# Spatial Big Data and Machine Learning in GIScience

*Workshop at GIScience 2018*

Melbourne, Australia
28 August 2018

Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, and Peter Kiefer (Eds.)

# Editors

**Martin Raubal**
ETH Zurich
Institute of Cartography and Geoinformation
Stefano-Franscini-Platz 5
CH-8093 Zurich
Switzerland
mraubal@ethz.ch

**Shaowen Wang**
University of Illinois at Urbana-Champaign
Department of Geography & Geographic Information Science
1301 W. Green St.
Urbana, IL 61801
USA
shaowen@illinois.edu

**Mengyu Guo**
University of Illinois at Urbana-Champaign
Department of Geography & Geographic Information Science
1301 W. Green St.
Urbana, IL 61801
USA
mengyu@illinois.edu

**David Jonietz**
HERE Technologies
Seestrasse 356
CH-8038 Zurich
Switzerland
david.jonietz@here.com

**Peter Kiefer**
ETH Zurich
Institute of Cartography and Geoinformation
Stefano-Franscini-Platz 5
CH-8093 Zurich
Switzerland
pekiefer@ethz.ch

# Program Committee

| | |
|---|---|
| Ben Adams | University of Canterbury, NZ |
| Gennady Andrienko | Fraunhofer IAIS, DE & City Univ. London, GB |
| Lauren Bennett | Esri, US |
| Budhendra Bhaduri | Oak Ridge National Laboratory, US |
| Dominik Bucher | ETH Zurich, CH |
| Krzysztof Janowicz | University of California at Santa Barbara, US |
| Craig Knoblock | University of Southern California, US |
| Wenwen Li | Arizona State University, US |
| Tao Lin | Zhejiang University, CN |
| Omar Maher | Esri, US |
| Shawn Newsam | University of California at Merced, US |
| Ross Purves | University of Zurich, CH |
| Kai-Florian Richter | Umeå University, SE |
| Monika Sester | Leibniz Universität Hannover, DE |
| Fabio Veronesi | Harper Adams University, GB |
| Jan Dirk Wegner | ETH Zurich, CH |
| Stephan Winter | University of Melbourne, AU |
| Ningchuan Xiao | Ohio State University, US |
| May Yuan | University of Texas at Dallas, US |
| Chao Zhang | University of Illinois at Urbana-Champaign, US |

# Table of Contents

# Stay-Move Tree for Summarizing Spatiotemporal Trajectories

Eun-Kyeong Kim[1]

[1] University Research Priority Program (URPP) "Dynamics of Healthy Aging" & Department of Geography, University of Zurich, Zürich, Switzerland
{eun-kyeong.kim}@geo.uzh.ch

**Abstract.** Summarizing spatiotemporal trajectories of a large number of individual objects or events provides insight into collective patterns of phenomena. A well-defined data model can serve as a vehicle for classifying and analyzing data sets efficiently. This paper proposes *the Stay-Move tree (SM tree)* to represent frequency distributions for types of trajectories by introducing concepts of *stay* and *move*. The proposed tree model was applied to analyzing the Korean Household Travel Survey data. The preliminary results show that the proposed SM trees can potentially be employed to compare/classify spatiotemporal trajectories of different groups (e.g., demographic groups or species of animals). The methodology can potentially be useful to summarize big trajectory data observed from both human and natural phenomena.

**Keywords:** spatiotemporal trajectory, spatiotemporal data model, movement analysis.

## 1 Introduction

The rise of new sensor and communication technologies increasingly produces massive trajectory datasets and demands methodologies for analyzing big spatiotemporal trajectory data. Summarizing spatiotemporal trajectories can provide insight into collective patterns of individual moving objects or events. Despite of efforts made for summarizing spatiotemporal trajectories [1-6], there is still a scarcity of representation models for huge trajectory data.

A well-defined data model can facilitate classifying and analyzing data sets efficiently. Modeling movement patterns of moving objects has been carried out by [3-6]. Dodge *et al.* (2008) [3] developed a conceptual framework for moving objects' behavior and classification of movement patterns. Hornsby and Li (2009) [4] introduced a typology for spatiotemporal trajectories of a single object. Schneider *et al.* (2013) [5] characterized daily mobility patterns by directed networks of locations. Inoue and Tsukahara (2016) [6] proposed a hierarchical classification method for categorizing stays in movement trajectories by the frequency of stays. However, many existing studies do not consider a temporal dimension [5-6] or focus on obtaining representative paths of clustered trajectories [1-3].

To summarize large trajectory datasets in a concise but comprehensive way, this study develops a methodology, called a *Stay-Move (SM) tree*, to categorize spatiotemporal trajectories into a simplified trajectory type, represent the (relative) frequency of trajectories of each trajectory type, and compare trajectories of different groups. This study proposes a *Stay-Move (SM) model* as a data model for spatiotemporal trajectories, which is described in Section 2. Then, the SM tree is proposed to arrange all types of trajectories that consist of *stay* and *move* elements and represent a (relative) frequency distribution of types of trajectories of an entire dataset (Fig. 2), which is illustrated in Section 3. The preliminary analysis on Korea Household Travel Survey data collected in 2006 in the Seoul Metropolitan area shows that the proposed methodology allows comparing collective patterns of trajectories of different groups.

## 2 Stay-Move Model

This study employs a *Stay-Move (SM) model* proposed in author's previous work [7], a typology of spatiotemporal trajectories. To simplify spatiotemporal trajectories, this model adopts the

Spatial big data and machine learning in GIScience, Workshop at GIScience 2018, Melbourne, Australia. Editors: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, Peter Kiefer.

two concepts of *stay*, defined as a behavior of staying at a place for a significant time, and *move*, defined as a behavior of moving from one place to another by a significant distance because they are basic elements of movements of objects and events (Fig. 1b). This model represents a trajectory as an ordered set of stays and moves. The order of places is determined by the order of visits, regardless of absolute locations of each place (Fig. 1). For example, if one stays at home from the beginning of a day, their home becomes the first place. If they move to places where they already visited, the model takes the previously labeled order of places, which is regressive (Fig. 1b). If they move to a new place, the order of the place increases by one, which is non-regressive (Fig. 1c). A trajectory type is represented as a sequence of numbers where the numbers denote identical stays in order of appearance for the observed time period. For instance, a travel of 'home, work, home' in a day is represented as '121', as is 'work, gym, work.'

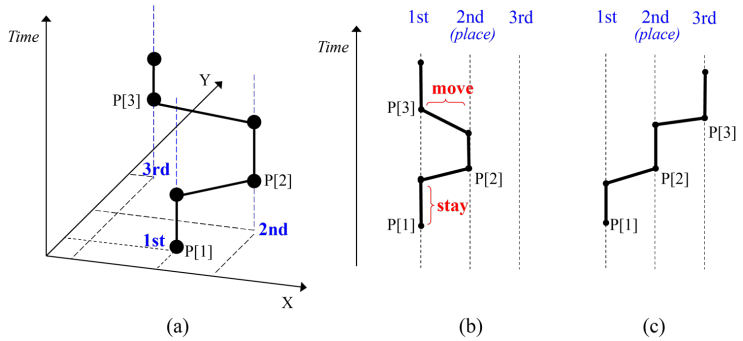



**Fig. 1.** Concepts of the Stay-Move model. (a) A trajectory in space and time, (b) A regressive type of SM model, and (c) A non-regressive type of SM model.

The SM model is useful to simplify complicated spatiotemporal trajectories. Although this model can be applied to any trajectory data, one needs to be cautious in applying the SM model to noisy data. Datasets containing no descriptions (e.g., home, work, stop-over) for each location, e.g., GPS data, need to be preprocessed. A set of location points that constitutes a daily trajectory needs to be discretized into move(s) and stay(s) according to an analyst's definition of moves and stays. If the definitions allow capturing stays and moves at a micro scale of space and time, the number of segments of a SM model may increase to the excessive degree, which is not ideal to summarize trajectory data concisely.

## 3 Stay-Move Tree

This study proposes a *Stay-Move (SM) tree* to organize all types of trajectories in an SM model by a tree structure. As seen in Fig. 2, each node of a tree represents a specific trajectory type constructed by the SM model. The root node at the top of the tree represents a trajectory without any moves for a period of time (e.g., a day), and its child node represents a trajectory type with one more move, extending from its parent node (Fig. 2). Each parent node branches into its child nodes along with a possible place set that includes previously visited places or a new place. Trajectory types with the same number of moves are located at the same depth of a tree. The height of a tree is determined by the maximum number of moves of a single trajectory in the data.

Here is a more formalized description. Let a list of locations where the $n$-th object once stayed until the $m$-th move be $\mathrm{Loc}_m[O_n] = \{\mathrm{location}[i] \mid i \le m\}$, and let a sequence of ordinal numbers that substitutes elements of $\mathrm{Loc}_m[O_n]$ by order of appearance of locations be $\mathrm{LocOrder}_m[O_n] = \{\mathrm{order}[i] \mid i \le m\}$. Then, the combination of elements belonging to $\mathrm{LocOrder}_m[O_n]$ can represent each type of trajectory as illustrated in Fig. 2. In addition, a possible location set can be defined as $\mathrm{PossibleLoc}_m[O_n] = [\mathrm{Loc}_m[O_n] - \{\mathrm{location}[m]\}] \cup \{\mathrm{location}[m+1]\}$.

Due to its predefined structure embracing all types of trajectories, the SM tree enables to represent the (relative) frequency distribution of trajectory type (Fig. 2). A (relative) frequency distribution over an SM tree gives insights into the nature of movements including the level of activities and the diversity of visited places. Based on a (relative) frequency distribution, SM trees can be used to compare different sets of trajectories grouped by demographic characteristics (e.g., age group), regions, time periods (e.g., Monday vs. Saturday), or animal species.

## 4 Classification of Stay-Move Trees

With Stay-Move trees, it is possible to investigate how similar different groups of trajectories are by measuring the similarity between a pair of SM trees constructed from a pair of groups of trajectories. A similarity function measuring the degree of similarity between two SM trees can be defined as follows:

$$100 - \sum_{i=1}^{depth} \sum_{j=1}^{node} \alpha_{ij} \left| P_{ij}[Tree_1] - P_{ij}[Tree_2] \right|$$

$$\alpha_{ij} = \text{weights for } j\text{th node at } i\text{th depth (default} = 1)$$

$$P_{ij} = \text{the relative frequency (i.e., percentage) of } j\text{th node at } i\text{th depth}$$

Similarity functions can be defined based on either the frequency or the relative frequency (i.e., percentage), but the similarity function based on the percentage is better to compare different groups of trajectories because it is normalized by the total number of trajectories.

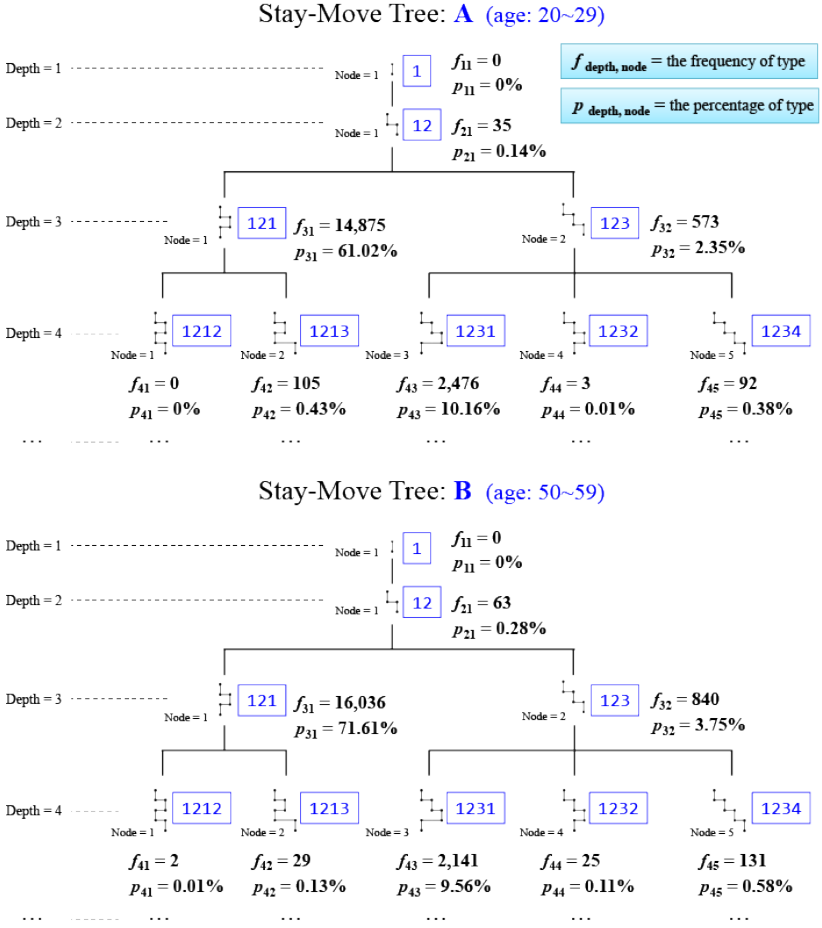



**Fig. 2.** The frequency distribution ($f_{ij}$) and the relative frequency distribution ($P_{ij}$) of daily trajectory type of a Stay-Move tree for two groups of people (A: age of 20-29; B: age of 50-59). Nodes beyond the depth of 4 are not represented in this figure for both trees, A and B, due to the limited space.

## 5 Preliminary Result

To show the utility of the proposed methodology, I conducted preliminary analysis on Korea Household Travel Survey data collected in 2006 in the Seoul Metropolitan area. The data contains about 100k daily trajectories and about 300k trips. Each trip has attributes including administrative area codes, time, and place type of departure and arrival as well as mode of transportation. To keep SM trees in a manageable size, this analysis does not include stop-over places but only destination places.

The frequency distributions and relative frequency (percentage) distributions of trajectory types of SM trees were constructed for age groups in the Seoul Metropolitan area (Fig. 2), and then, the similarity of each pair of SM trees was calculated (Table 1). The similarity matrix shown in Table 1 reveals that age groups of 10-19 and 20-29 have different patterns in relative frequency (percentage) distributions of trajectory types from the rest of age groups. A close look into the percentage of each type enabled a better understanding of those patterns. Age groups of 10-19 and 20-29 show the higher percentage of two types of trajectories: '12131' (e.g., home–school–home–extracurricular activities–home) and '1231' (e.g., home–school–extracurricular activities–home) than other age groups. While age groups of 40-69 have similar relative frequency distributions each other, the age group of 30-39 are different from them.

In the future, I plan to extend the data analysis to other big trajectory data including large GPS data.

**Table 1**. Similarity of relative frequency (percentage) distributions of different age groups

| | Age Group | | | | | | |
|---|---|---|---|---|---|---|---|
| | 10~19 | 20~29 | 30~39 | 40~49 | 50~59 | 60~69 | 70~119 |
| 10~19 | - | - | - | - | - | - | - |
| 20~29 | 84.4% | | | | | | |
| 30~39 | 48.8% | 56.0% | | | | | |
| 40~49 | 52.3% | 57.6% | 89.2% | | | | |
| 50~59 | 55.7% | 60.4% | 82.9% | 91.5% | | | |
| 60~69 | 55.2% | 59.9% | 87.1% | 94.6% | 93.7% | | |
| 70~119 | 52.6% | 59.1% | 91.2% | 93.1% | 87.9% | 91.8% | |

| Legend |
|---|
| 100.0% |
| 40.0% |

## 6 References


1. Laube, P., Imfeld, S., & Weibel, R. (2005). Discovering relative motion patterns in groups of moving point objects. *International Journal of Geographical Information Science*, *19*(6), 639-668.
2. Sinha, G. & Mark, D. M. (2005). Measuring similarity between geospatial lifelines in studies of environmental health. *Journal of Geographical Systems*,*7*(1), 115-136.
3. Dodge, S., Weibel, R., & Lautenschütz, A.-K. 2008. "Towards a taxonomy of movement patterns". *Information Visualization* **7**: 240–252.
4. Hornsby, K. & N. Li. 2009. "Modeling spatiotemporal paths for single moving objects". In *Research Trends in Geographic Information Science*, Ed. Gerhard Navratil: 151-167. New York: Springer, Dordrecht, Berlin 2009.
5. Schneider, C. M., Belik, V., Couronné, T., Smoreda, Z., & González, M. C. (2013). Unravelling daily human mobility motifs. *Journal of The Royal Society Interface*, *10*(84), 20130246.
6. Inoue, R., & Tsukahara, M. (2016). Travel pattern analysis from trajectories based on hierarchical classification of stays. In *International Conference on GIScience Short Paper Proceedings* (Vol. 1, No. 1).
7. Kim, E.-K. (2009). *3D sequential object clustering for spatio-temporal data analysis*. (master's thesis, Kyung Hee University, 2009).

# Detection of Unsigned Ephemeral Road Incidents by Visual Cues

**Alex Levering[1], Kourosh Khoshelham[2], Devis Tuia[1], and Martin Tomko[2]**

1 **Laboratory of Geo-information Science and Remote Sensing, Wageningen University**
alex.levering|devis.tuia@wur.nl
2 **Department of Infrastructure Engineering, University of Melbourne, Australia**
k.khoshelham|tomkom@unimelb.edu.au

## 1 Introduction

Traffic is a highly dynamic environment and ephemeral changes to the on-road conditions impact it continuously. Research in Autonomous Vehicles (AVs) is currently highly focused on dynamic changes, due to the high safety requirements [12]. AVs must be able to detect and respond appropriately to incidents. In a connected traffic data ecosystem [3], AVs will further share traffic information about the state of the road network.

Our research addresses the detection of ephemeral incidents affecting road networks by *first-on-scene vehicles*. An incident is *"... an event, which directly or indirectly can result in considerable reductions or interruptions in the serviceability of a link/route/road network."* [2]. For any unsigned incident (i.e. not yet signposted or tagged incidents such as a recently detected fire), the sensors of the vehicle that is *first on the scene* should detect and assess the danger to the traffic and report it to the connected traffic ecosystem. Efforts have been made to recognize *signed changes* to the environment (i.e., pedestrian and traffic signs) [6, 14], as well as the avoidance of dynamic scenarios (e.g., pedestrians or animals stepping into the traffic [9, 10, 11]). Yet, there has been no research covering the systematic classification and autonomous detection of various types of incidents by first-on-scene vehicles.

In this paper we address the problem of identifying and classifying *unsigned ephemeral on-road incidents* from street-level imagery, such as those acquired by dashcams. We present the approach to the creation of an extensive, labeled street-level image library that supports the detection and classification of on-road incidents using a deep convolutional neural network (CNN). The full results of the experiments will be reported at the workshop.

## 2 Methodology

Image classification using CNNs has had enormous success [1]. Pre-trained CNNs can now be used for fast image classification from camera frames onboard a vehicle. CNNs require vast amounts of labeled data to be trained. We describe the systematic approach to the collection of a detailed and comprehensive image dataset of on-road incidents, from a taxonomy of incidents through collection from Web sources, incl. data with a partial geographical stratification.

### 2.1 Taxonomy of Incidents

To label the incident images used by the CNN model, we propose an approach grounded in a systematic taxonomy of incident types, providing a semantic structure for an adaptable set of incident labels (Figure 1). Such adaptability is desirable to for alterations required to

Spatial big data and machine learning in GIScience, Workshop at GIScience 2018.
Editors: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, Peter Kiefer;

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

reflect changing local deployments (i.e, wildlife, meteorological events). We only consider physical incidents, as non-physical incidents cannot be identified in imagery (e.g., GPS or traffic information system failure). At the top level, we distinguish between signed physical incidents (not tackled here) and unsigned physical incidents. We further distinguish between man-made and natural physical incidents to distinguish the objects and covers altering the status of the road. In total, we consider eight incidents in this initial study: crashes (MAN-O-1), road collapse (MAN-C-1), animal on road (NAT-O-1), treefall (NAT-O-2), snow on road (NAT-C-1), flooding (NAT-C-2), landslides (NAT-C-3), and fires (NAT-C-4).

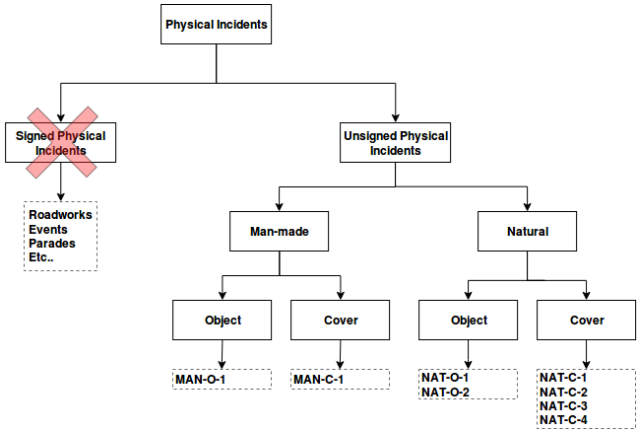



**Figure 1** Taxonomy of incidents and their semantic groupings. Each lowest-level indicator (e.g. MAN-O-1) refers to an incident under consideration.

## 2.2 Data

We collect RGB images captured by image sensors. We opted for optical recognition of incidents as cameras record information that cannot be extracted from LIDAR point clouds (e.g., smoke, fire), and because of the abundance of labeled images and datasets. RGB images are also readily accessible through online search engines. We collect and clean images from four sources: Google Custom Search API [1], Bing Image Search API v7 [2], the Flickr API [3], and the Geograph UK project [4]. We construct queries by grouping synonyms of each topic, such as 'snow on road' and 'street blizzard'. We retain the top-100 images that match queries crafted for each category of taxonomy, as the relevance and quality of images drops noticeably beyond this number. In a subsequent manual cleaning process images are restricted to relevant on-road incidents. We only retain images that have a viewport height of on-road vehicles, comparable to vehicle-mounted cameras. Lastly, we retain only imagery that is *spatially relevant*, i.e., every image relates to a possible incident on the road network. As an example, an animal in itself may not be an incident. It only becomes an incident when it is in a certain spatial context, in this case *close to or on the road*. The detection of such spatial relationships in image classifiers is currently in its infancy [7], and few datasets are built with this intrinsic property in mind – a shortcoming addressed in our collection. In

[1] https://developers.google.com/custom-search/json-api/v1/overview
[2] https://docs.microsoft.com/en-us/rest/api/cognitiveservices/bing-images-api-v7-reference
[3] https://www.flickr.com/services/api/
[4] https://www.geograph.org.uk/

total, we have collected 12.500 images spread across all incident classes. The set contains the following amount of images per class: *Animal on road*: 1321, *Collapse*: 491, *Crash*: 1478, *Fire*: 865, *Flooding*: 2155, *Landslide*: 825, *Snow*: 4744, *Treefall*: 751. The dataset and the query strategy is elaborated upon in more detail during the workshop.

An additional dataset of true-negative images to be used during classification will be gathered from the same sources and from image frames from benchmark datasets such as CityScapes [4]. In total, we aim for a dataset of 40,000 true-negative images to capture a great variety of environments and regular driving conditions. We maintain a 70:20:10 training, validation, and testing split respectively for both true-positive and true-negative subsets.

## 2.3 CNN Classification

The incident recognition is implemented using CNNs in the PyTorch environment in a multiclass classification task performed on a pre-trained ResNet-34 model [8]. We chose the ResNet architecture to leverage skip connections to reduce overfitting, as well as its state-of-the-art performance and ease of training. We use a model pre-trained on the ImageNet dataset [5] which is then re-trained across all layers. We don't consider multi-label cases (e.g. a fire and a car crash visible in the same image). During training we track the loss and classification accuracy, retaining the model with the lowest validation loss as the best model. The final overall accuracy is given by the F1-score and the overall accuracy. A qualitative analysis of visual triggers will be supplied through the use of *Gradient-weighted Class Activation Mapping* [13], which will help to highlight the model's visual cues for each class. Lastly, we also report the accuracy of an experiment using a geographical stratification for the classes *Animal on road*, *flooding*, and *Snow*, which contain geotagged images from the Geograph dataset.

# 3 Discussion and Conclusions

Matches for images tagged with *incident* – a somewhat jargon term for events on roads – are rare. Our searches therefore combine terms that can either be identified in the images, or in the text nearby. The applied restrictions limit the amount of relevant images, leading to problems with sourcing examples for certain classes. This may noticeably limit the amount of possible representations of incidents covered by the final dataset. Currently, we are training our CNN model, varying all relevant hyperparameters until satisfactory convergence. The final model performance will be reported at the workshop using the F1-score and the top-1 accuracy. Lastly, we will report the geographically stratified accuracy of the three classes (*animals, flooding, snow*) present in the Geograph dataset.

We anticipate problems due to classification correlation for themes present in certain images (e.g., landslides may correlate with rocky cliff-faces). Results for all classes may thus not be equally robust. Possible continuations of this research may focus on more efficient use of image samples from sparsely represented classes to increase their representational power. We envisage to publish the subset of images licensable under a CC-by license along with the pre-trained network. This will allow for further experimentation and improvements on the benchmark for this task that has so far not been broadly covered in the literature and where significant progress is possible.

## References


1 Rodrigo Benenson. Classification datasets results, 2016. URL: http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html#434946415 22d3130.

2 Katja Berdica. An introduction to road vulnerability: what has been done, is done and should be done. *Transport Policy*, 9(2):117–127, April 2002. URL: http://www.sciencedirect.com/science/article/pii/S0967070X02000112, doi:10.1016/S0967-070X(02)00011-2.

3 Riccardo Coppola and Maurizio Morisio. Connected Car: Technologies, Issues, Future Trends. *ACM Comput. Surv.*, 49(3):46:1–46:36, October 2016. URL: http://doi.acm.org/10.1145/2971482, doi:10.1145/2971482.

4 Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. *arXiv:1604.01685 [cs]*, April 2016. arXiv: 1604.01685. URL: http://arxiv.org/abs/1604.01685.

5 Jia Deng, Wei Dong, Richard Socher, Li-jia Li, Kai Li, and Li Fei-fei. Imagenet: A large-scale hierarchical image database. In *In CVPR*, 2009.

6 Zoltán Fazekas, Gábor Balázs, László Gerencsér, and Péter Gáspár. Locating roadworks sites via detecting change in lateral positions of traffic signs measured relative to the ego-car. *Transportation Research Procedia*, 27:341–348, January 2017. URL: http://www.sciencedirect.com/science/article/pii/S2352146517309018, doi:10.1016/j.trpro.2017.12.004.

7 Mandar Haldekar, Ashwinkumar Ganesan, and Tim Oates. Identifying Spatial Relations in Images using Convolutional Neural Networks. *arXiv:1706.04215 [cs]*, June 2017. arXiv: 1706.04215. URL: http://arxiv.org/abs/1706.04215.

8 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

9 M. Jeong, B. C. Ko, and J. Y. Nam. Early Detection of Sudden Pedestrian Crossing for Safe Driving During Summer Nights. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(6):1368–1380, June 2017. doi:10.1109/TCSVT.2016.2539684.

10 B. Pan and H. Wu. Urban traffic incident detection with mobile sensors based on SVM. In *2017 XXXIInd General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS)*, pages 1–4, August 2017. doi:10.23919/URSIGASS.2017.8104994.

11 Khaled Saleh, Mohammed Hossny, and Saeid Nahavandi. Kangaroo vehicle collision detection using deep semantic segmentation convolutional neural network. *DICTA 2016 : Proceedings of the IEEE International Conference on Digital Image Computing: Techniques and Applications*, January 2016. URL: http://dro.deakin.edu.au/view/DU:30092160, doi:10.1109/DICTA.2016.7797057.

12 Brandon Schoettle and Michael Sivak. A survey of public opinion about autonomous and self-driving vehicles in the US, the UK, and Australia. 2014.

13 Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *arXiv:1610.02391 [cs]*, October 2016. arXiv: 1610.02391. URL: http://arxiv.org/abs/1610.02391.

14 H. Yong and X. Jianru. Real-time traffic cone detection for autonomous vehicle. In *2015 34th Chinese Control Conference (CCC)*, pages 3718–3722, July 2015. doi:10.1109/ChiCC.2015.7260215.

# Convolutional Neural Network for Traffic Signal Inference based on GPS Traces

**Y. Méneroux[1], V. Dizier[1], M. Margollé[1], M.D. Van Damme[1], H. Kanasugi[2], A. Le Guilcher[1], G. Saint Pierre[3], and Y. Kato[4]**

1 **Univ. Paris-Est, LASTIG COGIT, IGN, ENSG, Saint-Mandé, France**
2 **CSIS, Institute of Industrial Sciences, The University of Tokyo, Japan**
3 **Cerema, Toulouse, France**
4 **Transport Consulting Division, NAVITIME JAPAN Co., Ltd**

## Abstract

Map inference techniques aim at using GPS trajectories collected from a fleet of vehicles, to infer geographic information and enrich road map databases. In this paper, we investigate whether a Convolutional Neural Network can detect traffic signals on a raster map of features computed from a large dataset of GPS traces. Experimentation revealed that our model is able to capture traffic signal pattern signature on this very specific case of unnatural input images. Performance indices are encouraging but need to be improved through a more refined tuning of the workflow.

**Keywords:** Map Inference, Machine Learning, GPS Traces, Traffic Signal, Deep Learning

## 1 Introduction

Along with the proliferation of electronic devices equipped with a Global Positioning System (GPS) receiver large datasets of daily trajectories are collected by specialized fleets of vehicles (such as taxis, bus, delivery trucks...) or more recently, through collaborative driving mobile applications. When available, they constitute a valuable source of information to infer or enrich road map databases. With traditional cartography techniques, roads are manually digitized on orthorectified aerial images. Despite the fact that automatic or semi-automatic detection from images is getting increasingly efficient [8], road maps update speed is still limited by the (generally consequent) period separating two successive image acquisitions. Indeed aerial image campaigns are typically conducted every several years, which is to be put in perspective with the fast evolution of cities [2]. Furthermore, the process often needs to be completed with field surveys to acquire features that cannot be captured in the images.

Conversely, by the means of GPS traces coupled with efficient mining algorithms, road maps can be inexpensively generated from scratch and updated on a daily basis. This observation gave birth to a new field of research, known as *map inference* which aims at completing or replacing traditional survey techniques with floating car data [1]. Though it was initially restricted to the geometry of roads, refining topology, inferring attributes and now detecting road infrastructure (traffic signs, traffic calming devices...) are becoming achievable [5, 7]. Detailed road maps are of utmost importance for traffic flow simulation and subsequent urban planing, for autonomous vehicles or merely for computing accurate routing time estimations. On the other hand, supervised statistical learning techniques [4] have been successful in a wide range of applications, from signal and image classification and segmentation to medical diagnosis. In our application, they guarantee the genericity of the approach (provided that labeled instances are available, the process may be customized to infer other types of information, in different environments without having to design a new algorithm *ex nihilo*). Considering the large number of GPS traces that are typically recorded every day, this raises two questions: firstly how to design accurate algorithms to extract the relevant information among such massive amount of data, and secondly how to implement them efficiently.



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper, we propose to train a convolutional neural network (CNN) to detect stop lines associated to traffic signals on a rasterized map of trajectories. The remaining of the paper is organized as follows: the workflow of our application is presented in the next section, then section 3 provides implementation details and preliminary results obtained on a real-world dataset. Eventually, section 4 concludes the paper.

## 2 Workflow

The key idea of our approach is to reduce the full dataset of GPS traces into a regular grid of aggregated features, which is much more tractable from the computational view point (the amount of information to process grows only with the spatial extent of the concerned area and no longer with the GPS dataset size). The overall workflow is depicted on figure 1 hereafter.

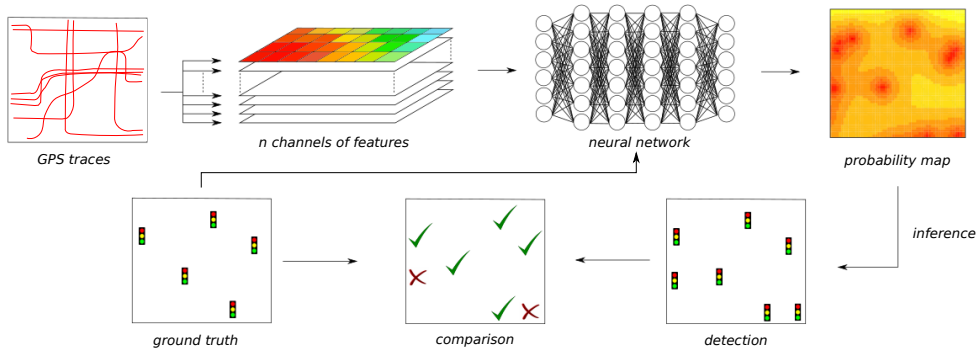



**Figure 1** Overall workflow

The first step allocates segments of GPS trajectories on a regular grid. In each cell, $n$ aggregated features (such as mean and standard deviation of speeds, number of traces...) are computed to produce a set of raster maps, which may be seen as a single image with $n$ channels. This operation may be performed efficiently through parallelization techniques and thus is scalable with the daily number of GPS traces recorded. The resulting image is input into a CNN, trained with labeled instances to detect traffic signals. CNN is an extension of artificial neural networks, where each layer of neurons performs a series of convolutions on the output of the previous layer, decreasing the number of parameters in the model while keeping the ability to extract very expressive features from spatially structured data [3, 4]. This makes CNN perfectly adapted to image-based learning. In this work, we investigate CNN capabilities in the context of recognition on images rasterized from GPS traces. The output of the network is a probability map, *i.e.* pixel values are proportional to the traffic signal presence probability. Note that here we compute a per-pixel estimation often referred to as *dense learning* in the literature. Local maxima of the output map are eventually extracted, and compared to the ground truth for validation.

## 3 Implementation and preliminary results

The experimental data represent a set of 11862 GPS traces, located in Mitaka city (16 km$^2$), suburbs of Tokyo (Japan). They were provided by NAVITIME JAPAN, a private company developing navigation technologies and providing various kinds of web application services such as route navigation, travel guidance, and other useful information services for moving people. The municipality has 669 traffic signals, reported by photo interpretation.

Traces containing fewer than 10 points (approximately 1% of all traces), were rejected. Subsequently, for each trace, a set of indices were computed from individual points: speed, acceleration, bearing, whether the point is a stop (*i.e.* if its acceleration is positive and the preceding GPS record has a zero speed or is decelerating). The bearing is used to distinguish the flow direction on two-way streets, and takes discrete values in the 8 directional sectors. A 5 meter-resolution base grid is made. The size of 5 m can reliably locate traffic signals. In each grid cell, computed indices on points are aggregated: mean and standard deviation of speeds, means of accelerations, number of stop points and most frequent orientation. These 5 features will provide 5 raster images. Feature values are converted on a 8-bit grayscale. To differentiate the value 0 from the absence of data, the minimum threshold taken by the values is 16 (*e.g.* the mean of speeds is distributed between 16 and 255). The ground truth is a binary image: cells containing a traffic signal take the value 255 (white) and remaining cells are set to 0 (black).

Feature maps and ground truth images are cut by a sliding-window into small 60×60-pixel output images, (corresponding to 300 × 300 m on the field). The overlap between images is equal to 20 pixels, hence the traffic signals have different positions in the image, so the spatial context changes. This artificial increase in the number of training instances may help the model to deal with a large variety of traffic signal configurations.

We validated the model with cross-validation on 8 geographical zones, each containing approximately 90 traffic signals and 200 windows. Areas have distinguishing morphologies like downtown, residential district, motorway environment, etc. (e.g. the zone 5 contains many unsignalized intersections).

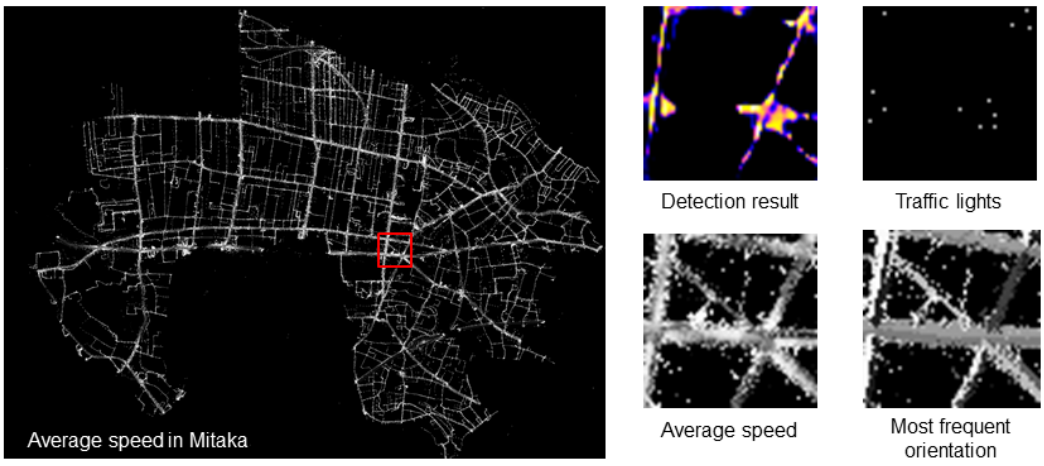



**Figure 2** Example of 2 features and the result for one image

Our neural network architecture is inspired from U-Net [6]. Since traffic signals are overwhelmingly under-represented in comparison with background, we used, as proposed in the U-NET framework, a weighted loss function, in order to penalize false negatives. The weight, corresponding to a rough average estimate of 3 traffic signals per image, was set to 1:1200 (3/60×60). It is important to note that, to produce a probability map with the same resolution as the input image, in the architecture pooling operators were replaced by upsampling operators. The network model has been implemented using Keras[1], a Python library that overloads TensorFlow[1].

Since a numerical validation of the model would require to have an efficient method to extract

[1] Keras: `https://keras.io/` TensorFlow: `https://www.tensorflow.org/`

traffic signals from probability map (which has to be addressed in future research), numerical validation is not possible yet. However a surrogate manual validation provided the following results: the recall rate stands at 75% with a precision of 60% (65% F-measure). The experiments were performed using a personal computer with 3.30 GHz CPU, 4-core processor and 8 GB of memory. The training step took 70 minutes for 54900 images for a batch size of 100 instances and 130 epochs. The validation step took 1.136 seconds for testing 60 images. This benchmark suggests that the processing time required to cover the entire Tokyo area (2190 $km^2$) is below 7 minutes, which highlights the scalability of our approach.

Though not satisfying yet for a fully automatic detection, these preliminary results are encouraging. We believe that the GPS dataset is not sufficiently large to ensure the convergence of the results. Particularly, a significant number of intersections are only covered by a few traces. Besides, with a more extensive dataset, we may calibrate the model more thoroughly without falling into the trap of overfitting. Results may be improved also by adding new features (congestion, road type, etc.). It would be interesting as well to input the aerial image in addition into the network to investigate whether this can further help the detection and localization process.

## 4 Conclusion

We presented a method based on CNN to detect traffic signals on a map of aggregated features computed from a dataset of GPS traces. Preliminary results demonstrated the potential and the scalability of the approach. The extraction of most likely locations from the probability map is an important aspect that needs to be addressed in future works. Let us note that the validation is as well problematic since it requires to measure the tradeoff between detection performance (recall and precision) and geometric accuracy. An important perspective of improvement lies in extending this approach to detect other types of infrastructure elements (such as yield signs, pedestrian crossings, speed bumps...) including more ephemeral events (roadworks zones...).

**Acknowledgements.** We would like to thank Professor Xuan Song, for his precious advice.

## References

1 James Biagioni and Jakob Eriksson. Inferring road maps from global positioning system traces: Survey and comparative evaluation. *Transportation Research Record: Journal of the Transportation Research Board*, (2291):61–71, 2012.

2 Yihua Chen and John Krumm. Probabilistic modeling of traffic lanes from gps traces. November 2010.

3 Zhiling Guo, Qi Chen, Guangming Wu, Yongwei Xu, Ryosuke Shibasaki, and Xiaowei Shao. Village building identification based on ensemble convolutional neural networks. *Sensors*, 17(11):2487, 2017.

4 Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

5 Seth Rogers, Pat Langley, and Christopher Wilson. Mining gps data to augment road models. In *KDD*, 1999.

6 O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015.

7 KBA Van Winden. Automatically deriving and updating attribute road data from movement trajectories. 2014.

8 Weixing Wang, Nan Yang, Yi Zhang, Fengping Wang, Ting Cao, and Patrik Eklund. A review of road extraction from remote sensing images. *Journal of Traffic and Transportation Engineering (English Edition)*, 3(3):271 – 282, 2016.

# Using Stream Processing to Find Suitable Rides: An Exploration based on New York City Taxi Data

**Roswita Tschümperlin[1], Dominik Bucher[1], and Joram Schito[1]**

1 **Institute of Cartography and Geoinformation, ETH Zurich, Stefano-Franscini-Platz 5, CH-8093 Zurich, Switzerland**
troswita@ethz.ch, dobucher@ethz.ch, jschito@ethz.ch

## Abstract

Changes in the mobility landscape, such as the emergence of shared or autonomous mobility, or the increasing use of mobility as a service, require an adaption of information technologies to satisfy travelers' demands. For example, the deployment of autonomous taxis will necessitate systems that automatically match people with available cars. Combined with the enormous increase in available data about transport systems and the people using them, mobility providers have to think about how to build flexible and scalable systems that satisfy their customers' information demands. In our work we explore how streaming frameworks (as part of Big Data infrastructures) can be used to process mobility data, and how the spatiality of the underlying problem can be used to improve the performance of the resulting systems. We currently experiment with different approaches on how to build a mobility data stream processing pipeline, and evaluate the different approaches against each other in terms of scalability, ease of use, and possiblity to integrate with data from other mobility providers.

## 1 Introduction

Mobility and transport must and already started to undergo large transformations due to sustainability goals set by many countries, new technological advances and the saturation of existing transport systems (3). Technological advances are not restricted to fields such as electric or autonomous cars, but also cover an increasing miniaturization of sensors and computing technology, which leads to enormous amounts of data about transport systems and the people using them. Several trends, including the emergence of shared and autonomous mobility, lead to new requirements for information technology (IT) to support travelers (6). In this work we look at taxis, which are often regarded as prime candidates for replacement with autonomous cars in the near future and will require automated IT to match people with taxis.

The use of Big Data streaming frameworks makes it possible to build systems that automatically scale (depending on the problem size), and makes them more easily deployable compared to having to build a distributed mobility processing framework from scratch. We currently experiment with different architectures for processing taxi data and evaluate them in terms of scalability, ease of use, and their suitability for integration with data from external providers (e.g., environmental data). In the work presented here, we are focusing on finding the nearest taxis for any client request (essentially solving a nearest neighbor problem), and discuss several important additions to the streaming system required for future autonomous taxi matching.

In a geographical context, Big Data frameworks were recently used to process location-based social network data, street network data and mobility and movement trajectories (e.g., (5)). There is only little public research on using *streaming* (in contrast to "plain" Big Data) frameworks to process geographical data: For example, Domoney et al. (2) look at the processing of sensor data streams, while Maarala et al. (4) use streaming to process and analyze traffic data. We argue that there is a lack of best practices when employing streaming frameworks to process spatio-temporal data, in particular with regards to traffic and mobility, an aspect that will be increasingly important for future forms of mobility. With this work, we show how spatial

Spatial big data and machine learning in GIScience, Workshop at GIScience 2018, Melbourne, Australia.
Editors: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, Peter Kiefer

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

indexing structures can be used for mobility data, and hope to initiate a discussion on the crucial components within a streaming pipeline for matching autonomous taxis.

The two best-known approaches used by Big Data streaming frameworks are *microbatching* and *true continuous streaming* (7). In microbatching, data is aggregated into smaller batches (usually of several hundred milliseconds) which are then sequentially processed. As taxi matching only has to be near real-time and vehicle and client data have to be *joined*, we are using Apache Spark[1], which adopted a microbatching approach in order to reuse its Hadoop-based structures for streaming. In addition to supporting well-known operations of Big Data systems (such as *map*, *reduce*, *join*, etc., which are automatically distributed among worker nodes) Spark Streaming offers functionality for stateful streams (e.g., to continuously update the position of a taxi) and checkpointing (for fault recovery), both essential to process mobility data in a realistic setting.

## 2 Method

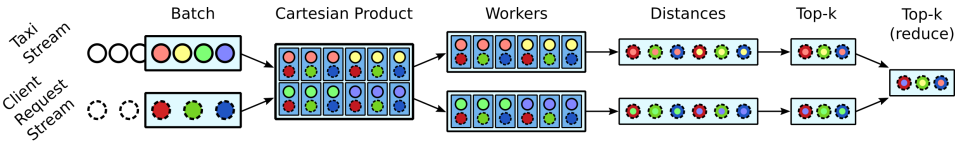
We present two approaches how to match a stream of taxi data with incoming and pending transport requests (i.e., to find the closest $k$ taxis to an incoming client request). Both the taxi data and client request streams consist of a tunable number of updates per second, sent as JSON-encoded packets over a WebSocket connection. Each packet of the incoming taxi data stream contains the *id* of the taxi it refers to (one out of several thousand), its current *location (longitude, latitude)*, the *number of people* currently traveling in it, and the *next destination* (if known). On the other hand, the stream of incoming client requests consists of the *origin (longitude, latitude)* and the *destination* of a desired journey. It is beneficial to maintain a *state* of the locations, as otherwise only the information within a single batch would be available. Technically, it would suffice to only either maintain the state for the taxis or the client requests, but this would inevitably lead to longer delays before finding the optimal match (e.g., if solely taxi locations were stateful, they could only be matched with client requests whenever those appear in the stream, and not when at a later point in time updated taxi locations would lead to a better match).

Figure 1 shows two different streaming pipelines evaluated as part of this research. Distributed processing of big data can easily be done if the individual records resp. the computations performed on them are independent. An evident approach to compute the $k$ nearest taxis for a number of client requests is to simply combine all taxis with all requests, compute the distance between each pair, sort them according to this distance, and take the top $k$. Figure 1a shows this approach, which does not use any spatiality inherent to the problem. Optimally, in order to compute the top-$k$ matches, one would not have to compute the distances between all taxis and client requests. Spatial indexing structures, such as R-trees (1), can help reducing the number of computations that need to be performed. In the approach shown in Figure 1b, we build an index for each batch of taxi data, and use these indexes to reduce the number of computations performed on each worker (the workers get a subset of client requests for which they have to find the closest taxis).

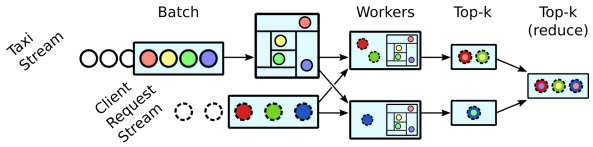
## 3 Data and Experiment

We are currently performing experiments based on real taxi data from New York City[2]. Because this dataset only contains dropoff and pickup coordinates, we used the Open Source Routing

[1] See spark.apache.org for details.
[2] The NYC taxi data is publically available starting from 2009 and can be downloaded from www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.




**(a)** The *cartesian pipeline*: Incoming taxi updates and client requests are processed in batches of several hundred milliseconds. The cartesian product is computed for all taxis and client requests in a (stateful) batch. These tuples are distributed on an adjustable number of workers which compute the distances between taxis and clients. Each worker then takes the top-$k$ pairs, and uses the streaming framework to perform a reduction step, where ultimately the top-$k$ taxis for each request remain.




**(b)** The *separated KNN pipeline*: On the master (resp. in the driver program), a spatial index is built for the taxis. This index is then sent to the workers, which use it to minimize the number of computations needed per client request.

**Figure 1** The two different mobility stream processing pipelines analyzed within this work.

Machine (OSRM)[3] to compute exact trajectories. As the dataset does not contain any indication about which taxi served a particular request, and what taxis did between two served clients, we used a heuristical approach to create a daily driving schedule (without gaps) for each taxi. A streaming application developed in Go[4] constantly reads the taxi schedules and outputs packets with an adjustable frequency to WebSocket streams. The streaming application that tests the different pipelines is thus able to receive taxi and client request streams in a controlled fashion (with respect to throughput, number of taxis and clients, time window, etc.).

Figure 2 shows the influence of the taxi and client request throughputs on the time that is required to process a single batch (normalized by the number of taxis; the batch size is 500 ms). On the one hand, the processing time decreases with an increasing number of taxi updates per second as the overhead of the streaming framework is spread across more taxi updates. On the other hand, an increase in client requests leads to an increase in processing time, as proportionally more computations have to be performed. The effects of an increase of the client requests are less prominent for the separated KNN pipeline, as the spatial index reduces the computational complexity at the expense of having to build it for every batch.

## 4 Discussion and Future Work

The first results presented in this work show that it is possible and likely beneficial to use a streaming framework for finding potential matches between taxis and the travelers requesting them. However, it is still unclear what kind of pipeline architecture will perform well, in particular when considering the integration of external services and data providers. Our findings suggest using the spatiality inherent in the problem of finding the nearest taxis can greatly reduce the computational efforts, leading to a speedup in the order of one magnitude in our case.

The use of realistic taxi data in combination with a tuneable streaming source allowed us to quickly test various streaming pipelines in a reproducible setting. As next step we planned a

[3] The Open Source Routing Machine (OSRM) can be downloaded from project-osrm.org.
[4] Go is a systems programming language developed by Google, see golang.org.

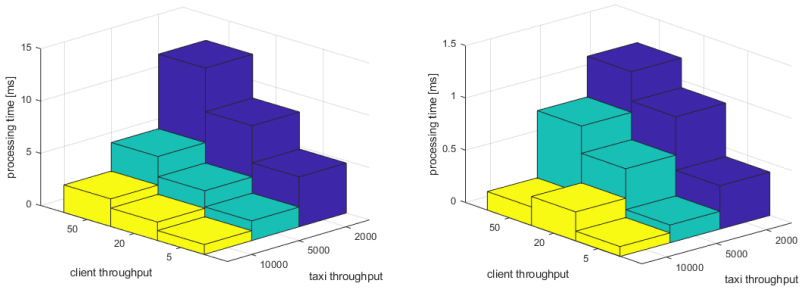



**Figure 2** Normalized processing times of the two pipelines presented in Figure 1 (cartesian pipeline on the left, separated KNN on the right).

more in-depth examination of the scalability of various architectures, e.g., by varying the number of workers in the cluster, by changing the partitioning (of work resp. data), and by employing a reusable spatial index that can be updated with new positions in every batch. In addition, we here omitted the integration of other services (e.g., OSRM for exact routing). Integrating such a service is particularly problematic as it can lead to enormous (and unpredictable) delays which in turn lead to instabilities in the streaming framework. Finally, in line with the goal of enabling people to share their taxis, we work on an addition to the streaming pipeline that considers potentially occupied taxis and lets taxis pick up multiple passengers. Especially the latter part requires a more complex approach, as solely solving the nearest neighbor problem often will not lead to a globally optimal solution.

**Acknowledgements.** This research was supported by the Swiss National Science Foundation (SNF) within NRP 71 “Managing energy consumption” and is part of the Swiss Competence Center for Energy Research SCCER Mobility of the Swiss Innovation Agency Innosuisse.

## References

1 Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The r*-tree: an efficient and robust access method for points and rectangles. In *Acm Sigmod Record*, volume 19, pages 322–331. Acm, 1990.

2 W Frank Domoney, Naseem Ramli, Salma Alarefi, and Stuart D Walker. Smart city solutions to water management using self-powered, low-cost, water sensors and apache spark data aggregation. In *Renewable and Sustainable Energy Conference (IRSEC), 2015 3rd International*, pages 1–4. IEEE, 2015.

3 Moshe Givoni and David Banister. *Moving towards low carbon mobility*. Edward Elgar Publishing, 2013.

4 Altti Ilari Maarala, Mika Rautiainen, Miikka Salmi, Susanna Pirttikangas, and Jukka Riekki. Low latency analytics for streaming traffic data with apache spark. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 2855–2858. IEEE, 2015.

5 Paras Mehta, Christian Windolf, and Agnès Voisard. Spatio-temporal hotspot computation on apache spark (gis cup). In *24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2016.

6 Harvey J Miller. Beyond sharing: cultivating cooperative transportation systems through geographic information science. *Journal of Transport Geography*, 31:296–308, 2013.

7 Maninder Pal Singh, Mohammad A Hoque, and Sasu Tarkoma. Analysis of systems to process massive data stream. *CoRR, abs/1605.09021*, 2016.

# Classification of regional dominant movement patterns in trajectories with a convolutional neural network

**Can Yang[1] and Győző Gidófalvi[2]**

1 **Department of Urban Planning, KTH, Royal Institute of Technology in Sweden**
cyang@kth.se

2 **Department of Urban Planning, KTH, Royal Institute of Technology in Sweden**
gyozo@kth.se

## 1 Introduction

Various movement patterns have been discovered in trajectories, which are valuable in studying the contextual behavior of tracked objects [15]. Most of the previous studies have been concentrated on detecting of specific patterns in trajectories, such as flock, leadership and convergence [9] or sequential patterns [14]. In practice, movement patterns can be much more diverse and complex, e.g., clockwise, and zigzag, which calls for a unified, effective and robust approach for classification.

In deep learning field, convolutional neural network (CNN) have achieved superior performance in classification of image [8], video [6], text [7] and voice data [4]. In terms of vector data, computer vision approaches have been designed to process point set data for shape matching and classification [1, 11, 12, 13]. However, applying CNN for classification of trajectory data is relatively unexplored, which confronts two challenges. Firstly, trajectory data includes two additional pieces of information, namely the connectivity between points and direction, which can be primary features in movement pattern classification. Secondly, both the number of trajectories in a set and points in a trajectory can be variable. They should be considered together with the variations in point positions.

To address the above two challenges, this paper proposes a deep learning approach for classifying *regional dominant movement pattern* (RDMP), which is defined as the movement pattern followed by the *majority* of a trajectory set within a target *region*. To achieve this objective, the proposed approach defines a *directional flow image* (DFI) by mapping a trajectory set to a grid space according to its extent and storing *local directional flow* in multiple channels at each grid, i.e., the pixel of DFI. The benefit is that a trajectory set with the aforementioned variations can be transformed into an image in *fixed* shape. Subsequently, a CNN called TR-Net is designed for classification of DFI, which is trained on synthetic trajectory data and a considerably high accuracy is achieved. In summary, the approach adds a bridge between deep learning and trajectory pattern classification.

## 2 Methodology

### 2.1 RDMP category definition

Based on the commonly encountered movement patterns summarized in [3], this paper firstly defines 10 RDMP categories as move straight, turn left, turn right, converge, diverge, cross, reverse, zigzag, clockwise and counter-clockwise. Some examples are displayed in Figure 1.

Spatial big data and machine learning in GIScience, Workshop at GIScience 2018.
Editor: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, Peter Kiefer;

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

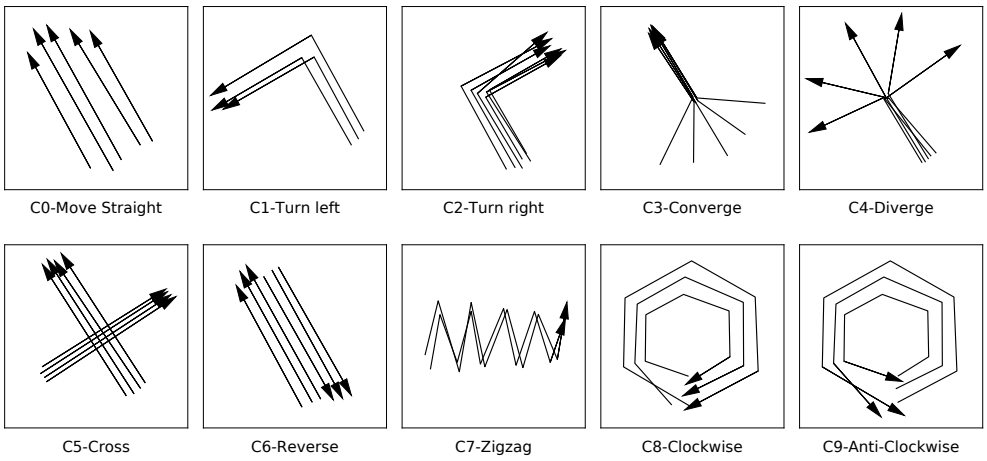
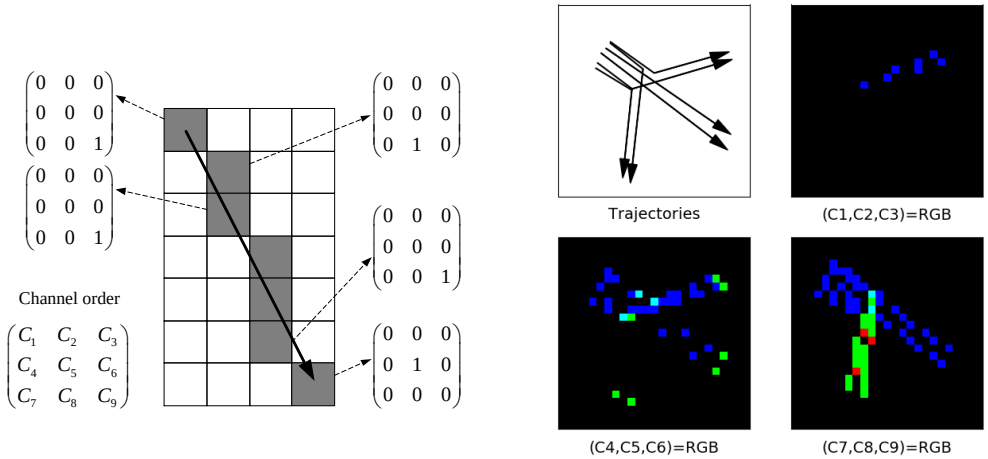



**Figure 1** Category definition of regional dominant movement pattern (category ID and name)




**(a)** DFI of a single trajectory **(b)** DFI ($25 \times 25$) of a trajectory set

**Figure 2** Illustration of directional flow image.

These definitions are highly conceptual and fuzzy where variations in shapes, the number of points and the number of trajectories are tolerated.

## 2.2 Directional flow image and TR-Net

A DFI is created from a trajectory set to store the local directional flow information. Given a trajectory set, its extent can be firstly partitioned into a grid space $G$ in shape of $H \times W$ indicating the number of grids in height and width. From $G$, a DFI is created as an image with 9 channels where each pixel represents a grid $g \in G$. The shape of DFI is written as $9 \times H \times W$. The channel $C_i$ at grid $g$ stores the number of traversals in the trajectory set from $g$ to its neighbouring grid indexed by $C_i$ as illustrated in Figure 2(a). The 9 channels cover the $\mathcal{N}_8$ neighbours and the grid itself, where $C_5$ represents traversals to the grid itself indicating the end of a trajectory.

When it comes to classification of DFI, the task is similar to the handwritten digit recognition problem studied in [10] where a CNN called LeNet was designed. TR-Net adjusts

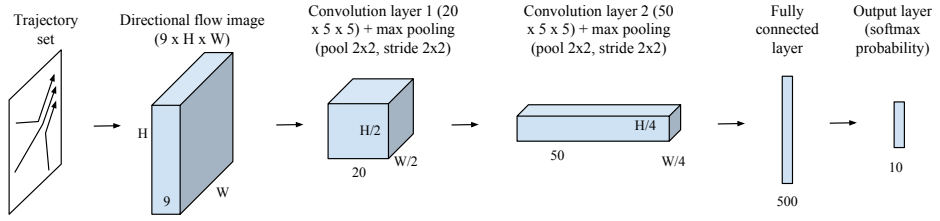



**Figure 3** Architecture of TRNet

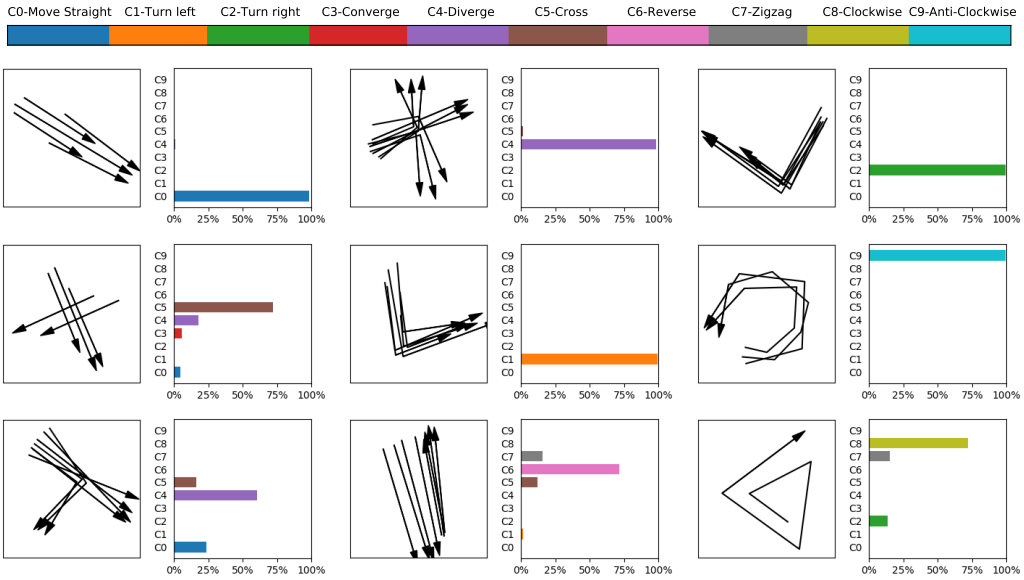



**Figure 4** Demonstration of some classification results using DFI size of $10 \times 10$

LeNet by replacing the first input layer as a $9 \times 10 \times 10$ DFI. The detailed information is shown in Figure 3. In this paper, TR-Net is implemented with Keras [2].

## 2.3 Synthetic training data generation

The synthetic training data generation is divided into two steps. In the first step, trajectory seeds are drawn manually according to the definition of 10 categories, as shown in Figure 1. Each seed is a set of trajectories following a specific RDMP. Different types of variations in a trajectory set are considered including the number of points inside a trajectory, the number of trajectories in a set and variations in shape. To reduce the complexity, all the trajectory seeds are draw within a fixed area bounded by point (0,0) and (10,10) in the Cartesian coordinate system. By setting a grid size of 1.0, the shape of DFI is thus fixed to be $9 \times 10 \times 10$. When applying TR-Net to other datasets, an adaptive grid size can be set proportional to the extent of a trajectory set.

In the second step, data augmentation is performed by rotating the trajectory seeds. It can be observed from Figure 1 that all the classes are invariant to rotation. In other words, a group of trajectories rotated by angle $\alpha$ still belong to the same class. Given a specific $\alpha$, sample size can be increased by $360/\alpha$ times. In this paper, 34 seeds are drawn manually for each of the 10 classes and rotated by 20 degrees. Finally, $34 \times 10 \times 360/20 = 6120$ trajectory sets and DFIs are generated.

## 3 Results and discussion

In training TR-Net, the 6120 DFIs generated from the synthetic trajectory sets are shuffled and split into a validation dataset (60%) and a test dataset (40%). The accuracy on test dataset increases from 20% to 90% after 9 epochs of training and finally arrives at 99.3% after 20 epochs. Some results are displayed in Figure 4, which demonstrates the effectiveness of the approach.

Several limitations of TR-Net are discussed below. In this paper, all the training data are created within an extent of (10,10) and the shape of DFI is fixed to be $9 \times 10 \times 10$. The restriction primarily comes from the consideration that a conventional CNN generally requires a fixed input layer in order to train the weights of fully-connected layer. Advanced architecture such as spatial pyramid pooling layer [5] has been designed recently to eliminate the constraint. Improving TR-Net along this direction will be planned as a future work. In addition, the patterns currently classified are regional dominant, which implies that it cannot discover a small area exhibiting a different pattern inside a regional dominant one. Similarly, a common phenomenon is that a large trajectory set can exhibit multiple patterns at different scales in space, which cannot be discovered by TR-Net. These problems may be addressed with by introducing object detection approaches, which is also left as a future work. Moreover, the performance of TR-Net will be further evaluated with different types of real-world data.

## 4 Conclusion

The paper proposed a deep learning approach to discover regional dominant movement patterns in a trajectory set. A directional flow image was firstly defined to store local movement information of trajectories in a multi-spectral image. Subsequently, a convolution neural network called TR-Net was designed for classifying that type of image. Experiments on synthetic data demonstrated that the approach achieved a considerably high accuracy that was robust to different types of variations in trajectories.

**Acknowledgements.** The first author is a Ph.D. student funded by the Chinese Scholarship Council (CSC) and KTH. Acknowledgements are also given to Kartografiska Sällskapet in Sweden for providing the travel grant.

## References

1 S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, Apr 2002.

2 François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

3 Somayeh Dodge, Robert Weibel, and Anna-Katharina Lautenschütz. Towards a taxonomy of movement patterns. *Information Visualization*, 7(April):240–252, 2008.

4 Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, May 2013.

5 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1904–1916, 2015.

**6** Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1725–1732, 2014.

**7** Yoon Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014.

**8** Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems*, NIPS'12, pages 1097–1105, 2012.

**9** Patrick Laube, Stephan Imfeld, and Robert Weibel. Discovering relative motion patterns in groups of moving point objects. *International Journal of Geographical Information Science*, 19(6):639–668, 2005.

**10** Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

**11** Xin Shu and Xiao-Jun Wu. A novel contour descriptor for 2d shape matching and its application to image retrieval. *Image and Vision Computing*, 29(4):286 – 294, 2011.

**12** Xinggang Wang, Bin Feng, Xiang Bai, Wenyu Liu, and Longin Jan Latecki. Bag of contour fragments for robust shape classification. *Pattern Recognition*, 47(6):2116–2125, 2014.

**13** Chunjing Xu, Jianzhuang Liu, and Xiaoou Tang. 2D shape matching by contour flexibility. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):180–186, 2009.

**14** Can Yang and Győző Gidófalvi. Mining and visual exploration of closed contiguous sequential patterns in trajectories. *International Journal of Geographical Information Science*, 32(7):1282–1304, 2018.

**15** Y U Zheng. Trajectory Data Mining : An Overview. *Tist*, 6(3):1–41, 2015.

# Spatial Big Data for Human-Computer Interaction

**Ioannis Giannopoulos**
Vienna University of Technology
igiannopoulos@geo.tuwien.ac.at

**Abstract**

The importance of spatial data for the area of human-computer interaction is discussed in this vision paper as well as how machine learning and spatial big data can be utilized for optimizing and adapting the interaction modalities in outdoor spaces. This paper briefly introduces and tries to connect previous work in order to highlight the vision towards a space adaptive personalized system and list important research questions.

**2012 ACM Subject Classification** Human-centered computing

**Keywords and phrases** interactive systems, urban structures, human behavior

## 1 Introduction

We often use several types of digital devices in outdoor spaces, e.g., smartphones, tablets, smartwatches, in order to interact with the surrounding environment [4] and other people which are not necessarily co-located. Depending on several characteristics of our surroundings, e.g., environmental complexity [5], type of space (e.g., dense urban area, open green field) as well as our activity, e.g., interaction with a map [7], the interface of such a device could adapt accordingly when we ask for it [6] in order to ease the interaction for the current situation.

Our vision is a personalized interactive system that is able to recognize the activity of its user, the user's cognitive load [1], the user's familiarity with the surrounding environment as well as the type of the environment and adapt accordingly.

## 2 Relevant Big Spatial Data

In order to come closer to the presented vision, several data sources are necessary for the prediction of the relevant classes. In our current approach we utilize environmental data, human behavior data as well as geo-spatial semantics[1] [9]. The environmental data we utilize are based on street networks, e.g., types of intersections [2] in the surroundings of the user, the geo-spatial semantics of this area, or even the probability for certain routes from the user's current location [3]. Furthermore, we use eye tracking technology in order to track the visual attention of the user [8], e.g., what is the user looking at, as well as location tracking in order to record the user's mobility patterns, aiming at understanding the underlying human behavior.

All this data is tracked with different frequencies and the amount of data that has to be analyzed at each relevant time instance varies according to the source. For instance, although typical mobile eye tracking devices run with a frequency of 120 Hz, the size of data is very lightweight compared with the tracked environmental data [2]. For instance, the types and number of intersections in the spatial proximity of the user can be extracted and compared

[1] `http://linkedgeodata.org/About`

Spatial big data and machine learning in GIScience, Workshop at GIScience 2018.
Editors: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, Peter Kiefer

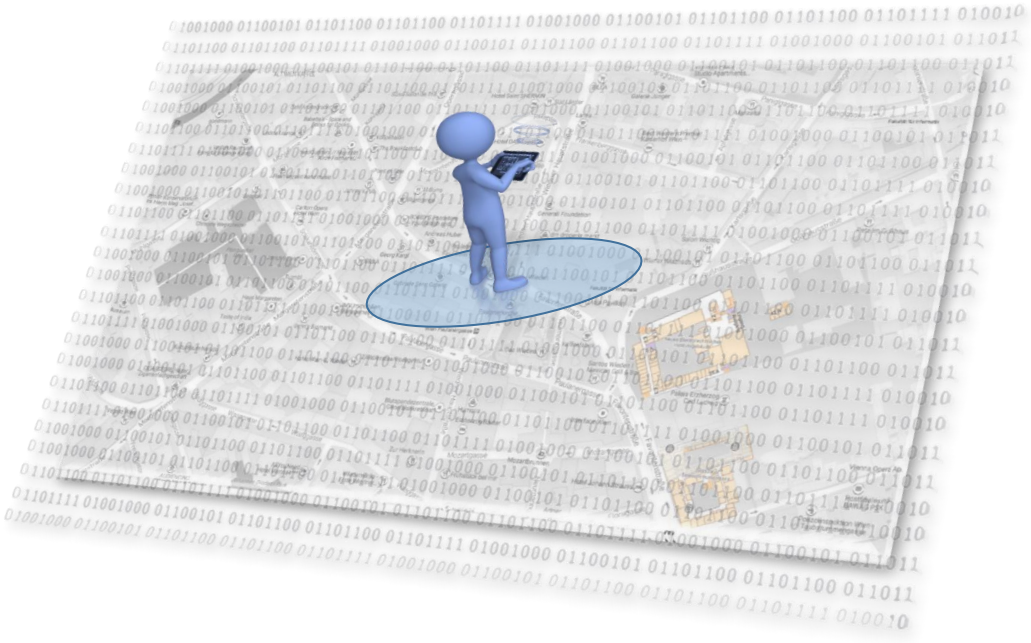


**Figure 1** A hypothetical scenario of a user interacting with a digital system at a specific location. The system adapts based on the users activity and the spatial characteristics in the local proximity of the user, e.g., based on the spatial semantics and the surrounding urban structure.

with the same type of data of a larger relevant region in order to extract features for machine learning. Figure 1 illustrates such an example.

Although the size and variety of the mentioned data sources is cumbersome to deal with, the big challenge is to find existing and novel programming paradigms to deal with all these different sources and decide for an efficient data fusion that would allow to come closer to the presented vision.

## 3 Research Questions

- Which types of urban classes (e.g., historic and touristic places) can we predict and what type of data do we need?
- Which land-cover classes can be of potential interest for adaptive interfaces?
- What type of human behavior and spatial data are necessary in order to detect familiarity with the surrounding environment?
- Which classes of human spatial behavior should we be able to predict?

The above research questions are not complete by any means. These questions demonstrate important aspects of the presented vision and highlight parts of the research we are currently conducting in order to come closer to the presented vision.

## 4 Current and Future Challenges

Of course there are several challenges when presenting research which is based on predictions. How good can these predictions be? How good do we need them to be? How much from these big data do we actually need for good predictions? Can we be fast enough for real time

applications? What is the elasticity between good predictions and real time application? These are only a few of the important challenges that demonstrate the decision making process that we have to go through when we have to cope with real time applications, the size and variety of data.

## References


1 Andrew T. Duchowski, Krzysztof Krejtz, Izabela Krejtz, Cezary Biele, Anna Niedzielska, Peter Kiefer, Martin Raubal, and Ioannis Giannopoulos. The index of pupillary activity: Measuring cognitive load vis-à-vis task difficulty with pupil oscillation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 282:1–282:13, New York, NY, USA, 2018. ACM. URL: `http://doi.acm.org/10.1145/3173574.3173856`, `doi:10.1145/3173574.3173856`.

2 Paolo Fogliaroni, Dominik Bucher, Nikola Jankovic, and Ioannis Giannopoulos. Intersections of our world. In Stephan Winter, Amy Griffin, and Monika Sester, editors, *International Conference on Geographic Information Science*, Leibniz International Proceedings in Informatics (LIPIcs), pages 3:1–3:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

3 Paolo Fogliaroni, Marvin Mc Cutchan, Gerhard Navratil, and Ioannis Giannopoulos. Unfolding urban structures: Towards route prediction and automated city modeling. In Stephan Winter, Amy Griffin, and Monika Sester, editors, *International Conference on Geographic Information Science*, Leibniz International Proceedings in Informatics (LIPIcs), pages 34:1–34:6, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

4 Ioannis Giannopoulos, Peter Kiefer, and Martin Raubal. Mobile outdoor gaze-based geohci. In *Geographic Human-Computer Interaction, Workshop at CHI 2013*, pages 12–13, 2013.

5 Ioannis Giannopoulos, Peter Kiefer, Martin Raubal, Kai-Florian Richter, and Tyler Thrash. Wayfinding decision situations: A conceptual model and evaluation. In Matt Duckham, Edzer Pebesma, Kathleen Stewart, and Andrew U. Frank, editors, *Geographic Information Science*, pages 221–234, Cham, 2014. Springer International Publishing.

6 Peter Kiefer, Ioannis Giannopoulos, Vasileios Athanasios Anagnostopoulos, Johannes Schöning, and Martin Raubal. Controllability matters: The user experience of adaptive maps. *GeoInformatica*, 21(3):619–641, 2017.

7 Peter Kiefer, Ioannis Giannopoulos, and Martin Raubal. Using eye movements to recognize activities on cartographic maps. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 488–491. ACM, 2013.

8 Peter Kiefer, Ioannis Giannopoulos, Martin Raubal, and Andrew Duchowski. Eye tracking for spatial research: Cognition, computation, challenges. *Spatial Cognition & Computation*, 17(1-2):1–19, 2017.

9 Marvin Mc Cutchan and Ioannis Giannopoulos. Geospatial semantics for spatial prediction. In Stephan Winter, Amy Griffin, and Monika Sester, editors, *International Conference on Geographic Information Science*, Leibniz International Proceedings in Informatics (LIPIcs), pages 63:1–63:6, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

# Unsupervised Clustering of Eye Tracking Data

**Fabian Göbel**
Institute of Cartography and Geoinformation, ETH Zurich, Zurich, Switzerland
goebelf@ethz.ch

**Henry Martin**
Institute of Cartography and Geoinformation, ETH Zurich, Zurich, Switzerland
martinhe@ethz.ch

**Abstract**

The reading behavior on maps can strongly vary with factors such as background knowledge, mental model, task or the visual design of a map. Therefore, in cartography, eye tracking experiments have a long tradition to foster the visual attention. In this work-in-progress, we use an unsupervised machine learning pipeline for clustering eye tracking data. In particular, we focus on methods that help to validate and evaluate the clustering results since this is a common issue of unsupervised machine learning. First results indicate that validation using the silhouette score alone is a poor choice and should, for example, be accompanied by a visual validation using t-distributed stochastic neighbor embedding (t-SNE).

**2012 ACM Subject Classification** Computing methodologies → Cluster analysis

**Keywords and phrases** eye tracking; unsupervised machine learning; clustering; map task

**Funding** Swiss National Science Foundation Grant No.: 200021_162886.

## 1 Introduction

Human factors such as background knowledge or mental model as well as the cartographic design of a map have a strong impact on how people read and understand maps. Therefore, eye tracking experiments have a long history in cartographic research (refer to [5] for an overview). Answering where and how map readers spent their visual attention on a map is crucial to evaluate its design and usability. However, due to the high data rates of up to 1500 Hz that are needed to track the quickest movements of the human eye, these experiments usually result in large log files, which requires using automated data analysis.

There are many examples of the successful application of supervised machine learning methods on eye tracking data. For instance, Kiefer et al. recognize six predefined map activities based on gaze patterns [4]. However, the application of supervised learning methods requires ground truth information. In cases where these are not available, unsupervised learning techniques are an option to explore the data. An example is the work by Jonietz et al. who used GPS trajectory data to search for user groups with similar changes in mobility behavior [3].

This work evaluates the suitability of unsupervised machine learning methods for clustering eye tracking data. In the following, we first propose a clustering framework and discuss each of its steps with the help of real data from a prior eye tracking experiment. This allows to exemplify the framework and test a wide-range of different hyperparameters. Finally, the results are validated and evaluated by comparing the clustering results with t-SNE visualization, before we draw a conclusion and give an outlook on future work.

Spatial big data and machine learning in GIScience, Workshop at GIScience 2018.
Editor: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, Peter Kiefer;

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

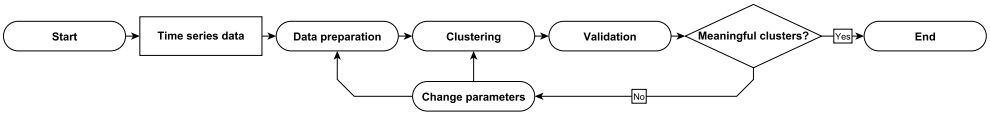



**Figure 1** Overview of the clustering pipeline.

**Table 1** Features used for clustering. Columns are interpreted as {statistic x indicator x type}. For instance, the first column reads: "total duration of trial".

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| statistics | total | mean, variance, median, 95th percentile, 5th percentile, total | total | rate of | mean, variance, weighted mean | covariance of | variance of | total |
| indicator | duration of | distance between succeeding, duration, angle between succeeding | number of, changes of curvature of, changes of direction of | | dispersion x of, dispersion y of | dispersion of | duration on | dwells on |
| type | trial | fixations | | | | | City A, City B, Legend, Map | |

## 2 Framework for clustering Eye Tracking Data

The proposed framework provides a structured approach to test different options of data preparation methods and clustering algorithms together with a wide-range of hyperparameters. Figure 1 gives a general overview of the iterative pipeline of the framework.

In order to explore patterns in gaze behavior, we use the data from an experiment published previously [2]. Participants performed a common comparison task on a map, which involved interacting with the legend. Three different maps with varying symbol density and three legend types were tested. In total, 18 samples were extracted for the clustering process.

### 2.1 Data Preparation

Because most clustering algorithms require tabular data, the pipeline start with a feature extraction step. We manually create features, that are specially tailored to reflect the scan pattern of a user's gaze behavior. Table 1 gives an overview of the features which quantify the spatio-temporal characteristics of the eye tracking data.

Feature extraction from time series data quickly leads to a high number of features: In our case, we create 37 features, which is a high value regarding 18 samples. Therefore, we include a feature selection step in our framework which filters features based on their dispersion. We use the interquartile ratio as it is robust to outliers [3].

### 2.2 Clustering

The next step is the clustering itself. Our framework applies three common clustering algorithms: *KMeans*, *spectral clustering*, and *DBscan*. KMeans is a simple and well-known clustering algorithm, however, it can only represent convex clusters. To address the case of non-convex clusters, we also apply spectral clustering. Furthermore, in this step, we test DBscan, which is often used for GPS trajectory analysis and does not require the definition of the number of clusters.

### 2.3 Validation

In general, unsupervised learning problems, such as clustering, are very hard to evaluate, because of the missing ground truth information. Although, in our scenario, the clustering algorithms produce a valid result for most hyperparameter configurations, the challenge to identify meaningful results remains. We propose a two-step approach to evaluate resulting

clustering configurations: At first, we apply internal clustering validation measures, namely the silhouette score (SIL)[7] and the distribution of users per class (e.g. uniform distribution of users over all classes vs. all but one user in the same class) to exclude meaningless or trivial results. One disadvantage of the SIL is that it favors convex clusters, which might not reflect the structure of the data. Therefore, we do not use the SIL[1] for ranking but as a threshold to filter the results.

In a second step, we evaluate the clustering results by visual inspection similar to [1]. A dimension reduction method projects high-dimensional data to a human readable two-dimensional space. We chose t-distributed stochastic neighbor embedding (t-SNE), which has been proven to be a powerful tool for the visualization of high-dimensional data [6, 8]. Although, t-SNE can unveil structure in the data the method can not be used as a quantitative clustering algorithm, as it does not preserve distances [8].

In our framework, we use t-SNE to visualize structure in the data by projecting the unlabeled, high-dimensional data into two dimensions. We then color the data points in the resulting two-dimensional plot according to the clustering results. This allows to visually evaluate if the clustering result corresponds to the result of the t-SNE projection. If both results are corresponding, the clustering result is likely to be meaningful. If the results are not in line, it is still possible that both algorithms have uncovered different meaningful patterns but we can not validate our clustering result.

## 2.4 Example

We calculated all possible clustering results of the example dataset, by running the different algorithms with a wide range of hyperparameters (comparable to an extensive grid search). 29700 parameter combinations were tested, which results in 9952 valid clusterings (a result is valid if it has more than 1 cluster and if the algorithm converged). Following the framework, results with a SIL below 0 were excluded and only the ones with an interesting distribution of users among the different classes were manually picked. We then visualize the promising picks using the t-SNE algorithm and color the data points according to the label of the clustering result.

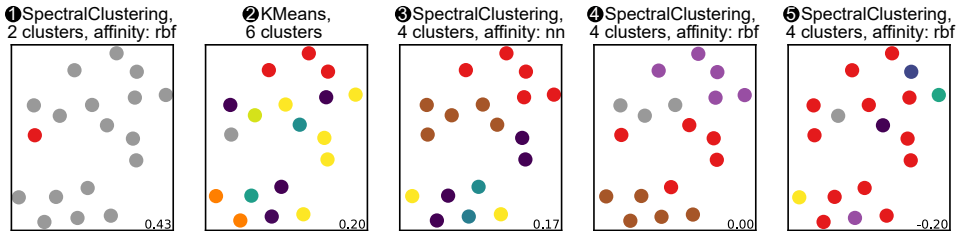
Figure 2a shows one single t-SNE projection colored with five different clustering results (result with negative SIL is included for comparison). The graphs show that the solution with the highest SIL (0.43) is a trivial solution where all points but one are in the same cluster. Although the forth clustering has a rather low silhouette score, it looks very promising. This might be an example of the inability of the SIL to acknowledge non-convex clusters.

Figure 2b shows the t-SNE label distribution of the forth clustering from Figure 2a with different perplexity values. Due to the stochastic nature and the dependency on the perplexity of the t-SNE, the position of a point and the distance between points varies greatly between the different results. The graphs show that the results by the spectral clustering are fairly robust and correspond well to the t-SNE projections with different hyperparameters.

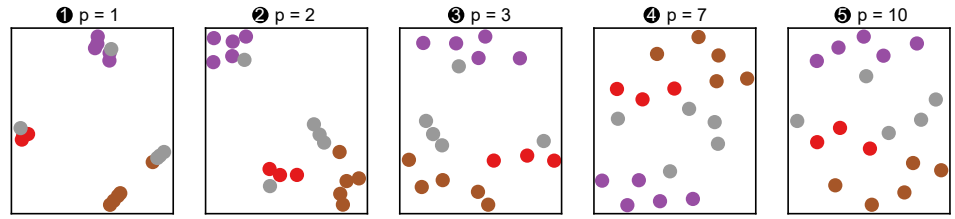
# 3 Discussion and Conclusion

In this work, we proposed a framework for unsupervised clustering with a special focus on the validation of the clustering results. The approach was exemplified with a data set of an eye tracking experiment on cartographic maps. We showed that a validation based on

[1] The SIL ranges between -1 and 1; values closer to 1 indicate well defined and separated clusters




**(a)** T-SNE graphs with a constant perplexity of 7. Colors indicate results of different clusterings. Headers state the respective algorithms. The SIL is shown at the bottom right of each plot.




**(b)** T-SNE graphs with varying perplexity (p). Coloring uses the same labels as in plot 4 from a).

the silhouette score alone can be misleading and should be accompanied by other validation methods. Promising results could be achieved by visual inspection based on t-SNE. Although we have demonstrated the framework with only very few data points, the approach is suited for large-scale datasets. The main contribution of this work-in-progress is the introduction of a technique to find a theoretically good solution in a structured way.

However, the question of how to finally choose the best clustering result is an ongoing topic of discussion. Although we could demonstrate the potential of t-SNE for evaluation of a clustering result, domain knowledge is crucial for both, choosing meaningful parameters and extracting more elaborate features that describe the interaction of a human with a map and the inspected map content in more detail.

## References


1 K. Chen and L. Liu. Validating and refining clusters via visual rendering. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 501–504. IEEE, 2003.

2 F. Göbel, P. Kiefer, I. Giannopoulos, A. T. Duchowski, and M. Raubal. Improving Map Reading with Gaze-Adaptive Legends. In *ETRA '18: 2018 Symposium on Eye Tracking Research & Applications*. ACM, 2018.

3 D. Jonietz, D. Bucher, H. Martin, and M. Raubal. Identifying and Interpreting Clusters of Persons with Similar Mobility Behaviour Change Processes. In *The Annual International Conference on Geographic Information Science*, pages 291–307. Springer, 2018.

4 P. Kiefer, I. Giannopoulos, and M. Raubal. Using Eye Movements to Recognize Activities on Cartographic Maps. *Proceedings of the 21st SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 498–501, 2013.

5 P. Kiefer, I. Giannopoulos, M. Raubal, and A. Duchowski. Eye Tracking for Spatial Research: Cognition, Computation, Challenges. *Spatial Cognition & Computation*, 17(1-2):1–19, 2017.

6 L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9:2579–2605, Nov 2008.

7 P. J. Rousseeuw. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

8 M. Wattenberg, F. Viégas, and I. Johnson. How to Use t-SNE Effectively. *Distill*, 2016.

# Collections of Points of Interest: How to Name Them and Why it Matters

**Gengchen Mai[1], Krzysztof Janowicz[1], Yingjie Hu[2], Song Gao[3], Rui Zhu[1], Bo Yan[1], Grant McKenzie[4], Anagha Uppal[1], and Blake Regalia[1]**

1 **STKO Lab, University of California, Santa Barbara, CA, USA**
{gengchen_mai | ruizhu | boyan | anagha.uppal | regalia | jano}@geog.ucsb.edu
2 **Department of Geography, University of Tennessee, Knoxville, TN, USA**
yhu21@utk.edu
3 **Department of Geography, University of Wisconsin, Madison, WI, USA**
song.gao@wisc.edu
4 **Department of Geography, McGill University, QC, Canada**
grant.mckenzie@mcgill.ca

## Abstract

The Web-accessible of large, global-coverage databases of Points of Interest (POI) as well as social sensing techniques to study how humans behave towards these POI, i.e., when they visit them, how they write about them, the sequences in which they visit these POI, and so forth, have lead to researchers and companies utilizing POI to represent regions and their affordances. For instance, one can characterize neighborhoods by the types of POI they contain as well as their frequency or try to extract spatial footprints of functional regions based on the spatial patterns in which these POI are distributed over geographic space. Such perspectives, however, ignore the spaces and places in-between these POI as well as the transportation infrastructure that defines many kinds of regions and their interaction. Consequently, it is often more beneficial to take a step back and explicitly state that one considers collections of POIs to delineate an area, e.g., to associate it with activity types such as tourism. To give a concrete example, the area enclosed by the locations from which people are frequently taking pictures in a city such as New York may be characterized as an area of interest (e.g., Time Square) to differentiate it from a merely point-based perspective. Unfortunately, the community has not yet agreed on a common term for such areas and instead uses similar terms from domains such as human geography or remote sensing as proxies. While they often have overlapping meanings, we believe that it would be beneficial to discuss the similarities and differences of terms such as neighborhood, functional region, vague cognitive region, and so forth as compared to areas of interest and other POI collections.

**1998 ACM Subject Classification** Information systems → Information systems applications → Spatial-temporal systems → Geographic information systems

**Keywords and phrases** area of interest, vague region, POI, neighborhood, data-driven methods.

## 1 Introduction

*Points of interest* (POIs) refer to point locations in urban or rural environments that support human activities or attract the attentions of people. POIs have been widely used in academic research and industrial applications as a proxy to understand human activities in an area. However, POIs are also criticized because they force the spatial footprints of geographic features to be points, thereby essentially ignoring their extent. This geometric generalization


Spatial big data and machine learning in GIScience, Workshop at GIScience 2018, Melbourne, Australia.
Editors: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, Peter Kiefer;



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

works in many applications but can also cause problems especially at large cartographic scales. In addition, it often makes sense to study collections of Points of Interest and their spatial distribution. These collections necessarily cover an area and this area may be of interest in various contexts such as reasoning about the activity spaces of people, delineating regions frequented by tourists, estimating the boundaries of neighborhoods and functional regions and so on. Such POI collections, however, can only act as proxy for the above mentioned concepts and therefore it is often beneficial to talk about POI collections and explain how they are used to derive further information. Recently, the term *area of interest* (AOI) has been used [6, 5] to describe one kind of POI collections, namely areas that attract and support various human interests and activities. At a first glance, AOI seems to be a simple extension of POI by enabling polygon representations of spatial footprints. However, POIs and AOIs do not have a one-to-one mapping relationship. For example, an AOI can be defined as a concave polygon containing locations photographed by tourists. Another example would be an AOI formed by all POI within a certain area that afford nightlife activities. Clearly, while such an area is a worthwhile nit of study, it should not be confused with concepts such as neighborhood.

AOI is also related to some other spatial concepts, and their distinctions are not always clear. One of such concepts is *administrative region* which is often defined by governments and used for various socioeconomic and political purposes. The well-defined semantics and the clearly-delineated boundaries make it easy to distinguish one administrative region from another. While administrative regions are defined in a top-down manner by authorities, other spatial concepts have been proposed to represent geographic regions extracted in a data-driven manner or from a community/culture-centered perspective. Examples include *(vague) neighborhoods* [2], *functional regions* [3], and *vague cognitive regions* (VCR) [7]. Vague cognitive regions, for instance, emphasize the cognitive aspect of places and reflect informal ways people organize their knowledge about places and how they share common attitudes towards them [7]. Functional regions focus on one or multiple functions provided by a specific area and model the (often hierarchical) spatial interactions among the places within this area. Neighborhoods are used to represent the structure and composition of the urban space in social perspectives such that spatial units within the same neighborhood share a similar character [1]. Neighborhoods also have a strong historical perspective as they often arise by smaller communities being incorporated.

AOI is also a type of data-driven geographic regions [6, 5], and some similarities can be found between AOI and other concepts. Similar to *vague cognitive regions*, AOIs exist in people's perception and have vague boundaries. Different AOIs may also support different activities (e.g., recreational areas for entertainments and commercial centers for shopping activities) and provide different functions similar to *functional regions*. The spatial units within an AOI can also share similar attributes such as the concept of *neighborhood* (e.g., higher housing price for commercial centers). Another concept that is similar to AOI are *regions of interest* (ROI) which is a term frequently used in remote sensing and other domains to refer to the study regions defined by researchers. These similarities makes it difficult to distinguish AOI from other concepts. An even more important question is whether AOI is a trivial concept that can be simply replaced by other terms or whether it adds something that cannot be expressed otherwise. This paper is intended to start a discussion around this topic by reviewing the similarities and differences between AOI and related concepts.

## 2 Comparison of AOI and Related Concepts

In this section, we review AOI and related spatial concepts and discuss their similarities and differences. Table 1 provides a comparative summary of concepts from six important perspectives: 1) *Consensus*: are the semantics and spatial footprints of these concepts commonly agreed among people? 2) *Footprints*: which type of geometry is typically used to represent this concept? 3) *Boundary*: does this concept have a vague or crispy boundary? 4) *Relatively static*: is this extent of the region relatively static or does it change frequently over time? 5) *Spatial Interaction*: is this concept defined based on the interactions among the places within this area? 6) *Named region*: it naming a key aspect of this type of region? While the concepts of *POI* and *administrative region* are also compared in the table, they are not discussed in separate sections since they have clear distinction with AOI outlines before.

**Table 1** A comparison among different spatial concepts. (Yes(Y)/No(N))

| | Consensus | Footprints | Boundary | Relatively static | Spatial Interaction | Named Place |
|---|---|---|---|---|---|---|
| Administrative Region | Semantics and footprints | Polygons | Crispy | Y | N | Y |
| Point Of Interest | Semantics and footprints | Points | - | Y | N | Y |
| Vague Cognitive Region | Semantics | Polygons | Vague | Y | N | Y |
| Functional Region | Applications dependent | Polygons | Crispy/Vague | Y/N | Y | N |
| Neighborhood | Applications dependent | Polygons | Crispy/Vague | Y/N | N | Y/N |
| Region Of Interest | Applications dependent | Polygons | Crispy | - | N | N |
| Area Of Interest | Applications dependent | Polygons | Vague | N | N | N |

### 2.1 Vague Cognitive Regions

While AOI is also a cognitive concept since the word "interest" can refer to various themes and the same theme can be interpreted or perceived differently by different people, it embodies less cognitive flavor compared with VCR [7]. VCRs are driven by individual and cultural beliefs about thematic properties of geographic places. VCR as a term is strongly tied with the intrinsic vagueness and fuzziness of many geographic concepts, such as "downtown Santa Barbara" and "Southern California". Each VCR refers to a named cognitive geographic region which is commonly agreed on by some part of the population whereas the vagueness or disagreement arises when it comes to the regions spatial boundary. In contrast, since each AOI it not necessarily associated with a name, AOIs are not formed by consensus or common reference, let alone consensus on their spatial representations. Another distinction is that VCRs are relatively static whereas people will be interested in different areas at different times of the day which forms different AOIs. Note that terms such as *activity spaces* typically refer to a single-persons perspective and are mostly defined by travel patterns.

### 2.2 Functional Regions

The concept of *functional region* has different definitions in the literature of regional science and urban geography studies. One popular definition is that *functional regions* are usually characterized by connections or spatial interactions between different areas and locational entities. Another one is from Hartshorne [4] with emphasis on supporting the particular set of activities and depending on the structure of the area. The functions of certain (sub)regions are originally defined in urban planning and then reshaped by needs arising from human activities. Researchers use POIs that afford specific types of human activities as a proxy to delineate functional regions with various co-location patterns of place types from a bottom-up perspective [3]. Because of the uncertainty in this bottom-up process, the boundaries of functional regions can be also vague. The distinctive character of functional regions is that

they focus on the (often hierarchical) spatial interactions between places within them and different combinations of functions which are not captured by any other concepts (and cannot be fully represented by collections of POI).

## 2.3 Neighborhoods

*Neighborhood* is a phrase frequently used in our everyday life. In most cases, administrative units at the country, state, and city level are well defined, but sub-city level tends to be where hard geospatial boundaries fall apart. Neighborhoods have become a common term used to describe an urban region that shares a similar *character*. In this way they can be considered areas of interest with specific focus on aspects of an urban setting that often differentiate parts of a city. Features such as socioeconomic status, demographics, housing prices, or even crime rate all play a role in how neighborhoods are perceived. There is a deluge of work exploring this topic, with many researchers focusing on a combination of characteristics as the basis for setting a boundary. For instance, work by Cranshaw et al. [2] explored the use of social media check-ins for clustering Points Of Interest which they refer to as *livehoods*. Neighborhood boundaries are a contentious issue in many parts of the world and have been historically used for redlining or other means of separating groups of people. One clear distinction between AOIs and neighborhoods it that the last mentioned captures the general characters of certain range of space in a slowly evolving, cultural, and history-based fashion, while AOIs are more similar to perspectives or views on space that highlight aspects along one or a few dimension. Their cultural relevance stems from usage patterns (e.g., sightseeing), not the historical development of an urban area.

## 2.4 Regions Of Interest

*Region of interest* is a general concept for describing a subset of data that is under study or serve some specific purposes. It is commonly used in domains such as medical imaging and computer vision. In geography, ROI is used to analyze remotely sensed imagery, and has been widely accepted as term in most commercial remote sensing software like ENVI. Compared to other concepts, ROI is usually defined by researchers from a perspective of data processing and analysis rather than conceptualization. Therefore, it is difficult and sometimes meaningless to have a consensus on the identification of ROI. Furthermore, most ROIs have crispy boundaries which makes them different from AOIs (which have rather vague boundaries). Nonetheless, researchers that delineate a specific area based on a type of human activity also define a region of interest, thereby showing similarities between the terms ROI and AOI. Put differently, an AOI sits in between areas merely selected for the purpose of a study, e.g., during sampling, and the shared experience of place associated with other terms such as vague cognitive regions. To refer back to the tourism and photography example, AOI can be defined by researchers based on geo-tagged images but are also experienced by citizens seeing how places attract tourists and the activities associated with tourism.

## 2.5 Areas Of Interest

The concept of AOI has two major emphases: *area* and *interest*. The first emphasis suggests that an AOI is geometrically represented as an *area* rather than a *point*. Based on this emphasis, AOI can be considered as an extension of the concept of POI. Meanwhile, an AOI is more than a collection of spatially co-located POIs. An AOI may also include regions that do not have any POI in the classical (static) sense but simply provide a scenic view [5]. Thus,

an AOI may also include the reversed viewshed of a POI. The second emphasis is on *interest*. This emphasis adds a layer of human cognition: an AOI does not refer to any geographic region but regions that can attract attention from people, e.g., by affording certain activities. Thus, an AOI may be an attractive regions that contain landmarks, commercial centers, and recreational areas as well as famous natural tourism resorts in a rural area, such as nature parks and famous mountains. Given its cognitive emphasis, AOI can be subjective: for a geographic region, different people may have different lists of AOI in mind due to their different cultures, education backgrounds, and personal interests, and so forth. In addition, agreement on the existence of an AOI does not mean agreement on their spatial extents nor the need for a name. Clearly, a town may have a region that is often included in the routes of bikers and runners without them sharing a name for this common path-fragment.

## 3 Conclusion

It is good scientific practice to question the necessity of a new concept. In this paper, we looked into the question whether the concept of AOI can be simply replaced by other existing spatial concepts. To do so, we compared the similarities and differences between AOI and related concepts from six perspectives, namely consensus, footprint, boundary, dynamics, spatial interaction, and name. Although some concepts like administrative region, ROI, and POI have rather clear distinctions from AOI, it is easy to confuse AOI with other concepts like VCR, functional regions, and neighborhoods, which are often extracted in a similar data-driven manner by researchers interested in Big Data. We believe that AOI (and other types of POI collections) is an unique concept which cannot be simply replaced by others, most importantly because it may lead to an oversimplification or even misrepresentation of other types of regions. AOI, for example, can be used as means of approximation of these other concepts. AOIs do not necessarily have names (compared with VCR), do not necessarily focus on spatial interactions (compared with functional regions), and have dynamic patterns (compared with neighborhoods). We hope that this comparison can start a useful discussion for future studies to select the most suitable term.

## References


1 Claudia J Coulton, Jill Korbin, Tsui Chan, and Marilyn Su. Mapping residents' perceptions of neighborhood boundaries: a methodological note. *American journal of community psychology*, 29(2):371–383, 2001.

2 Justin Cranshaw, Raz Schwartz, Jason Hong, and Norman Sadeh. The livehoods project: Utilizing social media to understand the dynamics of a city. In *ICWSM*, 2012.

3 Song Gao, Krzysztof Janowicz, and Helen Couclelis. Extracting urban functional regions from points of interest and human activities on location-based social networks. *Transactions in GIS*, 21(3):446–467, 2017.

4 Richard Hartshorne. *Perspective on the Nature of Geography*. Rand McNally, 1959.

5 Yingjie Hu, Song Gao, Krzysztof Janowicz, Bailang Yu, Wenwen Li, and Sathya Prasad. Extracting and understanding urban areas of interest using geotagged photos. *Computers, Environment and Urban Systems*, 54:240–254, 2015.

6 Jiajun Liu, Zi Huang, Lei Chen, Heng Tao Shen, and Zhixian Yan. Discovering areas of interest with geo-tagged images and check-ins. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 589–598. ACM, 2012.

7 Daniel R Montello, Alinda Friedman, and Daniel W Phillips. Vague cognitive regions in geography and geographic information science. *International Journal of Geographical Information Science*, 28(9):1802–1820, 2014.

# A Multi-scale Spatio-temporal Approach to Analysing the changing inequality in the Housing Market during 2001-2014

**Yingyu Feng[1] and Kelvyn Jones[2]**

1 **University of Sydney, Sydney, Australia**
yingyu.feng@sydney.edu.au

2 **School of Geographical Sciences, University of Bristol, Tyndall Avenue/Bristol, United Kingdom**
kelvyn.jones@bristol.ac.uk

## Abstract

The spatial inequality in house prices between the UK regions has been extensively researched in the existing literature, but the analysis at a sub-regional level is scarce. This working paper presents an integrated spatio-temporal approach that allows for the examination of the changing inequality in house prices at multiple spatial scales and over time simultaneously. By utilising a large-scale non-aggregated spatial housing dataset in England during 2001-2014, this study provides the most thorough and systematic spatiotemporal analysis of the changing patterns in house prices in England.

**1998 ACM Subject Classification** I.5.2 Design Methodology

**Keywords and phrases** Spatio-temporal Analysis, Inequality, Multilevel Models, House Prices, Spatial Big Data

## 1 Background

The spatial inequality in house prices between the UK regions is usually characterised as the North-South divide – lower in the former, higher in the latter [1-3]. This gap emerged in the 1970s, widened in the 1980s, and then somewhat reversed in the early 1990s. Until then, the regional house price movements were approximately aligned with national house prices [4-5], but since the mid-1990s some sub-housing markets in the country have become 'de-coupled' from the national market [6].

Apart from the wide regional price disparities, substantial price variations also started to emerge within regions[7]. The existing literature on the house price variation, however, commonly analysed the patterns cross-sectionally at one spatial scale at regional level, or temporal trends at national level. Spatio-temporal patterns were commonly analysed by applying the same cross-sectional methodology repeatedly at different time points.

The purpose of this study is to develop a new multi-scale spatio-temporal approach that allows the consideration of varying temporal trends at multiple geographic scales and thus to offer a thorough and systematic analysis of the changing patterns in house price differentials between and within regions at multiple spatial scales for entire England.

Spatial big data and machine learning in GIScience, Workshop at GIScience 2018, Melbourne, Australia).
Editors: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, Peter Kiefer;

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 2 Data and Methodology

A complete list of historical house transactions (13.6 million) in England during 2001-2014 was obtained from the Land Registry of England and Wales. Multiple spatial scales were investigated: region, Local Authority District (LAD), Middle- and lower-Super Output Areas (MSOA and LSOA), and Output Areas(OAs). MSOAs, LSOAs and OAs are the UK census geographies, which were designed to be socially homogeneous in terms of household numbers, housing type and tenure.

An integrated model-based spatio-temporal approach was developed to investigate the changing patterns at multiple scales over time. First, a fixed-effects approach[8] is employed to examine housing price inequality at the regional level. Regions were modelled as having fixed effects on house price by using dummy indicator representing each English region. A five-level random-effects model incorporating varying temporal effects across multiple-scale geographical areas were then developed to examine the changing geographical variations at sub-regional level over time.

The sub-regional units under investigation have a strictly hierarchical structure, where houses were nested within OAs, which, in turn were nested within LSOAs, MSOAs, and then LADs. These units were specified as having random effects through a five-level random-effects model with the level 1-5 units being houses, OAs, LSOAs, MSOAs and LADs, respectively. The stochastic variations between houses at level-1 as well as the spatial variations at sub-regional levels net of other levels and after removing regional differences can be examined simultaneously through this modelling technique.

The potentially different temporal trajectories of places are explored by allowing the effects of time to vary across space. This is achieved by allowing the coefficient of time variables to vary randomly across the units at all five scales. This approach not only allows the identification of the most marked spatial scale that accounts for price variation but also allows the changing patterns of spatial variations in house prices to be modelled as a function of time.

This spatio-temporal approach allows the investigation of temporal trends at each spatial level and spatial variations of house price at multiple scales simultaneously. It also produces more precise and robust estimations, which are shrunken towards the overall mean of the next higher-level units[9].

## 3 Results

### 3.1 Trends in Regional Housing Markets

Substantial price differentials were found between the nine English regions. The variations in the aggregated average house price between regions were three-fold in January 2001, dropped to 2.6-fold in January 2008 and rose to four-fold in Dec 2014. The modelled mean house prices at the regional level were presented in Figure 1. London was found to have seen major and extensive increases over the period and pulled away from the rest of English regions.

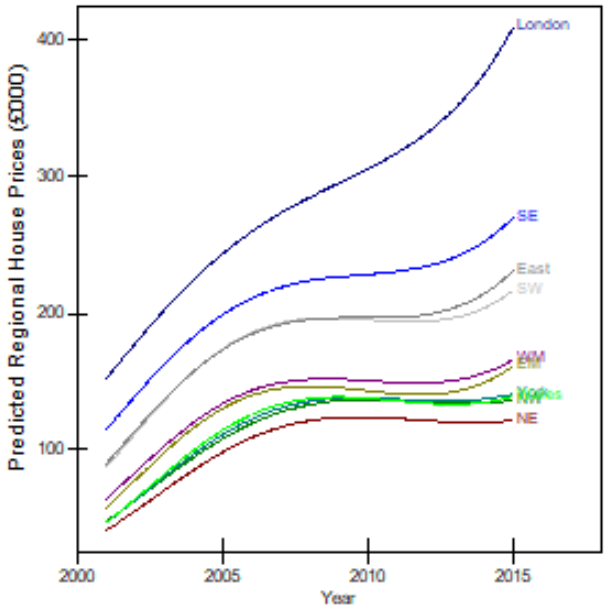



**Figure 1** Modelled mean house price for each English region.
Note: SE=South East, East=East of England, SW=South West, WM=West Midlands, EM=East Midlands, York=Yorkshire and Humber, NW=North West, NE=North East

## 3.2 Chaning Inequality in House Prices within Regions

The changing inequality in house prices at a sub-regional level is explored between the LADs within regions, between MSOAs within the LADs, between LSOAs within MSOAs and between OAs within LSOAs. The estimated variations in house prices at these four scales were presented in two steps: without and with adjusting for housing stock attributes, presented in Figure 2 and 3, respectively.

The overall changing patterns between the two graphs are broadly similar; controlling for the housing stock characteristics simply further brings out the importance of geography, particularly at the broad district or borough level in London and the East of England. The between-LAD variations in London the largest amongst in all English regions and they have risen sharply during the overall period, while the between-LAD variations in all the other regions are comparatively much smaller. Geographies in London, especially at the macro level, are much more important in house price determination than in any other regions and become more so over time.

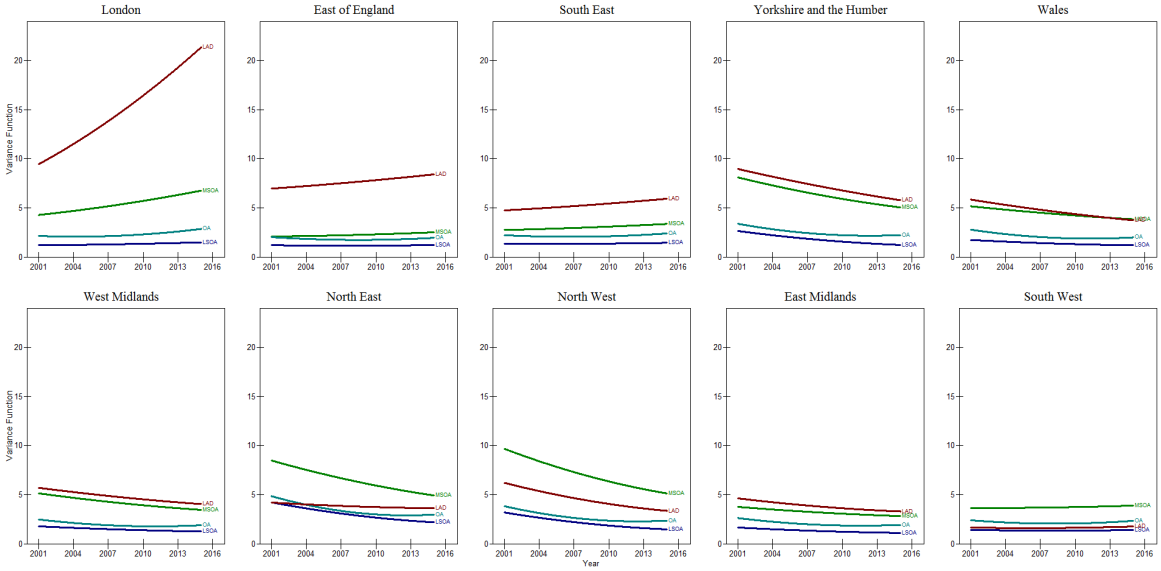



**Figure 2** Changing within-region price variations without adjusting for housing attributes.

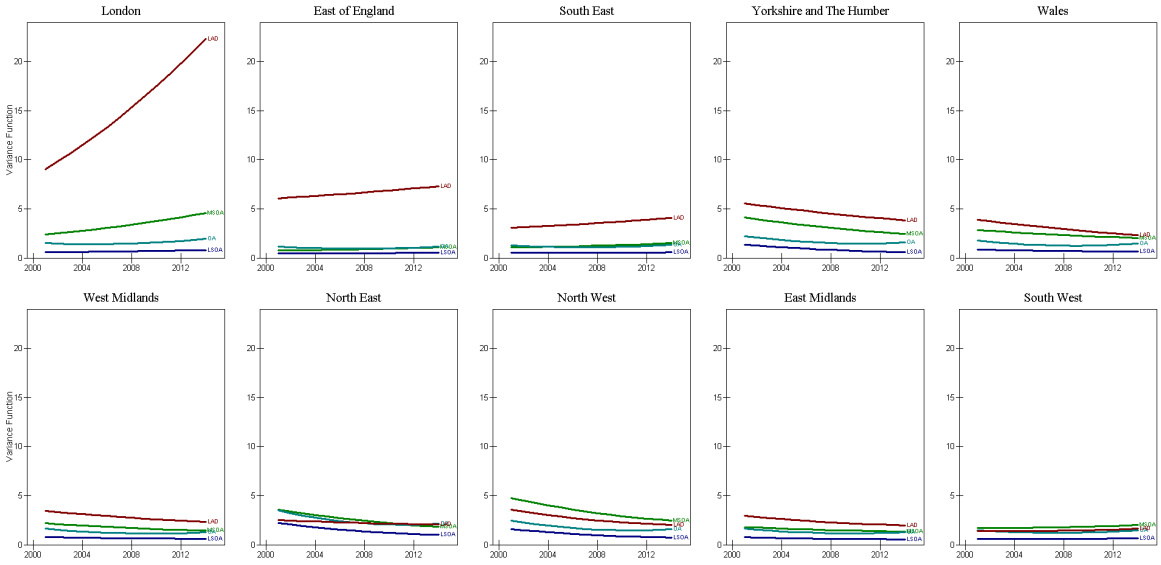



**Figure 3** Modelled mean house price for each English region.

## 4 Discussion and Conclusion

Many previous studies found that substantial house price disparities exist between regions, and there is a clear North-South divide. The analyses at a sub-regional level, including the changing patterns in house price disparities, are however under-researched. This study presents a new spatio-temporal approach and applied it to historical individual house-level transactional data in England over a 14-year period and the results suggest that the geography of house prices is much more complex than a simple North-South divide.

We have three major findings. First, house price disparities exist at multiple spatial levels: at both regional and various sub-regional levels. Second, The regional price differences narrowed during the first half of the study period but widened after the global financial crisis in 2007/08. Third, the changing patterns of price variations within regions depend on the region and spatial level concerned. At the LAD scale, spatial disparities have mostly decreased. The magnitude of the disparities is substantial for some regions.

It was found that London is very different from the rest of English regions in many aspects. Its spatial inequality is the largest compared to the rest. This inequality has also increased substantially during the overall study period, irrespective of the level of the deprivation of the place. At any given time in London, greater price variations exist between affluent areas than deprived areas, while it is completely the opposite for other English regions. In addition, urban houses do not appear to be more expensive than the urban ones, after adjusting for the housing characteristics. This study also demonstrates the advantages of a random-effects approach to analysing the changing effects of the spatial units at multiple scales on house prices over time.

**Acknowledgements.** I want to thank the support and funding provided by the Economic and Social Research Council, United Kingdom.

## References


1. Macdonald, R. & Taylor, M. P. 1993. Regional House Prices in Britain - Long-Run Relationships and Short-Run Dynamics. *Scottish Journal of Political Economy,* 40, 43-55.
2. Alexander, C. & Barrow, M. 1994. Seasonality and Cointegration of Regional House Prices in the UK. *Urban Studies,* 31, 1667-1689.
3. Meen, G. 1999. Regional House Prices and the Ripple Effect: A New Interpretation. *Housing Studies,* 14, 733-753.
4. Buckley, R. & Ermisch, J. 1982. Government Policy and House Prices in the United Kingdom: An Econometric Analysis. *Oxford Bulletin of Economics and Statistics,* 44, 273-304.
5. Meen, G. 1990. The Removal of Mortgage Market Constraints and the Implications for Econometric Modeling of UK House Prices. Oxford Bulletin of Economics and Statistics, 52, 1-23.
6. Ferrari, E. & Rae, A. 2011. *Local Housing Market Volatility*, York: Joseph Rowntree Foundation. Available from: https://www.jrf.org.uk/report/local-housing-market-volatility
7. Jones, C. & Leishman, C. 2006. Spatial Dynamics of the Housing Market: An Interurban Perspective. *Urban Studies,* 43, 1041-1059.
8. Bell, A. & Jones, K. 2015. Explaining Fixed Effects: Random Effects Modeling of Time-Series Cross-Sectional and Panel Data. *Political Science Research and Methods,* 3, 133-153.
9. Jones, K. & Bullen, N. 1994. Contextual Models of Urban House Prices : A Comparison of Fixed- and Random-Coefficient Models Developed by Expansion. *Economic Geography,* 70, 252-272.

# Automated social media content analysis from urban green areas – Case Helsinki

**Vuokko Heikinheimo[1], Henrikki Tenkanen[1], Tuomo Hiippala[1], Olle Järv[1], and Tuuli Toivonen[1]**

1 **Digital Geography Lab, University of Helsinki, Finland**
vuokko.heikinheimo@helsinki.fi

## 1 Introduction

Location-based social media data are increasingly used to understand spatial and temporal patterns of human activities in different environments. Mining of the continuous flow of social media posts has the potential to provide up to date information about where, when and how people use space [3]. Recent advances in the field of computer science such as machine learning enable us to analyze these spatial big data in unprecedented volumes. Here, we discuss the use of location-based social media data together with automated content analysis techniques to support smart spatial planning in cities. In the presentation we focus particularly on urban green areas, using parks of the Helsinki Metropolitan Region in Finland as the study area.

## 2 Social media data on urban greens

Social media data has been used to study extensively the use of urban areas [13, 1, 10, 5]. In national parks and other recreational sites, social media has been shown to work as a proxy for visitation [16, 15], and as an indicator of people's activities and preferences [7, 8] when compared with official visitor information. In urban green areas, social media has also been used as a proxy for park and trail use with mixed results [4, 6, 17].

## 3 Need for automated content analysis methods

Various different approached have been used to automatically analyze textual and image content in social media data (Figure 1 and Figure 2), but machine learning techniques are still rarely utilized in addressing environmental questions [3, 2]. Traditional data collection methods such as surveys, activity diaries, GPS tracking to study the use of green areas are costly and usually time consuming to implement, and thus social media analysis has potential to support these efforts. However, most social media content analysis from green areas have been based on manual work, which limits the extent and repeatability of the analysis. Promising examples have shown that automated text content analysis [11] and image content analysis [12] may help filtering content relevant to nature recreation. Also, the combination of different modalities (such as image, text and emojis) can be utilized to enhance the performance of deep-learning algorithms [14], as exemplified in our case study from Helsinki.

## 4 Platforms vary in their data

Social media platforms differ in popularity and the type of content, and thus the choice of platform(s) may affect greatly the analysis output. Different platforms provide varying

Spatial big data and machine learning in GIScience, Workshop at GIScience 2018, Melbourne, Australia.
Editors: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, Peter Kiefer;
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

spatial accuracies for geotagged data: Instagram posts are associated with the predefined Facebook places, Flickr provides GPS coordinates and Twitter a combination of these. Users also post different contents: Instagram is used for sharing momentary experiences, Flickr more professional content on e.g. species and Twitter discussions on topical issues. Urban social media studies are often based on Twitter or Foursquare, whereas environmental studies have used most widely Flickr and Panoramio. Users also share content across platforms. For example, a geotagged tweet might actually be an Instagram posts (in a sample dataset from Helsinki, over 50 % of the geotagged tweets originated from Instagram). The way in which users communicate in each platform (using text, emojis, image and video) should be taken into account when designing and training classifiers [9].

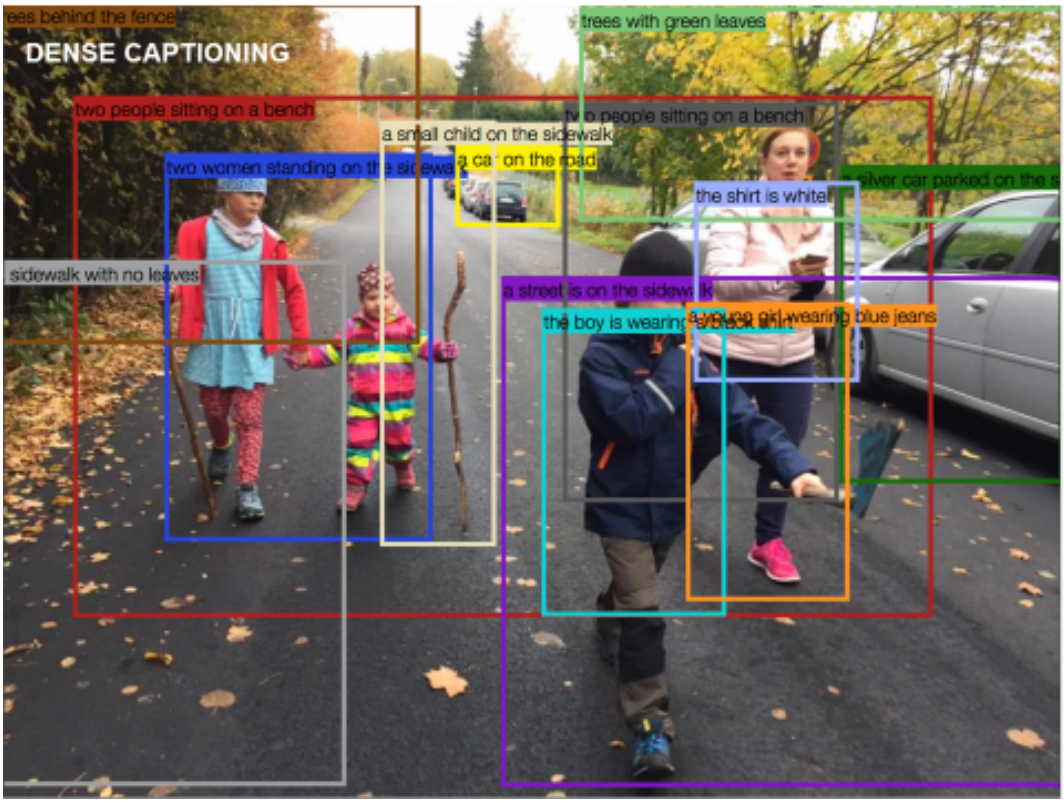



**Figure 1** Dense captioning allows verbalizing image content to natural language that can be analyzed more easily.

## 5 Multimodal analysis (text + image) brings added value when detecting activities

Our dataset from Helsinki contained 11 000 geotagged Instagram photos located within regionally important green areas (all available data from 2015). The dataset was manually annotated for the presence and absence of human activities (24 % of the data contained an activity), and split to training, validation and testing samples for the automatic activity detection. Automatic activity detection was done separately for images and captions (monomodal classification), and trough fusion of features extracted from images and captions (multimodal classification). Images were analyzed using NasNet-L pre-trained on ImageNet,

**Figure 2** Identifying objects and people allow further analyses of what people have been imaging (an example using MASK R-CNN)

and Captions were analyzed using fastText word embeddings trained on all available captions. Our preliminary analysis shows that multimodal methods are most effective in identifying activities automatically: automated detection of activities from the posts was most successful when combining both the textual and image content. With text only, we reached F1 score of 0.71, with image only 0.76 and with their combination 0.84 [9].

Future work will include the comparison of content from different platforms, and further analysis of the spatial activity patterns in urban green areas.

## References


1 Andrew Crooks, Dieter Pfoser, Andrew Jenkins, Arie Croitoru, Anthony Stefanidis, Duncan Smith, Sophia Karagiorgou, Alexandros Efentakis, and George Lamprianidis. Crowdsourcing urban form and function. *Int. J. Geogr. Inf. Sci.*, 29(5):720–741, 2015.

2 Enrico Di Minin, Christoph Fink, Henrikki Tenkanen, and Tuomo Hiippala. Machine learning for tracking illegal wildlife trade on social media. *Nat. Ecol. Evol.*, 2(3):406–407, 2018.

3 Enrico Di Minin, Henrikki Tenkanen, and Tuuli Toivonen. Prospects and challenges for social media data in conservation science. *Front. Environ. Sci.*, 3, 2015.

4 M. L. Donahue, B. L. Keeler, S. A. Wood, D. M. Fisher, Z. A. Hamstead, and T. McPhearson. Using social media to understand drivers of urban park visitation in the Twin Cities, MN. *Landsc. Urban Plan.*, 175:1–10, 2018.

**5** Alexander Dunkel. Visualizing the perceived environment using crowdsourced photo geodata. *Landsc. Urban Plan.*, 142:173–186, 2015.

**6** Zoe Hamstead, David Fisher, Rositsa T Ilieva, Spencer A Wood, Timon McPhearson, and Peleg Kremer. Geolocated social media as a rapid indicator of park visitation and equitable park access. *Comput. Environ. Urban Syst.*, 2018.

**7** Anna Hausmann, Tuuli Toivonen, Rob Slotow, Henrikki Tenkanen, Atte Moilanen, Vuokko Heikinheimo, and Enrico Di Minin. Social Media Data Can Be Used to Understand Tourists' Preferences for Nature-Based Experiences in Protected Areas. *Conserv. Lett.*, 11(1), 2017.

**8** Vuokko Heikinheimo, Enrico Di Minin, Henrikki Tenkanen, Anna Hausmann, Joel Erkkonen, and Tuuli Toivonen. User-Generated Geographic Information for Visitor Monitoring in a National Park: A Comparison of Social Media Data and Visitor Survey. *ISPRS Int. J. Geo-Information*, 6(3), 2017.

**9** Tuomo Hiippala, Christoph Fink, Vuokko Heikinheimo, Henrikki Tenkanen, and Tuuli Toivonen. Applying deep learning to multimodal data in social media, 2018. URL: `http://www.helsinki.fi/~thiippal/presentations/2018-04-aag.pdf`.

**10** Qunying Huang and David W. S. Wong. Activity patterns, socioeconomic status and urban spatial structure: what can social media data tell us? *Int. J. Geogr. Inf. Sci.*, 30(9):1873–1898, 2016.

**11** Andrew Jenkins, Arie Croitoru, Andrew T. Crooks, Anthony Stefanidis, CA Zucco, P Cascon, X Li, M Barthelemy, and G Lamprianidis. Crowdsourcing a Collective Sense of Place. *PLoS One*, 11(4):e0152932, 2016.

**12** Daniel R Richards and Bige Tunçer. Using image recognition to automate assessment of cultural ecosystem services from social media photographs. *Ecosyst. Serv.*, 2017.

**13** Taylor Shelton, Ate Poorthuis, and Matthew Zook. Social media and the city: Rethinking urban socio-spatial inequality using user-generated geographic information. *Landsc. Urban Plan.*, 142:198–211, 2015.

**14** Nitish Srivastava and Ruslan R. Salakhutdinov. Multimodal Learning with Deep Boltzmann Machines, 2012. URL: `http://papers.nips.cc/paper/4683-multimodal-learning-with-deep-boltzmann-machines`.

**15** Henrikki Tenkanen, Enrico Di Minin, Vuokko Heikinheimo, Anna Hausmann, Marna Herbst, Liisa Kajala, and Tuuli Toivonen. Instagram, Flickr, or Twitter: Assessing the usability of social media data for visitor monitoring in protected areas. *Sci. Rep.*, 7(1):17615, 2017.

**16** Spencer a Wood, Anne D Guerry, Jessica M Silver, and Martin Lacayo. Using social media to quantify nature-based tourism and recreation. *Sci. Rep.*, 3, 2013.

**17** Xinyi Wu, Greg Lindsey, David Fisher, and Spencer A Wood. Photos, tweets, and trails: Are social media proxies for urban trail use? *J. Transp. Land Use*, 10(1):789–804, 2017.

# Long short-term memory networks for county-level corn yield estimation*

**Haifeng Li**
School of Geosciences and Info Physics, Central South University, Changsha, China
lihaifeng@csu.edu.cn

**Yudi Wang**
School of Geosciences and Info Physics, Central South University, Changsha, China

**Renhai Zhong**
College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou, China

**Hao Jiang**
College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou, China

**Tao Lin**
College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou, China
lintao1@zju.edu.cn

## Abstract

Quantifying the response of crop yield to climate factors is important in agricultural systems. Many studies have worked on yield prediction through process-based simulation models and statistical models. Given the spatiotemporal explicit features of crop production, there exists a need to better understand the cumulative temporal effects of climate factors on crop production. To fill this gap, we build a Long Short-Term Memory (LSTM) model for weather-impacted corn yield prediction. The results show that LSTM model has a better performance (RMSE = 0.61 Mg ha$^{-1}$) in yield prediction than two other models: Lasso (RMSEP = 1.07 Mg ha$^{-1}$) and RF (RMSE = 0.64 Mg ha$^{-1}$) on the same test set. The results illustrate the potential of LSTM in crop yield prediction by considering the cumulative temporal impact of weather factors on crop yield.

**2012 ACM Subject Classification** Spatio-temporal modeling in machine learning applications

**Keywords and phrases** LSTM, Corn yield prediction, climate factors, cumulative temporal effect

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2018.xx

## 1 Introduction

In 2015, FAO estimated that 795 million people live without an adequate food supply [3]. With the increased global population, maintaining sustainable food supply becomes an international issue. How to improve crop productivity is critical to address food security. Accurate in-season yield prediction can support farmers to improve management and reduce yield loss caused by unfavorable weather conditions.

Process-based biophysical modeling and statistical modeling [4] are two popular approaches to quantify corn yield based on climate factors. Biophysical modeling is more suitable for site-specific yield analysis, whereas statistical modeling is often adopted in large-scale spatial analysis. Some researches further look into the integration of process-based and statistical

* corresponding author: Tao Lin, lintao1@zju.edu.cn.

Spatial big data and machine learning in GIScience, Workshop at GIScience 2018.
Editors: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz and Peter Kiefer;


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

modeling [6]. Under the rapid development of computing capabilities in recent years, artificial intelligence methods, such as Artificial Neural Network (ANN) [2] and Bayesian Network (BN) [5], have gradually been applied for agricultural yield prediction. These studies, however, often simply the temporal variations of yield-weather relationship and the cumulative effects of weather factors. In practice, not only the crop itself is growing over time, the impact of weather on the crop might also vary and accumulate throughout the growing season. Especially, the damage of extreme weather would possibly impose a long sequential impact on crop growth throughout the season. No remediations afterward can be applied to resolve the damage. There is a critical need, therefore, to integrate the cumulative effects of climate factors on crop production to better understand the interactions between crop and environmental factors.

Long Short-Term Memory (LSTM) model is a deep neural network that has been successful in learning sequence and tree structures [7]. It facilitates time-series analysis and handles complexity and nonlinearity functions by its unique structure. LSTM was developed to deal with the gradient vanishing and exploding problems. Previous studies have demonstrated that LSTM has a good performance in dealing with long sequential data in natural language modelling [7] and human trajectory prediction [1]. We would like to evaluate the performance of LSTM modeling in capturing dynamic temporal yield-weather relationships and yield prediction.

The objectives of this study are to: (i) develop a LSTM model to predict corn yield by cumulative climate factors; (ii) compare the prediction accuracy among LSTM, Random Forests (RF), and Lasso regression methods.

# 2 Methodology

## 2.1 Study Area and datasets

This study focuses on rainfed corn yield in the central and northern 11 states in the U.S. from 1970 to 2016. These 11 states are: Illinois, Indiana, Iowa, Kansas, Kentucky, Michigan, Minnesota, Missouri, Nebraska, Ohio and Wisconsin. The county-level non-irrigated corn yield data is from USDA's National Agriculture Statistics Service [8]. To capture the impact of climate factors on corn yield, we calculate the detrend yield (yield influenced by climate factors) by linear regression of corn yield versus year as the predictable variable in the models. The daily county-level climate data is obtained from Applied Climate Information System (ACIS) Web Service [9]. The climate data used in this study include: maximum daily temperature (Tmax), minimum daily temperature (Tmin), and daily precipitation (PRCP). During the corn growth period (from week 20 to 39), weekly Tmax, Tmin and PRCP is calculated and transferred into sequential vectors as the input of LSTM model. In addition, Min-Max scaling is used to scale the input into a range from zero to one.

## 2.2 Models

The structure of LSTM model in this study includes four layers: input layer, hidden layers and output layer (Figure 1). The input is a time sequence $X = \{x_1, x_2, \cdots, x_T\}$, $x_t$ is a vector which includes climate factors.$x_t = [\mathrm{Tmax}_t, \mathrm{Tmin}_t, \mathrm{PRCP}_t]$, T is 20, the length of time sequence and t is the time, represent the week in corn growth period, from week 20 to 39. The hidden layers are two LSTM layers composed of LSTM cells, in which information is selectively transported and stored. The output is detrend yield $y$ calculated out by all input vectors in one time sequence of corn growth period, from week 20 to 39.

To make a comparison between LSTM and other models, we build a Lasso regression model ($\lambda = 0.003$) and a RF model as baselines. All three models are trained based on the same training set, which is randomly selected 80% of the total sample set. The remaining 20% of the total sample set is used as the test set, where RMSE is used as the performance indicator of yield prediction accuracy.

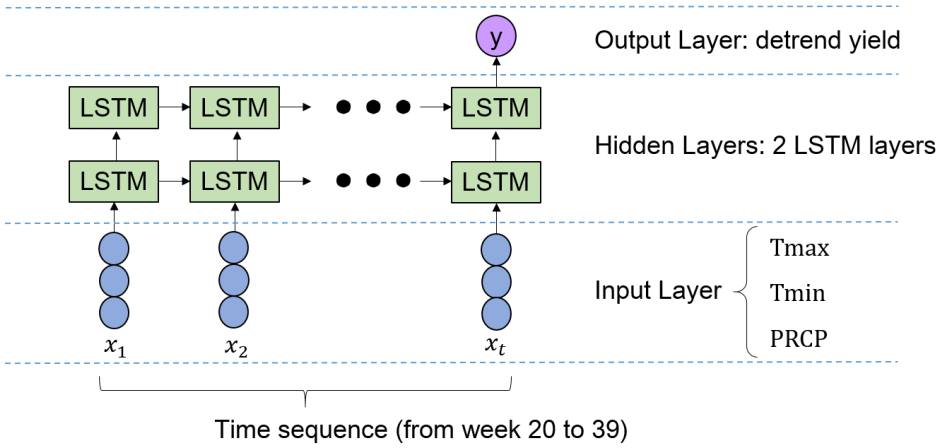



**Figure 1** The structure of LSTM.

## 2.3 Results and Discussion

We compare the accuracy of LSTM model with two typical models: RF and Lasso regression (Table 1). The results show that machine learning models, LSTM (RMSE = 0.61 Mg ha$^{-1}$) and RF (RMSE = 0.64 Mg ha$^{-1}$), outperform traditional linear regression model, Lasso (RMSE = 1.04 Mg ha$^{-1}$) in yield prediction on the same test set. Compared to RF, LSTM has a less degree of overfitting and a slight improvement on prediction accuracy. The improved accuracy by the LSTM model is possibly due to its structure designed for capturing not only the direct impact at each time period but accumulated effect of weather on crop yield throughout the entire growing season. The degree of impact by weather may vary temporally as the requirement of water and nutrients by crop varies at different growing stages. In addition to the accumulated temporal impact, LSTM is more suitable to capture nonlinear impact of weather on crop, especially considering the extreme events such as heating and drought during the growth period.

| | RMSE | |
|---|---|---|
| Model | Training Set (34868 obs) | Test Set (8717 obs) |
| LSTM | 0.53 | 0.61 |
| RF | 0.31 | 0.64 |
| Lasso | 1.07 | 1.04 |

**Table 1** The RMSE of LSTM, RF and Lasso models.

To better understand the accumulated and nonlinear impact of climate factors on corn

yield, some future work are proposed: (i) evaluate the performance of in-season prediction; (ii) quantify the sensitivity of yield to climate factors in different corn growing stages by adding an Attention mechanism into the LSTM model; (iii) incorporate the spatial correlation features of the crop yield-weather relationship in the LSTM model.

## 3 Conclusions

We build a Long Short-Term Memory model to predict the county-level rainfed corn yield in response to climate factors for 11 states in the U.S. from 1970 to 2016. The results show that the LSTM model has a better prediction accuracy than traditional Lasso regression model. Compared to the RF, LSTM has a slightly better prediction accuracy and less degree of overfitting. The results demonstrate the potential of LSTM in crop yield prediction and temporal analysis of yield-weather interactions. The mechanism of how the LSTM model captures the cumulative temporal effect of climate factors on corn yield needs to be further studied.

## References


1 Al. Alahi, K. Goel, V. Ramanathan, et al. Social lstm: Human trajectory prediction in crowded spaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016.

2 R. Alvarez. Predicting average regional yield and production of wheat in the argentine pampas by an artificial neural network approach. *European Journal of Agronomy*, 30(2):70–77, 2009.

3 FAO. The state of food insecurity in the world. Meeting the 2015 international hunger targets: Taking stock of uneven progress, 2015.

4 D.B. Lobell and M.B. Burke. On the use of statistical models to predict crop yield responses to climate change. *Agricultural and Forest Meteorology*, 150(11):1443–1452, 2010.

5 N.K. Newlands and L. Townley-Smith. Predicting energy crop yield using bayesian networks. In *The Fifth IASTED International Conference*, volume 711, pages 014–106, 2010.

6 M.J. Roberts, N.O. Braun, T.R. Sinclair, et al. Comparing and combining process-based crop models and statistical models with some implications for climate change. *Environmental Research Letters*, 12(9):095010, 2017.

7 M. Sundermeyer, R. Schlüter, and H. Ney. LSTM neural networks for language modeling. In *13th Annual Conference of the International Speech Communication Association*, 2012.

8 USDA-NASS. Quick Stats 2.0. SDA-NASS, Washington, DC. `http://www.nass.usda.gov/quickstats/`, 2017. [Online; accessed 20 August].

9 RCC-ACIS web service. `http://www.rcc-acis.org/`, 2017. [Online; accessed 21 August].

# Conditional Adversarial Networks for Multimodal Photo-Realistic Point Cloud Rendering

**Torben Peters**[1] **and Claus Brenner**[1]

1 **Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Germany**
{peters,brenner}@ikg.uni-hannover.de

## Abstract

We investigate whether conditional generative adversarial networks (C-GANs) are suitable for point cloud rendering. For this purpose, we created a dataset containing approximately 150,000 renderings of point cloud image pairs. The dataset was recorded using our mobile mapping system, with capture dates that spread across one year. By parameterizing the recording date, we are showing that it is possible to predict realistically looking views for different seasons, from the same input point cloud.

**1998 ACM Subject Classification** D.1.3 Concurrent Programming, I.2.6 Learning, I.2.10 Vision and Scene Understanding, I.3.3 Picture/Image Generation, I.4.8 Scene Analysis.

**Keywords and phrases** Deep learning, MapReduce, GAN, image-to-image translation.

## 1 Introduction

Laser scanned point clouds are difficult to handle when it comes to photo-realistic rendering. First, a camera calibration is needed in order to colorize each scanned point. However, this does not guarantee that each 3d point is captured by a camera viewpoint. Secondly, since point clouds are sparse, it is difficult to exclude occluded points, e.g. behind walls and buildings. In order to create a continuous surface, splats can be drawn instead of points. A splat is defined as elliptical surface with a size according to the local point density. Lastly, the colorized point cloud does not contain any information about the sky and lighting.
Our approach tries to circumvent the whole process of model-based point cloud rendering, by learning how a possible representation of the point cloud could look like in reality. Our key contributions in this work are:

- Predicting photo-realistic views from point clouds which are containing only (laser) reflectance information.
- Extending a C-GAN to parameterize different seasons and months in order to predict multimodal images.

## 2 Related work

**Conditional GANs** attracted a lot of attention in recent years. Most notable are the *pix2pix* network by Isola et al. [3] and the improved version *pix2pixhd* [10], which is able to predict high resolution image-to-image mappings. Like traditional GANs, these networks are using an adversarial loss which is learned by a discriminator network. In contrast to $L_1$-loss, the adversarial loss leads to less blurry images [3, 10] by learning to distinguish between real and generated images. In addition to image-to-image translation, there exist a wide range

Spatial big data and machine learning in GIScience, Workshop at GIScience 2018, Melbourne.
Editors: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, Peter Kiefer;

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of conditional GANs, such as unpaired image-to-image translation [14] and text-to-image [2, 12, 7, 8].

**Multimodal image-to-image translation** defines the process of mapping one-to-many images, by modeling distributions of possible outcomes with an additional latent space vector or matrix, e.g. as used by *BicycleGAN* [15]. Other notable contributions are *iGAN* [13] and *Scribbler* [9] which directly encode the information.

Depending on the application, **point clouds** are handled in different ways in order to incorporate them into deep neural networks. They can be inferred as unordered 3d point sets [5, 6], using voxels [11, 4], or projected into images in order to use traditional 2d-convolutional networks [1].

# 3 Method

## 3.1 Preparing the dataset using MapReduce

To prepare a training dataset, we used mobile mapping data which was captured during 21 measurement campaigns, over the duration of one year. To illustrate the size of the problem, the subset of the data we are using contains 15 billion $(15,017,586,980)$ 3d points and $123,047$ images. Each image capture is given in terms of position (in UTM coordinates) and orientation (roll, pitch, yaw angles). Additionally, the intrinsic parameters of each camera are known. The task is then to project each of the 15 billion 3d points to each of the $123,047$ images. To solve this task, we created a massively parallel point cloud renderer, using the MapReduce framework on an Apache Hadoop cluster.

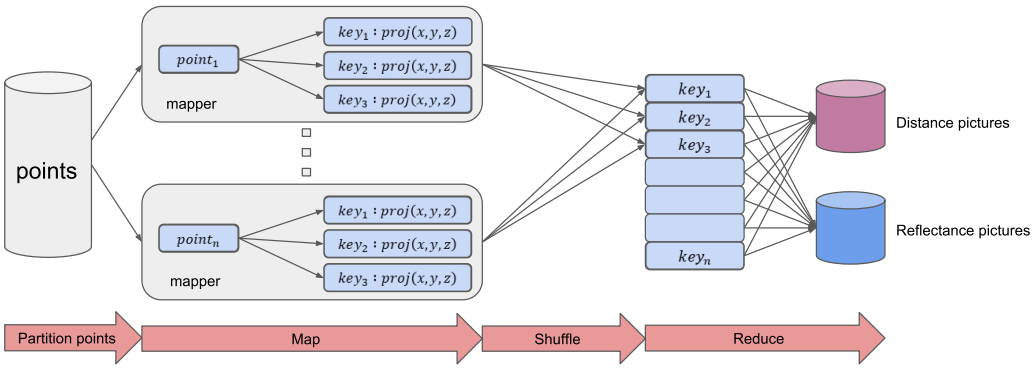



**Figure 1** Our MapReduce approach for rendering large point clouds.

In order to apply MapReduce, each mapper has a list of all image orientations. According to the MapReduce principle, it receives a subset (split) of 3d point coordinates and their reflectance values (reflectance is an entity measured by the laser scanner). In order to reduce the amount of points emitted by the mapper, we exclude points that are behind the camera or are further away than 500 m. The mapper possibly emits multiple key-value pairs per incoming 3d point, depending on the number of images the point appears in. The key is defined by the image name, identifying a single image take, whereas the value contains the distance, reflectance and the image coordinates of the point. Each reducer receives all necessary information, grouped by image (key), and computes two 16-bit gray-value images per key, one containing the distance and the other one containing the reflectance values per

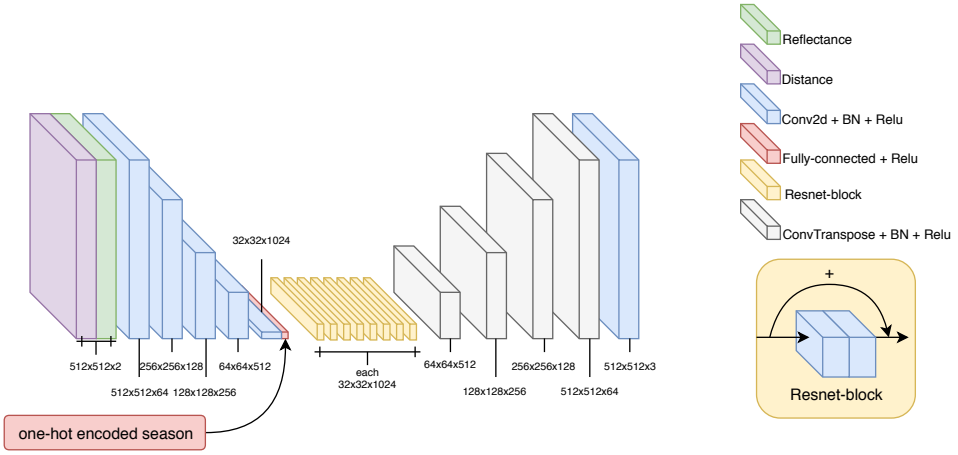



**Figure 2** Our adapted generator network. Note how the capture date is injected by concatenating a fully connected layer.

point. Depending on the scanning situation, the points appear more or less sparse on the image plane.

## 3.2 C-GAN

Our approach is heavily inspired by the *pix2pixhd* network by Wang et al. [10]. The *pix2pixhd* network incorporates instance segmentation information and label-maps in order to enable object manipulation. By encoding the features of one instance, it is able to generate diverse images from the same input.

We modified this generator network as follows. We removed the instance- and labels-maps from the network architecture because we don't have any information about the class or instance of each point. We also reduced the number of generator networks to one $\{G\}$ and the number of multiscale discriminators to two $\{D_1, D_2\}$. However, the discriminator networks have exactly the same architecture as defined by Wang et al. [10]. Each discriminator operates on a different image scale, $D_1$ at the original scale $512 \times 512$, and $D_2$ at $256 \times 256$. We adopted the $\mathcal{L}_{GAN}$ part of the loss function as follows:

$$\min_{G} \max_{D_1,D_2} \sum_{k=1,2} \mathcal{L}_{GAN}(G, D_k) = \sum_{k=1,2} E_{(x,y)}[logD_k(x,y)] + E[log(1 - D_k(x, G(x,s)))]. \quad (1)$$

The training dataset is given as a set of tuples of corresponding images and dates $\{(x_i, s_i, y_i)\}$, where $x_i$ is the input-/reflectance-image, $y_i$ is the real image, taken by a camera of our mapping van, and $s_i$ is the date the image was taken. In order to encode the capture date $s_i$, we added a fully connected layer as latent vector of the generator network. Similar to the findings of Zhu et al. [15], we also observed that noise fed additionally and directly into the generator was completely ignored by the network. As shown in figure 2, the fully-connected layer was instead concatenated to the bottleneck after convolving the input. The following layers of the generator network are identical to the *pix2pixhd* network. We used a one-hot encoding for each capture date $s_i$, as follows:

$$f(s_i) = \begin{cases} 1 + \mathcal{N}(\mu, \sigma^2), & \text{if } s_i = date \\ 0 + \mathcal{N}(\mu, \sigma^2), & \text{otherwise.} \end{cases} \quad (2)$$

In order to induce stochasticity, we added noise to each input, using $\mu = 0$ and $\sigma^2 = 0.1$.

## 4 Results

We trained the Networks for 20 epochs with a batch size of 1. Figure 3 shows an example of a predicted image. Remember that this is computed using only the reflectance and distance information from the point cloud. Note that the predicted building is colored in a typical color (white walls and red roof), while in reality, the building has quite different colors (red walls and dark roof). We believe that the color information is thus mostly derived by the spatial information and not by the (laser scanner) reflectance of the points themselves.

**Figure 3** Input image (reflectance, left), synthesized image (middle) and real image (right).

Figure 4 shows that by shifting the value in the one-hot encoded season vector, we are able to predict different seasons for the same laser scanner input. In this case, we used a point cloud that was recorded in Germany in March and predicted an image for June and December. However, we think that some features will stay encoded in the picture itself. For example, the amount of leaves which are captured by the laser scanner directly. It is also worth to mention that there are a large number of occluded points in the left pane of figure 4. From the middle and right pane of figure 4, it can be seen that the generator has learned to hide occluded points.

**Figure 4** Summer (middle) and winter (right) representation of the same input point cloud (left).

In Figure 5 we are showing additional examples of synthesized images from the same input but different season. We created also a video `https://youtu.be/mQINboXOvRM` which shows the difference between summer and winter.

**Figure 5** Examples for different seasons with the same input per row.

## 5 Conclusion and outlook

In this work we have shown that it is possible to predict realistically looking images, using only point cloud data. By parameterizing the different capture dates of the images and point measurements, we were able to map the same point cloud to different seasons. We have shown that the GAN was able to encode seasonal information like snow in winter or green trees in summer. Furthermore, the generator was able to hide occluded points.

For future work we would like to test if this procedure is able to colorize point clouds without using cameras. Provided that our point cloud is labelled, it is imaginable that this framework allows us to project high precision labels to the generated images in order to create or enrich data sets for semantic segmentation. Since our approach allows to define arbitrary view positions and angles, this would enable us to generate an infinite amount of training examples.

## Acknowledgement

This work was funded by the German Research Foundation (DFG) as a part of the Research Training Group GRK2159, ‘Integrity and collaboration in dynamic sensor networks’ (i.c.sens).

## References


1 Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert. Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. *Computers & Graphics*, 2017.

2 Ayushman Dash, John Cristian Borges Gamboa, Sheraz Ahmed, Marcus Liwicki, and Muhammad Zeshan Afzal. Tac-gan-text conditioned auxiliary classifier generative adversarial network. *arXiv preprint arXiv:1703.06412*, 2017.

3 Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

4 Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015.

5 Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

6 Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5105–5114, 2017.

7 Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

8 Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. In *Advances in Neural Information Processing Systems*, pages 217–225, 2016.

9 Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

10 Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *arXiv preprint arXiv:1711.11585*, 2017.

11 Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

12 Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE Int. Conf. Comput. Vision (ICCV)*, pages 5907–5915, 2017.

13 Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

14 Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.

15 Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *arXiv preprint arXiv:1711.11586*, 2017.