Identifying the Geographic Location of an Image with a Multimodal Probability Density Function

Jamie Davies
jagd1g11@ecs.soton.ac.uk

Jonathon S. Hare
jsh2@ecs.soton.ac.uk

Sina Samangooei
ss@ecs.soton.ac.uk

John Preston
jlp1g11@ecs.soton.ac.uk

Neha Jain
nj1g12@ecs.soton.ac.uk

David P. Dupplaw
dpd@ecs.soton.ac.uk

Electronics and Computer Science, University of Southampton, United Kingdom

ABSTRACT

Knowing the location that a photograph was taken provides us with data that could be useful in a wide spectrum of applications. With the advance of digital cameras, and with many users exchanging their digital cameras for their GPS-enabled mobile phones, photographs annotated with geographical locations are becoming ever more present on photo-sharing websites such as Flickr. However there is still a wide majority of online content that is not geotagged, meaning that algorithms for efficient and accurate geographical estimation of an image are needed. We present a general model for using both textual metadata and visual features of photos to automatically place them on a world map.

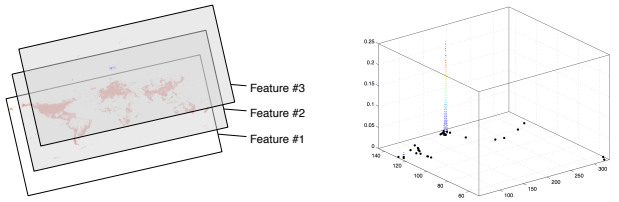
1. INTRODUCTION AND MOTIVATION

The primary goal of the 2013 MediaEval placing task [3] was to develop techniques for accurately predicting the geolocation of a set of Flickr images in terms of latitude and longitude. In addition, a secondary goal was to enhance predictions by estimating the error of the predicted location of each image. The task organisers provided a set of about 8.5 million images with metadata and locations for training, and a set of 262,000 images without geotags for testing.

The motivation for the techniques we have developed for the task was twofold; we firstly wanted to develop a technique that can operated using either the visual content or the metadata, but which also seamlessly allowed blending of information across modalities and also allowed information from external gazetteers to incorporated. Secondly, we wanted the technique to be scalable and efficient, with the aim of being able to estimate the position of an image in well under a second using standard desktop hardware.

2. OVERALL METHODOLOGY

The basic idea of our approach is that we estimate a continuous probability density function (PDF) over the surface of the Earth from a number of *features* extracted from the query image and/or its metadata. To estimate the PDF, each feature provides a fixed size set of *points* (in terms of latitude and longitude) which are then combined (Figure 1a), and a kernel density estimator can be used to estimate the probability density at any arbitrary position (Figure 1b). By finding the modes of the PDF we create an estimate the location of the photograph from the position of the mode with the highest probability. By fitting a univariate Gaussian over the support of the highest probability mode, we can estimate the accuracy of the estimated geolocation as a function of the variance of the Gaussian.



(a) (b)

Figure 1: (a) Sets of points representing the probability density of each feature are overlaid. (b) A kernel density estimator is applied and we estimate the photograph's location at the position of the largest peak in the density function.

In practice, density estimation and mode-finding can be combined by applying the mean-shift algorithm. Mean-shift has been used in the context of geolocation estimation in the past; for example, Hays and Efros [4] used mean-shift on the results of content-based image search to determine probable locations. Whilst Hays and Efros's approach is similar to ours, it differs in a number of important ways. In particular, whereas they only considered single (high recall/low precision) content-based features, we consider the fusion of multiple features from different modalities. In addition, Hays and Efros used the mean-shift algorithm for coarse-grained location estimation, with a very large kernel bandwidth. In our technique, because of the way we are using features we are able to use a much smaller kernel bandwidth for fine-grained location estimation.

3. EXPERIMENTS

The implementation of our methodology was realised in Java using OpenIMAJ[1] [1] and Lucene[2]. For speed, we used an approximate mean-shift implementation inspired by the one in scikit-learn[3]. The approximations stem from using a

[1] `http://openimaj.org`
[2] `http://lucene.apache.org`
[3] `http://scikit-learn.org/stable/modules/clustering.html#mean-shift`

MediaEval 2013 Workshop, October 18-19, 2013, Barcelona, Spain

Table 1: The feature configuration for the run submissions.

	Prior	Tags	PQ-CEDD	SIFT-LSH
Run 1	✓	✓	✓	✓
Run 2	✓		✓	✓
Run 3	✓	✓		
Run 4	✓	✓		✓
Run 5	✓	✓	✓	✓

regular grid for determining the seed points from which to seek modes (rather than using the actual data), and using nearest-neighbours to assign data points to modes, rather than actually assigning them to the mode they converge to. A KD-Tree was used for efficient nearest neighbour lookup.

3.1 Features

The following features were used in the our experiments. Each feature provides a set of geographic points in response to a query image:

Location Prior. A constant prior feature built by sampling 1000 geographical coordinates from the training data.

Tags. Every tag in the query image is associated with the coordinates of the training images in which the tag appeared. If a tag in the query was unseen in the training data, then it contributes no points. Each tag is considered to be an independent feature. No filtering of tags was performed.

PQ-CEDD. In order to provide a high-recall/low-precision image search we indexed the provided CEDD features with a product quantiser (18 products of 256 clusters) to enable fast in memory search of the complete training data using the asymmetric distance computation method [5]. The locations of the 100 top images to a query formed the point set returned by the feature.

LSH-SIFT. High-precision (low recall) image content search was performed using a variant of the approach we developed in [2]. DoG-SIFT features were extracted from the images and hashed using Locality Sensitive Hashing. A graph was built with the images as the nodes, and edge weights were based on the number of collisions. In order to perform a query, the directly connected nodes to the query were detected in the graph, and their geo-coordinates returned.

3.2 Runs

For all of our submitted runs, the methodology described in Section 2 was applied. In all runs we used a flat kernel with bandwidth of 0.01°. All runs except for run #5 used a constant number of 1000 points per feature (effectively making all features have a fixed weight); any feature that produced more or less points was randomly and uniformly sub- or super-sampled to reach 1000 points. In total, 5 runs were submitted. The configurations are summarised in Table 1. Notes about each run are listed below:

Runs 1-3: Provided data. The first three runs used features extracted from the provided dataset only. No external data was used.

Run 4: Text+Visual, bigger dataset. The fourth run used features extracted from a larger dataset of 46 million geotagged Flickr images we crawled last year. As with the provided data, only images with an accuracy of 16 were crawled. For the purposes of fair experimentation, we removed all photos from the users who appeared in the test set.

Run 5: Text+Visual, provided data with tag boosting. The fifth run was the same as run 1, but we used the GeoNames[4] gazetteer to boost the weight of tag features that were likely to belong to a specific geographic location; any textual tag that could be matched against the the GeoNames "name" or "alternate-name" field was boosted by doubling its number of points from 1000 to 2000. Non textual features, and tags that didn't match remained at 1000 points/feature.

3.3 Results and Discussion

4. CONCLUSIONS AND FUTURE WORK

We had started to experiment with product-quantised PCA VLAD encodings of SIFT features, temporal features and query expansion of LSH-SIFT features, however we ran out of time to optimise and include these in the runs. It will be interesting to investigate these further, together with other features such as GIST in the future. It would also be interesting to try some other approaches to incorporating structured knowledge from GeoNames.

5. ACKNOWLEDGMENTS

The described work was funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreements 270239 (ARCOMEM), and 287863 (TrendMiner).

6. ADDITIONAL AUTHORS

Additional author: Paul H. Lewis (email: `phl@ecs.soton.ac.uk`)

7. REFERENCES

[1] J. S. Hare, S. Samangooei, and D. P. Dupplaw. OpenIMAJ and ImageTerrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In *Proceedings of ACM Multimedia 2011*, MM '11, pages 691–694. ACM, 2011.

[2] J. S. Hare, S. Samangooei, D. P. Dupplaw, and P. H. Lewis. Twitter's visual pulse. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, ICMR '13, pages 297–298, New York, NY, USA, 2013. ACM.

[3] C. Hauff, B. Thomee, and M. Trevisiol. Working Notes for the Placing Task at MediaEval 2013. In *MediaEval 2013 Workshop*, Barcelona, Spain, October 18-19 2013.

[4] J. Hays and A. A. Efros. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[5] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, Jan. 2011.

[4] http://www.geonames.org

Table 2

	1	2	3	4	5
No. est. within 1km of the ground truth	53449	891	60190	68050	61631
No. est. within 10km of the ground truth	81988	1453	98032	106370	100009
No. est. within 100km of the ground truth	93838	2711	113937	123233	114986
No. est. within 500km of the ground truth	109117	9174	132655	139136	129721
No. est. within 1000km of the ground truth	122752	19129	147443	151876	141767
Median error in km	1352.897154	6898.266561	451.8928271	254.4838372	540.109773
linear correlation (est. error vs. true error)	0.1568	0.0594	0.3693	0.3721	0.0406