CAMERA GEO-CALIBRATION USING AN MCMC APPROACH

Menghua Zhai, Scott Workman, Nathan Jacobs

University of Kentucky

{ted, scott, jacobs}@cs.uky.edu

ABSTRACT

We address the problem of single-image geo-calibration, in which an estimate of the geographic location, viewing direction and field of view is sought for the camera that captured an image. The dominant approach to this problem is to match features of the query image, using color and texture, against a reference database of nearby ground imagery. However, this fails when such imagery is not available. We propose to overcome this limitation by matching against a geographic database that contains the locations of known objects, such as houses, roads and bodies of water. Since we are unable to find one-to-one correspondences between image locations and objects in our database, we model the problem probabilistically based on the geometric configuration of multiple such weak correspondences. We propose a Markov Chain Monte Carlo (MCMC) sampling approach to approximate the underlying probability distribution over the full geo-calibration of the camera.

Index Terms— geo-calibration, Monte Carlo

1. INTRODUCTION

Automatic image localization continues to grow in importance as a direct result of the increasing amount of imagery available via the Internet. Conceptually the task is straightforward; given an image, identify the location it was captured in the world directly from image data. Solving this problem is of great value for a wide variety of fields, with potential applications ranging from the forensic sciences [1] to crowd-sourced environmental monitoring [2].

However, recognizing the geo-location and geo-orientation of an arbitrary outdoor image is an extremely challenging task. Many methods have been proposed; the most common approach is to build a large database of images with known location and localize a query image using either local [3, 4] or global [5, 6] image features. This approach is not applicable when no nearby ground-level imagery exists in the reference database, such as when the image was not captured near a popular tourist destination. Even when reference imagery is available, the appearance of the objects may not be visually distinctive, for example a train track or a body of water.

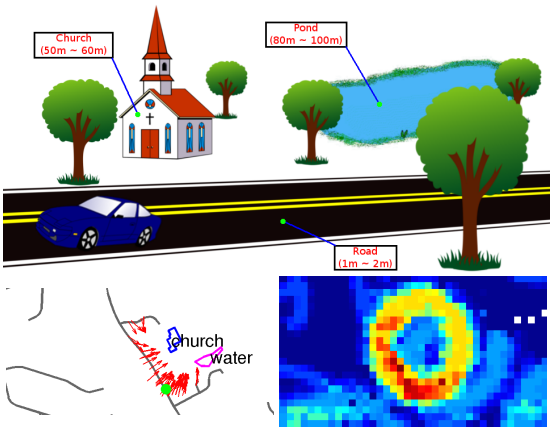


Fig. 1: We match objects in an image (top) to GIS reference data (left) to estimate a probability distribution over locations (right). The map (left) also shows the true camera location (green dot) and the top scoring sampled cameras (red arrows).

Instead of matching visually against a reference image set, we exploit the large quantities of publicly available geospatial data to build a geographic database containing geometric information for objects of interest in the world, such as roads, churches, bodies of water, water towers and golf courses. Given a query image, we identify visible objects of interest in the image and apply the Metropolis-Hastings MCMC algorithm to randomly sample possible cameras. We assign a score to each hypothetical camera and use these samples to approximate the probability distribution over the camera parameters and extract candidate locations. Fig. 1 gives an brief view of our approach.

Our key contributions are: 1) a flexible approach to the camera geo-calibration problem that supports priors over camera parameters and constraints that relate image annotations, camera geometry, and a geographic database and 2) an extensive comparison of this approach to uniform and grid-based sampling on real-world data.

1.1. Related Work

Self-localization has been heavily studied in the robotics community. The task is to estimate the probability density

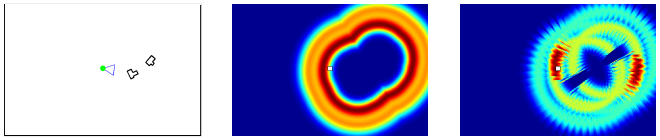


Fig. 2: (left) A map showing two houses (black) and a hypothetical ground-truth camera frustum. (middle) The PDF (red is high probability) over location for a 2D camera model, in which constraints are based on distances between the camera and GIS objects. (right) The 4D model we propose enables richer constraints, which leads to a more accurate PDF.

function (PDF) over the robot's state space (location and orientation). Early methods attempted to estimate this density by discretizing the state space [7]. These grid-based approaches suffered from large computational overhead and memory requirements. To overcome these limitations, probabilistic particle-based methods like MCMC were investigated [8, 9, 10, 11]. The idea is to approximate the probability density using randomly drawn samples, while maintaining performance even for high dimensional state spaces.

Extracting location-dependent features from image data has drawn a great detail of attention from the vision community [12, 13, 14]. The common trend amongst these methods is that they take advantage of a large dataset of geo-referenced images. Hays and Efros [5] use a data-driven scene matching approach to localize a query image using a large dataset of geo-tagged images. Doersch et al. extract location-dependent features that capture the relative appearance differences of large cities. Lin et al. [15] localize a ground-level image by learning the relationship between pairs of ground and aerial images of the same location. Other techniques focus on urban environments and infer location using local image descriptors [4, 16]. Li et al. [17] exploit geo-registered 3D points clouds to estimate camera pose. Many other cues exist, such as the skyline [18, 19], sky appearance [20, 21], and shadows [22, 23].

Our work attempts to combine these two research directions. We use publicly available geospatial data to build a large geographic database containing geometric information for objects in the world, identify objects of interest in the query image, and use a probabilistic approach to estimate camera geo-calibration. To our knowledge, our approach is the first to apply a probabilistic particle-based MCMC algorithm for single image camera geo-calibration using a large geographic database.

2. APPROACH

Given a query image, captured at an unknown location in a known region of interest (ROI) and annotated with the location of geographic objects (e.g., buildings and roads), our goal is to estimate the intrinsic and extrinsic camera parameters. We base these estimates on a geographic information system (GIS) database that contains the location and extent of objects in the ROI. Assuming a simplified pinhole camera model with square pixels and zero skew, the full camera geo-calibration problem is seven-dimensional: three position parameters, three orientation parameters, and the field of view. For this work, we assume that most photos are taken about five feet above the ground with little tilt or roll and reduce our model to four dimensions: $\Theta = (Latitude, Longitude, Azimuth, FOV)^{\mathsf{T}}$. Adapting to higher or lower dimensional camera models is straightforward and may be useful depending on the available image annotations and GIS data. We find our proposed 4D state space to be good trade-off between the lack of descriptive power of lower-order camera models and higher-order camera models that require more computational time and memory resources. See Fig. 2 for an example that compares our proposed model to a 2D model using only location.

Since one-to-one matching between image objects and GIS objects is likely not possible, we instead seek to estimate, $f(\Theta; \mathbf{C})$, the probability distribution function (PDF) over the camera parameters, Θ, given a set of constraints, $\mathbf{C}$. In the following section, we propose a function (an unnormalized density) that encodes constraints on the geo-calibration. This score function, $S(\Theta; \mathbf{C})$, encodes both prior knowledge about the camera and the geometric relationship between image annotations and the geographic database. In Sec. 2.2 we show how to estimate $f(\Theta; \mathbf{C})$ by sampling from this scoring function using an MCMC-based strategy.

2.1. Scoring Function

Proportional to the probability distribution of the camera parameters, $f(\Theta; \mathbf{C})$, the scoring function, $S(\Theta; \mathbf{C})$, is defined as a linear combination of multiple constraint functions:

$$S(\Theta; \mathbf{C}) = \sum w_i g(c_i, \Theta) \propto f(\Theta; \mathbf{C}), \qquad (1)$$

where w_i is a weight and $g(c_i, \Theta)$ is a constraint function which is larger if the calibration, Θ, is more consistent with the image information, c_i. While the exact set of constraints depends on the image and database contents, we define a variety of functions in our implementation, including geometric configuration of multiple weak correspondences, constraints on geographic location, and priors over the field of view and camera orientation. We focus on the first constraint, the latter are simple Gaussian and uniform distributions.

Given the extent of an object in the image, and an estimated range of distances, (d_{min}, d_{max}), from the camera to the object, we propose a constraint function that measures the geometric consistency between the object in the image and the GIS database. Given a hypothetical camera, Θ, we compute the distance, d, from the camera, through the object pixel, to the closest point of the object (see Fig. 3), then the consistency function, $g(c_i, \Theta)$, takes form of a Gaussian distribution

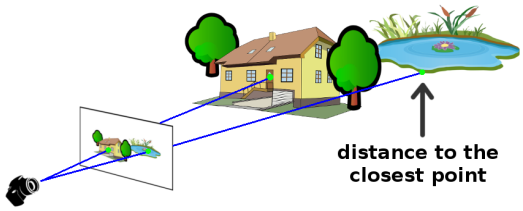


Fig. 3: Since the exact correspondence between an image pixel and a point on a GIS object is unknown, we compute the distance to the closest point of the object to compare with the estimated distance.

function, with mean $\mu = d - (d_{min} + d_{max})/2$, and standard deviation $\sigma = d_{max} - d_{min} + o$, where o is a constant offset ($o = 10m$ for all our experiments) to avoid dividing by zero. If no object of the correct type intersects the pixel ray, let $g(c_i, \Theta) = 0$.

2.2. Monte Carlo Markov Chain

The MCMC method is an efficient approach for sampling from high dimensional spaces. Its principal advantage over naive sampling strategies is that it visits high probability areas more frequently than the areas of low probability. We use the Metropolis-Hastings (MH) algorithm [24], a popular member of the MCMC family. The algorithm generates a set of possible cameras as follows: randomly sample a camera and compute its score; randomly sample (propose) a new nearby camera and compute its score; if the score of the new camera is higher, replace the old camera with the new camera, otherwise replace the old probabilistically; repeat this process many times, recording samples along the way. After a sufficiently large number of iterations, this process generates a set of possible cameras that are independent samples from the PDF, $f(\Theta; \mathbf{C})$, subject to some technical conditions. See Alg. 1 for details of the MH algorithm. We run multiple chains to mitigate issues with poor initial samples and local maxima.

Aside from the scoring function, $S(\Theta; \mathbf{C})$, the proposal distribution, $p(\Theta)$, which specifies how new cameras are sampled from a current camera (Alg. 1.4), is the most important choice when using an MCMC approach. We use the joint distribution of independent Gaussian random variables:

$$p(\Theta_{i+1}|\Theta_i) = \prod_{j=1}^{N_{dim}} \frac{1}{\sigma_j} \phi\left(\frac{\Theta_{i+1,j} - \Theta_{i,j}}{\sigma_j}\right),$$

where σ_j denotes the sampling step size on the j-th dimension, and $\phi(x)$ denotes the PDF of the standard normal distribution. As with all MH-based algorithms, the sampling performance is sensitive to the choice of step size. If the step size is too large, the algorithm converges slowly; if too small, the Markov chain may be trapped in a local maximum.

Require: $\mathbf{C}$ (constraint set), and $maxIter$

1: Initialize camera parameters, Θ_0
2: $i \leftarrow 0$, $\mathbb{S} \leftarrow \emptyset$, $s_0 \leftarrow S(\Theta_0; \mathbf{C})$
3: **while** $i < maxIter$ **do**
4: sample new camera parameters: Θ_{i+1}
5: scoring: $s_{i+1} \leftarrow S(\Theta_{i+1}; \mathbf{C})$
6: $\mathbb{S} \leftarrow \mathbb{S} \cup \langle \Theta_{i+1}, s_{i+1} \rangle$
7: **if** $s_{i+1} < s_i$ **then**
8: $s_{i+1} \leftarrow s_i$, $\Theta_{i+1} \leftarrow \Theta_i$ with prob. $\frac{s_i - s_{i+1}}{s_i}$
9: **end if**
10: $i \leftarrow i + 1$
11: **end while**
12: **return** $\mathbb{S}$

Algorithm 1: Metropolis-Hastings Algorithm (MCMC)

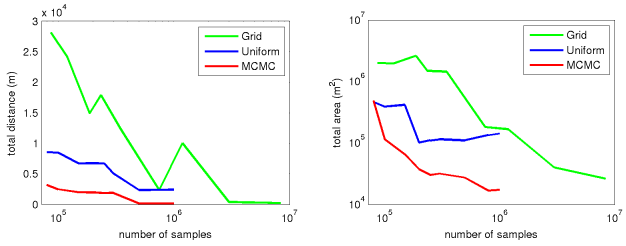


Fig. 4: Performance comparison among methods. (left) Total distance. (right) Total area. For the grid method the finest sampling over spatial dimensions is 30m.

3. EVALUATION

We compare our approach with two baseline methods, grid sampling and uniform random sampling, on real-world GIS data. The quantitative and qualitative results demonstrate the value of the proposed MH-based approach.

3.1. Methods

Dataset: We build a reference geographic database from OpenStreetMap[1] data, which contains the location and extent of many types of objects around the world. We only include roads, water, churches, residential buildings, and commercial buildings. For each query, we focus on a different $5km \times 5km$ ROI, in Kentucky, USA, each containing approximately 900 objects.

Metrics: We propose two evaluation metrics that simulate the process of manually verifying the camera location. For both, we generate a candidate list by sorting samples by their scores, then scan from the top until the ground-truth location is found. Given a set of samples from the state space and their corresponding scores, we greedily select the top-N candidates in terms of score, subject to the constraint that each of them is at least 200 meters away from the others spa-

[1] http://www.openstreetmap.org

tially. Then, if the k-th candidate is the first candidate that is d ($d < 100$) meters away from the ground truth, we define its *total distance* to the ground truth as follows: $T_d = 200(k-1) + d$. Intuitively, this metric enforces the diversity of candidate locations. The first term penalizes the ranking of the ground-truth candidate. The distance, d, distinguishes the accuracy of predictions in the case that two ground-truth candidate locations have the same ranking. By definition, a lower total distance to find the ground-truth candidate is better. The definition of the *total area*, T_a, is similar: we center a $l \times l$ patch around each candidate and compute the union of the areas from the top candidate patch to the patch that covers the ground-truth location. We set l to be $50m$ initially, if no patch covers the ground truth, we double the size of l and redo the computation until the ground-truth location is covered.

Implementation Details: We use the following fixed set of parameters, which we selected empirically, for all experiments. The ranges from which azimuth and FOV are sampled are $[0, 2\pi]$ and $[\pi/3, 2\pi/3]$ respectively. For the grid method, 50 angles are evenly sampled for azimuth, and 6 for the field of view. Thus at each geographic location, a total of 300 scoring samples are generated. For MCMC, 200 chains are independently processed in parallel. The step size is 100m for location, $\pi/20$ for azimuth, and $\pi/30$ for FOV.

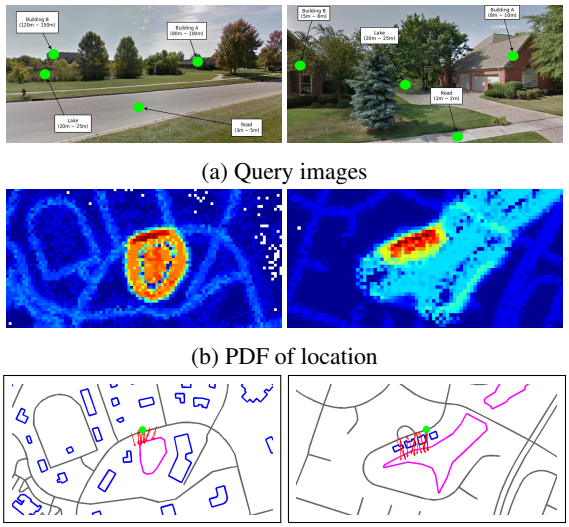
3.2. Evaluation on Synthetic Data

We manually constructed twenty-five synthetic queries using our geographic database. We hand-picked a location for each camera, such that there exists nearby objects in the database and adjusted the camera azimuth and field of view such that objects were visible in the view frustum. We then projected the objects onto the image frame to obtain labeled pixels, and for each provided a min/max distance from the camera to them (consistent with the actual distance). To simulate the estimation error in real life, we set the distance range to be $1/20$ the length of the actual distance.

For each query, we confined the search area to a $5km \times 5km$ neighborhood that includes the ground-truth location. We computed the accuracy in terms of the total distance, T_d, and the total area, T_a, and the computation time in terms of the number of samples. The results of this experiment are shown in Fig. 4. MCMC outperforms the other two baseline methods in both accuracy and speed. On average, our method only requires $1/10$ of computational time to converge to the same order of accuracy than the grid method.

3.3. Evaluation on Real Data

We evaluate our method using two real query images obtained from Google Street View. We hand-picked two locations, downloaded the corresponding equirectangular panoramas and extracted a perspective image from each. For each query, we labeled objects and estimated the min/max distance from the camera to the object in the world. Fig. 5 shows the qualitative result of this experiment. Our method generates high scoring samples that are close to the ground truth. We also found that simple prior constraints can dramatically affect localization accuracy. By restricting the range of the azimuth and FOV to be within 5° of the ground truth, the distance from the top sample to the ground-truth location decreased from $12.5m$ to $0.6m$ and from $1.73m$ to $0.9m$, respectively, for two test cases.



(a) Query images

(b) PDF of location

(c) Local geographic database and top-15 samples

Fig. 5: Qualitative results for two query images (top). The resulting PDFs (middle) and map visualization of the top-15 samples (bottom) show that the proposed scoring function realistically captures the uncertainty.

4. CONCLUSIONS

We proposed an MCMC-based approach framework for single image camera geo-calibration which leverages a large geographic database. Our results demonstrate the superiority of our method versus several baseline methods, without requiring nearby ground-level imagery as is typical for most vision techniques. For future work, we will explore methods for integrating approaches which use image-based matching.

Acknowledgements: Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory, contract FA8650-12-C-7212. 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, AFRL, or the U.S. Government.

5. REFERENCES

[1] Abby Stylianou, Austin Abrams, and Robert Pless, "Finding jane doe: A forensic application of 2d image calibration," *Imaging for Crime Detection and Prevention*, 2013. 1

[2] Haipeng Zhang, Mohammed Korayem, David J Crandall, and Gretchen LeBuhn, "Mining photo-sharing websites to study ecological phenomena," in *International World Wide Web Conference*, 2012. 1

[3] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher, "Location recognition using prioritized feature matching," in *European Conference on Computer Vision*, 2010. 1

[4] Grant Schindler, Panchapagesan Krishnamurthy, Roberto Lublinerman, Yanxi Liu, and Frank Dellaert, "Detecting and matching repeated patterns for automatic geo-tagging in urban environments," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 1, 2

[5] James Hays and Alexei A Efros, "Im2gps: estimating geographic information from a single image," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 1, 2

[6] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A. Efros, "What makes paris look like paris?," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 31, no. 4, 2012. 1

[7] Wolfram Burgard, Dieter Fox, Daniel Hennig, and Timo Schmidt, "Estimating the absolute position of a mobile robot using position probability grids," in *National Conference on Artificial Intelligence*, 1996. 2

[8] Frank Dellaert, Wolfram Burgard, Dieter Fox, and Sebastian Thrun, "Using the condensation algorithm for robust, vision-based mobile robot localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1999. 2

[9] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," in *AAAI Conference on Artificial Intelligence*, 1999. 2

[10] J-S Gutmann, Wolfram Burgard, Dieter Fox, and Kurt Konolige, "An experimental comparison of localization methods," in *International Conference on Intelligent Robots and Systems*, 1998. 2

[11] Sang Min Oh, Sarah Tariq, Bruce N Walker, and Frank Dellaert, "Map-based priors for localization," in *International Conference on Intelligent Robots and Systems*, 2004. 2

[12] Nathan Jacobs, Scott Satkin, Nathaniel Roman, Richard Speyer, and Robert Pless, "Geolocating static cameras," in *IEEE International Conference on Computer Vision*, 2007. 2

[13] Nathan Jacobs, Kylia Miskell, and Robert Pless, "Webcam geo-localization using aggregate light levels," in *IEEE Workshop on Applications of Computer Vision*, 2011. 2

[14] Nathan Jacobs, Nathaniel Roman, and Robert Pless, "Toward Fully Automatic Geo-Location and Geo-Orientation of Static Outdoor Cameras," in *IEEE Workshop on Applications of Computer Vision*, 2008. 2

[15] Tsung-Yi Lin, Serge Belongie, and James Hays, "Cross-view image geolocalization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 2

[16] Noah Snavely, Steven M Seitz, and Richard Szeliski, "Photo tourism: exploring photo collections in 3d," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 835–846, 2006. 2

[17] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua, "Worldwide pose estimation using 3d point clouds," in *European Conference on Computer Vision*, 2012. 2

[18] Georges Baatz, Olivier Saurer, Kevin Köser, and Marc Pollefeys, "Large scale visual geo-localization of images in mountainous terrain," in *European Conference on Computer Vision*, 2012. 2

[19] Srikumar Ramalingam, Sofien Bouaziz, Peter Sturm, and Matthew Brand, "Geolocalization using skylines from omni-images," in *IEEE Workshop on Search in 3D and Video (S3DV)*, 2009. 2

[20] Jean-François Lalonde, Srinivasa G Narasimhan, and Alexei A Efros, "What do the sun and the sky tell us about the camera?," *International Journal of Computer Vision*, 2010. 2

[21] Scott Workman, R. Paul Mihail, and Nathan Jacobs, "A Pot of Gold: Rainbows as a Calibration Cue," in *European Conference on Computer Vision*, 2014. 2

[22] Imran N Junejo and Hassan Foroosh, "Estimating geo-temporal location of stationary cameras using shadow trajectories," in *European Conference on Computer Vision*, 2008. 2

[23] Lin Wu and Xiaochun Cao, "Geo-location estimation from two shadow trajectories," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2

[24] Siddhartha Chib and Edward Greenberg, "Understanding the metropolis-hastings algorithm," *The American Statistician*, vol. 49, no. 4, pp. 327–335, 1995. 3