GIS-Assisted Object Detection and Geospatial Localization

Shervin Ardeshir, Amir Roshan Zamir, Alejandro Torroella, and Mubarak Shah

Center for Research in Computer Vision at the University of Central Florida
http://crcv.ucf.edu/projects/GIS-Object/

Abstract. Geographical Information System (GIS) databases contain information about many objects, such as traffic signals, road signs, fire hydrants, etc. in urban areas. This wealth of information can be utilized for assisting various computer vision tasks. In this paper, we propose a method for improving object detection using a set of priors acquired from GIS databases. Given a database of object locations from GIS and a query image with metadata, we compute the expected spatial location of the visible objects in the image. We also perform object detection in the query image (e.g., using DPM) and obtain a set of candidate bounding boxes for the objects. Then, we fuse the GIS priors with the potential detections to find the final object bounding boxes. To cope with various inaccuracies and practical complications, such as noisy metadata, occlusion, inaccuracies in GIS, and poor candidate detections, we formulate our fusion as a higher-order graph matching problem which we robustly solve using RANSAC. We demonstrate that this approach outperforms well established object detectors, such as DPM, with a large margin.

Furthermore, we propose that the GIS objects can be used as cues for discovering the location where an image was taken. Our hypothesis is based on the idea that the objects visible in one image, along with their relative spatial location, provide distinctive cues for the geo-location. In order to estimate the geo-location based on the generic objects, we perform a search on a dense grid of locations over the covered area. We assign a score to each location based on the similarity of its GIS objects and the imperfect object detections in the image. We demonstrate that over a broad urban area of >10 square kilometers, this semantic approach can significantly narrow down the localization search space, and occasionally, even find the correct location.

1 Introduction

Currently, the accurate locations of many static objects in urban areas, e.g., bus stops, road signs, fire hydrants, ATMs, subway stations, building outlines, and even trees, are documented in databases, commonly known as GIS. Such databases provide valuable semantically meaningful information, and their coverage and accuracy is constantly increasing. Therefore, it is natural to develop computer vision systems which can effectively leverage this information.

D. Fleet et al. (Eds.): ECCV 2014, Part VI, LNCS 8694, pp. 602–617, 2014.

In this context, we propose a method for improving object detection in images of outdoor urban scenes using GIS. The metadata of images are commonly available with the image in Exif tags. Using the GIS databases and the metadata, we project the GIS objects onto the image and use them as priors for object detection. However, a slight inaccuracy in the metadata or the GIS database, which is quite common, leads to completely misplaced projections which makes fusing them with the image content challenging. We resolve this issue by formulating our object detection problem as a higher-order graph matching instance which we solve using robust estimation techniques.

Furthermore, in the same context, we extend the use of GIS to even the cases with unknown camera locations and show that the objects in the image and their relative spatial relationship can be used for finding the GPS location where an image was taken. As an example, assume a traffic signal, a bus stop, and a trash can are visible in an image. At a city scale, this information would narrow down the feasible geo-locations for this image to some extent if the geo-locations of all traffic signals, bus stops, trash cans are known. Now assume it is known that the trash can is towards the north of the traffic signal, and the bus stop is on their east. One would imagine that there will not exist many locations in the city consistent with this arrangement, even though the objects are generic and common. We will show that, interestingly, the geometric relationship between these generic objects in fact provides distinctive cues for discovering the location of an image. However, the inaccuracies in GIS and the imperfect performance of state-of-the-art-object detectors are some of the critical issue which we address in our framework by employing a robust matching method.

GIS has been used for various applications in computer vision [21,22], in particular registration of aerial images based on semantic segments. As another example, Matzen et al. [14] use geographic and geometric information for detecting vehicles using a set of viewpoint-aware detectors. However, the majority of existing image geo-localization techniques do not leverage any type of semantic information, e.g., semantic objects. They are often based on establishing low-level feature correspondences, e.g., SIFT [13] or BoVW histograms, between the query image and a reference dataset of images. Several methods which adopt this approach for Street view[24,18,9,25], satellite or Birdseye view [1,12], or crowdsourced images [17,16,11,23] have been developed to date. Several other methods for identifying landmarks in images, e.g., [3], have been proposed which to some extent leverage the location-level semantic information (i.e., the landmarks). Recently, Shafique et al. [15] developed a system for coarse geographical information estimation from an image based the annotations provided by the user. Lee et al. [10] proposed a method for discovering a set of "visual styles" which can be linked to historical and geographical information. Bioret et al. [2] localized images based on matching them to building outlines. Even though some of the aforementioned methods leverage high-level information to some extent, in general, very little work towards rigorous integration of semantic information in the geo-localization process has been reported to date.

In the context of object detection, several techniques for assisting the process of detection using various type of prior context have been proposed [4,19,20]. For instance, Hoiem et al. [8] proposed a method for utilizing a set of general spatial priors for improving object detection (unlike our approach which uses location-specific priors and a graph matching formulation). However, the use of large scale and detailed GIS databases available for urban area has not been explored to date. This paper provides the first method for unifying *wide area* image localization and object detection in one robust framework which is centered around semantically tangible information.

2 GIS-Assisted Object Detection

Given the metadata, we gather a set of priors about the objects that are potentially in the field of view of the camera. Since some of them might be occluded by buildings or other objects, we perform occlusion handling to find a set of objects which are expected to be visible in the image. In addition, we perform object detection in the image to have some candidate bounding boxes using the content of the image. Then we fuse these two utilizing graph matching in order to find the final object detections.

2.1 Obtaining Priors from GIS

We want to extract a set of priors about the spatial locations of the objects in the image. We extract the metadata (e.g., the camera location, focal length, camera model (yielding the sensor size), and possibly compass direction) from the Exif tag of the images and use it for forming the camera matrix, $\boldsymbol{C}$, which maps the 3D world coordinates to the 2D image coordinates. Camera matrix has the standard form of $\boldsymbol{P} = \boldsymbol{C}[\boldsymbol{R} \mid \boldsymbol{T}]$, where $\boldsymbol{R}$, $\boldsymbol{T}$, and $\boldsymbol{C}$ are the rotation, translation, and calibration matrices, respectively. Using the information in the Exif tag, $\boldsymbol{C}$ can be obtained using the following equation:

$$C = \begin{bmatrix} f \times s_x & l_y/2 & 0 \\ 0 & 1 & 0 \\ 0 & l_x/2 & f \times s_y \end{bmatrix}, \tag{1}$$

where f is the focal length of the camera, s_x and s_y are the sensor size and l_x and l_y are the number of pixels is x and y coordinates of the image. We assume a fixed height for the camera (1.7m) and also zero roll and pitch as they are not recorded in the Exif tag (we will later see that graph matching will handle the shift caused by this approximation), but yaw is recorded by the compass. Therefore, $\boldsymbol{R}$ and $\boldsymbol{T}$ can be formed employing these approximations and the geo-location and compass information recorded in the Exif tag. The following equation yields the homogeneous coordinates of the projections of GIS objects in the image:

$$\begin{bmatrix} x_i \\ 1 \end{bmatrix} = \boldsymbol{P} X_i, \tag{2}$$

where X_i denotes the 3D GPS coordinates[1] of the i^{th} object in the GIS database which is in the field of view of the camera, and x_i represents an estimation of the two dimensional spatial location of the i^{th} object in the image plane.

In GIS, each object is represented by a single GPS location. However, the typical height of the objects, such as fire hydrants or bus stops, are known and often fixed. Thus, We assume fixed heights for the objects (e.g., 5.5m for traffic signal) and compute two projections points for each GIS object, one for its bottom and another one for its top. This process is shown in figure 1. Figure (a) shows the position and the view of the camera, in addition to the GIS objects[2]. Figure 1 (b) shows the projections, i.e., x_i. However, many of the GIS objects will be occluded by buildings or other GIS objects, so we need an occlusion handling mechanism to eliminate the occluded GIS objects from our projections which is described next.

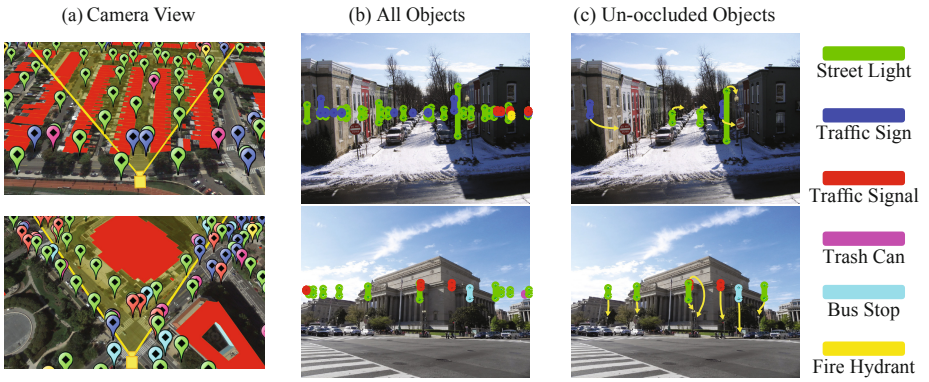

Fig. 1. The process of acquiring GIS projections for two sample images. Part (a) illustrates the camera view (yellow lines) shown in the overhead view with the occlusion segments (red) and GIS objects. Part (b) shows the projections prior to occlusion handling. Part (c) shows the final projections after occlusion handling along with their corresponding objects.

2.2 Occlusion Handling

Two main resources of the occlusions are buildings and other GIS objects. GIS database includes very accurate data about the outline of buildings. Therefore, we define an occlusion set as the union of the GIS projections and the projections of the building outlines. We perform the occlusion handling by enforcing that

[1] All GPS positions are converted to East-North-Up metric Cartesian coordinates system for the sake of simplicity.

[2] In the entire paper, street lights, trash cans, traffic signal, fire hydrants, bus stops, and traffic signs are shown with green, purple, red, yellow, cyan and blue markers, respectively.

the ray connecting the GIS object to the camera is not blocked by any member of the occlusion set.[3]

The non-occluded objects are shown for two samples in figure 1 (c). As apparent in the figure, many GIS projections are found to be occluded which indeed matches the content of the image.

2.3 Graph Matching

In practice, the GIS projections are rarely at the correct object positions in the image. That is because even a slight inaccuracy in the metadata or the employed approximations about the camera rotation parameters (roll and pitch) may considerably shift the projections from their correct position. This is can be clearly observed in figure 1 (c). However, the *relative geometric relationships* of the objects are yet preserved.

In order to compute a set of potential bounding boxes for the objects in the image, we perform a content-based detection (in our experiments using DPM [7,6]). These detections are usually far from perfect and contain a significant number of false positives due to the complexity of the scene and the nontrivial appearance of our objects. However, a subset of these DPM detections is expected to match the projections acquired from GIS. We formulate the problem of identifying this subset as graph matching.

In graph matching (assume a bipartite graph for the sake of simplicity), two sets of nodes and a number of edges between the two node sets are given; the goal is to exclusively assign the nodes of the first set to the nodes of the second set in a way that an objective function is minimized. Our aim is to assign each GIS projection to one of the DPM detections in a way that the overall geometry of the scene is preserved, even though the projections are not necessarily in the correct locations and the DPM detections include a lot of incorrect bounding boxes. Thus, we define our problem as a graph matching instance in which the first set of nodes represents the GIS projections that survived occlusion handling and the other set represents the DPM detections. Each DPM detection or GIS projection is represented by one node which is connected to all of the nodes of the *same class* in the other set through edges. In other words, each edge denotes one potential correspondence between a GIS projection and a DPM detection. Graph matching selects a subset of the edges in a way that the correspondences they induce best fit a *global affine model*. This process is illustrated in figure 2.

We assume the geometric model between the projections and the correct subset of detections can be approximated using an affine transformation. Since the inaccuracies in the metadata lead to a global transformation in the image, the projections are often translated, rotated, and scaled. Also, the GIS objects are often on the ground plane and the images in the urban area usually show a relatively wide view. Therefore, the affine model, even though not perfect, is

[3] Since the GIS projections include more than one pixel, this process is performed for all of the pixels on the projection, and an object is assumed to be occluded if more than 50% of its pixels are occluded.

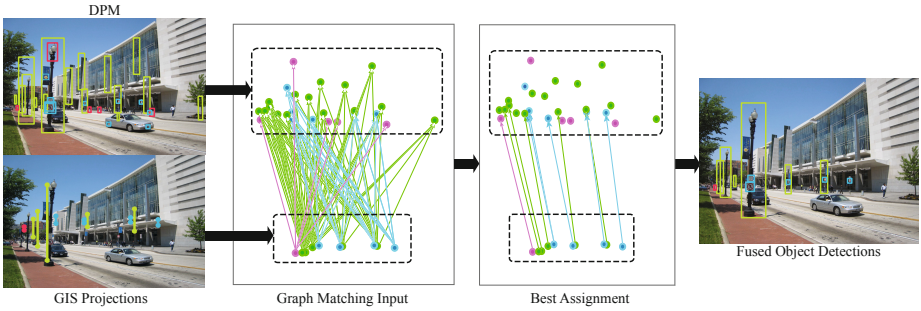

Fig. 2. An example of the graph matching process. The DPM detections and GIS projections are used to form a bipartite graph encoding the feasible assignment of objects. Solving the higher-order graph matching problem yields the best assignment which has the least cost for fitting an affine transformation. The corresponding object detection results are shown in the rightmost figure.

a reasonable approximation. In addition, the degree of freedom of affine is less than its alternatives which is desirable especially for the locations where few GIS objects are found to be visible in the image.

We solve this graph matching instance using a method based on RANSAC. The main difference between regular affine fitting and our problem is that, the preliminary correspondences in regular affine fitting are typically one-to-one, while in our problem, one GIS projection is preliminarily matched to *any* of the detections of the same class and vice versa; however, only one of these correspondences can be included in the final matching. To solve this problem using RANSAC, we randomly select three correspondences *which do not involve assigning one node to multiple nodes*, and find their corresponding affine model. Then the number of correspondences which fit this affine transformation are counted and interpreted as the score of the estimated model; however, if one node is involved in two inlier correspondences, only one of them is be counted. This process is repeated until a termination condition (convergence or maximum number of iterations) is met.

Our graph matching formulation is in fact a higher-order graph matching problem and not a linear assignment problem, such as bipartite matching. That is because our objective function, i.e., affine error, can be approximated using at least three correspondences and not one (unlike linear assignment problems which assumes a cost/weight is assigned to each correspondence). As an alternative to RANSAC for solving our graph matching problem, the recent tensor-based hypergraph matching techniques [5,26] can sufficiently approximate the solution in a fraction of a second. However, they allow for local deformations in the affine model, while employing RANSAC would enforce a *global* affine model on the matching correspondences; a global model is favorable for our problem as we know that the reason behind the misalignment is mainly the imperfect camera parameters (which typically lead to global shifts) and not local deformations in the objects or scene elements.

The best assignment found by graph matching is typically represented using the assignment matrix χ. For the sake of simplicity, assume $\chi(i)$ determines the index of the detection which was matched to the i^{th} GIS projection.

We want to assign a score to each DPM detection in the image which survived graph matching. This is done using the following equation based on how well the corresponding GIS projection and DPM detection fit the found affine transformation:[4]

$$S_i^G = Sig\Big(\|\boldsymbol{A}\begin{bmatrix}O_i\\1\end{bmatrix} - \begin{bmatrix}D_{\chi(i)}\\1\end{bmatrix}\|\Big), \tag{3}$$

where $\boldsymbol{A}$ is the best found affine matrix, $\|.\|$ denotes Euclidean distance, and O_i represents the spatial location of the center of the i^{th} GIS projection. Similarly, D represents the spatial locations of the (center of) detections by DPM. Therefore, $D_{\chi(i)}$ is the spatial location of the DPM detection matched to the i^{th} GIS projection. sig is a sigmoid function of the form $sig(x) = 2\frac{1}{1+e^{-\sigma x}}$ with $\sigma = -0.05$, which transforms the affine error to a score ranging from 0 to 1. Besides the geometry of the arrangement of objects, we wish to incorporate the content-based confidence of the detection returned by DPM. This is done using a linear mixture of the score based on the geometry and the one based on the content (represented by S^I) which is the actual score returned by DPM normalized to range from 0 to 1:

$$S_i = \alpha S_i^G + (1-\alpha){S^I}_{\chi(i)}, \tag{4}$$

where α is a linear mixture constant. Therefore, S_i denotes an updated score for object i based on the image content and its geometric consistency with the GIS. In the experiments section, we will see that the precision-recall curves of the object detection obtained using this method significantly outperforms DPM and the other baselines.

3 Geo-localization Using Generic Objects

As discussed earlier, the objects which are visible in the image as well as their geometric relationship form a discriminative cue for finding the location that the image was taken at. We propose a method for finding the geo-location of an image based on the semantic objects therein and their relative geometry. However, two issues make this process challenging: First, as discussed before, the object detections found in the image are far from perfect. Second, the GIS projections are often off the right spatial location. Our method, which is in fact based on the object detection process described in section 2, is capable of handling these issues.

[4] This could be also applied to all DPM detections, and not only the ones which are selected by graph matching, by allowing multiple DPM detections to match to one GIS projection after finding the best affine model using graph matching, or assuming the geometric score of zero, $S^G = 0$, for the detections which were not selected by graph matching.

To find the location of the query image, we perform a search on a dense grid of geo-locations ($20m$ apart) for the covered area. On each location, we search over different compass orientations as well (20° apart). This dense grid is shown for a sample covered area in figure 3 (b). We perform the object detection method described in section 2 on the query image using each of these feasible location-orientation pairs (i.e., assuming these are the correct location and orientation of the camera) and obtain the following score for the location-orientation pair:

$$L = \beta \sum_{i=1}^{|\boldsymbol{O}|} S(i) + (1-\beta) \sum^{c} \frac{\min(|\,\boldsymbol{O}^c\,|, |\,\boldsymbol{D}^c\,|)}{\max(|\,\boldsymbol{O}^c\,|, |\,\boldsymbol{D}^c\,|)}, \tag{5}$$

where $|.|$ denotes the size of a set. The parameter β is the the linear mixture constant, and c represent the object classes $c \in \{\text{Traffic Signal, Trash Can,...}\}$ (e.g., $\boldsymbol{D}^{\text{Traffic Signal}}$ and $\boldsymbol{O}^{\text{Traffic Signal}}$ are the DPM detection and GIS projections of Traffic Signals for a particular location-orientation). The first term captures how well the matching subset of DPM detections fits the GIS data of that particular location (including the spatial geometry of the objects). The second term has an intersection-over-union form and quantifies the difference between the presence and absence of the objects in the image compared to the GIS database (no geometry). If for each GIS projection an inlier detection was found by the graph matching, the left term yields a high score. On the contrary, if most of the GIS objects were found to be matching to unreliable detections, this term will give a lower score. The linear mixture of these two terms is the overall geo-localization score for a particular location and orientation. We perform this search for all the grid points with different orientations and rank different location-orientation pairs based on the score.

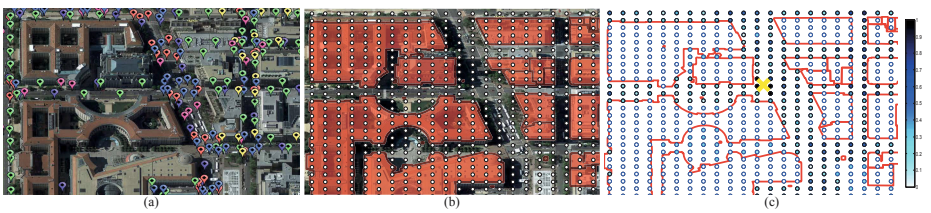

Fig. 3. The process of geo-localizing a query image based on the GIS objects in a sample region. Figure (a) shows the GIS objects in the region. Figure (b) illustrates the grid geo-locations along with the building outlines. Figure (c) shows the score of each grid geo-location obtained from equation 5. The ground truth location is marked with the yellow cross; notice that the grid points near the ground truth obtained a high score. The points which fall on buildings gain the score of zero.

This method is merely based on semantic objects and can significantly narrow down the search space for the location of the query image. We will show that the combination of the presence/absence of the objects and their geometric arrangement (i.e., the right and left terms of equation 5) is a distinctive cue for

finding the right geo-location. That means even though the objects are generic and can be found in many locations in the city, their geometric relationship contains distinctive cues about the location.

This process is illustrated in figure 3. The GIS objects and grid geo-locations are shown for a subregion covered by our GIS database. The score each location achieves is shown in figure 3 (c). As apparent in the figure, the ground truth location marked with the yellow cross obtains a higher score compared to the majority of other locations.

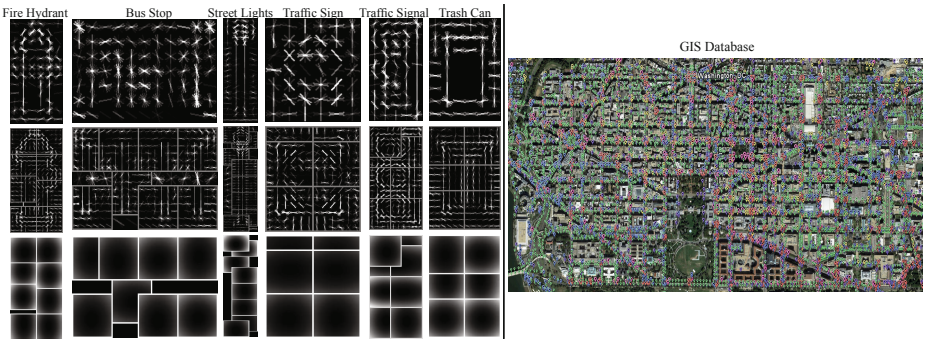

Fig. 4. Left: the learned DPM models for our six GIS objects. **Right**: part of the coverage of our dataset for Washington, DC. The GIS object are overlaid on the map.

4 Experiments

We performed our evaluations using a GIS database of over 10 square kilometers area of Washington DC. Figure 4 (right) shows part of the coverage of our dataset along with the locations of GIS objects. Our GIS database includes accurate locations of street lights, bus stops, traffic signals, traffic signs, trash cans, and fire hydrants as well as the building outlines (the color-coding of all markers and bounding boxes of the objects is specified in the footnote 2). We trained our DPM detectors (using the code of authors [7,6]) for each object, illustrated in figure 4 (left), by annotating more than 6000 urban area images downloaded from Panpramio and Flickr including 420 instances of bus stops, 921 street lights, 1056 traffic signals, 1264 traffic signs, 625 fire hydrants, and 646 trash cans.[5] Our test set includes 223 consumer images downloaded from Panoramio and Flickr. We set the linear mixure constants, β and α, to 0.5 in our experiments.

4.1 Object Detection Results

Figure 5 illustrates the results of different steps of our framework for the task of object detection. The first column shows the output of DPM on the images. It can be observed that there are a notable number of false positives and mis-detections

[5] For the GIS datab, training images, annotations, DPM models, and further information, please visit `http://crcv.ucf.edu/projects/GIS-Object/`

Fig. 5. Sample object detection results. For each sample, the detections found by DPM and our method, along with the GIS projections and the ground truth are shown. Our method significantly improved the results in these challenging urban-area images.

Table 1. Quantitative comparison of the proposed object detection method vs. the baselines

	DPM	GIS Proj.	Top DPM	Ours
Traffic Sign	0.087	0.002	0.095	**0.190**
Traffic Signal	0.543	0.027	0.561	**0.760**
Trash Can	0.043	0.000	0.041	**0.125**
Fire Hydrant	0.010	0.000	0.012	**0.090**
Street Light	0.123	0.001	0.129	**0.270**
mAP	0.1612	0.006	0.1676	**0.287**

due to the small size of the objects, poor lighting conditions and partial occlusions in the objects. The second column shows the projections extracted from the GIS data. As apparent in the figure, the locations of the projections on the image are not that accurate, while their relative geometry is consistent with the image content. The output of our method is illustrated in the third column. Comparing it with the last column (ground truth), the significant improvement of our method over the DPM and GIS projection results can be observed.

The precision-recall curves of the object detection can be seen in figure 6, in which the blue curve corresponds to the output of our method. The black curve shows the accuracy for the projections if we consider them as the detections. The red curve shows the performance of DPM, and the green curve illustrates the results of a naive fusion method in which the top k detections of the DPM model were maintained and the rest were eliminated from the detections (k is the

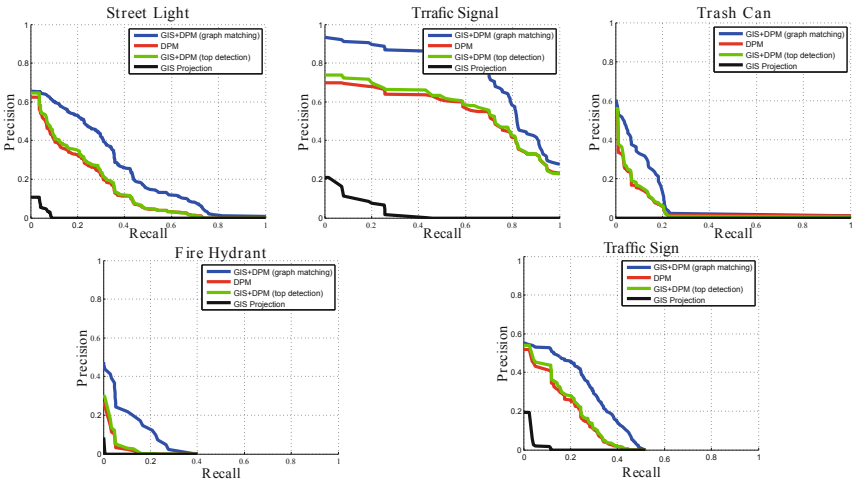

Fig. 6. The PR curves of our object detection method and the baselines. The DPM detector of bus stops yields very poor detection results because of the vast diversity and complexity of their appearance. Thus, we exclude it from the rest of our experiments.

number of projections of one class which are visible in the image). Table 1 provides the quantitative comparison of our method vs. DPM and GIS projections in terms of mAP which shows the significant improvement by our method.

Impact of the Number of Object Classes: since our method is leveraging the geometric relationship among different classes of objects, the more number of object classes we have, the higher overall accuracy we expect to obtain. Table 2 shows the effect of the number of object classes on the overall accuracy of our method. "n objects" means we ran our method using all feasible n-class combinations of object classes (i.e., $\binom{5}{n}$) and averaged their mAP. It can be observed that an increase in the number of object classes leads to notable improvement in the overall results.

Table 2. Effect of the number of object categories

	All Objects	Four Obj.	Three Obj.	Two Obj.	One Obj.
mAP	0.28	0.26	0.17	0.12	0.08

Significance of Each Class: the goal of this experiment is to evaluate the contribution of each object class in the overall accuracy. For this purpose, the object detection performance of our method was calculated by excluding one of the object classes from the process. Table 3 shows how the accuracy would be affected after neglecting each object. As an example, not incorporating GIS

information about the traffic signals makes the overall object detection accuracy drop significantly, thus traffic signals have a notable contribution in the overall accuracy. That is primarily because they are rarely occluded (similar to street lights and unlike trash cans and fire hydrants) and have a discriminative shape which can be detected easily. On the other hand, some objects, such as trash cans and fire hydrants, have a negative effect on the accuracy due to being frequently occluded and having a less distinctive appearance.

Table 3. The contribution of each object to the overall object detection results. Some of the more robust classes, e.g., Traffic signal, have a positive contribution, whereas the less reliable classes, e.g., trash can, have a negative contribution.

	All Objects	Street Light	Traffic Sign	Trash Can	Traffic Signal	Fire Hydrant
mAP	0.28	0.26	0.26	0.33	0.17	0.29

4.2 Geo-localization Results

For the task of geo-localization, a grid of > 25000 candidate points ($20m$ apart) was overlaid on the map. However, many of these points were placed on buildings, so only 4134 candidate points needed to be searched over. Each grid point was evaluated with 18 different orientations (20° apart), and the highest score among them was then assigned to that particular location.

Figure 8 shows four different localization examples. We use the GPS-tag of the query image in the Exif tag (after manual verification of the correctness) as the ground truth. The first column shows the query image and the objects present in it. The second column shows the objects detected by the DPM, and the last 5 columns show the top 5 location candidates for the image. The ground truth location, marked with the red boundary, obtains a high score due to the good matching between the content of the image and the GIS data of the location. The reason some arbitrary locations may get a high score is the poor object detection

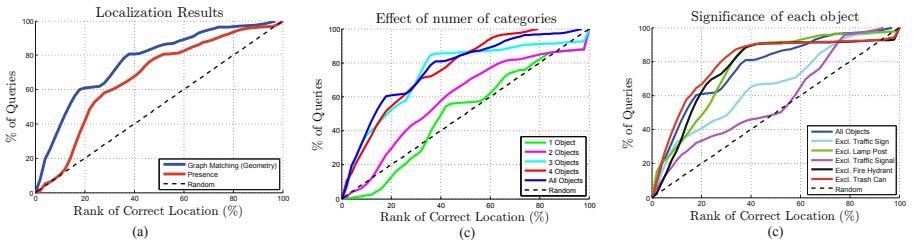

Fig. 7. Quantitative geo-localization results. Figure (a) shows the overall results (blue) along with the results obtained using only the presence/absence of objects (i.e., the left term in equation 5). Figure (b) illustrates the impact of the number of incorporated object classes. Figure (c) details the contribution of each object class.

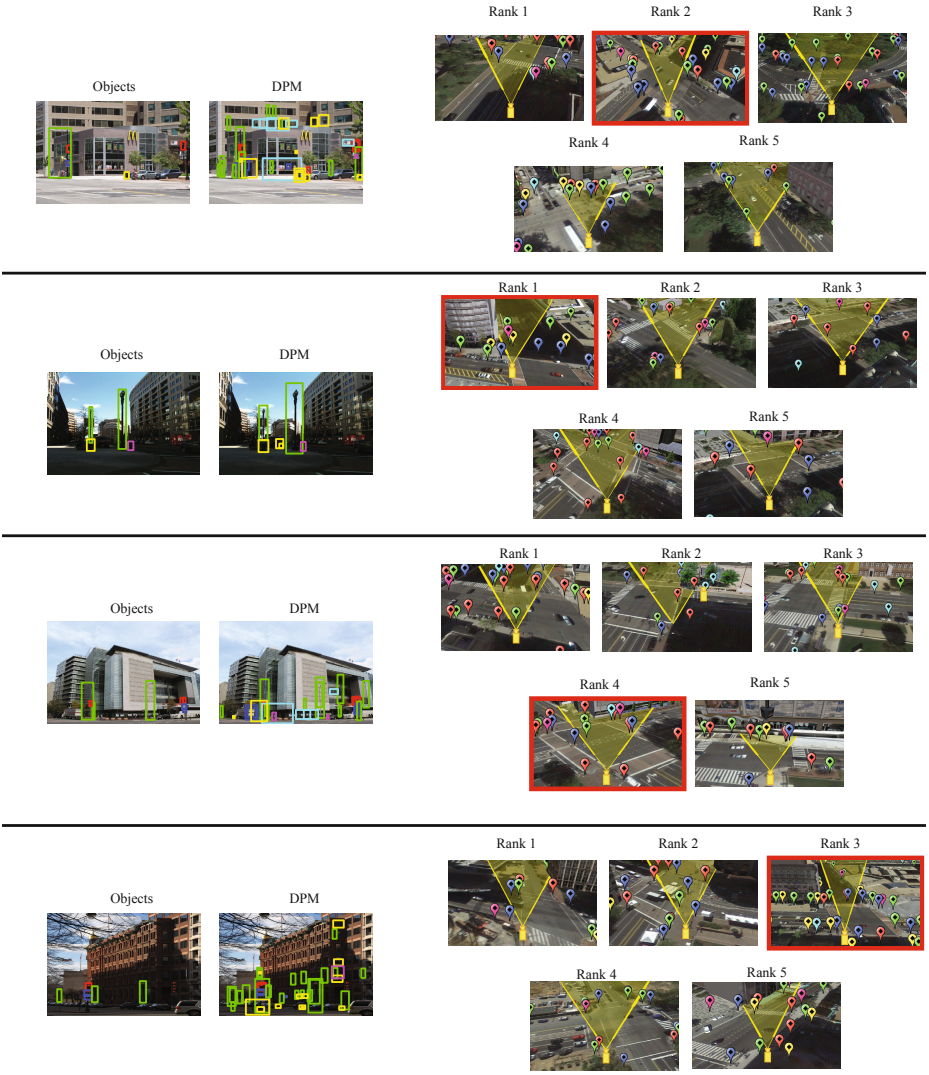

Fig. 8. Sample geo-localization results for four query images from Panoramio and Flickr. The images corresponding to the five best matching locations as well as the DPM detections and the ground truth objects are shown. The image corresponding to the correct location (i.e., the grid point within $20m$ of the ground truth GPS location.) is marked with the red boundary. As apparent in the figure, the correct location is among the best matching geo-locations as a results of the similarity between the objects visible in the image and the GIS database of the ground truth location.

results by DPM. That is because too many missed objects or false positives may form a random geometric arrangement in the image which matches the GIS objects of an arbitrary location.

Figure 7 (a) shows the quantitative localization accuracy of our method. We adopt the popular performance metric in image geo-localization literature [12,25] which employs a plot with the horizontal axis representing the rank of the ground truth location in the geo-localization results, and the vertical axis showing the percentage of test images with the ground truth location within a particular rank. The red curve in figure 7 (a) was computed by using only the information about the presence/absence of the objects in the image (i.e., the right term in equation 5). It can be observed that leveraging the geometric relationship leads to higher accuracy shown using the red curves.

Impact of Number of Object Classes: Similar to the experiments for object detection, we evaluated the localization results using fewer number of objects. Figure 7 (b) shows that, generally, utilizing more object classes leads to more accurate localization, as a results of incorporating more information.

Significance of Different Classes: similar to the experiment for object detection, we evaluated the importance of each object class in geo-localization. Figure 7 (c) shows localization results by excluding one of the object classes from the process; the amount of drop in the overall accuracy is indicative of the positive contribution of that particular class in the geo-localization results). Again, as shown in table 3, reliable objects, such as traffic signals, are confirmed to have a larger positive contribution in the localization results.

Failure Analysis: the root of the majority of the failure cases of our framework, for both object detection and geo-localization, is not being able to find the correct corresponding object bounding box among the DPM detections. In general, there are 3 main reasons for that: having no DPM detection for an object, misalignment by graph matching, and the inaccuracies of the GIS data (missing or misplaced objects). An experiment on a subset of our data showed the aforementioned reasons caused 25%, 41.6% and 33.4% of failures cases, respectively.

5 Conclusion

We proposed a method for improving object detection using a set of priors acquired from GIS databases. Given a database of object locations and a query image with metadata, we projected the GIS objects onto the image and fused them with candidate object detections acquired from DPM. In order to handle various inaccuracies and practical difficulties, we formulate our fusion as a higher-order graph matching problem which we robustly solved using RANSAC.

Furthermore, we proposed that the GIS objects can be used for discovering the GPS location from where an image was taken at. For this purpose, we performed

a search on a dense grid of locations over the covered area and assigned a score to each geo-location quantifying the similarity between its GIS information and the image content based on the objects visible therein. We showed that this intuitive and semantic approach can significantly narrow down the search space, and sometimes, even find the correct GPS location.

References

1. Bansal, M., Sawhney, H.S., Cheng, H., Daniilidis, K.: Geo-localization of street views with aerial image databases. In: Proceedings of the 19th ACM International Conference on Multimedia, pp. 1125–1128. ACM (2011)
2. Bioret, N., Moreau, G., Servieres, M.: Towards outdoor localization from gis data and 3D content extracted from videos. In: IEEE International Symposium on Industrial Electronics (ISIE), pp. 3613–3618. IEEE (2010)
3. Crandall, D., Backstrom, L., Huttenlocher, D., Kleinberg, J.: Mapping the world's photos. In: International World Wide Web Conference (2009)
4. Dasiopoulou, S., Mezaris, V., Kompatsiaris, I., Papastathis, V.K., Strintzis, M.: Knowledge-assisted semantic video object detection. IEEE Transactions on Circuits and Systems for Video Technology 15(10), 1210–1224 (2005)
5. Duchenne, O., Bach, F., Kweon, I.S., Ponce, J.: A tensor-based algorithm for high-order graph matching. Pattern Analysis and Machine Intelligence (PAMI) 33(12), 2383–2395 (2011)
6. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. Pattern Analysis and Machine Intelligence (PAMI) 32(9), 1627–1645 (2010)
7. Girshick, R.B., Felzenszwalb, P.F., McAllester, D.: Discriminatively trained deformable part models, release 5, `http://people.cs.uchicago.edu/~rbg/latent-release5/`
8. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. In: International Conference on Computer Vision (ICCV) (2008)
9. Knopp, J., Sivic, J., Pajdla, T.: Avoiding confusing features in place recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 748–761. Springer, Heidelberg (2010)
10. Lee, Y.J., Efros, A.A., Hebert, M.: Style-aware mid-level representation for discovering visual connections in space and time. In: International Conference on Computer Vision (ICCV) (2013)
11. Li, Y., Snavely, N., Huttenlocher, D., Fua, P.: Worldwide pose estimation using 3D point clouds. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 15–29. Springer, Heidelberg (2012)
12. Lin, T.Y., Belongie, S., Hays, J.: Cross-view image geolocalization. In: Computer Vision and Pattern Recognition (CVPR) (2013)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. In: International Journal of Computer Vision (IJCV) (2004)
14. Matzen, K., Snavely, N.: Nyc3dcars: A dataset of 3D vehicles in geographic context. In: International Conference on Computer Vision (ICCV) (2013)
15. Park, M., Chen, Y., Shafique, K.: Tag configuration matcher for geo-tagging. In: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 374–377. ACM (2013)

16. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2d-to-3D matching. In: International Conference on Computer Vision (ICCV) (2010)
17. Sattler, T., Leibe, B., Kobbelt, L.: Improving image-based localization by active correspondence search. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 752–765. Springer, Heidelberg (2012)
18. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: Computer Vision and Pattern Recognition (CVPR) (2007)
19. Torralba, A.: Contextual priming for object detection. International Journal of Computer Vision (IJCV) 53(2), 169–191 (2003)
20. Torralba, A., Murphy, K.P., Freeman, W.T., Rubin, M.A.: Context-based vision system for place and object recognition. In: International Conference on Computer Vision (ICCV), pp. 273–280. IEEE (2003)
21. Uchiyama, H., Saito, H., Servieres, M., Moreau, G., Ecole Centrale de Nantes - CERMA IRSTV: AR GIS on a physical map based on map image retrieval using llah tracking. In: Machine Vision and Application (MVA), pp. 382–385 (2009)
22. Wang, L., Neumann, U.: A robust approach for automatic registration of aerial images with untextured aerial lidar data. In: Computer Vision and Pattern Recognition (CVPR), pp. 2623–2630 (June 2009)
23. Zamir, A.R., Ardeshir, S., Shah, M.: GPS-Tag renement using random walks with an adaptive damping factor. In: Computer Vision and Pattern Recognition (CVPR) (2014)
24. Zamir, A.R., Shah, M.: Accurate image localization based on google maps street view. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 255–268. Springer, Heidelberg (2010)
25. Zamir, A.R., Shah, M.: Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI) (2014)
26. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. In: Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (June 2008)