Editorial

ACM SIGSPATIAL GIS 2012 PhD Showcases

Highlights from P-KAR 2012

Announcements
The SIGSPATIAL Special

The SIGSPATIAL Special is the newsletter of the Association for Computing Machinery (ACM) Special Interest Group on Spatial Information (SIGSPATIAL).

ACM SIGSPATIAL addresses issues related to the acquisition, management, and processing of spatially-related information with a focus on algorithmic, geometric, and visual considerations. The scope includes, but is not limited to, geographic information systems.

Current Elected ACM SIGSPATIAL officers are:
Chair, Walid G. Aref, Purdue University
Past Chair, Hanan Samet, University of Maryland
Vice-Chair, Chang-Tien Lu, Virginia Tech
Secretary, Divyakant Agrawal, University of California at Santa Barbara
Treasurer, Mohamed Mokbel, University of Minnesota

Current Appointed ACM SIGSPATIAL officers are:
Newsletter Editor, Andrew Danner, Swarthmore College
Conference Finances Coordinator, Yan Huang, University of North Texas
Conference Venue Coordinator, Shawn Newsam, University of California at Merced
Webmaster, Marco Adelfio, University of Maryland

For more details and membership information for ACM SIGSPATIAL as well as for accessing the newsletters please visit http://www.sigspatial.org.

The SIGSPATIAL Special serves the community by publishing short contributions such as SIGSPATIAL conferences’ highlights, calls and announcements for conferences and journals that are of interest to the community, as well as short technical notes on current topics. The newsletter has three issues every year, i.e., March, July, and November. For more detailed information regarding the newsletter or suggestions please contact the editor via email at adanner@cs.swarthmore.edu.

Notice to contributing authors to The SIGSPATIAL Special: By submitting your article for distribution in this publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance by the editor,
- to digitize and post your article in the electronic version of this publication,
- to include the article in the ACM Digital Library,
- to allow users to copy and distribute the article for noncommercial, educational or research purposes.

However, as a contributing author, you retain copyright to your article and ACM will make every effort to refer requests for commercial use directly to you.

Notice to the readers: Opinions expressed in articles and letters are those of the author(s) and do not necessarily express the opinions of the ACM, SIGSPATIAL or the newsletter.

The SIGSPATIAL Special (ISSN 1946-7729) Volume 3, Number 3, November 2011.
# Table of Contents

**Editorial**  

ACM SIGSPATIAL GIS 2012 PhD Showcases

- A Probabilistic Approach to Mining Geospatial Knowledge from Social Annotations  
- Automated Artifact-free Seafloor Surface Reconstruction with Two-Step ODETLAP  
- Mining Multi-object Spatial-temporal Movement Patterns

Highlights from Workshop on Place-related Knowledge Acquisition Research

Announcements

- SIGSPATIAL & ACM Membership Information
Dear Colleagues,

Welcome to the third issue of the fourth volume of the SIGSPATIAL Special for 2012. This issue is dedicated to the PhD Showcases from the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2012) held in Redondo Beach, California from 6-9 November 2012. Our goal is for the appearance of this issue to coincide with the conference so that all attendees will have simultaneous access to all of the presentations at the conference.

The issue begins with the presentation of four PhD Showcases from ACM SIGSPATIAL GIS 2012. We then include highlights from the International Workshop on Place-Related Knowledge Acquisition Research held in conjunction with Spatial Cognition 2012 on 31 August in Kloster Seeon, Germany. The issue concludes with membership information for ACM and SIGSPATIAL.

Andrew Danner, Editor
Department of Computer Science
Swarthmore College, Swarthmore, Pennsylvania
Tel: +1 610-328-8665
Fax: +1 610-328-8606
Email: adanner@cs.swarthmore.edu
ABSTRACT

Knowledge produced online often comes in the form of free-text labels, known as tags, with which users annotate the content they create, such as photos and videos. Increasingly, such content is also georeferenced, i.e., it is associated with geographic coordinates. The implicit relationships between tags and their locations can tell us much about how people conceptualize places and relations between them. However, extracting such knowledge from social annotations presents many challenges, since annotations are often ambiguous, noisy, uncertain and spatially inhomogeneous. We introduce a probabilistic framework for modeling georeferenced annotations and a method for learning model parameters from data. The framework is flexible and general, and can be used in a variety of applications that mine geospatial knowledge from user generated content. Specifically, we study two problems — extracting place semantics and predicting locations of photos from tags — and show that performance of our method is comparable to that of state-of-the-art approaches. Moreover, we show that combining the two problems leads to a better performance on the location prediction task than baseline.

Categories and Subject Descriptors
I.5.1 [Pattern Recognition]: Models—Statistical

General Terms
Algorithms

Keywords
information extraction, data mining, geo-spatial, social network

1. INTRODUCTION

The social Web sparked a revolution by putting knowledge production tools in the hands of ordinary people. Today on social Web sites such as Twitter, Flickr, and YouTube, large numbers of users not only create rich content, which includes photos and videos, but also annotate content with descriptive labels known as tags, and geo-reference it by associating geographic coordinates, or geo-tags, with it. The implicit relationships between tags and their locations can be mined to learn what places people talk about on the social Web and how much attention they give to these places, where they are, how far they extend, and how they relate to other places. Such knowledge will help us more effectively utilize the massive quantities of user-generated content across a wide spectrum of applications, from responding to disasters, to monitoring the environment, managing resources, and interacting with the world and one another.

Existing works have explored various methods for harvesting geospatial knowledge from geo-tagged user-generated content. They use clustering or information theoretic methods to analyze the spatial distribution of tags to extract place semantics [11], place boundaries [5], or suggest locations for photos based on their tags [3,12]. However, extracting knowledge from annotations created by many different people creates challenges that existing works did not address. Annotators generally differ in their expertise, expressiveness, and enthusiasm for the subject; therefore, the annotations they create are ambiguous, uncertain, noisy, and inhomogeneously distributed. Take the tag ‘victoria’, for example, which could refer to a place in Canada, Australia, or half a dozen other places called Victoria. It is also a popular female name, leading to this tag cropping up in photos taken all over the world. As another example, consider tourists who visit Disneyland on their trip to Los Angeles, causing the tag ‘LA’ to crop up in photos taken in Anaheim, a town in Orange County. As shown in [5], these factors can significantly distort the learned boundary of a place and lead to erroneous conclusions about its relationship to other places. Another challenge specific to geospatial data is the choice of scale at which data is discretized, which affects analysis [15].

In this paper we propose a probabilistic framework for mining geospatial knowledge from social annotations. We represent the spatial distribution of a tag by a mixture of Gaussian probability density functions. Such Gaussian mixture models (GMMs) can be estimated directly from data without using discretization parameters. Being probabilistic, the model also presents a natural way to deal with noise and uncertainty. However, using Gaussian mixture models presents its own challenges, as the number of Gaussian components in the model is not known a priori. Too high a number may lead to over-fitting the estimated probability density function, while too low a number will lead to an inaccurate probability density function. In this paper, we solve this problem by using Bayes information criterion to find the best number of components, and then estimate the parameters of each component using expectation maximization.

We use the proposed probabilistic framework to solve two tasks in geo-spatial data mining. The first task, place identification [11],
attempts to classify tags as place names or not place names. The second task, location prediction [3, 12], attempts to predict locations of photos given their tags. We show that the performance of the proposed method is superior to state-of-the-art algorithms. Moreover, the probabilistic framework enables us to eliminate tags that have low spatial information, e.g., ‘iphone’, leading to a much better performance on the location prediction task. Probabilistic models offer a general and flexible framework for solving a variety of geo-spatial data mining problems. The problems solved in this paper are just two illustrations of the utility of the approach.

2. PROBABILISTIC MODELING FRAMEWORK

We focus on analyzing the social photo-sharing site Flickr, which allows registered users to upload photos and videos and annotate them with descriptive labels, known as tags. Tags are used to describe the image’s subject (e.g., ‘animal’, ‘family’), properties (e.g., ‘cute’, ‘furry’), medium (e.g., ‘nikon’), as well as where the image was taken (e.g., ‘zoo’, ‘los angeles’, ‘california’). In addition to tagging photos and videos, Flickr also allows users to geo-tag or geo-reference, content by attaching geographic coordinates to it. The implicit relation between tags and locations of photos annotated with those tags can tell us much about how people think of places and relations between them.

We model tags and photos in terms of spatial probability distributions or density functions. Spatial distribution of tag \( w \) can be written as \( p(X|w) \), where \( X \) represents locations of photos tagged with \( w \). Then, we can easily model spatial probability distribution of a photo as a superposition of probability distributions of all tags in that photo.

We start by modeling probability distribution of each tag separately. Multivariate Gaussians, Gaussian mixture models (GMM) and kernel density estimation (KDE) are some of the more popular methods for estimating probability density of the observed data [6]. However, different tags may be expressed at different levels of granularity, from continent (e.g., ‘asia’) to the level of landmarks (e.g., ‘goldengatebridge’). Kernel density estimators require a scale, or bandwidth, parameter to be set, which may not be known a-priori and vary from tag to tag. Therefore, we prefer to use GMMs as the method for estimating the probability distribution of each tag. Using a sufficient number of Gaussian components, and by tuning the parameters of each component and adding them linearly, most continuous distributions can be approximated [2]. Moreover, probabilistic models can address the challenges of noise, uncertainty and ambiguity. However, there still remains a challenge to using GMMs to model spatial distribution of tags, as the number of components may not be known beforehand, and using too many components puts us in danger of overfitting the data. The section below proposes a solution to this problem.

2.1 Modeling Tag Distribution as Gaussian Mixture

Gaussian mixture model is a superposition of \( K \) Gaussians:

\[
p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k),
\]

where each Gaussian density \( N(x|\mu_k, \Sigma_k) \) is called a component of the mixture with mean \( \mu_k \) and covariance matrix \( \Sigma_k \). Parameter \( \pi_k \), called the mixing coefficient, gives the weight of the \( k \)th component, or the fraction of all data points that are explained by that component. It has a value between 0 and 1, and \( \sum_{k=1}^{K} \pi_k = 1 \). The distribution of photos tagged ‘victoria’, for example, will have one highly localized component in Canada and one in Australia. In addition, there will be random photos all over the world tagged ‘victoria’, where the term is being used as a female name. These points may be assigned to their own, noise component, with a low \( \pi_k \), because relatively few of all photos tagged ‘victoria’ will be classified as noise. Noise can also contribute to the variance of each Gaussian component.

The Gaussian mixture model is specified by the number of mixture components, each governed by parameters \( \pi_k, \mu_k, \Sigma_k \). For a given model, we use expectation-maximization (EM) algorithm to estimate model parameters \( \pi_k, \mu_k, \Sigma_k \). EM is an iterative algorithm with two major steps: expectation (E) and maximization (M) step. The E step estimate \( \gamma(z_{nk}) \) from the current parameter values where \( \gamma(z_{nk}) \) can be viewed as as responsibility of component \( k \) generate \( x_n \). The M-step updates the values of \( \pi_k, \mu_k, \Sigma_k \) from \( \gamma(z_{nk}) \) of the previous step. The process continues until convergence. The convergence is usually defined by when the log likelihood function or finding parameters changes below some threshold [2].

However, how many mixture components \( K \) should we use in a model of each tag? Intuitively, the optimal number of components is one that best explains the data. By adding more parameters to the model, one can usually get the model to better describe the data. However, this may lead to overfitting, where a very complex model explains every point in the training set, but has no predictive power for test data. We can reduce this problem by penalizing complexity in a model (e.g. AIC, BIC) or testing predictive performance on test data (e.g., cross validation) [8, 13, 9, 14]. Since BIC performs slightly better on our data than cross-validation, we focus our discussion on it.

The BIC is used as criterion for model selection in statistics. It avoids overfitting by introducing a penalty term for the number for parameters in the model [8]:

\[
BIC = -2 \cdot \ln L + k \cdot \ln(n).
\]

Here \( L \) is likelihood estimate of the model, \( k \) is number of parameters and \( n \) the number of observations, or data points. Our model is a mixture of \( K \) bivariate Gaussian components. Each component has a bivariate mean, which is specified by two parameters, co-variance matrix, which is symmetric, therefore, specified by three parameters, and a mixing coefficient, which contributes one parameter to the model. However, since mixing coefficients have to add to one, this constraint removes one parameter. Therefore, the total number of model parameters is \( k = (K - 1) + 2K + 3K = 6K - 1 \).

Our model selection process is very simple. We estimate model parameters using the EM algorithm to get the maximum likelihood estimate \( L(K) \) with respect to the number of components \( K \). We then choose the values of \( K \) that minimizes the BIC value of the model:

\[
K = \arg \min_{K} BIC(K).
\]

3. APPLICATIONS TO MINING GEO-SPATIAL DATA

Modeling tag distribution as a mixture of Gaussians offers a general and flexible framework for a number of geo-spatial inference tasks. In this section, we study two problems: place identification and location prediction. The goal of the first task is to learn whether or not a tag refers to a place based on its spatial distribution. The intuition behind the approach is that tags that are place names are more localized than those that are not place names, and the amount of localization can be estimated from the variance of the components of the tag. The goal of the second task is to geo-reference a photo based on its tags. Our approach exploits the structure in the data by probabilistically modeling the relations between a photo’s location and its tags using the Gaussian mixture model. We show
that the performance of the proposed method is better than state-of-the-art algorithms. Moreover, the probabilistic framework enables us to eliminate tags that have low spatial information, e.g., ‘iphone’, leading to improved performance on the location prediction task over the baseline method.

3.1 Data collection
We collected data for our experiments from the social photo-sharing site Flickr. We used the Flickr API to retrieve information about more than 14 million geo-tagged photos created by 260K distinct users. These photos were tagged over 89 million times with 3 million unique tags. Following [11], we represent each photo \( p \) as a tuple \( \{id_p,u_p,l_p,T_p\} \), where \( id_p \) is the site-specific id of the photo, \( u_p \) is id of the user who create it, \( l_p \) represents the photo’s location as lat-long pair, and \( T_p \) is set of tags in the photo. The photo has only one location, and one user, but it may have multiple tags.

As a preprocessing step, we filter out tags which were used by fewer than 100 people. As a consequence, photos containing only the less popular tags will not be included in the data set. We randomly select five thousand users whose photos will be the basis of the test data set. To create the training data set for learning GMM parameters of the distribution of a given tag, we sample a single photo from each of the remaining users uniformly from a 100 km grid. This sampling procedure helps reduce bias from users who take many more photos in some location compared to other users [3]. The same training set is used for both place place identification and location suggestion tasks. After these preprocessing steps, the training set contains 2.5M photos with 192K distinct tags.

3.2 Place Identification
One of main challenges of social computing is to extract structured knowledge from a set of annotations, such as tags. One way to extract such knowledge is by exploiting their usage patterns. Rattenbury et al. [11] analyzed spatial patterns to distinguish between place and non-place tags. We use the same intuition in our method, namely “place tags exhibit spatial usage patterns that are significantly geographically localized” [11].

3.2.1 Baseline: Entropy-based Method
[11] proposed a quantitative method to identify place tags. One of main challenges in determining whether the tags are localized is known as the Modifiable Areal Unit Problem, or MAUP [10]. The concern is that aggregate statistics are sensitive to the choice of spatial unit used to discretize data. For example, if we choose a 100 km \( \times \) 100 km grid, we may see samples of Los Angeles that are localized, i.e., are within the grid, while samples of California that are not localized. However, by increasing grid size to 1,000 km \( \times \) 1,000 km, both of them will be localized.

The solution proposed by [11] relied on multi-scale analysis method that they called Scale-Structure Identification. They cluster data samples with multiple radii \( r \). The intuition behind this method is if a tag is localized, most of samples should be found within a single cluster. For example, when we cluster samples of tag ‘san-francisco’ with radius \( r = 100 \) km, most of the samples will fall into a single cluster with few noisy samples in other clusters. Then, they compute the probability of samples in each cluster. Finally, they compute entropy from this probability mass function. Lower entropy means that most of samples fall within the same cluster. In extreme case, if all the samples fall in the same cluster, the entropy is zero. They tackle problem of MAUP by using multiple values of scale parameter \( r \) and combine entropy values at different scales. The entropy of place tag will converge to zero very fast for small \( r \). In contrast, the entropy of a non-place tag will converge to zero at very large \( r \). The method checks whether the entropy of a tag is below some threshold, and if yes, it classifies the tag as a place name. However, if the tag is ambiguous and refers to places that are far away from each other, e.g., ‘victoria’ may refer to a place in Canada or Australia, the method will not judge it as being well recognized, and hence, not recognize it as a place tag.

3.2.2 Model-Based Method
Instead of discretizing data at different scales, we work with a continuous probability density function. We judge whether a tag is well-localized or not by examining the average log-likelihood of its samples with respect to the learned GMM parameters. The intuition behind our method is as follows. If a tag is well-localized, its distribution will be highly peaked, so that most of the samples will be close to the mean of the Gaussian component, and as a result, the standard deviation of the component will be low. Since the model is a superposition of components, we need to combine them to get a single value which can be used to compare localization semantics.

We combine the components together using average log-likelihood. The likelihood function gives the probability of observed samples given model parameters. High average log-likelihood tags tend to have major components with low standard deviation. This may be explained by likelihood of Gaussian formula. When most of observed data is highly localized, the determinant of the covariance matrix is low. Consequently, the likelihood of these data points will be high from multiplication in both exponential and determinant of inverse covariance matrix term. However, each tag has different number of data samples. Thus, we use the normalized version of log-likelihood called average log-likelihood which can be computed by

\[
\bar{\mathcal{L}}_w(\pi_w, \mu_w, \Sigma_w | X_w) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \pi_{wik} N(x_{wi} | \mu_{wik}, \Sigma_{wik}) \tag{1}
\]

where \( X_w \) is an \( N \times 2 \) matrix of locations of photos which have been annotated with a tag \( w \). The vector \( x_{wi} \) represents row \( i \) of the matrix \( X_w \), namely the latitude and longitude of \( i \)th photo’s location. \( \pi_{wik}, \mu_{wik}, \Sigma_{wik} \) are parameters of component \( k \) of tag \( w \). Then, high average likelihood means that the observed data points are more clustered, which means they are more localized with respect to their maximum likelihood model parameters. Thus, if a tag’s average likelihood with respect to its maximum likelihood model parameters is higher than some threshold \( \theta \), we identify it as a place name:

\[
\bar{\mathcal{L}}_w(\pi_w, \mu_w, \Sigma_w | X_w) > \theta \tag{2}
\]

3.2.3 Evaluation
To compare the performance of different methods on the place identification task, we need a ground truth data about place names. We use GeoNames (geonames.org), a geographical database containing over eight million place names all over the world, for this purpose. We assume that if a tag exactly matches a name or an alternate name in GeoNames, then it is a place name. For example, tag ‘usa’ matches the alternate name of United States. If successful, the match function returns a geoid (geographic id) of the place. Unfortunately, any word, including non-place word, may match a name and an alternate name in GeoNames. For example, matching the tag ‘cake’ will return a place with geoid 7063723. We exploit geographic information in the Geonames database to solve this problem. We use two heuristics in building the ground truth.
### Table 1: Comparison of the model-based approach (gmm) to baseline on the place identification task.

<table>
<thead>
<tr>
<th>approach</th>
<th>AUC</th>
<th>Max F1</th>
<th>Min CE</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>0.3668</td>
<td>0.7530</td>
<td>0.0701</td>
</tr>
<tr>
<td>gmm</td>
<td>0.3719</td>
<td>0.7682</td>
<td>0.0683</td>
</tr>
</tbody>
</table>

First, tags that match names at administrative level 1 or higher are judged to be place name. The places that appear at this level are countries. Second, for landmarks such as Alcatraz, we list all possible matches with their locations. Then, for each photo with that tag, we find the nearest match location of the photo and compute the distance, or error, between the photo and the nearest match location. For example, a photo tagged with ‘victoria’ with location in Canada will be matched to Victoria, BC, while other photos may match to Victoria, Australia. If the median of the error distribution less than 100 km, we classify the tag as a place name.

We use standard precision and recall metrics to evaluate the performance of baseline and the proposed model-based method on the place identification task. Precision measures the fraction of tags that were correctly predicted to be place names relative to the number of all tags that were predicted to be place names. Recall measures the fraction of place names in Geonames that were predicted to be place names. However, precision and recall are sensitive to threshold value; therefore, we compute the performance as the precision-recall curve. Aggregate metrics, including AUC (area under precision-recall curve, maximum F1 score (Max F1), and minimum classification error rate (Min CE), are reported in Table 1. As we can see from these results, our method is competitive, and even has slight advantage over, the state-of-the-art place identification methods. However, the true advantage of our method is its flexibility, as the same probabilistic framework can be used to address different geospatial data mining questions, as shown below.

### 3.3 Location Prediction

We use the probabilistic framework to solve a different problem, namely, find the most likely geographic location of a photo given its tags. Many of the photos on Flickr were not taken by a GPS-enabled device. Can we leverage the annotations users create to automatically geo-reference them? Human can often easily infer a photo’s location from its tags, and assign it to the photo manually. For example, a set of tags {‘castle’, ‘smithsonian’, ‘dc’, ‘washington’} allows us to accurately place the photo, while the tags {‘flower’, ‘night’, ‘bug’, ‘insect’, ‘moth’} do not. However, this method has many disadvantages. First, it is time consuming. Second, a user has limited memory and knowledge of places, and may be able to guess the location accurately only for familiar places. For example, Europeans people may not give a good guesses about locations in America. In contrast, an automatic method could do this more cheaply.

#### 3.3.1 Baseline

Previous researchers framed the location prediction problem as a classification [3, 12]. Crandall et al. [3], for example, used the mean-shift clustering algorithm to separate photos into distinct clusters, with each clusters representing a class of locations. The tags of the photos in the cluster are used as prediction features. Their method computes the probability of a class and probability of each feature given the class. Then, given a new photo with some tags, they use a naive Bayes classifier to predict the class the photo belongs to. Their method is also sensitive to the scale parameter used to discretize data [15]. They used 100 km as the grid size for clustering photos.

Instead of measuring classification accuracy of the baseline method, as [3] did, we measure its performance in terms of the error between the photo’s actual and predicted locations. For example, the actual location of a photo could be Los Angeles, but the algorithm predicts its location to be San Diego. This prediction error is much less than one resulting from the predicted location being New York City. Since the Bayes Classifier gives only the class of the photo but not the numeric value of its geographic location, we take the location of the class to be the mode of the cluster from mean shift clustering procedure. While the original work reported predictions made by two classifiers, they were found to have similar performance. We chose to use the naive Bayes Classifier as the baseline, because it is easier for us to verify its results.

#### 3.3.2 Model-based Location Prediction

We model a photo as a probability density function \( p(x) \) in continuous space. Our model-based approach allows us to express the relationship between a photo’s location and its tags, whose distributions were learned from the training data. In previous section we have already computed \( p(x|w) \) where \( x \) is a geographic location and \( w \) is a tag. We will show that we can model a probability density function of a photo with this density function.

The assumption in our model is that tag frequency in one photo is one, i.e., a user does not repeatedly use the same tag in a photo. Thus, the probability distribution of each tag in a photo is uniform. Given \( W = \{w_i\} \), the set of tags in a photo, then using the assumption above, \( p(w) = 1/|W| \), where \( |W| \) denotes the number of tags in the photo. We can derive \( p(x) \) of a photo from marginal probability formula of joint probability between two random variables, location \( x \) and tag \( w, p(x, w) \).

\[
p(x) = \sum_{w \in W} p(x, w) = \sum_{w \in W} p(x|w)p(w) \\
= \sum_{w \in W} p(x|w) \cdot (1/|W|)
\]

Probability density function of a photo is a superposition of bivariate Gaussian distributions because each component, \( p(x|w) \), is superposition of bivariate Gaussian distributions. The distribution \( p(x) \) can be interpreted as probability of the photo’s location. Therefore, the best guess of location \( x \) of a photo is the mode of \( p(x) \), in other words, location \( x \) that corresponds to the maximum value of \( p(x) \).

\[
x_{\text{predict}} = \arg \max_x p(x) = \arg \max_x \sum_{w \in W} p(x|w)p(w)
\]

We can ignore the term \( p(w) \), because it is constant, independent of the photo’s location. However, \( p(x) \) of a photo is a non-linear function. To optimize this function numerically, in current implementation, we use Matlab implementation of a numerical method called Nelder-Mead Simplex Method [7]. We guess initial points to be means of each bivariate Gaussian component. The number of guessing trials is equal to number of component. Each trial will give the local optimal point \( x \). We pick \( x \) which gives the highest value of \( p(x) \) of photos from these local optimal points as the predicted location.

### 3.4 Evaluation

We use methodology described in [3] to prepare the test data set for evaluating the proposed method. Note that randomly selected 5000 users to be in the test set, the training and test data sets are disjoint on users. We then randomly pick single photo for each
user. Thus, in the location prediction test set there are 5000 photos from 5000 users who are not in the training set.

using the training data (e.g., $\mu = 77.4$ km and $\sigma = 812.4$ km). Finally, it uses these values to compute the confidence of the prediction for $b = 500$ km, which is 0.69.

### 3.4.1 Using Tag Likelihood in Location Prediction

Existing location prediction methods suggest using gazetteer to improve prediction accuracy [1, 4, 12]. The key idea is that place-related terms, e.g., ‘goldengate’, should have higher weight than non-place-related terms, e.g., ‘iphone’, in classification. However, there is no general agreement about how to weigh these terms: for example, should place and non-place terms be assigned binary values, or a continues value based on some criteria such as localization? Second, gazetteer may be incomplete or outdated. Also, we may encounter common terms such as ‘cake’ in a gazetteer that are not place names.

![Figure 1: Comparison of performance of different methods in terms of the error between the photos' predicted and actual locations. (a) Distribution of errors in the test set. (b) Cumulative distribution function of errors produced by the proposed method (gmm) and baseline.](image)

We hide the actual locations of photos in the test set and use our method and baseline to predict their locations. We compare performances between our method and baseline using Euclidean distance between predicted and actual locations:

$$\text{error} = [(\text{lon}_{\text{actual}} - \text{lon}_{\text{predict}})^2 + (\text{lat}_{\text{actual}} - \text{lat}_{\text{predict}})^2]^{1/2}$$

Figure 1(a) shows the distribution of prediction errors made by the model-based approach as a histogram, where each bin corresponds to a unit of 100 km. The bins corresponding to the lowest errors have the highest frequency, implying that our method results in small prediction errors most of the time. Figure 1(b) compares the cumulative probability distribution (CDF) of the errors made by the proposed method (gmm) and baseline. The proposed method has higher probability for lower errors, meaning that it produces better predictions than the baseline.

Unlike cluster-based approaches, probabilistic model-based approach allows us to estimate the confidence of the prediction, i.e., the probability that the predicted location falls within some radius $\delta$ of the actual location. The confidence value tells us how well the data constrains model parameters, and it is bounded by the best localized tag’s probability density. The formula for computing confidence is

$$\text{cdf}(b, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{b} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx,$$

where $\mu$ and $\sigma$ are the parameters of the lowest variance tag estimated from training data. For example, suppose a user would like to estimate a location from tags ['castle', 'smithsonian', 'dc', 'washington'], and get some idea about how likely it is that the predicted location falls within 500 km of the true location. Our method will first determine which tag has lowest variance (in this case ‘smithsonian’), and look up its parameters that were estimated

![Figure 2: Using tag likelihood to improve location prediction from tags. (a) Scatter plot of the prediction error vs likelihood of photo’s representative (highest likelihood) tag. (b) CDF of prediction errors after filtering out photos whose tags are not well localized. Each line corresponds to a different filtering threshold, i.e., the minimum likelihood value for the most localized tag in the photo.](image)
tion error for different values of likelihood threshold. The line that shows the CDF of the predicted error for threshold of -12 (which overlaps the results for threshold=−9) corresponds to the CDF of the model-based prediction error without filtering (Fig. 1(b)). The figure shows that we can get much more accurate predictions (lower errors) for photos that use well-localized tags, and the more localized the tags, the better the performance. In fact, the performance is much better than baseline.

4. RELATED WORK

Researchers propose several methods for finding locations of photos. Crandall et al. use the mean shift algorithm to cluster 20 million photos [3]. They called each cluster a “landmark” or “city” depending on the scale parameter they used. Then, they viewed each of these clusters as a “class” in supervised learning problem. Finally, they use standard machine learning tools to find what “class”, i.e., “landmark” or “city”, a photo belongs to. The possible location will be restricted by number of classes. Instead, Serdykov et al. [12] invented a system that places no restrictions on the particular landmarks. They model the world as an $M \times N$ grid. Each grid can be viewed as topic that generates words. Thus, they use bag of words model for finding the most possible location given tag which is analogy to the most plausible document given words.

The main problem of both state-of-the-art approaches is that they discretize continuous space into discrete classes or grids. The grid size was chosen experimentally. According to [15], however, different discretization techniques can affect classification performance. Our method, in contrast, works in continuous space and thus avoids the discretization problem.

One challenge in spatial data analysis is the problem of scale. Changing scale can affect the statistics significantly. In place identification problem studied by [11], researchers leveraged the problem of “scale” in the scale-structure method by computing statistics at every predefined scale, then used sum of these statistics for significance testing. In contrast, we manipulate the probability density function, which does not have the issues of scale.

5. CONCLUSION

We presented a probabilistic framework that models each tag in continuous space as a mixture of Gaussian distributions, the parameters of which we can estimate by analyzing a corpus of geo-referenced and annotated photos. The probabilistic framework is flexible and can used to solve a number of geo-spatial data mining problems. We illustrate how the same approach can be used in place identification and location prediction tasks. We showed that while method was competitive with state-of-the-art solutions to these problems, it went beyond existing solutions to enable us to compute a confidence of the prediction, as well as combine the two tasks to improve prediction performance on the location prediction task. The same framework can be used to address other problems, such as suggesting tags for geo-referenced photos and learning part-of-relations between places, as we intend to explore in future work. Our work shows the utility of probabilistic models in mining knowledge from social annotations.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. IIS-0812677 and CMMI-0753124. Suradej Intagorn gratefully acknowledges the support of the Thai Government.

6. REFERENCES

ABSTRACT
We present an updated artifact-free seafloor surface reconstruction scheme which preserves more terrain features than our previous attempt using overdetermined Laplacian Partial Differential Equation (ODETLAP) and automates the adjustment of smoothing parameter. The high resolution version of such a surface fitting problem remains a challenge since we are still confined to extremely unevenly distributed depth samples collected along and near the ships, in which case numerous generic reconstruction algorithms generate unacceptable surfaces featuring abnormal depth fluctuations which are correlated with the trackline locations. Previously we reported the use a modified ODETLAP scheme, which integrates data-density-dependent smoothing into the reconstruction process, to generate surfaces which are free from such acquisition footprint. However, that scheme still suffers from terrain feature loss due to smoothing, and the reliance of human to decide appropriate smoothing factor. This paper aims to fix these two problems with a two-step ODETLAP procedure. The procedure first applies an accuracy-biased ODETLAP to complete the missing depth data from the given samples. After that, the vigorous depth fluctuations along the tracklines are removed by applying a smoothing-biased ODETLAP on the completed depth grid. To decide the optimal smoothing factor automatically, the procedure computes the areas of the individual bumps on the reconstructed surface. A surface suffering heavily from the artifacts has many small bumps but few big ones. Smoothing reduces such skewness. We find that for many datasets, the artifact is mostly gone when the coefficient of variation of the areas drops to around 1.3. Using that value to gauge the smoothing factor, the automated scheme successfully generates artifact-free seafloor surfaces within a limited error budget.

Categories and Subject Descriptors
I.3.5 [Computing Methodologies]: Computer Graphics Computational Geometry and Object Modeling

Keywords
GIS, ODETLAP, bathymetry, surface reconstruction, sparse height grid

1. INTRODUCTION
A bathymetric chart, the underwater equivalent of a topographic map, represents the depth and features of the ocean floor. This data helps solve not only applications such as tsunami hazard assessment, communications cable and pipeline route planning, resource exploration, habitat management, and territorial claims under the Law of the Sea, but also fundamental Earth science questions, such as what controls seafloor shape and how seafloor shape influences global climate [12].

While wide-area, high-resolution ground heights can now be measured quickly with aerial electromagnetic survey technologies such as standard photogrammetry and LIDAR [10], the same is not true with seafloor depths. Good discussions on this issue are available at [12, 13]. In short, the problem lies in the 3000-5000 meters of salty water which masks the penetration of electromagnetic waves.

The altimeter method is an attempt for wide-area coverage. The method exploits the fact that the ocean surface has broad bumps and dips that mimic the topography of the ocean floor. By surveying the shape of the water surface instead, we avoid the need of shooting electromagnetic waves into the water, yet allowing us to deduce how the underlying seafloor looks like. However, features on the ocean floor that are narrower than the average ocean depth of 3-5 kilometers do not produce measurable bumps on the ocean surface.

To obtain high-resolution data, we have no choice but to sail across the ocean, and on the way send out acoustic pulses to the seafloor. From the time it takes the pulses to leave the ship, be reflected by the seafloor and eventually get back the ship again, the ocean depths can be estimated. However, since ships travel slowly, the ocean remains largely uncharted. It has been estimated that 300 years are needed to cover the whole ocean area. With the most popular multibeam bathymetry technique [15], we can collect many data points with 10m resolution in a swath up to 10km wide along a ship’s trackline. However, between the tracklines there is no data. In the southern oceans, these survey lines can sometimes be close together, but more often they are hundreds of kilometers apart [14]. Figure 1 demonstrates the spatial distribution of such shipboard data samples in a few 10 × 10 degree regions.
Since a full data grid is assumed for many terrain analyses, we need a surface reconstruction. The problem is defined over a spatial domain of dimensions $n \times n$. Available are the measured depth values of $k \ll n^2$ positions $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$, denoted as $h_{x_1, y_1}, h_{x_2, y_2}, \ldots, h_{x_k, y_k}$. The task is to predict the depths for all the $n \times n$ positions in the domain, including those of the $k$ known positions and those of the remaining $n^2 - k$ unknown positions. This assumes for each possible location $(i, j)$, the corresponding predicted depth $z_{i,j}$ is single-valued; caves or overhangs are not allowed.

Acquisition footprint is the major problem of using general reconstruction schemes on those extremely unevenly distributed depth data, even with a few current bathymetry charts such as the one published by National Oceanic and Atmospheric Administration (NOAA) [9]. It refers to the artifact which makes the tracklines visible. The artifact is especially obvious under shaded relief, a graphic technique which is often used to highlight terrain surface variations [17]. Figure 2 shows how the reconstructed seafloor surfaces look like under a few such schemes such as Kriging [4] and natural neighbor [16]. While both come up with a surface of a similar general shape, we observe abnormal high-frequency depth fluctuations which are correlated with the few trackline locations. We aim at a surface from which we cannot deduce the terrain heights better than conventional prediction schemes. Besides, it can work in two ways. First, we find that quite a few terrain features have been lost in the above variable-smoothing ODETLAP. It predicts neighboring terrain heights better than conventional algorithm. At that time, we further improved its accuracy by allowing the smoothing factor to vary according to local smoothing need. Section 2 will give information about this solution.

This paper aims to improve that ODETLAP implementation in two ways. First, we find that quite a few terrain features have been lost in the above variable-smoothing ODETLAP. We fix it by a two-step approach which reconstructs a preliminary surface with ODETLAP of a high-accuracy setting, and then applying a smoothing-biased ODETLAP over that preliminary surface. Second, our previously-reported implementation relied on human to set smoothing parameters. We automate the process based on the bump area distribution. Details will be given in Section 3, before we conclude the paper in Section 4.

2. ODETLAP

We first presented Overdetermined Laplacian Partial Differential Equation (ODETLAP) as a superior above-ground terrain reconstruction and lossy terrain compression [2, 20, 21], since it predicts neighboring terrain heights better than other conventional prediction schemes. Besides, it can work with contour lines (continuous or intermittently broken), infer mountain tops inside a ring of contours, and enforce continuity of slope across contours. All these are favorable features of natural-looking terrains.

Its formulation sets up an overdetermined system $Az = b$, as shown in Figure 3, to solve for the depths of the whole seafloor depth grid $z$ in the bathymetry case presented here. The system includes an exact equation for each of the $k$ known-depth positions. That equation aims to set the depth value
of the respective position to its known value. The system also contains an averaging equation for all $n^2$ positions. That equation attempts to regularize the respective depth to the average of its immediate four neighbors. Through adjusting the weights $r_{i,j}$ of averaging equations, we can change how the errors are distributed over the equations and hence obtain terrain surfaces of the desired accuracy-smoothness tradeoff.

If the $r$ values are low (e.g. 1), the system is accuracy-centric. The reconstructed values of the known-depth locations will be close to the measured values. However, acquisition footprint appears extensively as vigorous fluctuations along the tracklines, as shown in Figure 4, top left. Note that in areas with no data, the surface is relatively plain. This is superior to the surfaces done with a few conventional reconstruction schemes mentioned earlier, in which the artifact also affects those regions. To alleviate such artifact, we can use the smoothing-centric version in which the $r$ values are high (e.g. 50). Accuracy at known-depth locations (especially those with exceptional values) are sacrificed for a smooth surface as implied by the averaging equations, as shown in Figure 4, middle left.

In the original version of ODETLAP, we set all $r_{i,j}$ to the same value. In our previous paper [5], we demonstrated how unnecessary error budget could be saved by adjusting the $r_{i,j}$ values for individual locations. We observed that the trackline locations were usually of high data density and hence required relatively higher smoothing. We asked the users to specify the smoothing factors for locations with lowest local sample density and locations with highest sample density, and then allowed locations with intermediate data density to vary between these two values. As a result, we achieved a surface of similar smoothness with a smaller error budget. Figure 4, bottom left, shows a typical reconstruction results with such a variable-smoothing ODETLAP system.

The time complexity of ODETLAP is $O(n^3 + k)$. In practice, we transform the system to $A^T A z = A^T b$ before solving for $z$. In this equivalent system $A' z = b'$ where $A' = A^T A$ and $b' = A^T b$, $A'$ is symmetric positive definite. We can then take advantage of the fast Cholesky factorization to keep the actual solving time to within seconds even for large datasets [7]. Note that the matrix $A$ is indeed a $n^2 \times n^2$ sparse matrix because the number of non-zero entries in each row is upper bounded by the number of possible immediate neighbors, which is 4. With the recent advances of graphical display units (GPU) in solving sparse linear systems [1, 6, 8], this approach may offer even faster and more efficient solutions to large data grids.

### 3. AUTOMATED TWO-STEP ODETLAP

Figure 5 gives the flow diagram of the two-step ODETLAP as our solution of the automated artifact-free seafloor surface reconstruction problem. Below we will describe the algorithm in terms of its two feature characteristics: terrain feature preservation and automated smooth factor determination.

#### 3.1 Terrain feature preservation

Our new algorithm first reconstructs a highly-accurate preliminary surface from the available depth data, and then smooths that preliminary surface.

In the first step which reconstructs a preliminary seafloor surface, we choose ODETLAP since this reconstruction scheme works best in deducing the missing seafloor depth data, as described in Section 2. In case the resultant surface needs no further smoothing, we are likely to have obtained the most
accurate surface. We set the weighting between the exact equations and the averaging equations to 1:1. Beyond that point, increasing the weightings of the exact equations does not change the accuracy of the preliminary terrain too much.

To investigate the effect of switching the terrain reconstruction scheme in the first step from ODETLAP to the others, we first remove the height values of a few full terrains except those falling on the tracklines shown in Figure 1. (Note we do not use seafloor surfaces as no ground truth is available for error comparison.) Then we reconstruct the terrain with different techniques and compute the errors with respect to the ground truth. ODETLAP does the best in guessing the missing heights, as reflected by the generally lower mean errors and maximum errors shown in Table 1. This result supports our choice of ODETLAP even if we are now working on extremely unevenly distributed depth samples.

In the second step which smoothes the preliminary surface, we once again pick ODETLAP since this scheme provides better smoothing of the artifacts than others under the same error budget. Figure 4, right, compares the smoothing results with a mean filter (similar to the one used in CleanTOPO2 mentioned in Section 2) and ODETLAP under similar error budget. While the acquisition artifact is almost gone with ODETLAP smoothing, it is not the case with the mean filter. Table 2 shows the respective mean error budgets needed by the sample seafloors to achieve the respective optimal smoothing levels (using the metric defined in the next subsection). Our ODETLAP smoothing scheme just needs half of the error budget as the average filter counterpart. The results above demonstrate the capability of our scheme in distributing the limited error budget to smoothing locations.

The only variable of this algorithm is the smoothing factor in the second step. Figures 6–8, left, show how the surface varies as smoothing increases. A higher smoothing factor means a higher mean error but at the same time better smoothing-out of the small bumps along the tracklines. When compared with our original implementation which requires the specification of the lower and upper smoothing factors, we now have one fewer degree of freedom, make it easier to adjust. Also note that on raising the error budget, the high-frequency acquisition footprint is gone before those lower-frequency terrain features which account for the general terrain shape. Such a smoothing priority makes the scheme superior over our previously-proposed variable-smoothing ODETLAP. In the surface reconstructed with variable-smoothing ODETLAP such as the one in Figure 4 middle right, even though the acquisition footprint is also almost gone, we also lose quite a few terrain features.

### 3.2 Automated smoothing determination

The above two-step procedure features a single parameter controlling the smoothing of the final reconstructed seafloor surface. To allow automated determination of the optimal smoothing level, first we need to convert the acquisition footprint to a form that is recognizable by computers. After several experiments, we find that the graph of Gaussian curvature may be used. As shown on the right hand side of Figures 6–8, the artifact appears as relatively small patches of positive Gaussian curvature concentrated at the trackline locations. In fact, patches of Gaussian curvature is regarded as a view-independent indicator of regions with potential shaded relief in a few other research [5].

As observed from the same set of figures, smoothing helps enlarge those patches or simply remove them. With a small smoothing factor, along the tracklines we have a huge number of small such patches. This is in contrast with regions with no data where there are few, much bigger patches. On increasing smoothing, mean error at known-height location increases. Meanwhile, the bumps along the tracklines become fewer and bigger, while those in no-data regions hardly change in terms of both size and quantity. This leads to a drop of the variations among the patch areas.

To utilize the above phenomenon in automated smoothing factor determination, we first apply a morphological erosion [19] with a 3-pixel-width square component on the positive Gaussian curvature graph to help discriminate the patches. We then compute the coefficient of variation cv, which is a normalized measure of dispersion of a probability distribution [18], of the patch areas. The acquisition footprint is found to be almost gone when that coefficient drops to around 1.3. Figures 6–8, bottom, correspond to a smoothing level with

<table>
<thead>
<tr>
<th>Region</th>
<th>ODETLAP Mean filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region 1</td>
<td>130m 297m</td>
</tr>
<tr>
<td>Region 2</td>
<td>75m 178m</td>
</tr>
<tr>
<td>Region 3</td>
<td>214m 360m</td>
</tr>
</tbody>
</table>

Table 2: Mean error budgets to reach optimal smoothing levels.
Figure 6: (Left) Region 1 reconstructed seafloor surfaces. (Right) Respective positive Gaussian curvature locations. The mean errors, from top to bottom, are around 44m, 78m and 130m respectively.

around that coefficient value. Note that different datasets may need different error budgets to remove the artifacts. For example, while Region 3 requires a mean error budget as high as 214m to have the artifact removed, Region 2 needs 75m only. Using that coefficient as a gauge helps reduce unnecessary smoothing and hence reduce the errors needed to achieve an artifact-free surface.

4. CONCLUSION
We have presented an improved ODETLAP procedure for the automated reconstruction of artifact-free seafloor surfaces within a limited error budget. It has the smoothing factor as its only parameter, making it easier to adjust than our previous attempt which requires two parameter inputs. By smoothing an accurate reconstruction with ODETLAP, we allow terrain features to be better preserved than our previously-reported scheme that reconstructs from the given measured values directly. To automate the adjustment of the smoothness parameter, we analyze the Gaussian curvature of the reconstructed surfaces. Areas of positive Gaussian curvatures highly resemble the locations of the bumps that we observe on the shaded reconstructed surface. Increasing smoothing enlarges the small bumps and reduces their numbers along the tracklines, thus alleviating the acquisition footprint. When the coefficient of variation of such areas is around 1.3, the artifact is almost gone. We use this observation to determine the minimum smoothing needed for the artifact-free surface.

In the future, we will look into the automatic stopping criterion further. More tests will be done on a variety of trackline depth samples. Even more accurate stopping criteria will be investigated. Our current work embraces data from the tracklines but not the altimeter. It is interesting to see how data from different sources could combine.

This research was partially supported by NSF grants CMMI-0835762 and IIS-1117277.

5. REFERENCES
Figure 8: (Left) Region 3 reconstructed seafloor surfaces. (Right) Respective positive Gaussian curvature locations. The mean errors, from top to bottom, are around 43m, 86m and 214m respectively.


PhD Showcase: Mining Multi-object Spatial-temporal Movement Patterns

PhD Student: Htoo Htet Aung
h2aung@comp.nus.edu.sg
Phd Supervisor: Kian-Lee Tan
tankl@comp.nus.edu.sg
School of Computing, National University of Singapore

ABSTRACT
In this paper, we present an ongoing PhD research on mining Multi-object Spatial-temporal Movement Patterns (M-STEM Patterns) from a Trajectory Database (TJDB). Information of the M-STEM Pattern instances has numerous applications in epidemiology, ecology, location-based services, transportation, and social and behavior sciences since it supplements the information provided by a traditional GIS. We describe the research we had conducted to find instances of two M-STEM Patterns, namely the Meeting pattern and the Convoy pattern. We conclude this paper after introducing our ongoing research on discovering instances of another M-STEM pattern called Tried-and-True Route pattern.

Categories and Subject Descriptors
H.2.8 [Database Management]: Database Applications—Data mining, Spatial Databases and GIS

General Terms
Algorithms, Experimentation

Keywords
Movement Patterns, Trajectory Databases, GPS

1. INTRODUCTION
A Global Positioning System (GPS) receiver is a location-sensing device that allows its users to access time-stamped locations of the device. Advances in GPS technology enable a user to maintain a highly accurate record of the locations visited by the tracked object in high temporal resolutions. Since the GPS service was open for civilian use, GPS receivers have been installed in land and sea vehicles, wild-animals, and even civilians for various purposes. The movement data of such tracked objects are often archived in Trajectory Databases (TJDBs) and can be used to extract instances of Multi-object Spatial-temporal Movement Patterns (M-STEM Patterns).

In this paper, we will present an ongoing PhD research on Mining Multi-object Spatial-temporal Movement Patterns (M-STEM Patterns) from a Trajectory Database. We focus our research on multi-object movement patterns in contrast to single-object movement patterns for two reasons. First, while knowledge of the instances single-object movement patterns does not shed much light on the trends in objects’ movement or information of the underlying terrain, that of the instances of multi-object movement patterns, which are embedded in the TJDB, such as (a) multiple objects traveling to and meeting in a specific spatial area—a meeting, (b) multiple objects forming and moving in a group—a convoy, and (c) multiple objects using the same route frequently—a tried-and-true route—will augment the insights provided by a traditional GIS in various applications such as epidemiology, ecology, location-based services, transportation, and social and behavior sciences. For instance, an ecologist can better understand the effects of climate change on wildlife by analyzing meetings, convoys and frequently used roaming routes of wild-animals in addition to information available in a traditional GIS such as vegetation cover and temperature. Second, they are usually more computationally expensive to discover than their single-object counterparts.

There are many challenges to discover instances of Multi-object Spatial-temporal Movement Patterns. Current literature lacks experimental study on algorithms to discover instances of the meeting patterns. Moreover, for each meeting instance, its associated meeting place is not well defined. Existing works on finding instances of the convoy pattern cannot handle real-life convoys as they assume the members of the convoy would always stay together during its lifespan while, in reality, convoys members occasionally dispatch themselves from their parent groups as well as new/existing members join/leave the convoy in its life-span. It is not a trivial task to discover tried-and-true routes as, given the continuous nature of the spatial-space and time, trips using the same route rarely have matching GPS location sequences in the TJDB. Therefore, efficient and effective mining of the instances of the M-STEM Patterns from a TJDB becomes a new and interesting research problem.

Our ongoing PhD research will contribute in three areas: (a) efficient algorithms to mine meeting instances along with their associated information like meeting places; (b) a new concept of convoy pattern that can capture the nature of real-life convoys and incremental algorithms to mine such convoy instances; and (c) scalable algorithms to mine tried-and-true routes frequently used by the moving objects. The algorithms we contribute will be generic algorithms, i.e. they
will work regardless of the type and movement nature of the tracked objects. Also, their designs will not assume any availability of additional terrain information such as road networks and semantic region labels.

Some of the research works we present in this paper have been conducted and published. The works in Sect. 3 on finding meeting instances appeared in [3] while those in Sect. 4 on discovering evolving convoy instances appeared in [2].

2. RELATED WORKS

Gudmundson and Kreveld [6] defined a Meeting pattern \( meet(m, r, w) \) as a pattern formed by a set \( G \) of at least \( m \) objects staying in a fixed circle of radius \( r \) for at least \( w \) consecutive time-stamps and a Flock pattern \( flock(m, r, w) \) as a pattern formed by a set \( G \) of at least \( m \) objects staying in a moving circle of radius \( r \) for at least \( w \) consecutive time-stamps. They reported that, for parameters \( m, r \) and \( w \), and a Trajectory Database storing spatial-temporal movement of \( n \) objects for \( \tau \) time-stamps, the complexity to compute the longest-duration instance(s) of meeting pattern \( meet(m, r, w) \) is \( O\left(n^2+\tau^2\log(n)+n^2w^2\right) \) and that to compute the longest-duration instance(s) of flock pattern \( flock(m, r, w) \) is NP-Hard.

Vieira et al. [14] reported polynomial-time algorithms to find Flock instances of fixed durations \( dur = w \). In [4], Buchin et al. gave an \( O(\tau) \) algorithm to compute all similar sub-trajectories of two trajectories spanning \( \tau \) time-stamps, where similarity is defined as starting at the same time and having equal duration. Since each pair of similar sub-trajectories forms an instant of \( flock(2, r, w) \), this implies, for parameters \( m = 2, r \) and \( w \), and a Trajectory Database storing spatial-temporal movement of \( n \) objects for \( \tau \) time-stamps, the complexity to compute the longest-duration instance(s) of flock pattern \( flock(2, r, w) \) is \( O(n^2\tau) \).

Given a Trajectory Database \( R \), a distance \( r \), a set of objects \( G \) containing at least \( m \) objects, and a time-interval \( I \) spanning at least \( w \) time-stamps, whether \( G \) forms a Meeting \( meet(m, r, w) \) during \( I \) can be verified by checking if the radius of the smallest circle covering the set \( P \) of all points, which the objects in \( G \) visited during \( I \), is not longer than \( r \). Elzinga and Hearn [5] proposed Euclidean Messenger Boy algorithm (EMB) in order to calculate the smallest circle enclosing a finite set of points. EMB monotonously increases the radius of the circle until all points are covered.

In order to form a Flock pattern, in each time-stamp during its life-span, the member objects must stay within a circle not larger than the user-defined size \( r \). Therefore, the number of members each Flock instance can have is limited as a typical real-world object has a volume and a circular area of a pre-defined size can hold only a limited number of objects. Jeung et al. [8] termed this situation as lossy-flock problem and defined a Convoy as a group of at least \( m \) objects being density-connected with each other throughout \( w \) consecutive time-points, where \( m, w \), and the two DBSCAN parameters \( \epsilon \) and \( min pts = m \) are provided by the user. Since a convoy can occupy a spatial region of arbitrary size and shape in its lifetime, there is no limit to the maximum number of members in a convoy. They proposed a filter-and-refinement algorithm called CuTS to find convoys.

Kalnis et al. [9] proposed two exact algorithms and an approximate algorithm to find Moving Clusters. Clusters found in consecutive time-slices are defined to be a Moving Cluster if their Jacquard similarity is higher than a user-defined threshold \( \theta \). Moving cluster model does not have an explicit notion of members and allows members to leave or enter the cluster in its lifetime. Therefore, moving cluster permits clusters to be completely different (larger, smaller etc) in a short period of time.

In a more recent work [11], a Swarm is defined as a group of at least \( m' \) objects, which are found together for \( k \) (possibly non-consecutive) time-stamps. Therefore, the definition of the Swarm pattern allows member objects to move away from each other yet it enforces a discipline such that all members must re-group frequently-enough to form a Swarm instance. They proposed ObjectGrowth algorithm to report instances of Swarm patterns.

Figure 1 compares all existing convoy models in a two-dimensional plot. Traditional notion of convoys (flocks and convoys) neither allows members to temporarily move away from the parent convoy, nor presents convoys in different evolving stages. Moving Cluster is able to capture the different stages of a convoy but does not explicitly enforce leaving members to return. Swarm allows members to move away from the group yet it cannot capture the evolving stages of convoy in an intuitive manner. To the best of our knowledge, there is no convoy model that can both allow dynamic members and capture evolving stages of a convoy.

Lee et al. [10] proposed a clustering-based technique to discover frequent routes. Their solution includes three steps: (i) it partitions the tracks into line-segments; (ii) it clusters the line-segments; (iii) it extracts a representative route from each cluster. Since original tracks are partitioned into multiple line-segments, long frequent routes are reported as multiple disconnected representative routes extracted from multiple line-segment clusters. It requires many parameters to efficiently obtain reasonably accurate results. Sacharidis et al. [13] suggested a strategy for on-line discovery of top-k frequent routes called "hot motion paths". It requires cooperation of moving objects to compute the results.

Recently, Ying et al. [15] proposed semantic trajectory mining to predict the next action of a user based on his historical movement data. In contrast to finding which routes (sequence of locations) in the spatial-space are frequently used by multiple tracked objects, their work emphasized on predicting what the user will do next and relied on the prior knowledge of the semantic region labels.

3. FINDING CLOSED MEMOS

Informally, an instance of the meeting pattern is formed when a group of objects comes to and stays in a fixed (circular) area for a while. Information of the meeting instances can light insights into the behavior and interaction of the tracked objects under analysis. For example, from the information of the commuters, who form meeting instances...
during lunch time, advertising agencies can discover their interactions – and deduce influence on purchasing decision.

Information of the instances of the meeting pattern can also be analyzed to discover knowledge on trends of meeting places (and times). For instance, meeting places, times and frequencies of wild-animals show changes in their natural habitats and behavior. By analyzing these changes and other GIS information such as changes in vegetation cover and population density, an ecologist can better understand the effects of human-induced change on wildlife. Similarly, meetings of commuters during lunch time show trends in patron preferences and such information can be used by marketers to plan advertisements and promotions.

**Definition. 1. MEMO –** Given parameters m, r, and w, a set of objects M forms a Meeting of Moving Objects, or a “MEMO(m, r, w)”, during the time-interval I at the circular region loc if (i) M has at least m objects, (ii) I spans for at least w consecutive time-stamps, (iii) loc has a smallest-possible radius r (loc) ≤ r and (iv) all objects o ∈ M reside in loc in each t ∈ I. The instance of MEMO formed by M during I at loc will be denoted as: memo(M, I, loc).

Definition 1 formally defines a MEMO. In Fig. 2(a), $M = \{o_1, o_2, o_3, o_4\}$ forms a MEMO(3, r, 2) during $[t_1, t_3]$ at the meeting place. loc, as $M$ has more than 3 objects staying together in loc, which is smaller than a circle of radius r, for a time-interval $[t_1, t_3]$, which is longer than 2 time-stamps. Although our definition of the meeting pattern, MEMO, is similar to the one found in [6], MEMO explicitly defines the meeting place as a smallest-possible circle in contrast to a circle of fixed radius r. As a result, the solutions we are going to present in Sect. 3.1 can report more accurate and non-ambiguous meeting places (hot areas). For example, for the meeting pattern formed by three animals in Fig. 2(b), our algorithm proposed in [6] may report multiple meeting instances, each of which is associated with a less accurate circular region of radius r (three shown as dotted circles) as its meeting place. Identifying an accurate meeting place from multiple meeting instances (and meeting places) requires extra effort.

**3.1 Solutions**

**Lemma. 1. Apriori-properties of MEMOs –** All subsets of the set of objects that forms a MEMO(m, r, w) during a time-interval I also form MEMO(m − 1, r, w) during the same time-interval I.

We use Lemma 1 to adapt Apriori-algorithm [1] into our Apriori-based closed MEMO Miner (A-miner), which systematically discovers the instances of MEMO formed by $k + 1$ objects only when those formed by its subsets exist. An outline of A-miner is given in Algorithm 1. The function CI-MEMO($\mathcal{R}$, w, r, $O_k$) returns all the instances of MEMO($k, r, w$) formed by the set $O_k$ containing k objects. The A-miner initializes the instances formed by exactly one object (lines 2-5). Starting with $k = 1$, larger MEMO instances formed by $O_{k+1}$ objects is built only if two of its subsets $O_k$ and $O_k'$ form some instances (lines 6-16). In doing so, if $k \geq m$, the instances are put into the result set $\mathcal{M}$ (lines 7-9), which is finally filtered (line 17).

**Algorithm 1 Apriori-based closed MEMO Miner.**

**Input:** $\mathcal{R}$, r, m and w. **Output:** A set of closed MEMO $\mathcal{M}$.  
1: $C_1 \leftarrow \emptyset, M \leftarrow \emptyset$ and $k \leftarrow 1$
2: for all object o do
3: \hspace{1em} $L(\{o\}) \leftarrow$ CI-MEMO($\mathcal{R}$, w, r, $\{o\}$)
4: \hspace{1em} if $L(\{o\})$ is not empty then
5: \hspace{2em} $C_1 \leftarrow C_1 \cup \{\{o\}\}$
6: \hspace{1em} while $C_k \neq \emptyset$ do
7: \hspace{2em} for all $O_k \in C_k$ do
8: \hspace{3em} if $k \geq m$ then
9: \hspace{4em} $\mathcal{M} \leftarrow \mathcal{M} \cup L(O_k)$
10: \hspace{4em} $C_{k+1} \leftarrow \emptyset$
11: \hspace{3em} for all $O_k, O_k' \in C_k$ s.t. $|O_k - O_k'| = 1$ do
12: \hspace{4em} $O_{k+1} \leftarrow O_k \cup O_k'$
13: \hspace{4em} $L(O_{k+1}) \leftarrow$ CI-MEMO($\mathcal{R}$, w, r, $O_{k+1}$)
14: \hspace{4em} if $L(O_{k+1})$ is not empty then
15: \hspace{5em} $C_{k+1} \leftarrow C_{k+1} \cup \{O_{k+1}\}$
16: \hspace{4em} $k \leftarrow k + 1$
17: $\mathcal{M} \leftarrow \mathcal{M} - \{M | M$ is not a closed-MEMO$\}$.

For a large Trajectory Database containing records of a large number of tracked objects, A-miner needs a huge memory. Therefore, Equivalent Class Transformation (ECLAT), proposed in [16], is used to recursively partition the search space in our ECLAT-based close MEMO Miner (E-miner). When a partition is small enough to fit into the memory, A-miner is used to extract the closed MEMO instances.

**Lemma. 2.** if the distance between location of object $o_i$ at $t_j$ and that of object $o_j$ at $t'_j$ is further than 2r, $o_i$ and $o_j'$ do not form a MEMO at any interval I containing $t_j$ and $t_j'$.

A-miner and E-miner query the Trajectory Database for locations the tracked objects visited in order to verify if they actually form a MEMO instance. Those queries (and computation of the minimum covering circle) are often wasted when the objects do not form a MEMO. Using Lemma 2, we introduced a filtering step, which needs less access to the TJDB, into A-miner to obtain Filter-And-Refinement-based Closed MEMO Miner (FAR-miner). The filtering step in FAR-miner helps to avoid wasted computation.
4. DISCOVERY OF EVOLVING CONVOYS

Knowledge of convoys has many practical applications ranging from traffic planning to wild-life research. Traffic planners can benefit from knowledge of truck-convoys moving between factories, warehouses and stores. Convoy discovery can be used to extract complex herding behaviours of wild animals from GPS-collar data.

**Definition 3.** *Traditional Convoy* – Given parameters $m$ and $w$, a set $C$ of at least $m$ objects forms a traditional convoy if all objects in $C$ are within proximity of each other for $w$ or longer period of time.

Definition 3 serves as a template for existing convoy definitions [6, 8], which only differ in the definition of spatial proximity. The concept of evolving convoys we will discuss applies to convoys defined using any spatial proximity. However, we focus our studies on convoys formed from density-connected objects because defining the spatial-proximity as density-connection is more relevant for real-world objects.

Although Def. 3 is intuitive, it mandates members of a convoy to be always together during its life-span. This notion of convoys having only persistent members does not reflect and cannot cope with the following real-life scenarios:

(i) Some members of the convoy may temporarily leave the group (and return); (ii) Some members may join/leave the convoy later/earlier than the convoy’s starting/ending time, i.e. the convoy may evolve into a larger/smaller convoy.

4.1 Solutions

The solution to the problem of finding instances of the convoy pattern has two parts: (i) A new model of convoy – since traditional convoy models cannot cope with the nature of real-life convoys, we propose a new definition of convoy called, *Evolving Convoy* (ECOCO); (ii) Algorithms to find EVOCOs – we developed three incremental algorithms that can be used with either offline or streaming Trajectory Databases.

4.1.1 Definition of Evolving Convos.

**Definition 4.** *w-Convoy* – For given parameters: $m$, $k$, and $w$, a set of objects $D$ forms a w-Convoy during a time-interval $I$ if (a) $D$ contains at least $m$ persistent members (denoted by $PM_D$), all of which are density-connected in each time-stamp $t \in I$ (b) $D$ contains (possibly zero) dynamic members (denoted by $DM_D$), each of which must be density-connected to $PM_D$ in at least $k$ time-stamps during $I$ and (c) the time-span $I$ spans for exactly $w$ time-stamps.

Definition 4 introduces flexibility to Def. 3 by defining a discipline for dynamic members in $DM_D$ concerning leaving and returning their parent convoy. The first condition ensures that a fixed set of the persistent members forms its main body. The second condition requires each dynamic member to stay close with the set of persistent members frequently enough. For smaller $k$ values, a dynamic-member can move away from the w-Convoy for a longer period while larger $k$ values prohibit a dynamic-member from being away from its parent w-Convoy for a long time.

**Definition 5.** *Evolution of w-convoy* – A w-convoy $D$ that exists from $t$ to $t + w - 1$ evolves into another w-convoy $D'$ that exists from $t + 1$ to $t + w$ if they have at least $m$ common persistent members, i.e. $|PM_D \cap PM_{D'}| \geq m$.

**Definition 6.** *EVOCO* – For given parameters $m$, $k$, and $w$, assume a sequence of w-convoys, $D = D_1, D_2, ..., D_z$ such that each $D_i$ evolves into $D_{i+1}$ for $1 \leq i < z$, there is no $D_i$ evolving into $D_1$, and $D_z$ does not evolve into any $D_{z+1}$. The sequence $D$ forms an* Evolving Convoy $V$ having multiple stages, where each stage is defined as the sub-sequence of w-Convoys having the same members.

Definition 5 defines how a w-convoy can evolve into the next w-convoy. Definition 6 ensures each maximal sequence of evolving w-Convoys to be covered in an Evolving Convoy (EVOCO) with stages. In this way, EVOCOs can gain/lose new-existing members throughout its life-span as w-Convoys can evolve into other w-Convoys with different set of (dynamic) members.

**Definition 7.** *Discovery of EVOCOs* – Given a Trajectory Database $\mathcal{R}$, DBSCAN parameters $\varepsilon$ and $\minpts$, and constraints $m$, $k$, and $w$, Discovery of EVOCOs (DEC) is to list all instances of EVOCOs.

4.1.2 Algorithms to find Evolving Convos

The first algorithm called Simple Slice-by-Slice algorithm, $S^3$, is directly obtained from the problem definition and is similar to MC2 in [9] and CMC in [8]. Details of $S^3$ is shown in Algorithm 2. EXTEND is a function that assigns different roles to each member $o$ for a convoy (based on the number of times it stays together with the convoy) and maintains a log of stages for each convoys to be outputted. For each time-stamp $t$, $S^3$ clusters all the objects in $\mathcal{R}$ according to their locations at $t$ using DBSCAN (line 3). $S^3$ tries to match each of the current convoys maintained in $\mathcal{V}_{\text{cur}}$ with the clusters found in the current time-stamp $\mathcal{L}$ (lines 4-8). If $m$ or more persistent-members of a convoy $V$ is found in a cluster $C$, $V$ is matched to $C$ – we say $V$ “extends” to $C$. For each evolving convoy $V$ matched to a cluster $L$, the roles of its member objects are maintained by EXTEND($V, L$) (line 7). When a matching cluster cannot be found, the convoy is put in the results if its life-span is at least $w$ (lines 9-12). Those clusters $L$, into which no convoy has extended, are made potential convoys and placed in $\mathcal{V}_{\text{cur}}$ (lines 13-16).

In $S^3$, all objects in each time-stamp are clustered using DBSCAN, which is an expensive operation. Therefore, for better performance, we need to minimize DBSCAN calls. TRAJ-DBSCAN (DBSCAN for trajectories) proposed in [8] uses the closest distance between each pair of trajectories as their distance. In order to prune objects which may not form a cluster at $t$, we borrowed TRAJ-DBSCAN to check the trajectory-clusters for trajectories in a time-partition containing $t$.

Our proposed Interleaved-DEC (ID) algorithms divide the dataset into partitions, each containing $\lambda$ consecutive time-stamps. For each partition, the ID algorithms operate in two steps – the first is to get the set of objects which likely to form convoys while the second is actual clustering of objects and extending of the convoys. ID algorithms, therefore, interleaves the two steps (hence their name) as they progress. The partition-length ($\lambda$) can be set independently.

The first interleaving algorithm, ID-1, is a simple extension of $S^3$. ID-1 brings performance improvement over $S^3$ by clustering a handful of objects each time-stamp. We developed another interleaving algorithm ID-2 to have tighter pruning than ID-1 by exploiting a heuristics.
5. MINING TRIED-AND-TRUE ROUTES

The knowledge of the routes frequently taken – or Tried-and-True Routes (TNT Routes) – is useful in many applications: the tried and true paths of local pedestrians can be used to suggest safe walking routes for tourists; tracks frequented by wild-animals can help scientists better understand their movement behaviours.

A TNT Route can be loosely defined as a track, along which the entities in question have travelled at least a number of times (a user-defined frequency threshold). Note that a single trip (an entity took) can involve multiple TNT Routes and non-TNT Routes. Given a Trajectory Database and a frequency threshold, finding TNT Routes is a challenging task since the spatial space is continuous and there is an infinite number of potential tracks.

**Definition 8.** Track-Clique – Given parameters m, l and r, a set of tracks J forms a track-clique if (i) each track j ∈ J is within r Frechet distance away from all tracks j’ ∈ J, (ii) the lengths of all tracks j ∈ J are longer than l and (iii) J contains at least m non-overlapping tracks.

Definition 8 defines a track-clique which includes m unique tracks, which are at least l units in length (not duration). The Frechet distance ensures tracks in the same clique are spatially close, similar in shape, and similar in direction. Frechet distance is invariant in speed, i.e. if the tracks are similar in shape and direction, they belong to the same track-clique even though the corresponding trips were taken at different speeds. In other words, a track-clique groups the tracks, which are on the same route. Therefore, a track-clique containing m tracks correspond to a Tried-and-True Route taken at least m times.

**Definition 9.** TNT Route – A track-clique is closed if there is no other track-clique covering it. The longest track of a track-clique is defined as the corresponding Tried-aNd-True Route (TNT Route) of the closed track-clique.

We are currently developing algorithms to discover all TNT Routes from a given Trajectory Database and user-defined parameters m, l and r. We expect to report efficient solutions in a forthcoming paper.

6. EXPERIMENTS

We conducted experiments to assess the performance of solutions we presented to list all MEMO and EVOCO instances. We give a summary of the outcomes of the conducted experiments here. For more complete sets of experiments Refer to [3] and [2]. Experiments of our proposed solutions for mining Tried-and-True Routes is underway and will be reported in a forthcoming paper.

We used five human movement datasets [12] – Statefair, Orlando, New York, NCSU and KAIST – to assess performance of A-Miner, E-Miner and FAR-Miner against the column-sweeping algorithm (CS-miner) proposed in [6]. For assessing S$^3$ and ID-family, along with X-CuTS, which is an extension of CuTS [8], we used larger datasets – Mob and Bus. Mob is obtained from merging the five human movement datasets [12], while Bus is constructed from a subset (from 30-Oct-2001 to 05-Nov-2001) of the data available in [7] by merging the bus movements during peak hours (0800-1600 hours) to obtain 4, 471 trajectories.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No. of Objects</th>
<th>Covers</th>
<th>No. of Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statefair</td>
<td>19</td>
<td>3hr</td>
<td>17,545</td>
</tr>
<tr>
<td>Orlando</td>
<td>41</td>
<td>14hr</td>
<td>133,076</td>
</tr>
<tr>
<td>New York</td>
<td>39</td>
<td>22hr</td>
<td>118,584</td>
</tr>
<tr>
<td>NCSU</td>
<td>35</td>
<td>21hr</td>
<td>128,417</td>
</tr>
<tr>
<td>KAIST</td>
<td>92</td>
<td>23hr</td>
<td>404,981</td>
</tr>
<tr>
<td>Mob</td>
<td>559</td>
<td>9hr</td>
<td>267,459</td>
</tr>
<tr>
<td>Bus</td>
<td>4,471</td>
<td>8hr</td>
<td>1,000,579</td>
</tr>
</tbody>
</table>

The outcome of the experiments, comparing the performance of the MEMO mining algorithms on human movement datasets, is shown in Fig. 3 (the y-axis is in log-scale). We were looking for meetings of at least two people (m = 2) lasting for at least 15 minutes (w = 15 minutes), which were reasonable choices of parameters for the corresponding datasets. In NCSU and KAIST, we even discovered meetings of up to 3 and 5 students. Our proposed data-driven algorithms, A-miner and E-miner, ran faster than CS-miner to find the closed MEMOs as they ignore the fast-moving objects in building MEMO instances while CS-miner attempts to build MEMOs containing them. FAR-miner outperforms A-miner and E-miner by a large order of magnitude due to its cheap pruning mechanism.

Table 2 shows a comparison of running time (in seconds) of the algorithms to find evolving convoys for each dataset. We found 10 and 153 convoys from Mob and Bus respectively. ID family (ID-1 and ID-2) always out-performs $S^3$ and X-CuTS. In general, ID-2 is faster than ID-1. $S^3$ runs faster than X-CuTS in Bus datasets. In all cases, it performs worse than ID-family due to the limitation it has on parameter λ. Therefore, we omit its results in further discussions.

More experiments are conducted using synthetic datasets (with 7.5K, 10K and 12.5K objects each) to assess how the algorithms scale on different sizes of data. Figure 4 shows the running time of each algorithm. ID algorithms outperform $S^3$ when the dataset contains more than 7,500 objects.
Figure 3: Performance of the algorithms on various human-movement datasets using $r = 10$ meters.

Table 2: Comparison of the Algorithms.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$S^3$</th>
<th>ID-1</th>
<th>ID-2</th>
<th>X-CuTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mob</td>
<td>174.39</td>
<td>137.41</td>
<td>113.00</td>
<td>156.959</td>
</tr>
<tr>
<td>Bus</td>
<td>1843.88</td>
<td>1765.95</td>
<td>1423.24</td>
<td>2470.83</td>
</tr>
</tbody>
</table>

We found that ID-2 performs best and scales much better than $S^3$ and ID-1.

Figure 4: Performance of the Algorithms on Synthetic Datasets.

7. CONCLUSION

In this paper, we proposed a thesis on “Mining Multi-object Spatial-temporal Movement Patterns”. We had conducted research on two such patterns – (a) meeting patterns and (b) evolving convoy patterns. We had developed three algorithms to find meeting instances in a Trajectory Database. We had defined evolving convoy pattern, which is able to capture different stages of a convoy’s evolution while allowing members to temporarily leave the parent convoy. We presented three algorithms to extract instances of evolving convoys from a TJDB. We are currently trying to develop a novel solution to extract Tried-and-True Routes.

8. REFERENCES

Knowledge Acquisition about Places
Stephan Winter\textsuperscript{a}, Krzysztof Janowicz\textsuperscript{b}, Kai-Florian Richter\textsuperscript{a}, Maria Vasardani\textsuperscript{a}

\textsuperscript{a}Department of Infrastructure Engineering, The University of Melbourne, Australia
\textsuperscript{b}Department of Geography, University of California at Santa Barbara, CA

Place, a concept originating from human geography \cite{1,2}, has recently become a hot topic in GIScience: place is important in human cognition and communication, and hence, is a high priority for human-computer interaction research. But place is also a challenging concept to model, reason with, and analyze in information systems, because of its fluency with context shifts, and its underspecification \cite{3}. Thinking about knowledge acquisition methods—from geographic information retrieval from big data to dedicated user generated content—they seem to involve several disciplines such as computational linguistics, data mining, artificial intelligence, and geographic information science.

The participants of the Workshop on Place-related Knowledge Acquisition Research were coming from these disciplines, accordingly. The workshop was held on 31 August 2012 in conjunction with the 8\textsuperscript{th} International Conference on Spatial Cognition, Germany. The workshop was jointly organized by three research teams from around the world: the University of Melbourne (Talking about Place), the University of California at Santa Barbara, and University of Edinburgh (SpaceBook), and attracted about 25 participants.

In the keynote, Ross Purves (U Zurich) laid out the challenges, but also pathways to approach place as a concept to be captured for databases and used in reasoning and human interaction. He especially was searching for useful definitions of place that allow to progress in formal knowledge acquisition methods, and adopted for this purpose the Panovsky-Shatford facet matrix by merging it with John Agnew’s conditions for placeness: (a) an identifiable location, in relation to everywhere else, (b) a locale, which is the actual shape of the space, and (c) a sense of place, which is formed by personal and emotional attachment of people to a place.

A number of (peer reviewed) position papers answered to the challenge set out by the workshop, which had been to “discuss methods to automatically estimate the location of things or events based on verbal or graphical descriptions, or photographs, or a combination of them”. Themes covered observations such as collections of text corpora, or big data, or attention and emotion mapping, analysis such as classification and pattern analysis, or semantic analysis, and applications such as guides for navigation, locating things or events, or mapping. The workshop proceedings are openly accessible from \url{http://ceur-ws.org/Vol-881}.

Thus inspired the participants developed in groups three “research proposals”, or approaches to further pursue place-related knowledge acquisition research, profiting from the interdisciplinary mix in the audience. The approaches were: (A) capturing signatures of places, (B) crowd-sourced place descriptions, and (C) context-dependent strategic navigation advice. A short description of the projects follows:
(A) SIM²PLE: Signatures from Multi-Modal Place Extraction (using big data)

The aim of the project is to calculate signatures (e.g., feature vectors, even though that is not necessarily the best representation) for places. These signatures are best on multiple modalities, e.g., visual information about the place, (typical) sounds at that place, and potentially even olfactory information. These signatures would allow answering how much change to the characteristics of a place might happen for it to still be the same place (or even a place). This would enable the maintenance of place databases, and more generally allow for predictions of 'placeness' for a location.

(B) Analysing Crowdsourced Natural Language Place Descriptions

The project sets out to examine two interrelated questions: (i) How do people produce descriptions in vista space in real-world urban contexts for different target audiences? and (ii) What are the more and less effective ways in which they do it, relative to a target audience? A modified version of the geocaching game is used to collect data, in which people are asked to produce: (a) a description of the place where a geocash is found, and (b) of the route for someone else to find it. The results of the data analysis are useful for a number of applications, such as personalised image retrieval and image annotation, and are pushing the boundaries of current natural language (NL) automated generation technology. The analysis would also lend insights into how people perceive places (cognitive dimensions) and how they construct such descriptions from a linguistic aspect. Finally, the results have the potential of generating more semantics for data, in the current data rich but semantic poor era.

(C) User-specific Contextual Landmark Selection for Navigation

This approach started from the observation that landmarks frequently have a sweet side, from where they are easily recognizable even by strangers, but can be used in route instructions for locals also when they approach from directions where the landmark is not easily recognizable. Expanding on this thought we identify the need for context-dependent selection of landmarks in strategic navigation, focusing on two contexts: the one of the landmark, such as its appearance at the time of the day or the season, its efficacy from all directions, or its cultural significance, and the one from the user, where mobility, gender, age, education, hometown and other socio-demographic properties are considered for computing the cultural preferences of users.

References

The ACM Special Interest Group on Spatial Information (SIGSPATIAL) addresses issues related to the acquisition, management, and processing of spatially-related information with a focus on algorithmic, geometric, and visual considerations. The scope includes, but is not limited to, geographic information systems (GIS).

The Association for Computing Machinery (ACM) is an educational and scientific computing society which works to advance computing as a science and a profession. Benefits include subscriptions to Communications of the ACM, MemberNet, TechNews and CareerNews, plus full access to the Guide to Computing Literature, full and unlimited access to thousands of online courses and books, discounts on conferences and the option to subscribe to the ACM Digital Library.

- SIGSPATIAL (ACM Member) .................................................. $ 15
- SIGSPATIAL (ACM Student Member & Non-ACM Student Member) ........................................... $6
- SIGSPATIAL (Non-ACM Member) ........................................... $ 15
- ACM Professional Membership ($99) & SIGSPATIAL ($15) ................................................... $114
- ACM Professional Membership ($99) & SIGSPATIAL ($15) & ACM Digital Library ($99) ........ $213
- ACM Student Membership ($19) & SIGSPATIAL ($6) ........................................................ $ 25
- Expedited Air for Communications of the ACM (outside N. America) ...................................... $ 50

The ACM occasionally makes its mailing list available to computer-related organizations, educational institutions and sister societies. All email addresses remain strictly confidential. Check one of the following if you wish to restrict the use of your name:

- ACM announcements only
- ACM and other sister society announcements
- ACM subscription and renewal notices only

Questions? Contact:
ACM Headquarters
2 Penn Plaza, Suite 701
New York, NY 10121-0701
voice: 212-626-0500
fax: 212-944-1318
email: acmhelp@acm.org

Remit to:
ACM
General Post Office
P.O. Box 30777
New York, NY 10087-0777

www.acm.org/joinsigs

Advancing Computing as a Science & Profession
The SIGSPATIAL Special