The SIGSPATIAL Special

Newsletter of the Association for Computing Machinery
Special Interest Group on Spatial Information

Volume 8 Number 3 November 2016

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.

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 chiychow@cityu.edu.hk.

The SIGSPATIAL Special (ISSN 1946-7729) Volume 8, Number 3, November 2016.

Table of Contents

Message from the Editor

Chi-Yin Chow
Department of Computer Science, City University of Hong Kong, Hong Kong
Email: chiychow@cityu.edu.hk

In the first section, we have the top four papers selected for the 1st ACM SIGSPATIAL Student Research Competition (SRC) held at the ACM SIGSPATIAL 2016. The SRC chair is Prof. Moustafa Youssef (Egypt-Japan University of Science and Technology).

The second section consists of two event reports from:

1. The 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems (ACM SIGSPATIAL MobiGIS 2016)
2. The 7th ACM SIGSPATIAL International Workshop on GeoStreaming (ACM SIGSPATIAL IWGS 2016)

I would like to sincerely thank all the 1st ACM SIGSPATIAL SRC authors, SRC chair (Prof. Moustafa Youssef), and event organizers for their generous contributions of time and effort that made this issue possible. I hope that you will find the newsletters interesting and informative and that you will enjoy this issue.

You can download all Special issues from:

`http://www.sigspatial.org/sigspatial-special`

.

The SIGSPATIAL Special

Section 1: The 1st ACM SIGSPATIAL Student Research Competition (SRC)

ACM SIGSPATIAL
http://www.sigspatial.org

ACM SIGSPATIAL 2016 Students Research Competition Report
San Francisco, USA - October 31, 2016

Moustafa Youssef
Department of Computer Science and Engineering
Egypt-Japan University of Science and Technology (E-JUST)
(Student Research Competition Chair)

For the first time, the ACM SIGSPATIAL 201 hosted the SIGSPATIAL ACM Student Research Competition (SRC) this year. SRC allows undergraduate and graduate students to share their research results and exchange ideas with other students, judges, and conference attendees; understand the practical applications of their research; perfect their communication skills; and receive prizes and gain recognition from ACM and the greater computing community.

Student Research Competition winners were selected in three phases: In the first phase, a two-page abstract was judged based on novelty, impact, approach, results, and contributions to the field of spatial systems and algorithms. Selected competitors prepared a poster for demonstrating their work during the conference in the second phase. Those selected for further competition at the final phase gave a short talk about their research project in front of the judging committee and conference attendees. All SRC participants at the conference received support to cover their travel to the conference. Three graduate category and one undergraduate category winners of the SIGSPTIAL 2016 SRC were announced at the conference banquet and received certificates, medals, as well as monetary awards from the ACM. In addition, the top winner from each category will advance to the SRC Grand Finals, where winners from various ACM SIGs are evaluated to nominate the ACM-wide SRC winners. The winners of the Grand Finals will be recognized at the Annual ACM Awards Banquet, the same banquet that also recognizes the Turing Award winners.

The winning entries cover different areas of interest to the SIGSPATIAL community including computational steering for geosimulations, accelerating the calculation of the minimum set of viewpoints for maximum coverage over digital elevation model data, dynamic indoor navigation, and city-scale mapping of pets using georeferenced images.

I would like to thank all the authors of papers and the SRC judging committee for their professional evaluation and help in the three phases of the competition. A special thanks goes to Microsoft Research for supporting the SRC across the different ACM SIGs. Finally, I hope that the first Student Research Competition will inspire new research ideas and encourage further participations from all students working in areas relevant to the SIGSPATIAL community.

SRC: Accelerating the Calculation of Minimum Set of Viewpoints for Maximum Coverage over Digital Elevation Model Data by Hybrid Computer Architecture and Systems

Chenggang Lai, Miaoqing Huang
CSCE Department
University of Arkansas
Fayetteville, Arkansas 72701, USA
Email: {cl004,mqhuang}@uark.edu

Xuan Shi
Department of Geosciences
University of Arkansas
Fayetteville, AR 72701, USA
Email: xuanshi@uark.edu

ABSTRACT
This paper introduces how to accelerate the calculation of the minimum set of viewpoints for the maximum coverage over digital elevation model data using Intel's Xeon Phi and a computer cluster equipped with Intel's Many-Integrated-Core (MIC) coprocessors. This data and computation intensive process consists of a series of geocomputation tasks, including 1) the automatic generation of control viewpoints through map algebra calculation and hydrological modeling approaches; 2) the creation of the joint viewshed derived from the viewshed of all viewpoints to establish the maximum viewshed coverage of the given digital elevation model (DEM) data; and 3) the identification of a minimum set of viewpoints that cover the maximum terrain area of the joint viewshed. The parallel implementation on the hybrid computer cluster was able to achieve more than 100x performance speedup in comparison to the sequential implementation. The outcome of the computation has broad societal impacts since the research questions and solutions can be applied to real-world applications and decision-making practice.

1. INTRODUCTION
Identifying a minimum set of observational viewpoints that can cover the maximum area of a given terrain has high values in many applications including civil engineering, infrastructure optimization and management, and military operations. Theoretically this minimum set problem can be elaborated as given a set U of n elements, and a collection I = [S_1, S_2, ..., S_m] of m subsets of U such that the union of S equals U. The set cover problem is to identify the smallest subset of S whose union covers U. Such an optimization problem is NP-hard [1]. More formally, no polynominal solution has been identified for such a set coverage problem.

Approximate solutions can be explored using heuristic strategies, which typically take a very long process. In order to reduce the computation time, we use computer clusters with coprocessors/accelerators to parallelize the application. To achieve the goal of this research, three computation tasks have to be implemented. *Firstly*, for any given DEM data, all potential control viewpoints will be extracted automatically through map algebra calculation and hydrological modeling approaches. *Secondly*, the viewshed calculation has to be implemented on each viewpoint to generate the joint viewshed of all viewpoints to establish the maximum viewshed coverage of the given DEM. The R3 [2] and the sweep line [3] algorithms are implemented in this study. *Thirdly*, the minimum set coverage computation is to derive the minimum set of viewpoints that have their joint viewshed equals to the maximum coverage, as shown in Algorithm 1.

Algorithm 1. Finding a Minimum Set of Viewpoints for the Maximum Coverage of Digital Terrain.

```
Initialize the solution set S to empty;
while (joint viewshed criterion is not satisfied) do
    for (each viewshed Pi in the potential points P) do
        Compute its overlap fractions and Euclidean distances
    between viewpoints and S;
        if (overlap fractions > overlap criterion or Euclidean
    distances < distance criterion)
            Pi cannot be added to S;
        Calculate the joint coverage;
        if (joint coverage > maximum joint coverage)
            Maximum joint coverage := joint coverage;
    end for;
    Add Pi to S and remove it from P;
end while;
```

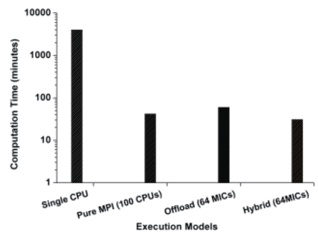
2. METHODS AND EXPERIMENTS
We conducted our experiments on two platforms, the NSF sponsored Arkansas High Performance Computing Center (AHPCC) computer cluster, which is a CPU cluster (i.e., Xeon E5-2670 8-core 2.6 GHz processors), and Beacon supercomputer, which is a hybrid cluster containing both CPUs (i.e., Intel Xeon E5-2670 8-core 2.6 GHz processors) and Intel MIC coprocessors (i.e., Intel Xeon Phi 5110P). Task 1 (i.e., identifying control points) runs on the CPU sequentially and only takes 1 minute. Both Task 2 (i.e., viewshed computation) and Task 3 (i.e., finding the minimum set of viewpoints) are time-consuming and thus have to be parallelized. The parallel solutions are implemented using the following three models.

- MPI: AHPCC cluster is employed for the viewshed calculation and the parallel minimum set calculation through MPI commands. In this implementation, a single-thread MPI process is directly executed on a CPU core. We used sweep line algorithm for viewshed calculation on CPU.
- MIC+Offload: In this model, the MPI processes are hosted on the CPU cores, which offload the computation including data to the MIC processors on Beacon. The host MPI process on CPU issues multiple threads to the MIC card using OpenMP so that each thread works on one or more coefficient vectors depending on the number of participating MIC cards. The computation tasks are done on MIC processors, while CPU cores just wait for the results. We used R3 algorithm for viewshed calculation on

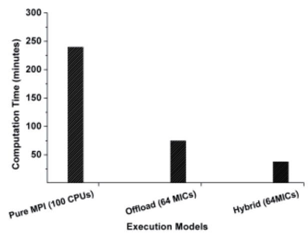
MIC cards as sweep line algorithm has data dependency issue. Algorithm 1 is applied for the minimum set calculation.

- MIC+Hybrid: In this model, both CPUs and MICs are utilized for data processing on Beacon. First the workload is distributed to CPUs through MPI. Then a host CPU will offload part of the workload to a MIC card using OpenMP. On the host CPU, we also use OpenMP to spawn multiple threads for parallel processing. The R3 algorithm is used for viewshed calculation, while Algorithm 1 is applied for the minimum set calculation.

3. RESULTS AND DISCUSSION

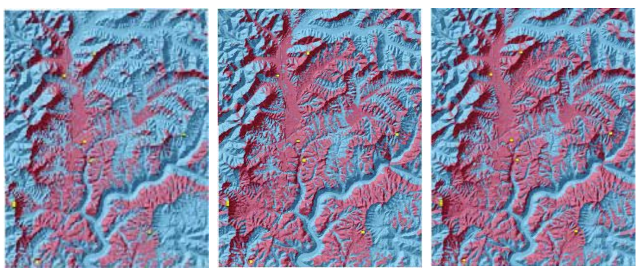


(a) Task 2: viewshed computation.

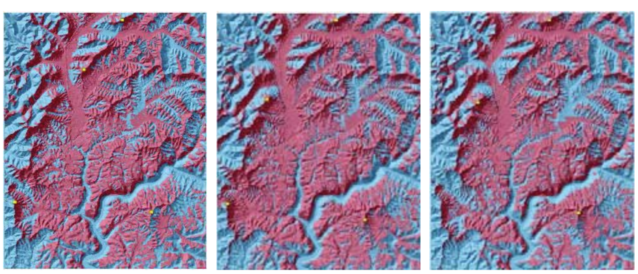


(b) Task 3: minimum set calculation.

Figure 1. Performance comparisons.



(a) 1 foot, 10 FCC antennas, 33% coverage (b) 30 feet, 10 FCC antennas, 46% coverage (c) 60 feet, 10 FCC antennas, 49% coverage



(d) 1 foot, 4 viewpoints, 50% coverage (e) 30 feet, 4 viewpoints, 50% coverage (f) 60 feet, 3 viewpoints, 50% coverage

Figure 2. Visibility coverage at different offset heights.

The 3-meter resolution DEM data for West Virginia is used in this study. Federal Communications Commission's antenna data are used for validation and comparison. Although multiple DEM tiles were tested in this experiment, only the result on *Summersville* is reported. This DEM tile has 3,646 × 4,626 pixels, in which 10 FCC antennas are installed in this DEM tile. 4,106 viewpoints are automatically derived from Task 1 calculation. The joint viewshed of these 4,106 viewpoints can cover about 99.5% of this tile of DEM. The remaining task is to identify the minimum set of 4,106 viewpoints that can cover the same joint viewshed area.

The performance of Task 2 (i.e., viewshed computation) under different execution models are shown in Figure 1(a). For the offload model, each host CPU will host one MPI process, which offloads the computation including data to the MIC coprocessors. We schedule 240 threads to a MIC card. For the hybrid model, both CPUs and MICs are allocated for data processing. We run 4 threads on the host CPU and evenly divide the workload between a host CPU and its corresponding MIC coprocessor. We also schedule 240 threads to a MIC card. From the result, the hybrid model has the best performance. However, the performance of pure MPI model (100 CPUs on AHPCC) is better than that of offload model, since the sweep line algorithm is more efficient than R3 and there is about 10 times performance difference between them. From Figure 1(b) we can see that the execute time of minimum set implementations under offload model and hybrid model are much shorter than the time under the pure MPI model.

The proposed workflow successfully derives the minimum set of 1,217 viewpoints to achieve the goal of the maximum coverage of 100%, which means the selected minimum set of viewpoints can achieve the same coverage of 4,106 viewpoints on the tile of Summersville DEM. Obviously such a minimum set still contains a large number of viewpoints because many single cells in the DEM grid can only be seen by one viewpoint. When the criteria of maximum coverage are changed, the number of minimum set can be reduced significantly.

Figures 2(a)-2(c) display the visibility coverage of current locations of FCC antennas at different offset heights. Even when the height of the antennas is set to 60 feet, these 10 FCC antennas can only cover 49.74% of this area. Figures 2(d)-2(f) display the result derived from the minimum set calculation. Only 4 antennas are required to cover 50% of the area even when the offset height is set to 1 foot, or only 3 antennas are required when the offset height is set to 60 feet.

4. CONCLUSIONS

While a few relevant works [4] were conducted in the past decades, we resume this challenging research on generating a minimum set of viewpoints for the maximum coverage over large-scale digital terrain data. The comprehensive workflow has been implemented and validated with satisfactory results in comparison to the current locations of FCC antennas. The computational bottleneck of the proposed workflow mainly lies in viewshed/joint viewshed calculation, counting visible pixels, computing the ratio of overlaid viewshed, and minimum set calculation. Although deploying CPU clusters can help reduce the computational time, modern accelerator technologies can achieve better efficiency and scalability when large volumes of high-resolution DEM data are to be processed.

5. REFERENCES

[1] Feige, U. 1996. A threshold of ln n for approximating set cover. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 314-318. ACM

[2] Sorensen, P.A. and Lanter, D.P. 1993. Two algorithms for determining partial visibility and reducing data structure induced error in viewshed analysis. *Photogrammetric Engineering and Remote Sensing*. 59(7):1149-1160.

[3] Warn, S. 2011. High performance geospatial analysis on emerging parallel architectures. PhD Thesis, U. of Arkansas.

[4] Franklin, W. R. 2002. Siting Observers on Terrain. In Advances in Spatial Data Handling: 10th International Symposium on Spatial Data Handling, edited by D. Richardson and P. van Oosterom, 109–120.

SRC: City-Scale Mapping of Pets Using Georeferenced Images

Aaron San Jose and Eduardo Hernandez (Mentor: Shawn Newsam)
University of California, Merced
5200 N. Lake Rd., Merced, CA 95343
[asanjose,ehernandez58]@ucmerced.edu

ABSTRACT

We investigate the mapping of pet activity using social media. Specifically, we perform cat and dog detection in a large collection of georeferenced images in San Francisco. We compare detection based on keyword search in user-supplied tags to detection based on image content using state-of-the-art deep-learning classification methods. The resulting city-scale spatial distribution of cat and dog activity makes sense based on our knowledge of the region. Our approach represents a general framework for mapping phenomena that are difficult to observe through traditional means.

CCS Concepts

•Information systems → **Geographic information systems;** **•Computing methodologies** → *Object detection;*

Keywords

Georeferenced images, geographic knowledge discovery

1. INTRODUCTION

There is a plethora of untapped data in Internet social media feeds that could be used to answer various interesting questions. For example, images uploaded on social media feeds are frequently of pets. Given a large number of such images, with their locations, one should be able to map where pets are. This is the focus of our project. We perform pet detection in a large number of georeferenced social media images. Mapping these detections allows us to analyze spatial trends of pet activity at a city scale.

The key technical challenge is automating the detection. We investigate two different approaches to this problem 1) applying text-based search algorithms to the user-submitted tag descriptors of the images, and 2) applying computer vision classification algorithms to the actual image content.

2. APPROACH

We seek to label each image as containing a dog or a cat. We then assign this detection to the location of the image in order to perform the spatial analysis.

2.1 Text-Based Detection

In order to classify the images through their text tags, we use keyword search algorithms. Each image has a varying number of text tags that have been provided by the user. If our search term, for example "dog", matches any of the tags, we mark the image as a detection. Text tags do not necessarily describe exactly what is in the image or completely prove that a pet is in the photo or not, though.

2.2 Image-Based Detection

Deep learning is recent, effective method of image classification that creates models based off of "learned features" of a visual class using convolutional neural networks (CNNs). CNNs are trained to recognize classes in a supervised fashion. A model is learned by feeding it labeled images. It can then be used to perform detection in unseen images. The training process involves tuning layers of neurons that perform simple tasks, like image convolution or subsampling, that culminate in a larger task, like image classification.

CNNs are useful as non-binary classifiers, or classifiers with multiple classes. During prediction, a CNN will produce a vector of size N, where N is the number of classes, of the probabilities that the image belongs to each class. A CNN usually will normalize the probabilities so they sum to one using a softmax function, and then return an encoding that gives the label for the class with the highest probability. The actual return value is another vector of size N that contains all zero values except for one index that holds the value one. This index indicates the predicted class.

We apply a CNN that has been trained to recognize a large number of visual classes including dogs and cats. This also includes specific breeds.

3. EXPERIMENTAL RESULTS

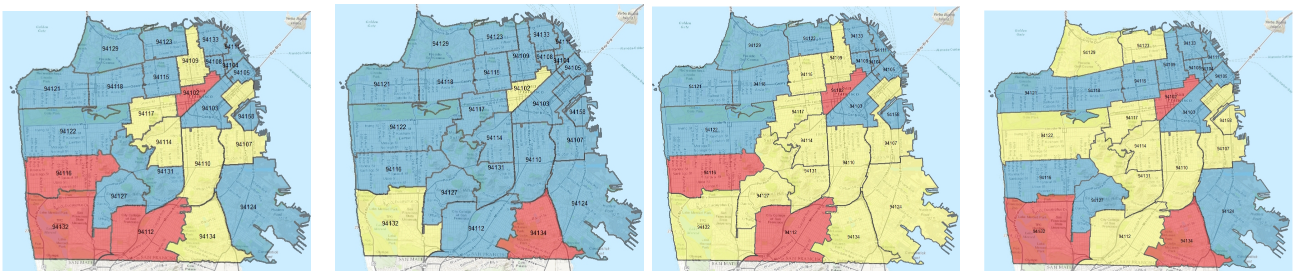
We applied the proposed approach to over one million Flickr images of San Francisco taken between 2008 and 2015.

For the text-based detection, we also used specific dog breeds (e.g., "terrier", "hound") and dog synonyms ("canine") as keywords to detect dogs. Similarly, we used specific cat breeds (e.g, "Siamese", "Egyptian") and cat synonyms ("feline") for detecting cats. Rows two and three of Table 1 show the number of cat and dog detections per year based on performing keyword search on the user-provided tags.

For the image-based detection, we used a CNN called Inception-v3 [2] that has been trained on the ImageNet Large

Table 1: Rows two through five indicate the number of detections. The last row shows the total images.

Approach	2008	2009	2010	2011	2012	2013	2014	2015	Total
Text Cat	217	272	78	235	128	177	217	177	1501
Text Dog	624	537	196	722	323	456	252	199	3309
Image Cat	258	313	79	426	240	464	351	396	2527
Image Dog	735	941	228	1554	891	1486	901	1069	7805
Total Photos	158720	184742	61942	202154	116826	195390	143824	72784	1136382



(a) Text Cat (b) Text Dog (c) Image Cat (d) Image Dog

Figure 1: Per-zip code activity mapping. Red, yellow, and blue indicate high, medium, and low activity. (a) and (b) are the results of text-based detection and (c) and (d) are of image-based.

Scale Visualization Dataset [1]. This dataset contains 1000 different classes, including various breeds of cats and dogs. The Inception-v3 CNN returns the five most likely classes for an image and we mark a detection if any of these five classes are related to cats or dogs. This network has been shown to be very effective, achieving an error rate of just 3.46% for the top five predictions [2] on the 1000 class ImageNet data set. Rows four and five of Table 1 show the number of cat and dog detections per year in our data set based on image content.

We aggregated the detections by zip code to perform our spatial analysis. We calculated a pet activity value for each zip code by normalizing the number of detections by the total number of Flickr images in that zip code. Fig. 1 shows the resulting maps where each of the 26 zip codes is labeled as having low (blue), medium (yellow), or high (red) activity.

4. DISCUSSION

We do not have a ground truth to evaluate our results. However, we make the following observations based on Table 1 and Fig. 1.

Our image-based method results in over twice as many detections as the text-based. This demonstrates the potential benefit of exploiting the image content through state-of-the-start image understanding.

Both methods, text- and image-based, result in more dog detections. This could indicate that there are more dogs in San Francisco than cats (or, really, that people take more pictures of dogs).

We observe the following spatial patterns in Fig. 1.

- The two methods result in very similar spatial distributions for each type of pet. Compare the similarities between the text- and image-based cat activity in Figs. 1(a) and 1(c) and text- and image-based dog activity in Figs. 1(b) and 1(d). While image content results in more overall detections, the spatial distributions of the two methods are very much in agreement.
- Dog activity is high where there are parks. Fig. 1(d) shows high dog activity in 94132 which includes the sizable Lake Merced Park and 94134 which includes the sizable John McLaren Park and medium dog activity in 94122 which includes Golden Gate Park and 94129 which include the Presidio. In contrast, as seen in Fig. 1(c), cat activity is lower in the zip codes with parks and higher in more residential zip codes such as 94116 which contains the Sunset District and 94112 which contains Ingleside, Excelsior, and the Outer Mission.
- Despite of there being more dog detections overall (Table 1), they are more concentrated. Compare the dog detections in Figs. 1(b) and 1(d) with the cat detections in Figs. 1(a) and 1(c).
- Fisherman's Wharf, North Beach, and the Embarcadero, tourist regions in 94133 and 94111, contain very little pet activity.
- We detect high pet activity in 94102 which is downtown and very urban. This is somewhat surprising and warrants further investigation.

5. CONCLUSION

We demonstrated a framework that uses georeferenced social media to measure phenomena that might not be observable through other means. Specifically, we explored two methods to detect pets in Flickr images and then mapped the results at the city-scale. The spatial distributions make sense based on our knowledge of the region.

6. ACKNOWLEDGMENTS

This work was funded in part by an NSF CAREER grant, #IIS-1150115, through an REU supplement. We would like to thank the UC Merced Spatial Analysis and Research Center (SpARC) for help with preparing the maps.

7. REFERENCES

[1] O. Russakovsky et al. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[2] C. Szegedy et al. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

SRC: Computational Steering for Geosimulations

Ashwin Shashidharan*
North Carolina State University
Raleigh, NC 27695
ashdharan@ncsu.edu

ABSTRACT

Geosimulations using computer simulation models provide researchers an effective way to study complex geographic phenomena and their outcomes. These simulations allow for scenario based exploration by capturing spatial and temporal relationships between various geographic processes in a region. However, current approaches to geosimulation limit manipulating model input and exploring alternative scenarios by controlling the simulation model at runtime. This paper proposes a computational steering system for geosimulation models and presents a prototype, *tFUTURES*, developed for the FUTURES Urban Growth Model (UGM). By allowing tangible inputs and implementing mechanisms to control model execution, this system solves the problem of lack of user-interactivity experienced at runtime. We develop a web interface and leverage the WMS, WFS-t and WPS OGC services to help visualize, modify and execute geosimulations. We define new *steering controls* within this interface and implement *application checkpointing*, allowing a user to provide new *steering input* and execute *steering actions* that can *pause*, *advance* or *rollback* a geosimulation and display the model outcomes in near real-time.

CCS Concepts

•**General and reference** → *Design;* •**Computing methodologies** → *Real-time simulation; Interactive simulation; Scientific visualization;* •**Computer systems organization** → *Special purpose systems;*

Keywords

Geosimulation, Computational steering, Visualization

1. INTRODUCTION

Modeling and simulation has revolutionized many scientific and engineering fields in the past two decades. In recent years, geosimulation has emerged at the intersection of Geographic Information Science, Complex Systems Theory and Computer Science. Geosimulations [1], where real-world processes are modeled and studied over time, have been successfully applied in urban studies, epidemiology, land use and land cover changes, and climate change studies. Using geosimulations, "what if" scenarios can be studied to understand potential impacts of geographic events. However, such scenario analyses rely on static inputs prepared beforehand by GI scientists.

Computational steering [3, 4] is a mechanism that supports interactivity in simulations while they are in progress. Specifically, it allows for manipulation of the internal state of a simulation and its inputs during execution. For instance, in a UGM geosimulation, computational steering mechanisms could be used to specify new zoning regulations and transportation networks to an in-progress simulation. Further, the ability to visualize the impact on development patterns in real-time, could be used to tweak the inputs for subsequent time-steps or in retrospect. Such interactivity helps improve the quality of simulations, allows on-the-fly "what if" scenarios, and improves computational efficiency. However, little work has been carried out to integrate computational steering and geosimulations with visualization support [5]. In our system, tFUTURES, we attempt to bridge this gap by supporting computational steering for geosimulations from a javascript enabled web browser. Finally, we enable *application checkpointing* in geosimulations and support *steering actions* that can *pause*, *advance* or *rollback* a geosimulation from any such browser.

2. tFUTURES SYSTEM

The tFUTURES system is designed to support practitioners and users who wish to simulate and understand urbanization under varying human decision scenarios. It supports tactile input to be provided to the FUTURES UGM [2] and the analysis of their outcomes in real-time. The tFUTURES computational steering system comprises of three components as shown in Fig. 1, namely (i) Monitoring server; (ii) Steering client; and (iii) Visualization service.

2.1 Monitoring Server

The monitoring server acts as an interface between the *visualization service* and the *steering client* in the system. It receives WPS requests generated by the *visualization service* and forwards them to the *steering client* in the UGM.

*The author acknowledges the support and guidance received from his advisors Dr. Ranga Raju Vatsavai and Dr. Ross K. Meentemeyer.

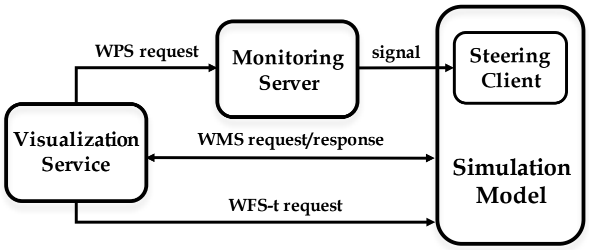


Figure 1: tFUTURES System Architecture.

2.1.1 Capabilities

The monitoring server is setup and initialized as a signal handler hub capable of routing *steering actions* to the UGM. It is aware of the *steering controls* available to a user at runtime and maps *steering actions* to specific signal handlers in the steering client.

2.1.2 Mechanism

Each *steering action* is an event that generates a particular type of signal. When a *steering control* is selected by a user, the monitoring server interrupts the current flow of UGM execution, and delivers a *steering action* to be executed by the UGM.

2.2 Steering Client

The steering client augments the UGM code to handle *steering actions* and user-defined *steering input*. It defines signal handling routines to service the *steering actions* forwarded by the *monitoring server*. Specifically, the steering client embedded in a UGM allows (i) modifying UGM simulation state; (ii) altering the control flow of the UGM simulation at runtime; and (iii) periodically checkpointing simulation state to enable *rollback* of the UGM simulation.

2.2.1 Capabilities

The steering client implements handling routines that asynchronously process signals delivered to the UGM during execution. The *steering controls* shown in Fig. 2a are defined as follows: (i) **skipPrev:** rollback the simulation by a single time-step; (ii) **restart:** reset the *steering input* and restart the existing simulation run; (iii) **play:** run the simulation from current state till completion; (iv) **skipNext:** advance the simulation by a single time-step; and (v) **pause:** pause the simulation at the end of the current time-step.

2.2.2 Mechanism

When a signal is received by the steering client, the specified *steering action* for that signal is taken. A handler function in the steering client implements this action for the UGM. In the FUTURES UGM, these handlers are predefined in the steering client as part of program annotation.

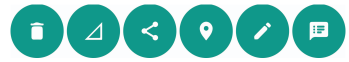
2.3 Visualization Service

The visualization component is a web service accessible from a web browser on a user's local machine. It provides the end-user with (i) web controls for interacting with the simulation; and (ii) on-line visualization of the simulation results. We use the OpenLayers JavaScript library for online map visualization, and develop the *steering controls* as web widgets using HTML, CSS and JavaScript.

2.3.1 Capabilities

The web interface provides dynamic rendering of output raster maps from the simulation. It also supports drawing vector data as input to the simulation and rendering them from within the web browser. A user can select controls from the "Steering Controls Menu" or the "Map Controls Menu" from within this web interface (Fig. 2).

(a) Steering Controls



(b) Map Controls

Figure 2: Map and Steering Controls Menu.

2.3.2 Mechanism

To experiment with various development scenarios, a user defines patterns using the *map controls* (Fig. 2b). These map controls trigger WFS-t requests directly modifying the UGM input. The *steering controls* provide the ability to run scenarios based on these inputs in time-steps as defined by the UGM. At the end of every *steering action*, the resulting urbanization map is refreshed in the browser. The visualization component thus, acts as an endpoint that accepts user input and displays simulation outputs in tFUTURES.

3. CONCLUSIONS

In this paper, we show that computational steering capabilities can be easily extended to geosimulations with a small set of interacting components and minimal changes to legacy model code. At a bare minimum, the set of interacting components must include 1) a visualization interface with *steering controls*; 2) a monitoring server to intercept and relay *steering actions* to the simulation; and 3) a steering client embedded in the legacy simulation code. Finally, by intertwining user interactions with geosimulations, we empower practitioners and novice users to dynamically vary model inputs at runtime and produce desired simulation results.

4. REFERENCES

[1] I. Benenson and P. M. Torrens. *Geosimulation: Automata-based modeling of urban phenomena*. John Wiley and Sons, 2004.

[2] R. K. Meentemeyer, W. Tang, M. A. Dorning, J. B. Vogler, N. J. Cunniffe, and D. A. Shoemaker. FUTURES: multilevel simulations of emerging urban-rural landscape structure using a stochastic patch-growing algorithm. *Annals of the Association of American Geographers*, 103(4):785–807, 2013.

[3] J. D. Mulder, J. J. van Wijk, and R. van Liere. A survey of computational steering environments. *Future generation computer systems*, 15(1):119–129, 1999.

[4] R. Van Liere, J. D. Mulder, and J. J. Van Wijk. Computational steering. In *International Conference on High-Performance Computing and Networking*, pages 696–702. Springer, 1996.

[5] R. R. Vatsavai and R. K. Meentemeyer. Lightning talk: Intertwining computational steering with geosimulations. http://ucgis.org/sites/default/files/document-sharing-form-files/209/Vatsavai_.pdf. University Consortium for Geographic Information Science (UCGIS), 2016.

SRC: Dynamic Indoor Navigation with Bayesian Filters

Wenlu Wang
Dept. of Computer Science and Software Engineering
Auburn University
Auburn,AL,36849
wenluwang@auburn.edu

Wei-Shinn Ku
Dept. of Computer Science and Software Engineering
Auburn University
Auburn,AL,36849
weishinn@auburn.edu

ABSTRACT

Indoor space navigation has always been an issue without GPS localization. Especially for complicated cases such as emergency evacuation and dynamic navigation, there is no existing efficient solution to the best of our knowledge. Localization in indoor spaces has to rely on sensing devices (e.g., Radio Frequency Identification(RFID) readers, WiFi routers, bluetooth beacons) rather than GPS, and indoor floor plans are more complicated than road networks. Consequently, existing spatial outdoor query techniques are not suitable for this new challenge. However, raw data generated by sensing devices suffers from false negatives and errors. As a result, filtering methods are necessary for accurate localization.

We propose a novel dynamic shortest path navigation strategy to enable efficient navigation for emergency evacuation in indoor spaces. This strategy achieves optimal time efficiency by: 1) using a Bayesian inference based concurrent model, which integrates dynamic shortest path searching into the filtering process, thus achieving an efficient and accurate search for any time-sensitive situation; 2) storing alternative parent nodes along the shortest path search for a fast, dynamic search.

We use both particle filters and the Kalman filter to study which one is more suitable for dynamic environments. In general, we develop an innovative, dynamic shortest path navigation solution based on Bayesian inference localization.

1. INTRODUCTION

People spend most of their time in indoor spaces. Indoor spaces are growing larger and more complex (e.g., multi-functional shopping malls, NYC subways, etc.). Therefore, users will be likely to use spatial navigation mobile apps to find friends or Points Of Interest (POI) in indoor places. In extreme cases like fires or terrorist attacks, indoor spatial navigation could even save lives. However, existing spatial query solutions [2] for Euclidean distances or road networks cannot be applied to indoor spaces because of the lack of GPS signals. Furthermore, indoor floor plans are more complicated with multiple levels involved. The uses of sensing devices have expanded beyond traditional fields and made indoor localization possible. Take RFID technologies as an example. When a tag is in the detection range of a reader, the reader recognizes the tag and generates a reading record. Several types of deviations can be observed from sensor devices, such as sensitivity errors, bias, noise and so on. As a result, the raw data generated by sensing devices can not be used for localization directly. Therefore, we use Bayesian inference based filtering methods, such as particle filters [4], to accurately calculate the position of a user.

More importantly, in our research, we focus on dynamic navigation rather than static spatial queries. Dynamic navigation is more suitable for indoor environments for the following two reasons: 1) Indoor routes could change at any time, especially during an emergency, during which a route could be blocked in a short time. In such scenarios, static solutions would not be workable, because the system has to calculate all over again; 2) Sensing devices localization is not as good as GPS. They have to correct themselves sometimes, which will affect the navigation process.

Based on the aforementioned reasons, we apply a concurrent model to Bayesian inference to accelerate dynamic navigation. There are two reasons why we apply this model: 1) In event-driven cases, there is a high possibility that the user is in a room or in a highly-recognizable space. We do not need to know the exact location of a user to navigate; especially in an emergency situation, we need to navigate as soon as possible; 2) Bayesian inference has an updating phase. In time-sensitive cases, it is crucial to take advantage of this phase and accelerate the whole process.

2. APPROACH AND UNIQUENESS

2.1 Design

Our accelerated dynamic navigation has two components: Bayesian-based Concurrent Navigation and Redundant Tracking method.

Bayesian-based Concurrent Navigation is based on Bayesian Inference methods, and it is combined with a Bayesian updating phase. More generally, it could be added to any localization method with an updating phase involved. *Redundant Tracking* is applicable to any dynamic queries. It is extremely suitable for indoor spaces.

2.1.1 Bayesian-based Concurrent Navigation

Each time a position is calculated, multiple resamplings /re-calculations are required for Bayesian Inference meth-

Algorithm 1 Bayesian inference-based dynamic navigation

```
retrieve Obj_s's readings from the data collector
for every second of readings do
    Initialization
    Normalize the weights of Obj_s
    Resampling/Re-calculation
    Store locations with weight > τ (τ is the threshold)
    Redundant Shortest Path Search
end for
return navigation based on weight and location
```

ods. In the process of resampling/re-calculation, a user has to wait for a response, which is unacceptable in an emergency. Here, we propose concurrent dynamic indoor navigation combined with Bayesian inference.

We use particle filters and the Kalman filter as examples.

Particle filters Traditional particle filters method consists of three phases: initialization, particles updating, and particles resampling. At first, a set of particles are uniformly distributed in the search area. Then the particles are updated with Gaussian distribution. After updating, the resampling process will remove particles with lower weight and replicate particles with higher weight. In the initial process, a user has to wait enough rounds of resampling for accurate localization. We take advantage of this time period to conduct dynamic shortest path searches on all potential locations. We start as soon as we get the first reading, and search a shortest path from every possible location. We also maintain all possible paths to avoid further calculation.

The Kalman filter The Kalman filter uses a series of measurements observed over time. It assumes an object's speed is a Gaussian variable, and for each reading, the algorithm recursively enumerates all possibilities. At last, the algorithm integrates the *pdf* (probability distribution function) of the object's possible locations. For each update phase, the current priori prediction is combined with current observation information to refine the state estimate. Same as particle filters, we integrate our dynamic navigation into the state prediction phase.

2.1.2 Redundant Tracking Approach

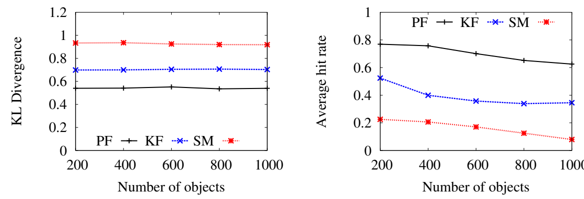
We assume the routes are constantly changing. In order to do the incremental search only, our algorithm uses "redundant" storage to keep track of potential shortest paths along the way. For each visited node, we store λ alternative parent nodes (λ is the number of alternatives, which is a parameter).

Even though redundancy is required for this approach, creating extra storage will not be a burden for the search process because indoor floor plans are relatively smaller than outdoor road networks.

Algorithm 1 shows the general structure to integrate redundant tracking into Bayesian inference. We check all possibilities for each Bayesian updating. When the current possibility is good enough for a search, we execute redundant shortest path search instantly.

2.2 Data Settings

In our research, we focus on the setting of an indoor environment and RFID technologies. A number of RFID readers are deployed along the hallway. A user is attached with an RFID tag, which can be recognized by any reader when the user passes the reader's detection range. The system will store all raw readings generated by readers.



(a) KL divergence (b) kNN success ratio

Figure 1: Varying the number of moving objects.

2.3 Uniqueness

While other researchers have made use of Bayesian Inference to localize indoor targets, to the best of our knowledge there is no previous work that combines Bayesian inference with dynamic indoor navigation. In addition, previous works pay more attention to static queries (e.g., kNN, range query), while our work focuses on dynamic queries to support indoor navigation. An approach based on dynamic settings may be more practical for indoor queries.

3. PRELIMINARY EXPERIMENTS AND FUTURE WORK

We carry out experimental evaluations using the data generated by real-world parameters, and compare the results with other symbolic model-based solutions [3].

We test the effect of particle filters and the Kalman filter with various parameters (e.g. query window size, number of particles, number of moving objects, activation range, continuous query, etc.). We use *PF*, *KF*, and *SM* to represent the curves of the particle filter-based method, Kalman filter-based method, and symbolic model-based method, respectively. Due to limitations of space, we only show 1) the Kullback-Leibler (KL) divergence of range query; 2) hit rate of kNN query by varying the number of moving objects. Figure 1 demonstrates both filtering methods have better scalability than the symbolic model based solution.

Our preliminary results show that particle filters and the Kalman filter based spatial queries are efficient and accurate enough to extend to dynamic navigation. We will compare the speed of particle filters and the Kalman filter to see which one is more suitable for dynamic navigation.

4. ACKNOWLEDGMENTS

This research has been funded in part by the National Science Foundation grant IIS-1618669.

5. REFERENCES

[1] S. R. Jeffery, M. Garofalakis, and M. J. Franklin. Adaptive cleaning for rfid data streams. In *PVLDB*, pages 163–174. VLDB Endowment, 2006.

[2] H. Samet, J. Sankaranarayanan, and H. Alborzi. Scalable network distance browsing in spatial databases. In *SIGMOD*, pages 43–54. ACM, 2008.

[3] B. Yang, H. Lu, and C. S. Jensen. Probabilistic threshold k nearest neighbor queries over moving objects in symbolic indoor space. In *EDBT*, pages 335–346. ACM, 2010.

[4] J. Yu, W.-S. Ku, M.-T. Sun, and H. Lu. An rfid and particle filter-based indoor spatial query evaluation system. In *EDBT*, pages 263–274. ACM, 2013.

The SIGSPATIAL Special

Section 2: Event Reports

ACM SIGSPATIAL

http://www.sigspatial.org

MobiGIS 2016 Workshop Report
The Fifth ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems
San Francisco, California, USA - October 31, 2016

Shashi Shekhar[1] Maria Luisa Damiani[2] Chi-Yin Chow[3] Reza Nourjou[4]
[1]Department of Computer Science and Engineering, University of Minnesota, USA
[2] Department of Computer Science, University of Milan, Italy
[3]Department of Computer Science, City University of Hong Kong, Hong Kong
[4]GIS Scientist, Washington D.C., USA
shekhar@cs.umn.edu, damiani@di.unimi.it, chiychow@cityu.edu.hk, nourjour@gmail.com

Combining the functionality of mobile devices (smartphones and tablets), wireless communication (Wi-Fi, Bluetooth and 3/4G), and positioning technologies (GPS, Assisted GPS and GLONASS) results in a new era of mobile geographic information systems (GIS) that aim at providing various invaluable services, including location-based services, intelligent transportation systems, logistics management, security and safety, etc. Many mobile GIS applications have been developed to solve challenging real-world problems and improve our quality of life.

MobiGIS 2016 (http://www.mobigis.org) was held in conjunction with the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL 2016) on October 31, 2016 in San Francisco, California, USA. It aims at bringing together researchers and practitioners from the GIS community, the mobile computing community, and the data management community. Many current research areas, such as spatio-temporal databases, spatio-temporal data mining, mobile cloud computing, remote sensing, participatory sensing, or social networks, raise research problems that lie at the boundary between these three communities. MobiGIS's goal is to foster an opportunity for researchers from these three communities to gather and discuss ideas that will shape and influence these emerging GIS-related research areas.

MobiGIS 2016 has accepted 11 research papers for oral presentations (30 minutes for each full paper and 20 minutes for each short paper). MobiGIS 2016 was a one-day workshop consisting of four sessions: (1) Trajectory Computing, (2) Keynote and Location-based Query Processing, (3) Mobile Data Analytics, and (4) Urban Computing, Mapping, and Positioning. We would like to express our special thanks to the keynote speaker, Prof. Maria Luisa Damiani (University of Milan, Italy), who gave a very interesting and inspiring talk “Spatial Trajectories Segmentation: Trends and Challenges”.

We would also like to thank the authors for publishing and presenting their papers in MobiGIS 2016, and the program committee members and external reviewers for their professional evaluation and help in the paper review process. We hope that the proceedings of MobiGIS 2016 will inspire new research ideas, and that you will enjoy reading them.

IWGS 2016 Workshop Report
The 7th ACM SIGSPATIAL International Workshop on GeoStreaming
San Francisco, CA, USA - October 31, 2016

Chengyang Zhang
Amazon
cyzhang@amazon.com

Farnoush Banaei-Kashani
University of Colorado Denver
farnoush.banaei-kashani@ucdenver.edu

Abdeltawab Hendawi
University of Virginia
hendawi@cs.virginia.edu

(Workshop Co-chairs)

The ACM SIGSPATIAL International Workshop on Geostreaming (IWGS) was held for the seventh time in conjunction with the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACMGIS 2016). The workshop has been a successful event that attracted participants from both academia and industry. The workshop addressed topics that are at the intersection of data streaming and geospatial systems. The workshop fostered an environment where geospatial researchers can benefit from the advances in geosensing technologies and data streaming systems.

We are entering the era of ”big data” thanks to the exponential growth and availability of structured and unstructured data, among which a large amount are real-time streaming data emitted from sensors, imagery and mobile devices. In addition to the temporal nature of stream data, various sources provide stream data that has geographical locations and/or spatial extents, such as geotagging twitter streams, mobile GPS location streams, spatial temporal image streams, and so on. On one hand, this amount of streamed data has been a major propeller to advance the state of the art in geographic information systems. On the other hand, the ability to process, mine, and analyze that massive amount of data in a timely manner prevented researchers from making full use of the incoming stream data. The geostreaming term refers to the ongoing effort in academia and industry to process, mine and analyze stream data with geographic and spatial information.

This workshop addresses the research communities in both stream processing and geographic information systems. It brings together experts in the field from academia, industry and research labs to discuss the lessons they have learned over the years, to demonstrate what they have achieved so far, and to plan for the future of geostreaming.

The workshop featured two keynotes. The first keynote was delivered by Roger Zimmermann from NUS, who reflected on the fascinating work at his research lab on fusion and analysis of data streams received from physical sensors and social media, discussing the the challenges in addressing this problem and corresponding solutions his team have developed. The second keynote was offered by Yu Zheng, a research manager at Microsoft Research China. He defined urban computing as the process of acquisition, integration, and analysis of big and heterogeneous data generated by a diversity of sources in cities to tackle urban challenges, e.g., air pollution, energy consumption and traffic congestion. Urban computing connects unobtrusive and ubiquitous sensing technologies, advanced data management and analytics models, and novel visualization methods, to create win-win-win solutions that improve urban environment, human life quality, and city operation systems. According to Zheng, this field is an inter-disciplinary field where computer science meets urban planning, transportation, economy, the environment, sociology, and energy, etc., in the context of urban spaces. In this talk, he provided an overview of a framework for urban computing, and discussed its key challenges and methodologies from computer science perspective. He also presented a variety of urban computing applications, ranging from big data-driven environmental protection to transportation, from urban planning to urban economy. This keynote was very well attended and engaging.

The call for paper resulted in 11 submissions of very high quality research papers. A program committee of 6 members reviewed the submissions and as a result 10 papers were accepted given the time constraints of the workshop. On average, over 20 attendees were present at every session of the workshop, although in certain sessions the attendance exceeded 60. The topics presented in the workshop include but are not limited to: Moving Object Queries, Geostream Data Processing, Mining Geostreams, and Trajectory Analysis.

join today!

SIGSPATIAL & ACM

www.sigspatial.org www.acm.org

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*, full and unlimited access to 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

payment information

Name ______________________

ACM Member # ______________________

Mailing Address ______________________

____

City/State/Province ______________________

ZIP/Postal Code/Country ______________________

Email ______________________

Mobile Phone ______________________

Fax ______________________

Credit Card Type: ❑ AMEX ❑ VISA ❑ MC

Credit Card # ______________________

Exp. Date ______________________

Signature ______________________

Make check or money order payable to ACM, Inc

ACM accepts U.S. dollars or equivalent in foreign currency. Prices include surface delivery charge. Expedited Air Service, which is a partial air freight delivery service, is available outside North America. Contact ACM for more information.

Mailing List Restriction

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

SIGAPP

www.acm.org/joinsigs

Association for Computing Machinery

Advancing Computing as a Science & Profession

The SIGSPATIAL Special