSV images, and all 41 discovered in at least 2 images enabling our method to infer their GPS coordinates.

Table 2 reports the empirical statistics of pole-to-camera distance estimation error: mean, median, variance, 95% empirical confidence interval as obtained with our triangulation procedure. Note the empirical 95%-confidence interval is at 2.07m, which is comparable with the 1.89m confidence interval for single-frequency GPS receivers [2].

In Table 2 we report the results of geotagging based on depth estimation from 89 monocular views without resorting to triangulation. We observe a substantial improvement of GPS accuracy obtained with our triangulation procedure. In both cases - triangulation and depth FCNN - the geotags rely entirely on the accuracy of the input camera coordinates provided with GSV imagery. Any outliers in camera geolocations result directly in object geotag errors.
5. Conclusions

We have proposed a technique for automatic detection and geotagging of recurring objects from street view imagery. The developed method is fully automatic and can be employed to geotag any kind of stationary road-side objects such as street furniture, small landmarks, etc.

The proposed solution takes advantage of two existing deep learning pipelines, one fine-tuned for our needs while the other employed as-is. A triangulation-based MRF has been formulated to extract object geolocations and avoid duplicates and false positives. The experimental analysis has demonstrated high accuracy in both discovering and geotagging the objects. The accuracy of the latter is empirically similar to that obtained with a single-frequency GPS receiver. We stress that the geotagging module is independent and can be combined with any segmentation and depth pipelines fine-tuned for any specific object type.

One of the directions for future work would be the design of a unified FCNN architecture for segmentation and depth which, in the absence of a priori depth data, can be trained at the second stage on the results of the triangulation procedure presented here. This may allow partial or complete relaxation of the sparsity assumptions by providing improved input to the triangulation-driven geotagging pipeline.

Acknowledgement

This research was supported by eir - the principal provider of fixed-line and mobile telecommunications services in Ireland - who are actively investing in research into machine learning and image processing for the purposes of network planning and maintenance.

References

[1] Mapillary Vistas Dataset. https://www.mapillary.com/dataset/vistas. 5, 7
[2] GPS Performance Analysis Report #96. Technical report, Federal Aviation Administration, USA, 2017. www.nstb.tc.faa.gov/reports/PAN96_0117.pdf. 6, 7
[3] A. Bulbul and R. Dahyot. Social media based 3d visual popularity. Computers & Graphics, 63:28 – 36, 2017. 2
[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In Proc IEEE Conf on CVPR, pages 3213–3223, 2016. 5, 7
[5] R. Du and A. Varshney. Social Street View: Blending Immersive Street Views with Geo-tagged Social Media. In Proc Int’l Conf on Web3D Technology, pages 77–85, 2016. 2
[6] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. 2017. 2
[7] K. Hara, S. Azenkot, M. Campbell, C. L. Bennett, V. Le, S. Pannella, R. Moore, K. Minckler, R. H. Ng, and J. E. Froehlich. Improving public transit accessibility for blind riders by crowdsourcing bus stop landmark locations with Google Street View: An extended analysis. ACM Trans Accessible Computing, 6(2):5, 2015. 1, 2
[8] J. Hays and A. A. Efros. IM2GPS: estimating geographic information from a single image. In Proc IEEE Conf on CVPR, pages 1–8, 2008. 2
[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proc IEEE Conf on CVPR, pages 770–778, 2016. 2, 3
[10] Z. Kato and J. Zerubia. Markov random fields in image segmentation. Foundations and Trends in Signal Processing, 5(1–2):1–155. 2012. 4, 5
[11] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In Proc Int’l Conf on 3D Vision, pages 239–248, 2016. 2, 3, 5
[12] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In Proc IEEE Conf on CVPR, pages 1119–1127, 2015. 2
[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single Shot MultiBox Detector. In Proc of ECCV, 2016. 2
[14] G. Mattyus, S. Wang, S. Fidler, and R. Urtasun. HD maps: Fine-grained road segmentation by parsing ground and aerial images. In Proc IEEE Conf on CVPR, pages 3611–3619, 2016. 1, 2
[15] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE T-PAMI, 39(6):1137–1149, 2017. 2
[16] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In Proc IEEE Conf on CVPR, volume 1, pages 519–528, 2006. 2
[17] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. IEEE T-PAMI, 39(4):640–651, 2017. 2, 3, 5, 10
[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015. 2, 3
[19] B. Soheilian, N. Paparoditis, and B. Vallet. Detection and 3D reconstruction of traffic signs from multiple view color images. ISPRS J Photogram Remote Sens, 77:1–20, 2013. 2
[20] R. Tron, X. Zhou, and K. Daniilidis. A survey on rotation optimization in structure from motion. In Proc IEEE Conf on CVPR Workshops, pages 77–85, 2016. 2
[21] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proc IEEE Conf on CVPR, volume 1, pages I–511–I–518 vol.1, 2001. 2
[22] J. D. Wegner, S. Branson, D. Hall, K. Schindler, and P. Perona. Cataloging public objects using aerial and street-level images-urban trees. In Proc IEEE Conf on CVPR, pages 6014–6023, 2016. 1, 2
[23] Z. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In Proc IEEE Conf on CVPR, pages 1529–1537, 2015. 2
Figure 6. Traffic light segmentation on Google Street View images. First two rows: correct segmentation. Third row—noisy segmentation, left image fourth row—false negative (traffic light from rear). Rest—false positives: reflected traffic light, other objects.

Figure 7. Telegraph pole segmentation on Google Street View images. First two rows: correct segmentation. Third row—partial detection, fourth row—partial detection and strong stitching, fifth row—false positives: ivy-covered tree and electricity pole.
Supplementary material

2-stage telegraph pole segmentation training

The challenge of telegraph pole segmentation is in teaching the FCNN to accurately distinguish this specific class of poles from any other poles and similar objects, in particular electricity poles, lampposts, trees (with little or no leaves). The considered telegraph poles are made of wood and visually have a natural ‘wooden’ texture (no artificial coloring due to paint). They are never used for mounting lamps and there is no shared use of poles with electricity distribution network. Hence, this typology of poles is a clearly defined subset of all tall poles, that include all lampposts and utility poles. There are several more subtle distinctive visual features about telegraph poles, like steps for climbing, particular types of insulation in use, and specific types of objects mounted on the poles, but none are necessarily present (or visible) on any particular pole. Telegraph poles are occasionally covered by vegetation: overgrown by ivy or closely surrounded by trees.

Strong stitching often does not allow us to accurately identify the pole. This occurs when the camera is located very close to the pole and its upper half (containing visual information necessary to identify its type) is stitched incorrectly so that the pole is cut into two disconnected parts.

For all of these reasons we rely on a 2-stage training: first, using the proprietary telegraph-pole GPS dataset and publicly available Cityscapes and Mapillary Vistas datasets to achieve detection of all poles, and, second, on our custom dataset with 500 fine telegraph pole segmentations and multiple negative examples to obtain the final segmentation pipeline. The first step is required since the 500-images custom dataset is not sufficient to train the FCNN to discriminate strong vertical features such as facade elements, antennas, fences, trees, etc.

The learning rates are 5e-11 (100 epochs) and 2e-11 (200 epochs) for the first and second stages, respectively. These fairly low learning rates are as suggested by Shelhamer et al. [17] and allow the employed VGG-16-like architecture (FCN-8s all-at-once) with deconvolutions and skip connections to be trained in an end-to-end fashion. Experiments with other (higher) learning rates and optimizers (ADAM, ADAGRAD) reported worse convergences.

In Fig. 8 we present multiple complex GSV scenes where our custom-trained segmentation FCNN reported accurate results. All the rejected objects in these scenes are not telegraph poles but rather electricity poles, lampposts or trees.

Figure 8. Examples of correct telegraph pole segmentation in the presence of other similar vertical objects, such as electricity poles, lampposts and trees.