

Overhead-based image and video geo-localization framework

Riad I. Hammoud, Scott A. Kuzdeba, Brian Berard, Victor Tom,
Richard Ivey, Renu Bostwick, Jason HandUber
Lori Vinciguerra, Nathan Shnidman, Byron Smiley

BAE Systems
Burlington, MA, USA
riad.hammoud@baesystems.com

Abstract

This paper presents a geo-localization framework of street-level outdoor images using multiple sources of overhead reference imagery including LIDAR, Digital Elevation Maps and Multi-Spectral Land Cover/Use imagery. We describe five different matchers and an adaptive linear fusion process which combines individual matchers' probability maps into a single map. These matchers exploit mountain elevation profiles, rendered camera views, landmarks, landuse classes and building heights. We successfully validated our framework on 100 queries with geographic truth in two world regions (each of 10,000km²) in the USA.

1. Introduction

In recent years, the problem of geo-localization of photos gained a lot of attention in the computer vision community [7, 2, 1, 9]. It is encountered in numerous applications including reliable geographic map augmentation with street-level photos and interactive image/video indexing and browsing [4]. Perhaps the most dominant approach is photo-to-photo matching where a street-level query image is matched against a set of geo-tagged photos and similar ones are retrieved and browsed for final user validation. In spite of the remaining challenges to be solved by such techniques, including robustness to changes in viewing angle, scale and illumination, these techniques are only applicable to highly photographed world regions [3] and hence not scalable to every region on Earth. In this paper, we describe an image/video geo-localization framework where matching is performed between the content of a street-level query image and corresponding information extracted from multiple sources of overhead reference imagery including LI-

DAR, Digital Elevation Maps (DEM), and Hyper-Spectral Land Use/Cover imagery. We tested our proposed framework in two different coast and desert regions in the United States using 100 street-level query photos. The problem is very challenging because we are trying to match two heterogeneous image sources: a street-level image to an overhead image. From the DEM reference data we will synthesize mountain skylines and camera ground-views, and in the LIDAR, we will extract meaningful data such as land-marks and building heights.

1.1. Related work

Recently, Baatz et al. [1] proposed an automated approach for large scale visual localization given a DEM of the searched place. Synthesized ground views are generated using a camera model and the DEM reference data. Their technique exploits visual information (curvelets) and geometric constraints (consistent orientation). They successfully validated their system on the whole Switzerland area. Bansal et al. [2] were able to match query street-level facades to airborne LIDAR imagery under challenging viewpoint and illumination variation by introducing a novel approach of selecting the intrinsic facade motif scale and modeling facade structure through self-similarity.

1.2. Paper organization

Section 2 provides an overview of the proposed system. In section 3 we present different sources of reference data. Then, in section 4, we describe and illustrate various modules and methods used to generate location hypotheses, henceforth known as “matchers.” The aggregation process of various matchers outputs is presented in section 5. In section 6, we describe the validation process of candidate regions by the end-user. The experimental setup, evaluation metrics, and results are summarized in section 7. Finally, we conclude with a brief discussion in section 8.

“Approved for public release; distribution is unlimited. Cleared for open publication on May 10, 2013.”

2. Proposed framework overview

The proposed framework consists of two stages. The offline stage consists of constructing the knowledge base where we index geo-referenced features relevant to the matching processes such as skylines and landmarks. These methods are detailed in the next sections. The online stage consists of segmenting and annotating a query image, matching extracted content to the reference knowledge base, generation of probability maps, fusion of various matchers outputs, and fused map thresholding to generate geographic candidates regions for the end-user to validate. Figure 1 illustrates the segmentation and annotation processes of three different query images.

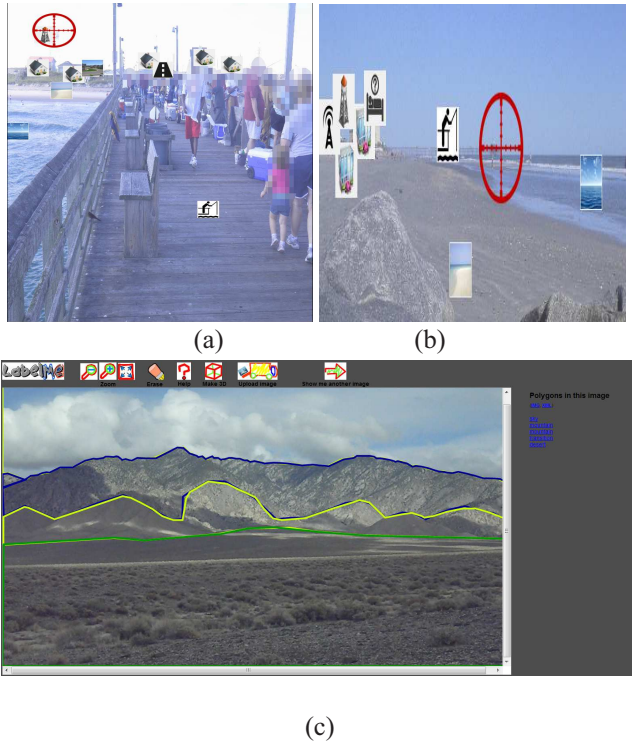


Figure 1: Example of three annotated query images. The user marked key landmarks (overlaid icons in (a) and (b)) such as piers, water towers, tall buildings and mountain elevation profile or skyline curve (c).

3. Knowledge base

The overhead reference imagery we utilized in this framework is from LIDAR, DEM, and Multi-spectral land cover imagery. This data covers various areas in the continental United States and the world, but our system tested two world regions within the United States. One region in the coast and one region in the desert each represents

a 10,000 square kilometer search space. The LIDAR data was particularly useful in the coastal world region where we extracted key landmarks such as piers, water towers, buildings, and other man-made structures. The geo-location of this extracted information is stored in the knowledge base (KB) for the coastal region. In the desert region we extracted mountain profiles from the DEM and stored them in the desert knowledge base. The DEM consists of ground positions sampled at regularly-spaced horizontal intervals. The underlying data consists of hypsographic data (contour lines) and/or photogrammetric methods using USGS topographic quadrangle maps. The DEM data used in this system was sampled at 30-meter resolution. The National Land-Cover Database (NLCD) provides 30-meter resolution land cover data comprised of 16 land classifications in the continental United States, with 4 additional land classes in Alaska [5]. The resulting data was created by unsupervised classification of multi-season Landsat 5 and Landsat 7 imagery, and Digital Elevation Model derivatives.

4. Matchers

In this section, we provide a high-level description of the five key matchers of the proposed geo-localization framework. Two of these matchers, the *Land use/cover* and *Landmarks* matchers, act like regions discounting matchers (RDMs) and generate binominal mask-like maps (in lat/lon coordinates). These matchers don't employ a similarity distance but rather rely on whether a query feature (landmark, land class) exists or not in the KB. In contrast, the other three matchers, *DEM-based skyline*, *Scene Configuration*, and *LIDAR-based Context* matchers, are candidate generating matchers (CGMs). These matchers assess a similarity measure between the geo-indexed KB features (elevation profiles, building types, etc.) and the searched query content to generate probability maps.

When output by a matcher, a probability map either represents object-location or camera-location estimates. A camera-location probability map is produced when the matcher relies on a virtual camera model and generates synthetic views. Most of the proposed matchers generate object-location probability maps because they match query content against geo-indexed objects such as water towers, piers, and mountain elevation profiles rather than geo-indexed synthesized views.

4.1. Land use/cover matcher

This matcher employs the National Land Cover Database information to eliminate large areas from the search regions. The purpose of this matcher is to remove all area that does not match the land class on which the camera is located. The source data is stored in raster form with approximately 30 meters per pixel. As input, this matcher asks the user to estimate the land cover class on which the camera is located.

By presenting examples of each land classification, the user has a very high rate of success in selecting the correct land type. In the event that the user is unable to determine the land classification under the camera, the system allows the user to select multiple possibilities. In this event, the Land Use Matcher will join all selected land classes together using a simple “OR” operator.

Additionally, in query images in which the land class is clearly identifiable, it is sometimes possible to discern that the camera is located at the intersection of two land classes. For example, if the camera is located on a beach within 50 meters of the water, the user may input both landclasses and specify that they are intersecting. In this event, we perform a simple dilation on each land class’s raster image and then perform an “AND” operation to calculate only those areas in which both land classes are adjacent.

The level of discrimination of these probability maps can vary greatly from query to query, based on the land class(es) seen in the image. Some land classes, such as scrub, represent a huge portion of our world regions - roughly 93% of our desert region is scrub. In this case, this matcher contributes very little to the overall result. Other land classes, however, can narrow the search space to 1% or less. Furthermore, since the user almost never selects the wrong region, this matcher is a very reliable region discounting matcher.

4.2. DEM-based skyline matcher

The skyline matcher is a curve-to-curve representation and matching scheme. Figure 2 summarizes the overall approach. It utilizes the Chamfer distance [6] to compute the similarity distance between the query skyline curve and the mountain profiles extracted from DEMs. This matcher is applied in the desert region only.

Figure 3 shows the skyline of query image before and after curve scaling and its corresponding top two mountain profile matches in the KB. A mountain-location probability map is generated from these matching scores and then converted to camera-location probability map using both orientation information stored in the knowledge base and the user’s best guess of mountain depth in the query image. In this mapping process, we used a *donut-like* convolution kernel to convert from object location (mountain peaks) to camera location. This map is further filtered out using a predefined threshold (see bottom left of Figure 3). This matcher is used as a coarse-matcher in our system.

4.3. DEM-based scene configuration matcher

This matcher is a method by which DEMs are used to generate synthetic versions of ground imagery and compare the rendered scene to the query image. Rather than using color information, we use terrain classes. Here we

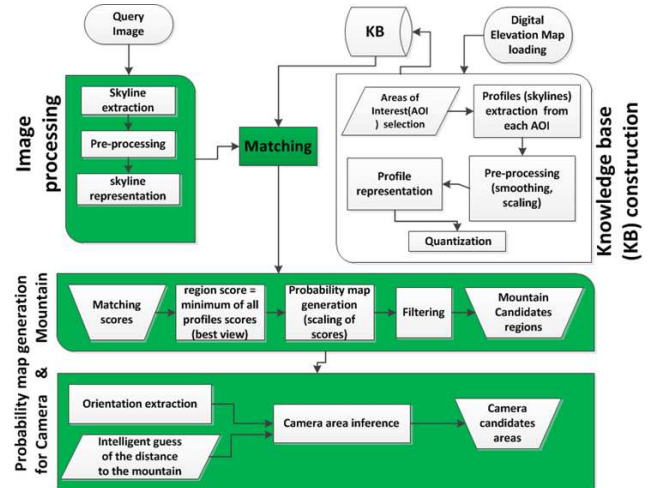


Figure 2: Skyline matcher diagram. The DEM-based Area-Of-Interests (AOIs) are generated using an elevation threshold. The skyline/DEM profile axis are normalized to [0,1].

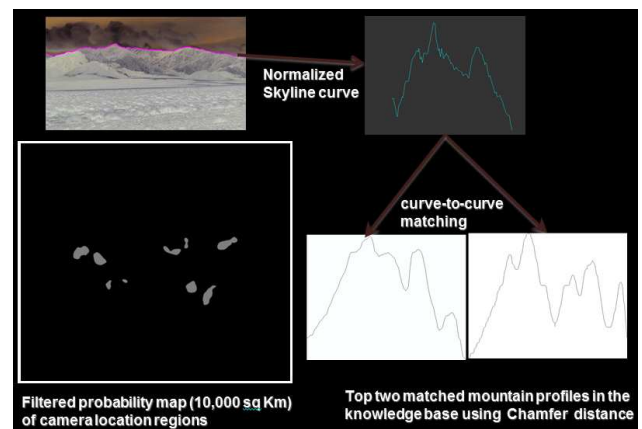


Figure 3: Illustration of the curve-to-curve matching using Chamfer distance in the skyline matcher.

show the use of flat (red), mountainous (green), and transition (olive) terrain, in addition to blue sky. Terrain classes obviate the need to handle subtle color shifts, illumination issues, and atmospheric attenuation. Synthetic scenes are rendered (Figures 4 and 5, right) and compared with the query image marked up by the user (Figures 4 and 5, center). High-scoring matches are retained and presented to the user as candidate matches. Because of an efficient OpenGL implementation of the scene rendering engine, the Scene Configuration Matcher (SCM) can generate and evaluate over 60 random scenes per second. This rate of search makes a brute-force search over the possible scenes feasible, although we implemented further improvements over the brute-force search through importance resampling,

whereby spatial locations near close candidate matches are resampled in order to further improve the best matches. This matcher is applied in the desert region only.

4.4. LIDAR-based context matcher

This matcher operates on the LIDAR data that is stored in the KB. The LIDAR data has been processed by BAE Systems Phase II URGENT algorithms [8] to extract information about the structures contained within the data - height, volume, perimeter, area. It uses this extracted data, along with user input, to create a probability map of estimated geo-locations for the query.

A user inputs two pieces of information for the Context Matcher. The first piece of information is the estimated structure height. Due to errors in estimating the height of a structure, the user is asked to label a building as falling into one of three bins according to the number of stories it has (1-2 stories, 3-8 stories, or 8+ stories). The second piece of information is for the user to estimate the distance between the structure and the camera. This distance estimate is also grouped into easy-to-use bins for the user.

This user information is then used to tap into the knowledge base and extract structures that fall within the bin(s) that the user input. Each structure location is then modified to find the camera location, as discussed in 5.1. The geo-location estimates for each structure type are then combined to only return areas that contain all the structure types selected. In the event that part of our search area is not represented by LIDAR data, we scale back this matcher in

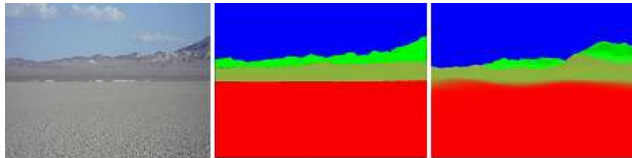


Figure 4: Example desert query image (left). User markup of query image using the landcover class color assignments, described in the text (center). Candidate match found by SCM (right). Although not a precise pixel-level match, the rendered scene is qualitatively similar to the query image.

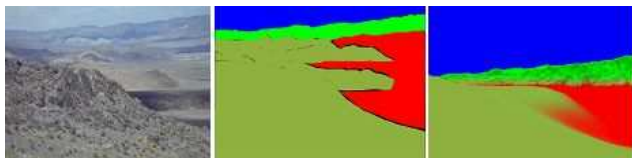


Figure 5: As with Figure 4, above. Note that the SCM matcher is robust to horizon shifts and does not require manual horizon estimation to generate valid candidate matches.

those areas such that those results do not contribute to the overall results when combined with other matchers.

Figure 6 illustrates the LIDAR reference data and the probability map generated by this matcher.

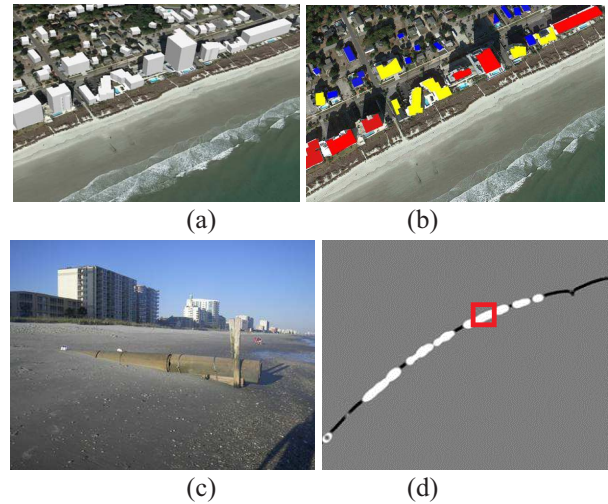


Figure 6: Context Matcher input and output illustration. (a) LIDAR structure data extracted with the use of the URGENT software [8]; (b) its corresponding data binned into the three structure categories, where blue represents 1 – 2 story buildings, yellow represents 3 – 8 stories, and red is for 8+ stories; (c) street-level query image with ground truth in the LIDAR tile (shown in (a)); and (d) Context Matcher probability map with LIDAR tile marked as a rectangular red box.

The Context Matcher discussed above is a fairly coarse matcher that returns larger areas for a trade-off of faster computational time and reduced complexity. To extract more information out of the structure heights, we also developed a Scene Geometry Matcher that operates on actual estimates of the structure heights as well as uses the relational information between different structures to geo-locate the query. This yields smaller predicted areas for the camera location, but takes longer to compute and requires more information.

4.5. LIDAR-based landmarks matcher

The Landmark Matcher employs named, geo-located objects of interest including *piers*, *bridges*, *water towers*, *communications towers*, *forts*, *beaches*, *hotels*, and *marinas*. We processed LIDAR data to automatically locate water towers and piers. The remainder of this data is automatically extracted from geo-referenced sources such as Wikipedia, Geonames, USGS, and OpenStreetMap.

“Not Export Controlled”

Using the user markup as described above, this matcher extracts the objects of interest from the query and their associated distances-from-camera estimates to convert each object’s geo-location to a camera estimate in the image space. The conversion from object location to camera location uses a large uncertainty to reduce the effects of human error in estimating the distance between the object and the camera location and also to account for the various possible zoom levels of the camera. As such, this matcher does not, and is not intended to, provide highly-localized results, but in scenes containing multiple objects of interest, it is able to significantly reduce the search space. The speed and efficiency of this matcher makes it a great candidate to generate region-reduction masks to be passed into other, more computationally intensive matchers.

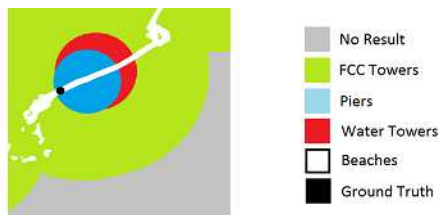


Figure 7: Illustration of a landmarks camera-location map for a query image

5. Adaptive fusion of multiple matchers

Here we describe the weighted linear aggregation method employed to fuse the probability maps produced by different proposed matchers. Prior to this process, every object-location probability map is converted to a camera-location-based probability map. The conversion mechanism and the fusion methodology are detailed in sub-sections 5.1 and 5.2.

5.1. Conversion to camera-location space

Some of the matchers described above estimate the locations of different objects, while others estimate the location of the camera. The final output desired by our system is an estimate of the camera location. Thus, the object estimation matchers need to be converted to camera estimates and we used various convolution kernels to achieve that (see Figure 8). Convolving the matcher probability map with a kernel transforms the probability map from an object estimation map to a camera estimation map. The kernel that is used to modify the map depends on (1) the estimated distance between the object and the camera and (2) an expected error region (or uncertainty). Figure 9 illustrates the result of using a donut kernel with a large error region on an object matcher’s probability map.

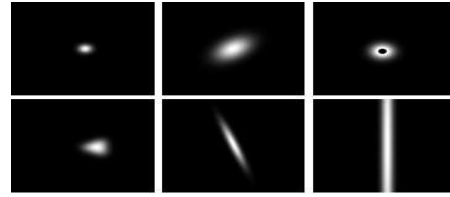


Figure 8: Illustration of various convolution kernels used in the conversion of probability maps from object location to camera location space. Going from left to right and top to bottom: Gaussian, Ellipse, Donut, Wedge, Narrow Ellipse and Ray kernels.

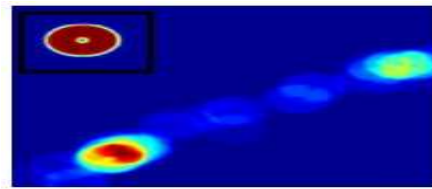


Figure 9: Illustration of a converted probability map (from object location to camera location) using the kernel shown in the top left corner.

5.2. Probability maps fusion

Now that all of the information from the different matchers has the same representation, we combine the output from the different matchers to get a final estimate. Fusing the output from the different matchers is done using two different methods.

The first method operates on region discounting matchers (RDMs). These matchers are “accurate” in the sense that their returned area rarely discounts the true location of the image. The total area returned, however, is sometimes fairly large. These matchers offer the ability to discriminate regions with a very high confidence, and, as such, we are able to treat them as masks. These masks are then combined with all the other matcher outputs in a multiplicative way. These matchers are helpful in cutting out large regions and only returning areas that usually contain the true location. An example of a matcher that is a region discounting matcher is the land use matcher, as discussed in 4.1, which matches the land class that the camera is over to the land classes in the knowledge base.

The second method applied to fuse the matcher outputs is a weighted linear combination of the non-region discounting matchers. This method, which operates on candidate generating matchers (CGMs), produces a range of probability values which allows us to generate and rank “candi-

date regions.” Matcher map outputs are scaled, based either off preset weights or by user input, to create new probability maps that are then summed with all other matchers of this type. Matchers that perform really well under a set of conditions are “stretched” out in their probability, i.e., their probabilities are increased so that they contribute more to the final output. Matchers that don’t perform well under a set of conditions are “flattened” in probability, i.e., their probabilities are decreased so that they don’t contribute much to the final output. Therefore, the final output produces highly probable areas where multiple high performing matchers have overlapping potential candidate regions (individual matcher high probable areas).

Putting both of these methods together results in the final output ($PMap$) as computed in Equation 1.

$$PMap = \left(\sum_{i=1}^m k_i CGM_i \right) * \prod_{j=1}^n RDM_j \quad (1)$$

Where the k_i ’s in Eq. 1 are the fusion weights that have been assigned to each of the CGMs, which are additive. The RDMs are multiplicative, as they take on the values zero or one and act as masks.

The fusion process is displayed pictorially in the following figures. Figure 10 shows an example of two region discounting matchers being combined (multiplied together). Note how these two matchers act as binary masks and cut out any region that is not included within both maps. Similarly, Figure 11 shows an example of two candidate generating matchers being combined (added together). Here you see that there is the addition of the weights, k_1 and k_2 , which are used to scale the maps amongst each other. Finally, in Figure 12, the final output is displayed. In this figure you see that the output of the region discounting matchers yield in regions being cut out of the candidate generating matchers to produce the final probability map.

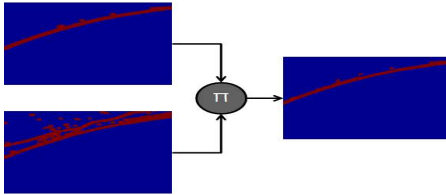


Figure 10: Fusion of Region Discounting Matchers

6. User feedback loop

Under the assumption that the true camera location can be found in the top candidate regions generated by the system, user feedback can be utilized to find the correct region using an overhead viewer, such as Google Earth or

“Not Export Controlled”

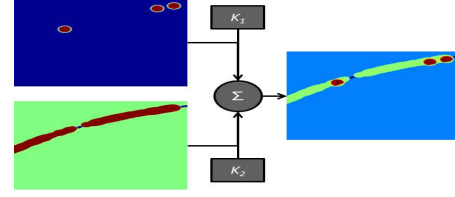


Figure 11: Fusion of Candidate Generating Matchers

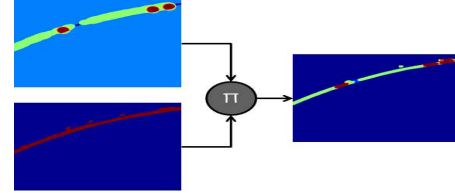


Figure 12: Fusion of RDMs and CGMs

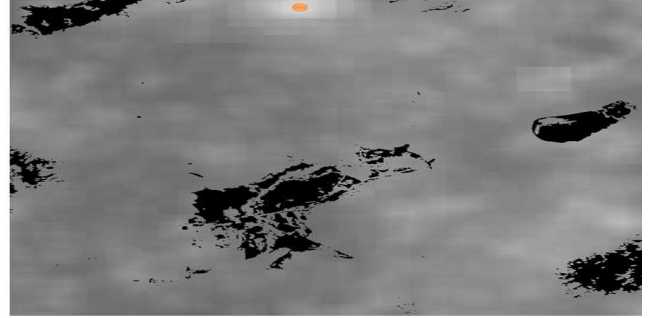


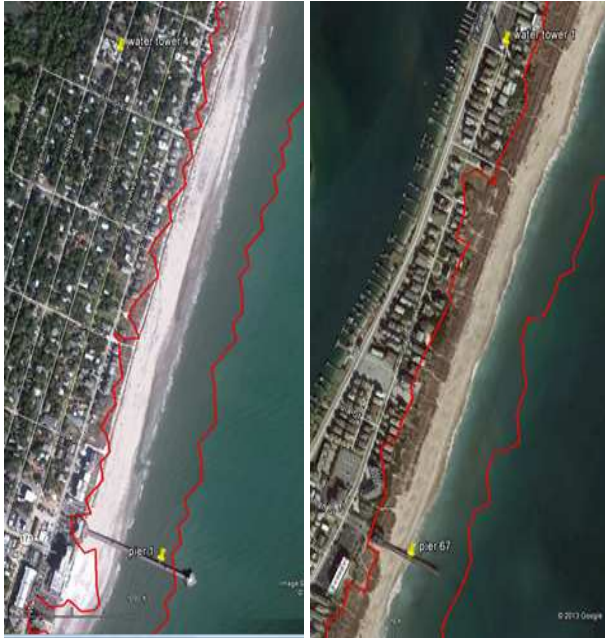
Figure 13: Illustration of the fused probability map for the desert query image in Figure 1 (c). Here we fused three probability maps generated by the skyline, land use and scene configuration matchers. The peak probability is marked with a pink dot near the top edge of the map.

Open Layers for many cases. The viewer enables the user to quickly step through the candidate regions, visually analyze the region content for relative placement of landclasses, shoreline, buildings, landmarks (highlighted by icons), in order to either reject or accept each region as a valid hypothesis. For example, for test query in Figure 14 (a), the top two candidate regions contain a shoreline, houses, and a water tower (Figure 14 (b) and (c)). Based on the content of this query image, the user imagines a camera field of view with houses on the left and water on the right and no piers in the field of view. The correct hypothesis was identified by the user as the right region (c) in Figure 14, in which the camera position is bounded in the south by the pier, the water tower is in the correct spatial placement relative to the houses and the shoreline, no pier in the field of view, and there is a discernible gap in the housing along the beach.

The rejected hypothesis (Figure 14 (b)) had the water tower too far inland, no housing gap, or the wrong type of housing (i.e., tall hotels) to be considered valid.



(a)



(b)

(c)

Figure 14: Illustration of the user validation process of generated candidate regions (second row) for a query image (first row). The user validates these candidate regions using Google Earth. Using the rationale described in the text, the candidate region (c) is kept while (b) is rejected.

This approach was also successful for the desert scenarios, in which landmarks such as buildings, roads, and barren salt flats were present. However, we determined that this aerial view approach for user feedback was not adequate for determining the validity of desert candidate regions when no such landmarks exist. For example, it was extremely difficult for the user to infer mountain ridge height from the

“Not Export Controlled”

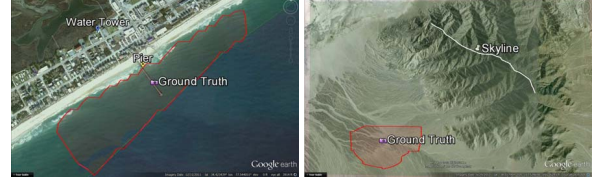


Figure 15: Illustration of the top candidate region validated by the user for queries shown in 1 (a) and (c), using Google Earth.

aerial color imagery. For those cases in which mountain skylines were the key factor in localizing candidate regions, we determined that the user needs to view terrain elevation (from above or even better a synthesized ground view) to validate a desert skyline query. In Figure 15, we illustrate the top candidate region positively validated by the user for a coastal and desert query image.

7. Experimental analysis

One hundred truthed query images (split evenly between desert and coast) were used to test matcher components and quantify system performance. These truthed images in two world regions (each of $10,000\text{km}^2$) and the metrics discussed in this section were used to measure performance improvements during the spiral development of this geo-localization framework.

7.1. Evaluation metrics

For system performance, we used the geographic area GA , in which one fixes the threshold a priori for thresholding the fused system overall probability map to generate candidate polygonal regions labeled and ordered from highest probability to lowest probability. Using this ranked candidate area list, we can define GA to be the sum of all candidate region areas up to and including the candidate region that includes the truth location.

We generated ROC curves by rank ordering GA values for the hundred query test images. The median rank value for the candidate region in which the truth resides is also an important metric to consider because the GA and the median rank give an indication of how easy or hard it will be to find the true camera location with user feedback on the automatically system generated candidate regions.

7.2. Experimental results

Using the baseline system described in this paper (input annotations, matchers, fusion, and user feedback) on 100 desert and coastal query images, we have achieved the following initial results.

	$GA(km^2)$		Top		Median	
	< 200	< 100	5	1	GA	CR
# queries	58	49	40	31	110.62	10^{th}

Forty nine queries returned less than $100km^2$ for the GA system metric. More importantly, for 40 queries, the true camera location was in the top five region candidates and the median candidate region rank was 10^{th} , i.e., for 50% of the queries the region ranking was 10^{th} or better. From our experience, whenever the true hypothesis is in the top 10 candidate regions, the likelihood of a user determining the correct solution is very high.

The various matchers in the proposed system vary broadly in the amount of time each takes to produce a set of probability map outputs for a single query image. The average run time per matcher and query is about 2 minutes on a standard laptop.

8. Discussion and conclusion

This paper presented a geo-localization framework of street-level outdoor images using multiple sources of overhead reference imagery including LIDAR, Digital Elevation Maps, and Multi-Spectral Land Use/Cover imagery. We described five different matchers and an adaptive linear fusion process which combines individual matchers probability maps into a single map. These matchers exploited mountain elevation profiles, rendered camera views, landmarks, land use/cover classes, and building heights.

The camera locating system described in this paper is part of an ongoing research effort that will continue to improve the content of its database: improving accuracy, simplify user interactions, refining matcher performance, and integrating new matchers into the system. Two new matchers that are being integrated include: 1) a scene geometry matcher that takes relative placement of buildings into account; and 2) a volumetric matcher that incorporates a full 3D volumetric database incorporating object information, map layers and overhead imagery. Ongoing efforts focus on refining our ability to reduce candidate regions by improving our underlying database (using higher resolution elevation data, land class and extracted map data), and utilizing a camera model in a fine search mode to localize camera position and look orientation using our most detailed matchers.

Acknowledgments

The authors would like to thank Diane Mills, Brad Rhodes, and Michael Seibert for their valuable support throughout this process.

“Supported by the Intelligence Advanced Research

“Not Export Controlled”

Projects Activity (IARPA) via Air Force Research Laboratory. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, Air Force Research Laboratory (AFRL), or the U.S. Government.”

“This material is also based upon work supported by United States Air Force under Contract FA8650-12-C-7211.”

“Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of United States Air Force.”

References

- [1] G. Baatz, O. Saurer, K. Koser, and M. Pollefeys. Large scale visual geo-localization of images in mountainous terrain. ECCV 2012. 1
- [2] M. Bansal, K. Daniilidis, and H. Sawhney. Ultra-wide baseline facade matching for geo-localization. Computer Vision ECCV 2012. Workshops and Demonstrations Lecture Notes in Computer Science Volume 7583, 2012, pp 175-186. 1
- [3] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes paris look like paris? ACM Transactions on Graphics (SIGGRAPH), 2012, vol. 31, No. 3. 1
- [4] R. I. Hammoud. Interactive video: algorithms and technologies. Springer book, ISBN: 978-3-540-33214-5, Signals and Communication Technology, 2006. 1
- [5] C. Homer, J. Dewitz, J. Fry, M. Coan, N. Hossain, C. Larson, N. Herold, A. McKerrow, J. VanDriel, , and J. Wickham. Completion of the 2001 national land cover database for the conterminous united states, 2007. Photogrammetric Eng. and Remote Sensing, Vol. 73, No. 4, pp 337-341. 2
- [6] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. TR2010-045 July 2010, <http://www.merl.com/papers/docs/TR2010-045.pdf>. 3
- [7] S. W. Pritt. Geolocation of photographs by means of horizon matching with digital elevation models. Geoscience and Remote Sensing Symposium (IEEE IGARSS), 2012. 1
- [8] E. Sobel, L. Vinciguerra, M. Rinehart, and J. Dankert. Urgent phase ii final report. BAE Systems April 30, 2012, Sponsored by DARPA, IPTO, 675 North Randolph Street Arlington, VA 22203-2114 Program: URGENT ARPA Order No. Y475/01, Program Code: P9X20 Issued by DARPA/CMO under Contract No. HR0011-09-C-0101. 4
- [9] R. Talluri and J. Aggarwal. Position estimation for an autonomous mobile robot in an outdoor environment. Robotics and Automation, IEEE Transactions on, 8(5):573584, 1992. 1