MOBILITY 2023

The Thirteenth International Conference on Mobile Services, Resources, and Users

ISBN: 978-1-68558-074-2

June 26 - 30, 2023

Nice, France

MOBILITY 2023 Editors

Jaime Lloret Mauri, Universitat Politecnica de Valencia, Spain

MOBILITY 2023

Forward

The Thirteenth International Conference on Mobile Services, Resources, and Users (MOBILITY 2023), held between June 26th and June 30th, 2023, continued a series of events dedicated to mobility-at-large, dealing with challenges raised by mobile services and applications considering user, device, and service mobility.

Users increasingly rely on devices in different mobile scenarios and situations. "Everything is mobile", and mobility is now ubiquitous. Services are supported in mobile environments, through smart devices and enabling software. While there are well known mobile services, the extension to mobile communities and on-demand mobility requires appropriate mobile radios, middleware, and interfacing. Mobility management is becoming more complex but is essential for every business. Mobile wireless communications, including vehicular technologies, bring new requirements for ad hoc networking, topology control, and interface standardization.

We take here the opportunity to warmly thank all the members of the MOBILITY 2023 technical program committee, as well as all the reviewers. The creation of such a high-quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to MOBILITY 2023. We truly believe that, thanks to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the MOBILITY 2023 organizing committee for their help in handling the logistics of this event.

We hope that MOBILITY 2023 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of mobile services, resources, and users.

MOBILITY 2023 Chairs

MOBILITY 2023 Steering Committee

Lounis Adouane, Université de Technologie de Compiègne / Heudisayc, France
Omar Sami Oubbati, University of Laghouat, Algeria

MOBILITY 2023 Publicity Chairs

Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

MOBILITY 2023
Committee

MOBILITY 2023 Steering Committee

Lounis Adouane, Université de Technologie de Compiègne / Heudisayc, France
Omar Sami Oubbati, University of Laghouat, Algeria

MOBILITY 2023 Publicity Chairs

Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

MOBILITY 2023 Technical Program Committee

Osama M.F. Abu-Sharkh, Princess Sumaya University for Technology, Amman, Jordan
Lounis Adouane, Université de Technologie de Compiègne / Heudisayc, France
Vijayan K. Asari, University of Dayton, USA
Mohamad Badra, Zayed University, Dubai, UAE
Matthias Baldauf, OST - Eastern Switzerland University of Applied Sciences, Germany
Chaity Banerjee, University of Central Florida, USA
Leandro Becker, Federal University of Santa Catarina (UFSC), Brazil
Olfa Ben Rhaiem, University of Sfax, Tunisia
Mohd Rashidi Che Beson, Universiti Malaysia Perlis, Malaysia
Khadija Bouraqia, ENSEM, Morocco
Peter Brída, University of Žilina, Slovakia
Ivana Bridova, University of Zilina, Slovakia
Simeon Calvert, Delft University of Technology, Netherlands
Carlos Carrascosa, Universitat Politècnica de València, Spain
Chao Chen, Purdue University Fort Wayne, USA
Daniel Delahaye, Ecole Nationale de l'Aviation Civile (ENAC), Toulouse France
Anatoli Djanatliev, University of Erlangen-Nuremberg, Germany
Mohand Djeziri, Aix Marseille University, France
Mohamed El Kamili, Higher School of Technology - Hassan II University of Casablanca, Morocco
Ayman El-Saleh, A'Sharqiyah University, Oman
Brahim Elmaroud, Higher Institute of Maritime Studies - Casablanca, Morocco
Javier Fabra, Universidad de Zaragoza, Spain
Anders Fongen, Norwegian Defence University College, Norway
Hassen Fourati, University Grenoble Alpes, France
Lokesh P. Gagnani, KSV University, India
Jordi Garcia, CRAAX Lab - UPC BarcelonaTECH, Spain
Abhinav Goel, NVIDIA, USA
Gelayol Golcarenarenji, University of the West of Scotland, UK
Shi-Jinn Horng, National Taiwan University of Science and Technology, Taiwan
Nornazlita Hussin, Universiti Malaya, Malaysia
Javier Ibanez-Guzman, Renault, France
Sergio Ilarri, University of Zaragoza, Spain

Jin-Hwan Jeong, SK telecom, South Korea
Christian Jung, Fraunhofer IESE, Germany
Ishmeet Kaur, Apple, USAs
Georgios Kambourakis, University of the Aegean, Greece
Hassan A. Karimi, University of Pittsburgh, USA
Hamzeh Khalili, Fundació i2CAT, Barcelona, Spain
Azad Khandoker, Johannes Kepler University Linz, Austria
Imran Khan, Insight Center for Data Analytics | University College Cork, Ireland
Hanane Lamaazi, C2PS Center - Khalifa University of Science and Technology, Abu Dhabi, UAE
Isa Maleki, Islamic Azad University, Iran
Francesca Martelli, IIT-CNR, Pisa, Italy
Ignacio Martinez-Alpiste, University of the West of Scotland, UK
Antonio Matencio-Escolar, University of the West of Scotland, UK
Subhasish Mazumdar, New Mexico Tech, USA
Weizhi Meng, Technical University of Denmark, Denmark
Farshad Miramirkhani, Isik University, Turkey
Javad Mohammadi, Carnegie Mellon University, USA
Alireza Morsali, McGill University, Canada
Tathagata Mukherjee, The University of Alabama in Huntsville, USA
Tatsuo Nakajima, Waseda University, Japan
Jose E. Naranjo, University Institute for Automobile Research (INSIA) | Universidad Politecnica de Madrid, Spain
Najett Neji, Université Paris Saclay - Univ. Evry - UFR Sciences et Technologies, France
Omar Sami Oubbati, University of Laghouat, Algeria
Sugandh Pargal, IIT Kharagpur, India
Hyoshin (John) Park,North Carolina A&T State University, USA
Wuxu Peng, Texas State University, USA
Alejandro Ramirez, Siemens AG, Germany
Victor Ramos, Universidad Autonoma Metropolitana, Mexico
Anna Lina Ruscelli, TeCIP Institute - Scuola Superiore Sant'Anna, Pisa, Italy
Mahsa Sadeghi Ghahroudi, Glasgow Caledonian University, UK
Antonio Arlis Santos da Silva, Hamm-Lippstadt University of Applied Sciences / Promotionskolleg NRW - PK NRW, Germany | Federal University of Rio Grande do Sul, Brazil
Viliam Sarian, Scientific Research Institute for Radio, Russia
Régine Seidowsky, COSYS-GRETTIA | Univ. Gustave Eiffel | IFSTTAR, France
Alireza Shahrabi, Glasgow Caledonian University, Scotland, UK
Haichen Shen, Amazon Web Services, USA
Danny Soroker, IBM T.J. Watson Research Center, USA
Harald Sternberg, HafenCity Universität Hamburg, Germany
Daxin Tian, Beihang University, Beijing, China
Markku Turunen, Tampere University, Finland
Dario Vieira, Efrei Paris, France
Ulrich Walder, Graz University of Technology, Austria
Shuliang Wang, Beijing Institute of Technology, China
Rainer Wasinger, University of Applied Sciences Zwickau, Germany
Mudasser F. Wyne, National University, USA
Cong-Cong Xing, Nicholls State University, USA
Daitao Xing, NYU Abu Dhabi, UAE

Renjun Xu, Zhejiang University, China
Hong Yang, Nokia Bell Labs, Murray Hill, USA
Paul Yoo, Birkbeck, University of London, UK
Lisu Yu, Nanchang University, China
Mariusz Zal, Poznan University of Technology, Poland
Jianhua Zhang, Clarkson University, USA
Xiao Zhu, University of Michigan, USA
Wolfgang Zirwas, Nokia Bell Labs, Munich, Germany
Kamil Zyla, Lublin University of Technology, Poland

Copyright Information

Table of Contents

A Real-time Multiple People Tracking System in a Complex Environment 1
Shi-Jinn Horng and Hu-Ke Li

Exploring Spatial Transformation-Based Privacy in a Small Town 8
Aidan Jacobs and Subhasish Mazumdar

A Real-time Multiple People Tracking System in a Complex Environment

Shi-Jinn Horng
Department of Computer Science and Information
Engineering, Asia University, Taichung Taiwan
Department of Medical Research, China Medical University
Hospital, China Medical University, Taichung Taiwan
e-mail: horngsj@yahoo.com.tw

Hu-Ke Li
Department of Computer Science and Information
Engineering,
National Taiwan University of Science and Technology,
Taipei, 10607, Taiwan, R.O.C
e-mail: 1742640235@qq.com

***Abstract*—Multiple Object Tracking is a major research field of computer vision due to increasing demand. Its application has become more and more extensive. The model proposed in this paper is an improved version of the traditional Deep Sort, which is mainly divided into two parts, the object detection part and the target tracking part. YOLOv5 (PA), the improved version of YOLOv5, is used as the front object detection model and it was trained specifically for the category of "person" in the CrowdHuman dataset, which greatly improved the detection accuracy of the model in a complex environment. Based on the Deep Sort tracking architecture, the Re-ID accuracy of the model was improved by using Mahalanobis distance, Hungarian algorithm, Aligned ReID, etc., and the tracking was predicted by Kalman filtering. In this paper, we use videos from the MOT20 dataset as the main test scenario. While achieving good MOTA and MOTP, the running speed of this model is guaranteed to achieve the effect of real-time.**

Keywords-Deep learning; target tracking; MOTA; MOTP.

I. INTRODUCTION

In recent years, with the rapid development of neural networks, deep learning has gradually received attention from all walks of life. In particular, with the rise of AlphaGo, developed by Google's DeepMind in 2017, deep learning has become a fierce competitor to other traditional algorithms.

Due to its rich and diverse datasets, computer vision has become one of the most important and rapidly developing application fields of deep learning, with a large number of applications with strong practicability, wide coverage and rapid development, for example, image classification, object detection, object tracking and semantic segmentation. These technologies have been fully integrated into every corner of our lives and continue to develop and gradually change our lives.

When taking monitoring as an example, whether it is outdoor traffic or indoor dense crowd, face recognition, ultra-distant object identification and target tracking technologies have been integrated into the monitoring system. They provide more accurate and diverse data, making surveillance more than just a picture. In particular, Multiple Object Tracking (MOT) has gradually developed and been more and more integrated into autonomous driving, pedestrian detection and other fields.

Based on the MOT20 dataset provided by MOT Challenge [1], this paper uses Deep Sort architecture with our YOLOv5 (PA) algorithm to carry out multi-target tracking. Through continuous trial and improvement, it can still complete high accuracy target tracking in the MOT20 dataset.

II. RELATED WORK

MOT has many related research directions, such as object detection, Single Object Tracking (VOT/SOT), Multi-Object Multi-Camera Tracking (MTMCT), Person Re-ID and multi-object single-camera tracking. The most commonly used datasets are MOT Challenge and Duke MTMC.

MOT Challenge, an advanced MOT competition with people as the main detection target, has been continuously developed since MOT15. In recent years, algorithms appearing on MOT Challenge are more and more accurate, among which the classic ones are Sort [2] and Deep Sort [3], which are dedicated to multi-target tracking. In addition, object detection algorithms such as Mask RCNN [4] also start to develop in this direction. All of these algorithms performed well in various competitions that year, until MOT20, the new dataset of MOT Challenge, appeared. Compared with previous MOT16 or MOT17, MOT20 has a huge difference. In MOT17, the number of persons in the same frame is roughly distributed between 10 and 30, but in MOT20, the number of persons in almost every frame is far more than 50, or even more than 100, and mutual occlusion between targets occurs more frequently. The proportion of pedestrians to the whole picture shrinks dramatically. All these present a more severe challenge to the original multi-target tracking algorithm.

In addition to Sort and Deep Sort, which are currently favored by the industry, many new algorithms have emerged in previous MOT Challenge competitions, attracting much attention. For example, MOTDT [5] made some modifications to the traditional Deep Sort. By learning Deep Sort, JDE [6] embedded the detection network into the model to improve the speed. However, these models are mainly developed in the direction of detection accuracy, using very complex algorithms, but often ignore the speed. Therefore, this paper not only focuses on improving the accuracy, but also controlling the running speed, so that the model can achieve real-time.

III. System Model

This model is divided into two parts when making MOT. The first part is called object detection block, which identifies persons in the picture with YOLOv5 (PA) and acquires Bounding boxes. The second part, called the object tracking block, does continuous tracking of objects through Deep Sort.

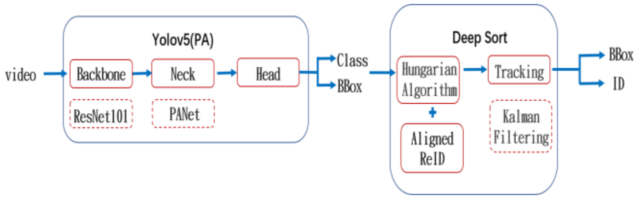

Figure 1. Model structure.

A. *Detection*

As shown in Figure 1, the object detection model we used is YOLOv5 (PA), which is a new model we improved based on the traditional YOLOv5 model. YOLO series is a very famous one-stage object detection model. Compared with the two-stage model, like R-CNN series, the detection accuracy is similar, but the detection speed is faster. Two-stage models divide the orientation of Bounding Box and object classification into two stages in the process of object detection, while YOLO combines them into one, which greatly improves the detection speed and meets the demand of real-time with high frame rate.

The YOLO series was continuously updated and improved from YOLOv1 [7], followed by YOLOv5, then Ultralytics [8] team, which is based on the YOLOv4 model for adjustment and improvement.

In this paper, a new YOLOv5(PA) model is obtained.

Backbone

This part is used to preliminarily extract the features of the target object. ResNet101 is used in this paper [9]. Because in this step, the network only learns some low-level features, and these features are often very similar. Therefore, it can save a lot of time in the initial training of the model by borrowing ready-made models and replacing the steps of pre-training with transfer learning.

Neck

The main function of this part is Feature enhancement. In YOLOv5 model, Feature Pyramid Networks are added [10], which is also called Feature Pyramid. In FPN of YOLOv5, as shown in Figure 2, FPN will convolve the input image first, and obtain feature maps of 76*76, 38*38 and 19*19 with different sizes in the process of convolution. Then, the feature map of 19*19 was upsampled twice (nearest neighbor interpolation method), and finally, these feature maps obtained by upsampling were added to the original feature map of the same size.

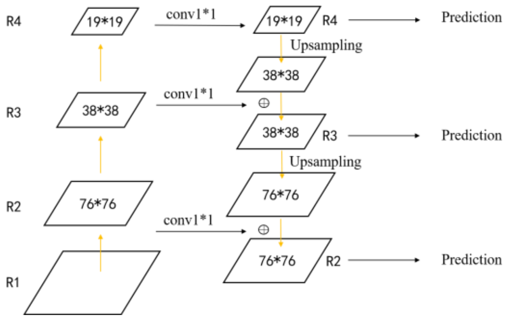

Figure 2. The FPN in YOLOv5.

Before addition, the model will conduct 1*1 convolution for the original feature map. Take R2 and P2 in Figure 2 as an example, the original size of R2 is 76*76*C1, while P2 is 76*76*C2. When C1≠C2, these two feature maps cannot be added together. Therefore, a 1*1 convolution of R2 is required to make C1 equal to C2. In this paper, the channel number is set to 225. Finally, three feature maps with different sizes were obtained, namely 19*19*225, 38*38*225 and 76*76*225. Object detection will be conducted for feature map model of each size, and the final detected target will be integrated into a map. These three feature maps of different sizes correspond to boxes of three sizes respectively. The method of convolution, up-sampling and finally addition adopted by FPN can combine the low-level features learned by neural network in the early stage and the high-level features learned in the later stage, so that the model can obtain more comprehensive target features in classification.

Head

It is the part responsible for output results in the whole model. In the YOLO series, only object classes and Bounding boxes are required. In this paper, we only focus on the object whose class is classified as "person".

B. *Tracking*

After receiving Bounding boxes from YOLOv5(PA), Deep Sort first selects detection targets and Bounding boxes according to confidence score. In this paper, MOT20 standard is adopted, and targets with confidence below 0.5 will be excluded (the range of confidence is 0~1).
Mahalanobis distance is mainly used to measure the feature distance between two persons during feature comparison in the subsequent Re-ID.

The Hungarian algorithm aims at finding the maximum number of matches and matching as many targets as possible. From the perspective of MOT, it means to associate each target as much as possible between adjacent frames.
Kalman filter is used for trajectory prediction in Deep Sort. It can predict the target's state at time t according to the target's state at time t-1.

IV. IMPROVED METHOD

A. Anchor Box Size

In the Neck part of YOLOv5, FPN provides three anchor boxes of different sizes for each feature map of different sizes, altogether nine anchor box sizes. These anchor boxes of different sizes are mainly used to match targets of different shapes and sizes, and match NMS to select the final Bounding box. However, NMS is not selected in this model, and the accuracy of Bounding Box in model is more dependent on the accuracy of Anchor Box given at the beginning. In addition, if the original anchor box size is continued to be used, plural targets may be boxed into a Bounding box. Therefore, we need to improve the size of the original Anchor box to get the anchor box that is more suitable for MOT20. The improved size of the anchor box will be closer to the size of individual pedestrian targets in MOT20, so as to improve the accuracy of Bounding box selection of model frames.

This is only a significant improvement on datasets like MOT20, which are mostly from a surveillance overhead view, with relatively small targets. Therefore, we adjusted the size of Anchor Box to be close to the target average size in MOT20. However, in MOT17, MOT16 and other datasets, a good result can be obtained even without this step adjustment. The reasons are as follows: The original anchor box size of YOLOv5 is suitable for traditional datasets such as COCO and ImageNet, and the target size of such datasets will not be too small compared to the whole image. The target sizes of MOT16 and MOT17 datasets are also close to those of COCO or ImageNet datasets, so better detection results can be obtained.

Although this change in this paper is only for the MOT20 dataset, its generality is not limited to this dataset. At present, MOT is more and more applied in the field of crowd monitoring and traffic flow monitoring.

As shown in Figure 3, the original Anchor Box of YOLOv5 can be roughly divided into three different proportions: a, b and c. We removed the Anchor Box for class c because people of this proportion are not possible in the MOT20 dataset. Category a and b are exactly corresponding to persons with standing and sitting positions. Therefore, we reserve these two categories and adjust their proportions to better match the average Bounding Box size of persons in MOT20. Among them, we further subdivide category a with different proportions to obtain two categories a1 and a2, which can take into account both taller persons and shorter persons.

B. PANet

In addition, we also improved FPN by adding the architecture of PANet [13] after FPN. As shown in Figure 4, we changed the model with PANet architecture into YOLOv5 (PA). FPN improves the traditional feature extraction method by adding high-level features to the shallow feature map, it is more conducive to classification. Also, PANet's improvement on FPN is to add low-level features into the deep feature map to improve the target positioning accuracy of the model, that is, Bounding Box accuracy. Why is the low-level feature helpful for target positioning? Because the low-level features are mostly features such as edge shapes, these features are particularly important when the model is doing instance segmentation, especially pixel-level segmentation. The improvement of PANet lies in its better transmission of low-level features than FPN.

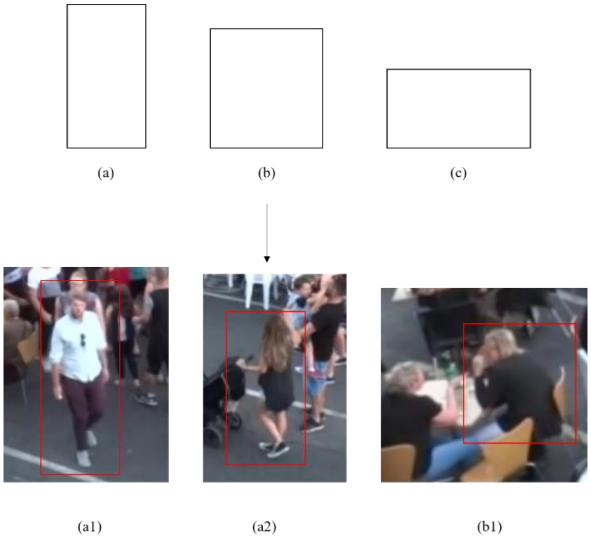

Figure 3. The types of new anchor box size

As shown in Figure 4, we can see the process of low-level feature transmission in FPN through the orange arrow. Low-level features can be transmitted to P4 only through the convolution of R2, R3 and R4. Although only THREE layers of R2, R3 and R4 are drawn in the schematic diagram, in fact, YOLOv5, R2, R3 and R4 all contain a large number of residual blocks. Therefore, the low-level features actually enter dozens or hundreds of network layers during transmission. In this process, the low-level features are inevitably lost, and few of them can be successfully transmitted. Compared with P4, although the process of low-level feature transferring to P3 and P2 has undergone fewer convolution, it still has dozens of layers.

As shown in Figure 4, the process of low-level feature transmission in PANet is represented by a green arrow. The low-level feature is first transmitted from R1 to N2 through P2 through 1*1 convolution twice, and then transmitted to N4 through two more convolution times. The convolutional network here is a very simple single-layer convolutional network, rather than a large set of residual blocks. Compared with FPN, it passes through very few layers and has few feature loss, so it can transmit more complete low-level features to the deep network. Experiments show that the accuracy of Bounding Box selection is improved after the addition of PANet. Moreover, compared with FPN, the amount of computation increased by PANet architecture is almost negligible, so it does not affect the overall running speed of the model.

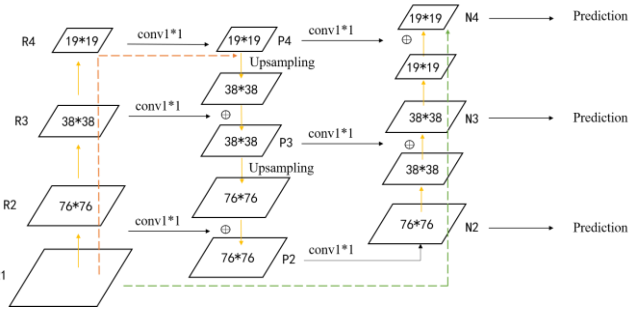

Figure 4. Architecture after adding PANet.

C. Aligned ReID

In the traditional Deep Sort architecture, kalman filter is used to predict the trajectory, and a simple CNN composed of six residual blocks acts as the function of Re-ID to do feature matching. This method has achieved good results in MOT16 and MOT17. However, in MOT20, due to the large number of targets, the trajectories of different targets overlap and a large number of interludes also appear. However, the original Re-ID model cannot perform feature matching well, which makes it easy for kalman filter to appear IDSW phenomenon due to matching errors in trajectory prediction. Therefore, we try to replace the Re-ID model and use a more effective model to enhance feature learning and clustering among targets with the same feature, so that the model can better track targets.

In the end, after many attempts, we chose Aligned ReID [14] as the new Re-ID approach and ResNet50 as the Re-ID model. Aligned ReID not only compared global features of persons, but also used dynamic programming to align local features. We will explain dynamic programming in the future, and the so-called local feature alignment refers to matching the local features of two targets one by one to facilitate the subsequent calculation of feature distance.

In general, Aligned ReID studied the local features of the target, associated all the local features as global features, and then did the final feature comparison between the two targets directly by global features. Compared with the Re-ID method which only considers the global features, the Re-ID method not only satisfies the integrity of the whole feature of the target but also gives good consideration to the local differences. Aligned ReID not only noted local features and human spatial structure as well as methods that considered only local features, but also greatly reduced computation time and cost by using only global features in the end.

As shown in Figure 5, the image is cut into multiple regions of the same size, and Aligned ReID sequentially compares the distances of each two small regions in the feature space. Starting from the comparison between the first layer in Figure 5(a) and the first layer in Figure 5(b), the first layer in Figure 5(a) corresponds to the head of the pedestrian while the first layer in Figure 5(b) is only the background. Obviously, the feature distance between the two layers is large. So Aligned ReID continues to compare the first layer of Figure 5(a) with the second layer of Figure 5(b). And so on, until a certain layer with the lowest feature distance of the first layer in Figure 5(b) is found in Figure 5(a). At this point, it is judged that the features of the two layers are similar, and then the two layers are matched together, as shown in the thick arrow in the figure. In Figure 5, we can see that matching the first layer in Figure 5(a) is the fourth layer in Figure 5(b).

Then, the second layer in Figure 5(a) is compared with each layer in Figure 5(b) in sequence, and the layer in Figure 5(b) with the shortest feature distance from the second layer is selected and matched. Finally, each layer in Figure 5(a) has the corresponding layer with the shortest feature distance in Figure 5(b), which completes the local feature alignment between the two images. However, in the final feature comparison of the two images, the first, second and third layers of Figure 5(b) will not participate in the comparison if they are not matched. In this way, the model will be less disturbed by environmental factors when comparing targets, especially its identification ability of the same pedestrian in different situations will be greatly improved, so that the feature distance between multiple images of the same pedestrian will be closer, and it will not be easy to lose or follow the wrong target in target tracking, effectively reducing IDSW.

a_i and b_j are used to represent the feature vectors of layer I in Figure 5(a), and layer J in Figure 5(b), respectively. After normalizing them, the feature distance $d_{a,b}$ between any two regions can be calculated by formula,

$$d_{i,j} = \frac{e^{\left\|a_i - b_j\right\|^2} - 1}{e^{\left\|a_i - b_j\right\|^2} + 1} \qquad i, j \epsilon\ 1, 2, 3, \cdots, H.$$

where H represents the number of region division. Take Figure 5 as an example, H=7.

Figure 5(c) shows us the process of feature alignment of Figure 5(a) and Figure 5(b) by dynamic programming. Each point in the figure represents a characteristic distance. For example, the point (1, 1) in the upper left corner represents the characteristic distance between the first layer in Figure 5(a) and the first layer in Figure 5(b). As shown in Figure 5(c), it takes at least 13 points to go from point (1, 1) to point (7, 7). Connecting these 13 points constitutes a "path", and the length of this path is the sum of the characteristic distances represented by each point passing through. We use I →j to indicate that the i-layer of 3-3(a) corresponds to the j-layer of Figure 5(b), and the final correspondence between Figure 5(a) and Figure 5(b) is 1→4, 2→5, 3→5, 4→6, 5→6, 6→7 and 7→7. For the model to find the shortest path, the points (1,4), (2,5), (3,5), (4,6), (5,6), (6,7), and (7,7) representing the seven sets of corresponding relationships must all be on that path.

We use $S_{i,j}$ to represent the shortest distance from (i, j) to (1,1) at any point in Figure 5(c).

ISBN: 978-1-68558-074-2

$$S_{i,j} = \begin{cases} d_{i,j} & i = 1, j = 1 \\ S_{i-1,j} + d_{i,j} & i \neq 1, j = 1 \\ S_{i,j-1} + d_{i,j} & i = 1, j \neq 1 \\ \min\left(S_{i,j-1}, S_{i-1,j}\right) + d_{i,j} & i \neq 1, j \neq 1 \end{cases}$$

For example, in Figure 5, Aligned ReID's goal is to find the shortest path from (7, 7) to (1, 1), which is the minimum $S_{7,7}$. Note that this shortest path does not represent the final characteristic distance of the two graphs in Aligned ReID.

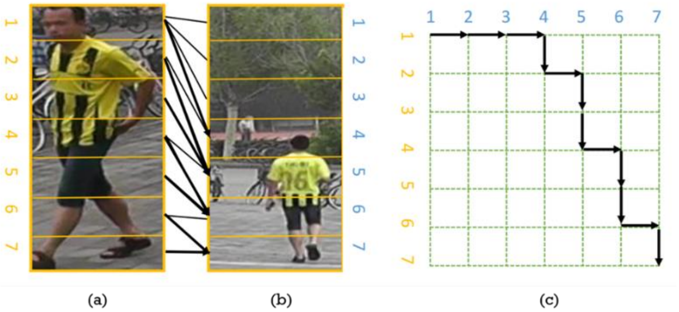

Figure 5. Aligned ReID to find shortest paths [14].

V. Experiments

A. Datasets

In this paper, there are two datasets involved in model training: CrowdHuman [15] for training object detection model YOLOv5 (PA), and Market-1501 [16] for training Aligned ReID.

1) CrowdHuman

We used the CrowdHuman database, which is open source by Cut Technology, a leading AI unicorn company in China. Why CrowdHuman?

Although the traditional YOLOv5 [18] model trained from COCO [17] dataset performed well in MOT16 and MOT17, due to the huge increase in the difficulty of MOT20 dataset, it could not achieve such satisfactory results in MOT20.In MOT20, the difficulty of multi-target tracking of this dataset is much higher than that of several MOT datasets due to the increase of the number of persons in the same frame, the decrease of the proportion of persons to the whole graph and the frequent occurrence of mutual occlusion among persons. Although COCO is a good dataset of quantity and quality, the models trained from it still do not meet the MOT20 requirements.

After comparing with several datasets, we finally chose to use CrowdHuman instead of COCO to train the model. We believe that CrowdHuman is currently the most suitable dataset for training pedestrian detection models, especially dense crowds. Here are some comparisons between the CrowdHuman dataset and the COCO dataset.

As shown in Table I, CrowdHuman far outnumbered COCO in the data volume of persons, and the average number of people in each graph was also higher than COCO, which was more suitable for MOT20 with a dense crowd. Although the COCO dataset is also a well-known object detection dataset with a wide range of content, the proportion of the dataset allocated to persons is not as large as other pedestrian-specific datasets.

The MOT20 dataset is taken in a very different way from the original MOT16 and MOT17 datasets, with cameras instead of close street shots. The pedestrian in the picture is relatively small and difficult to detect. Moreover, most of the persons in the picture are shot from a overlooking Angle, which is not common in COCO. In COCO dataset, the target size of persons is usually moderate relative to the whole image, so it is difficult for the model trained with COCO dataset to accurately detect the persons of small size, and they are often taken as the background, resulting in "missing report". However, in CrowdHuman's dataset, there are not only conventional images like COCO, but also many images taken at a distance or from an overhead perspective, which greatly improves the model's ability to detect persons in various situations.

There is a very deadly marking method for MOT tasks in the COCO data set when marking Ground Truth, which we call "crowd marking". COCO data assembly makes multiple persons with similar positions share an Bounding Box, and then gives this big Box a label of "person", that is, multiple targets are judged as one target. This is not acceptable in MOT, where any MOT task wants to isolate as many targets as possible.

TABLE I. COPARISION OF DATA AMONG DATA SETS

	Caltech	KITTI	City Persons	COCO Persons	Crowd Human
images	42,782	3,712	2,975	**64,115**	15,000
persons	13,674	2,322	19,238	257,252	**339,565**
persons/image	0.32	0.63	6.47	4.01	**22.64**

In addition, the most special part of The CrowdHuman data set is its Grounding Truth. It provides three different Bounding boxes, namely Head Box, Full Box, Visible Box, which are abbreviated as HBox, FBox, VBox in the following. This is a unique feature of the dataset. HBox is specifically for the head, which will not be used in this paper, while FBox and VBox are for the whole body of persons. The difference is the following: as the name implies, VBox boxes the part of persons that can be seen, while FBox boxes the whole pedestrian, including not only the visible part but also the blocked part. This labeling method does not exist in other datasets, as shown in Figure 6. The training of models using such datasets can well achieve the purpose of modal labeling (Amodal).

Modal labeling is a concept proposed by Deng Z. and Jan Latecki L. [19], originally used for 3D object detection, which refers to enclosing invisible parts of objects in Bounding boxes at the same time. For example, in the detection, only the upper body of a pedestrian is uncovered while the lower body is covered, but the Bounding Box area is extended to the foot of the pedestrian it deems to be completed completely by the modal labeling.

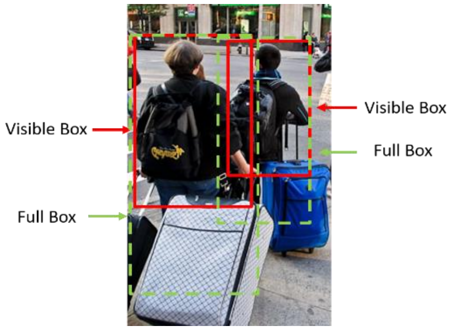

Figure 6. Two different boxes in the CrowdHuman dataset.

2) *Market-1501*

Since MOT20 was much more difficult than before, we retrained a better Re-ID model to replace the original Deep Sort module responsible for this function with the Market-1501 dataset. Market-1501 is a dataset dedicated to Person Re-ID released by Tsinghua University of China in 2015. There are 32668 images in this dataset, and each image is 64*128 in size. There is a totally 1501 different persons. There were 751 people in the training set with an average of 17.2 images per person, and 750 people in the test set with an average of 26.3 images per person. The dataset was captured by six cameras, each with a different resolution. Even the image of the same pedestrian has a large number of different features and environmental interference factors. As shown in Figure 7, each column shows the same pedestrian in different poses with different cameras and shooting angles.

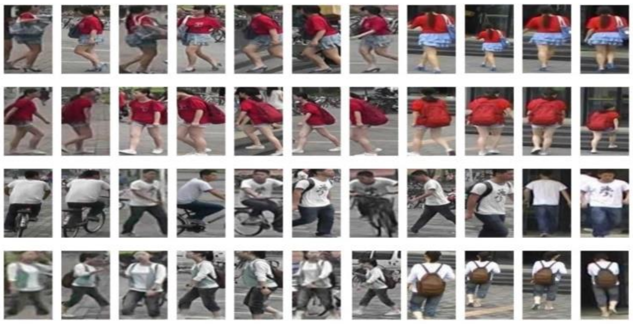

Figure 7. Market-1501 dataset.

B. *Results*

Using MOT20 video as input, the model identifies each pedestrian frame by frame, assigns each pedestrian a unique ID, and keeps the ID corresponding to the pedestrian until the pedestrian leaves the frame. In the MOT20 competition of MOT Challenge, MOTer [20] is the model with many leading data, and we take it as the benchmark for comparison. As shown in Table II, our model's MOTPI and MOTPC were 35.6 and 77.9, respectively, after the training of CrowdHuman dataset. By changing the size of Anchor Box and adding PANet architecture, we improved YOLOv5 and obtained YOLOv5(PA), which further improved the Bounding accuracy of Bounding boxes, contributing greatly to the improvement of MOTPI and MOTPC. But we are still 2% behind Moter in MopI.

With Aligned ReID, we compared local features between targets in addition to global features. Therefore, the model obtained excellent ReID ability, reduced the target mismatching between before and after frames, reduced the occurrence of IDSW, and MOTA was significantly improved to 69.7, surpassing the traditional Sort20 and MOTer.

Moreover, it is worth mentioning that the operation speed of MOTer model is 1 second/frame, that is, it takes about 1 second to process an image. Generally speaking, it is considered that 1 second /12 frames is about 12 images output per second to be a smooth picture. Therefore, it is difficult for MOTer model to achieve real-time, which is fatal in real-time monitoring. However, our model relies on YOLOv5 (PA)'s super fast computing speed to stabilize the processing time of each frame at about 0.1 seconds, which is far superior to MOTer in this point and can achieve basic real-time tracking. Figure 8 shows the results on MOT 20. Table II shows the results compared to some existing results.

TABLE II. THE EXPERIMENTAL RESULTS

	MOTA↑	MOTPI↑	MOTPC↓	s/ frame
MOTer [20]	58.6	**79.8**	/	1.0
Sort20 [2]	42.7	78.5	/	0.7
ours	**69.7**	77.9	35.6	**0.1**

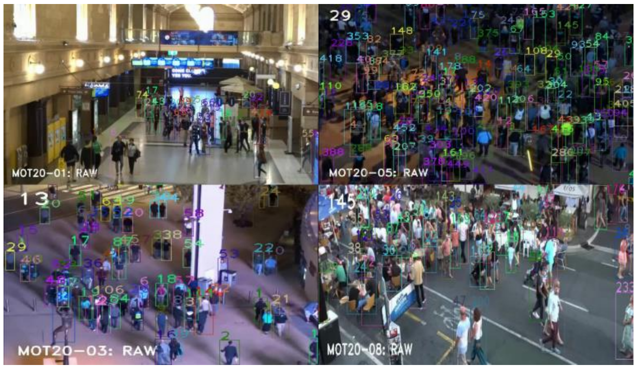

Figure 8. Results in MOT20.

VI. Conclusions and Future Work

In this paper, experimental results show that the performance of the proposed method outperforms that of other approaches. During the testing, for the dataset like MOT20, it is still hard to get better performance due to many people. Hence, in future work, a good method to resolve the crowded people needs to be proposed.

ACKNOWLEDGMENT

This work was supported in part by the National Science and Technology Council under contract numbers 111-2221-E-011 -134 -, 111-2218-E-011 -011 –MBK.

REFERENCES

[1] MOT challenge, https://motchallenge.net/

[2] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple online and realtime tracking," *IEEE International Conference on Image Processing (ICIP)*, 2016.

[3] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *IEEE International Conference on Image Processing (ICIP)*, 2017.

[4] K. He, G. Gkioxari, P. Dollar and R. Girshick, "Mask R-CNN," *IEEE International Conference on Computer Vision*, 2017, pp. 2961-2969.

[5] L. Chen, H. Ai, Z. Zhuang and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," *IEEE International Conference on Multimedia and Expo (ICME)*, 2018.

[6] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," arXiv preprint arXiv:1909.12605 2.3 (2019): 4.

[7] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779-788.

[8] Ultralytics： https://ultralytics.com/

[9] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[10] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature pyramid networks for object detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117-2125.

[11] B. Keni and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, 2008, pp. 1-10.

[12] R. Ergys *et al.*, "Performance measures and a data set for multi-target, multi-camera tracking," *European Conference on Computer Vision*, Springer, Cham, 2016.

[13] S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, "Path aggregation network for instance segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759-8768.

[14] X. Zhang *et al.*, "Alignedreid: Surpassing human-level performance in person re-identification," arXiv preprint arXiv:1711.08184, 2017.

[15] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," arXiv preprint arXiv:1805.00123, 2018.

[16] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang and Q. Tian, "Scalable person re-identification: A benchmark," *IEEE International Conference on Computer Vision*, 2015, pp. 1116-1124.

[17] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," *European Conference on Computer Vision*, Springer, Cham, 2014.

[18] Yolov5: https://github.com/ultralytics/yolov5

[19] D. Zhuo and L. Jan Latecki, "Amodal detection of 3d objects: Inferring 3d bounding boxes from 2d ones in rgb-depth images," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5762-5770.

[20] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "TransCenter: Transformers with dense queries for multiple-object tracking," arXiv preprint arXiv:2103.15145, 2021.

[21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[22] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," *18th International Conference on Pattern Recognition (ICPR'06)*, Vol. 3, IEEE, 2006.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, issue 6, June 2016, pp. 1137-1149.

Exploring Spatial Transformation-Based Privacy in a Small Town

Aidan Jacobs
Computer Science & Information Technology
San Juan College
Farmington, NM 87402. USA
Email: arjacobs1@hotmail.com

Subhasish Mazumdar
Computer Science & Engineering
New Mexico Institute of Mining and Technology
Socorro, NM 87801. USA
Email: Subhasish.Mazumdar@nmt.edu

***Abstract*—As mobile devices become increasingly prevalent in society, the expected utility of such devices rises; arguably, the most impact comes from location-based services as they provide tremendous benefits to mobile users. These users also value privacy, i.e., keeping their locations and search queries private, but that is not easy to achieve. It has been previously proposed that user location privacy can be secured through the use of space filling curves due to their ability to preserve spatial proximity while hiding the actual physical locations. With a space filling curve, such as the Hilbert curve, an application that provides location-based services can allow the user to take advantage of those services without transmitting a physical location. Earlier research has uncovered vulnerabilities of such systems and proposed remedies. But those countermeasures were clearly aimed at reasonably large metropolitan areas. It was not clear if they were appropriate for small towns, which display sparsity of Points of Interest (POIs) and limited diversity in their categories. This paper studies the issue focusing on a small university town.**

***Index Terms*—Mobile environments; Location-dependent and sensitive; Privacy; Query Processing.**

I. Introduction

Users are interested in location-based queries like “find me the nearest C” or “find me k nearest Cs,” where C is a service category like *restaurant* or *gas station*. However, they face the risk of compromising their privacy; consequently, researchers have suggested spatial transformation to address such concerns. The Hilbert curve-based approach maps every geographic coordinate (x, y) into a number $h(x, y)$ through a Hilbert function h that has useful properties: it fills the grid in such a way that consecutive mapped numbers are physically contiguous (though not necessarily the other way around). Accordingly, the *Location Based Server (LBS)* is given values of $h(x, y)$ instead of (x, y), thus encoding both the locations of *Points of Interest* (POIs) and those of users; further, instead of the categories of interest, it gets encrypted descriptions. Thus, the LBS is able to answer queries without being aware of either user locations or the categories of their interest.

Earlier works have explored strategies that a rogue LBS can adopt in order to defeat this method. In [1], semantic factors such as the distribution of POI categories and variations in POI density were considered and countermeasures offered. Further enhancements of the method, in the form of rotation and transposition, were offered in [2]. However, this approach is especially challenging when the grid represents a small town that has only a modest number of POIs — that too in a cluster or two — and limited diversity among the categories. In this paper, we explore that issue by choosing Socorro, New Mexico, our university town, with a population less than 9,000, on which to test the methods.

The paper is structured as follows. In the second section, we outline the basic strategy behind the spatial transform method. In the third, we present related work. The fourth covers the data describing Socorro as well as semantic attacks on that data. The fifth and sixth sections outline our proposed countermeasures based on adding fake POIs and a feature of the L1-variant respectively. We offer concluding remarks in the seventh section.

II. The setup

Suppose we have a $2^N \times 2^N$ grid (containing 2^{2N} cells) corresponding to geographic coordinates (x, y) where x and y are integers in $0 \cdot\cdot (2^N - 1)$. A Hilbert curve H of order N gives a bijective function h that maps each (x, y) into an integer in $0 \cdot \cdot (2^{2N} - 1)$, which is interpreted as the Hilbert cell number. Figure 1 shows an example with $N = 3$ (i.e., an 8x8 grid); it shows the cell numbers $h(x, y)$ inside each cell. The sequence $0 \cdot \cdot 63$ defines the *Hilbert curve* that fills the grid passing through each cell exactly once. We view it as a matrix where the cell at the bottom row and leftmost column corresponds to $H[0, 0]$ representing the geographic coordinate $(0, 0)$. Rotated representations of the curve are useful too; Figures 2, 3, and 4 depict rotations by 90°, 180°, and 270°, respectively.

This function h is contiguity-preserving, i.e., two cells numbered i and $(i + 1)$ in Hilbert space must represent geographical coordinates that are contiguous. However, h may map contiguous coordinates in 2D-space into Hilbert numbers that are *not* even close (e.g., numerically distant cells 5 and 58 in Figure 1 represent contiguous coordinates).

A. Utilizing the Hilbert Curve for Location Privacy

A Hilbert curve is generated by a Trusted Server (TS) after deciding the curve’s parameters. They are: (1) the *order* of the curve N; (2) the (physical location of the) point of

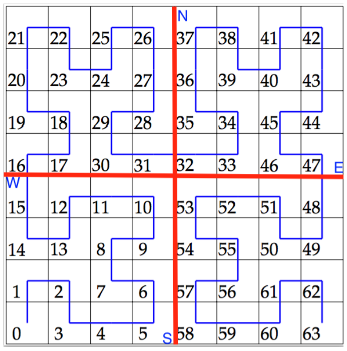

Figure 1. Normal Hilbert Curve for $N = 3$. The cell in the bottom row and leftmost column corresponds to geographical $(0, 0)$ and matrix element [0,0].

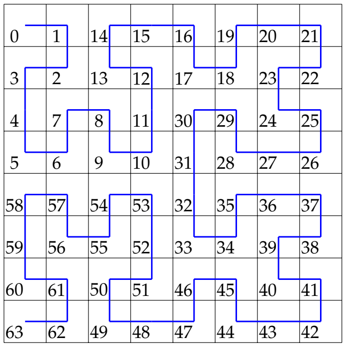

Figure 2. Hilbert Curve for $N = 3$ rotated clockwise by 90 degrees.

origin X_0, Y_0; (3) the *orientation* Θ (*normal* as in Figure 1 or *transposed* (the matrix transpose of Figure 1, not shown); and (4) a *scaling factor* Γ that captures the number of meters that each unit cell represents (in both figures, Γ is the distance in meters covered by the entire grid in either the X- or Y-direction divided by 8). Using Γ and the origin, any geographic location (x_0, y_0) (which could be latitude and longitude), can be converted into a grid cell or matrix element $H[(x_0^*, y_0^*)]$, where $x_0^*, y_0^* \in 0 \cdot \cdot (2^{2N} - 1)$. Thus, the transformation parameters (unknown to the LBS) are $[X_0, Y_0, \theta, N, \Gamma]$.

First, the TS sends a table of POIs with encrypted categories (see Table I). Later, the LBS gets a query containing the Hilbert cell number of the mobile user along with an encrypted (sub)category. The LBS searches for the numerically closest Hilbert cell number containing POIs of that (sub)category and returns it to the user.

III. Related Work

Khoshgozaran et al. [3] first suggested the use of a Hilbert curve for location-based services. Abel et al. [4] compared Hilbert curves with four spatial transformation orderings — row, row prime, Morton and Gray code — and found that Hilbert curves are weaker than Morton in some aspects but superior for preserving contiguity. Moon et al. [5] showed the effect of rotation of the Hilbert curve on clustering. The effects of shifts of the Hilbert curve on the loss of proximity in the Hilbert space were shown in [6] and [7].

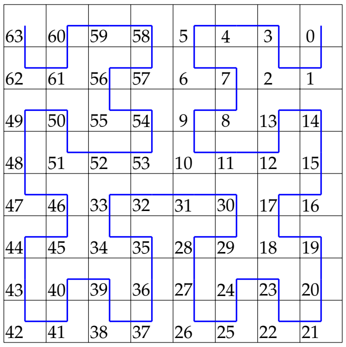

Figure 3. Hilbert Curve for $N = 3$ rotated clockwise by 180 degrees.

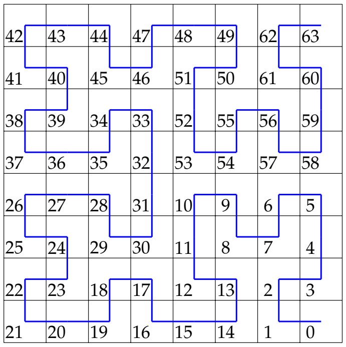

Figure 4. Hilbert Curve for $N = 3$ rotated clockwise by 270 degrees.

IV. Small town: Socorro, NM

We gathered 147 POIs from Socorro, New Mexico, USA, and stored them in a table. Each POI entry consisted of its latitude-longitude pair, category, and subcategories, if applicable. We used categories and subcategories used by *Yelp* and found 19 categories and 3 levels of subcategories. Table II shows a fragment of our table. Figure 5 shows the category hierarchy as graphs: each node contains the number of POIs that fall under that and descendant subcategories. Next, we mapped them with a Hilbert curve of order 4; we found a cluster of POIs with large sparse regions (see Figure 6).

TABLE I
Example of a Table Sent From the TS to the LBS

Hilbert Cell	(encrypted) POI description	(encrypted) Category	(encrypted) Subcategory
43	1547DA276CCDA	9A4027D	0032
...	...	...	...
15	07BB583A9FF46	1C011DD	0120
16	9577CC2D55B2F	9A4027D	0122

TABLE II
A Fragment of the Data for Socorro

Name	category	subcat. 1	subcat. 2	subcat. 3
P.B.S.	beauty / spa			
Nusenda	financial serv.	banking		
Solaro	home services	real estate	solar inst.	
YMG	restaurant	american	steak	seafood

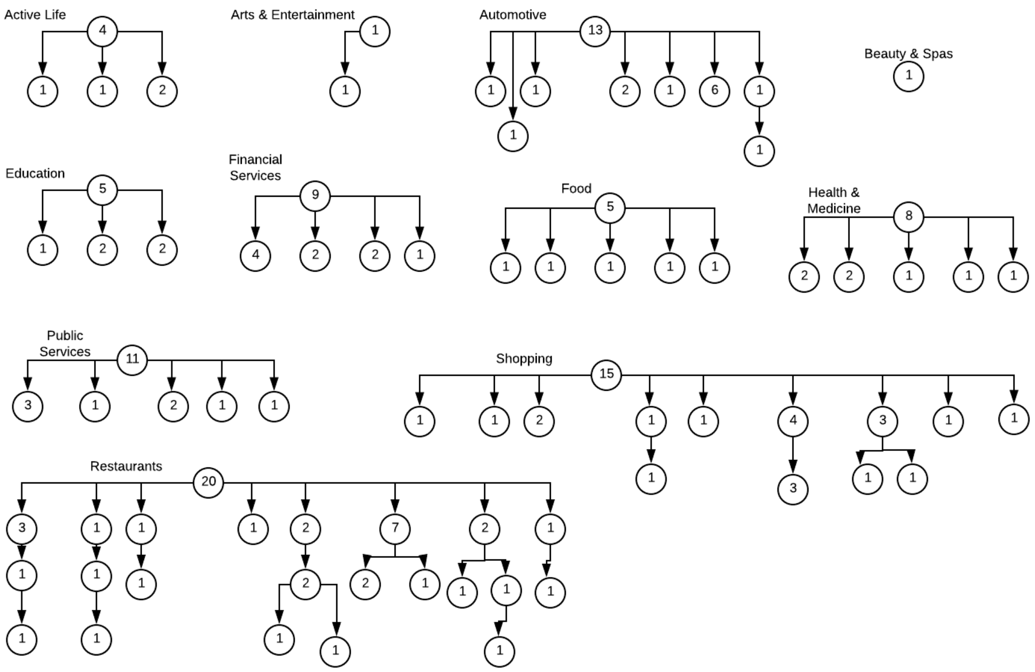

Figure 5. Some category trees for Socorro, NM. Each node represents a sub/category and shows the number of POIs corresponding to the subtree rooted at that node.

A. Attacking the Socorro data

Playing the role of a rogue LBS, we sorted the encoded data by category, sub-category, etc., and tallied the number of instances of each. Next, we associated each POI with a tuple that represented the categories and subcategories which it belonged to. For example, a POI with the tuple (6, 4, 1) would share its category with 5 other POIs, its sub-category with 3 other POIs, and be the only POI in its sub-subcategory.

The categories *Arts & Entertainment* and *Health & Beauty* each had one POI within them. The *Arts & Entertainment* POI was located on the outskirts of town while the *Health & Beauty* POI was located in the center. By counting the number of surrounding POIs, we could easily identify which entry in the table of POIs belonged to *Arts & Entertainment* and which entry belonged to *Health & Beauty*.

Pets was the only category that contained three POIs — two veterinary clinics and an animal shelter. The two clinics were close together while the animal shelter was on the opposite side of town. By comparing cell values within this category, we could easily identify which POI was the animal shelter.

Using these methods, we could identify 21 of the 147 POIs in Socorro. From just two of those identified POIs which fell in different cells, we could derive the scaling factor Γ (following the scheme outlined in [1]).

V. Countermeasures

A. Replication and Rotation Method

If the encoded map contains a significant amount of empty cells (i.e., cells containing no POIs), the LBS can use this feature to make educated guesses about the alignment of POIs. For example, the encoded map for Socorro shows that almost all the POIs fall on the left half (Figure 6).

One way to mitigate this issue is to generate fake POIs and place them in the empty cells. The fake POIs must be distributed in a manner such that the LBS cannot easily discard them. In other words, they must mimic the distribution of real POIs. Furthermore, since users cannot send queries from impossible locations, the LBS may identify some of these fakes if no user ever queries from those locations. (It is true that not every empty space is an impossible location.) So, we require that the TS should generate fake users to periodically query the fake POIs.

We propose a *Replication and Rotation Method*, whose goal is to fill empty space on the encoded map as well as to generate fake user queries that are indistinguishable from real user queries. The method is the following.

- The TS replicates the set of real POIs into four sets — the original set S, S encoded with 90° rotation, S with 180° rotation, and S with 270° rotation — and superimposes them onto the same grid. For example, suppose $N = 3$ (8x8 grid) and there are just 2 POIs in (2,3) and (5,0). From Figures 1, 2, 3, and 4, they are 11, 59 (in the 0° curve); 54, 44 (90°); 33, 25 (180°); and 28, 14 (270°). So, the list of POIs the LBS gets is [11, 14, 25, 28, 33, 44, 54, 59]. Furthermore, the categories in the 0° set are encrypted once; those in the 90° set are encrypted twice; and those in the 180° and 270° thrice and four times respectively. The LBS gets a single table containing all four sets.

- The LBS gets four queries for each user query; they contain the user's location encoded at 0°, 90°, 180°, and 270° rotations. Thus, if the user is at (4,1), the LBS gets four queries from 57, 46, 27, and 12. In addition, the POI category desired by the user also goes through this process: encrypted once for the first query (0°), twice, thrice, and four times for the other three (90°, 180°, and 270° rotations respectively).
- Since the LBS receives these four queries (not simultaneously), they appear to be from different locations.
- The user's computational module only accepts the response to the first. It rejects the other three, which are detected when (a single) decryption results in gibberish. The delays and arrival order of the responses are inconsequential.

We applied this system to our Socorro data set. The improvement in coverage of the curve can be seen by comparing the distribution of colored cells (presence of POIs) in Figure 7 against that in Figure 6.

To estimate the expected improvement in coverage, let p be the fraction of the grid that is filled before applying this method and P be the fraction of the grid that is filled after. Given p, the value of P is expected to be

$$P = p + p(1-p) + p(1-p)^2 + p(1-p)^3 \quad (1)$$

$$= 4p - 6p^2 + 4p^3 - p^4 \quad (2)$$

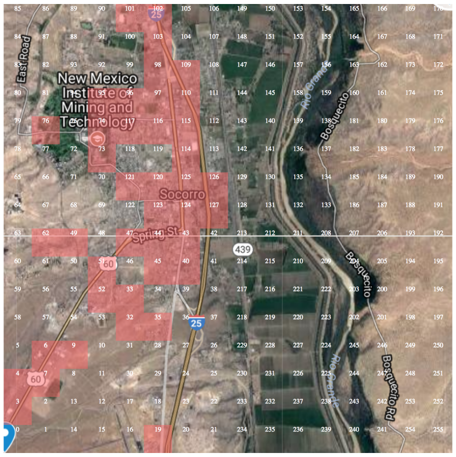

Figure 6. Socorro POIs encoded at order 4.

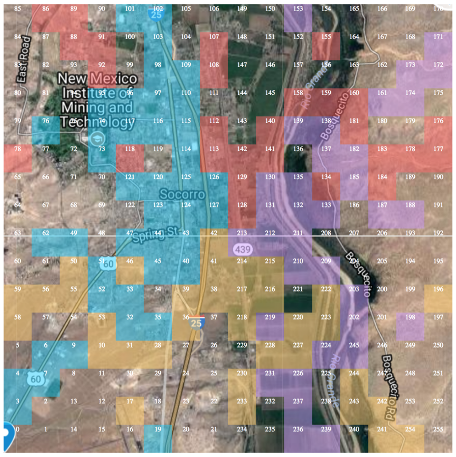

Figure 7. Replication and rotation method applied to Socorro POIs.

In (1), the first term represents the original data in which a fraction p of cells contain POIs. The next three terms reflect the three steps of *Replication and Rotation* where the fractions of still-empty cells are reduced by p. Figure 8 plots this improved coverage P against the initial coverage p.

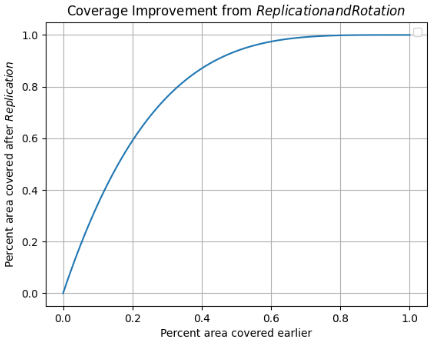

Figure 8. Improvement in coverage from *Replication and Rotation*.

B. Balancing the Category Distribution

We found that this system is still susceptible to semantics-based attacks arising from the asymmetry in the category / subcategory hierarchy. Also, if there is only one POI of a particular category and all other categories have more than one POI, then the LBS can find out that four is the minimum number of POIs of a particular (encrypted) category and guess a four-fold replication scheme.

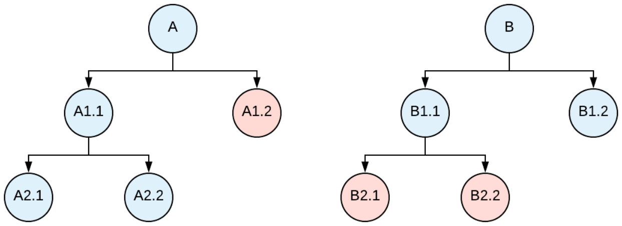

Figure 9. An example of balancing the category trees. Blue cells represent original categories with real POIs. Since trees A and B are different, fake categories (red cells A1.2, B2.1, and B2.2) with fake POIs are added.

The solution for the problem of variation in category hierarchy is to add fake sub-categories so that every category has the same number of sub-categories, sub-sub-categories, etc., thereby making the trees similar. For example, in Figure 9, the original trees for A and B consisting of the blue nodes are clearly different; we compensate for the difference by adding the red nodes.

Later, fake POIs would be added for those fake categories. Users should be presented with the appropriate lowest-level subcategories to choose from based on their initial query. Thus,

ISBN: 978-1-68558-074-2

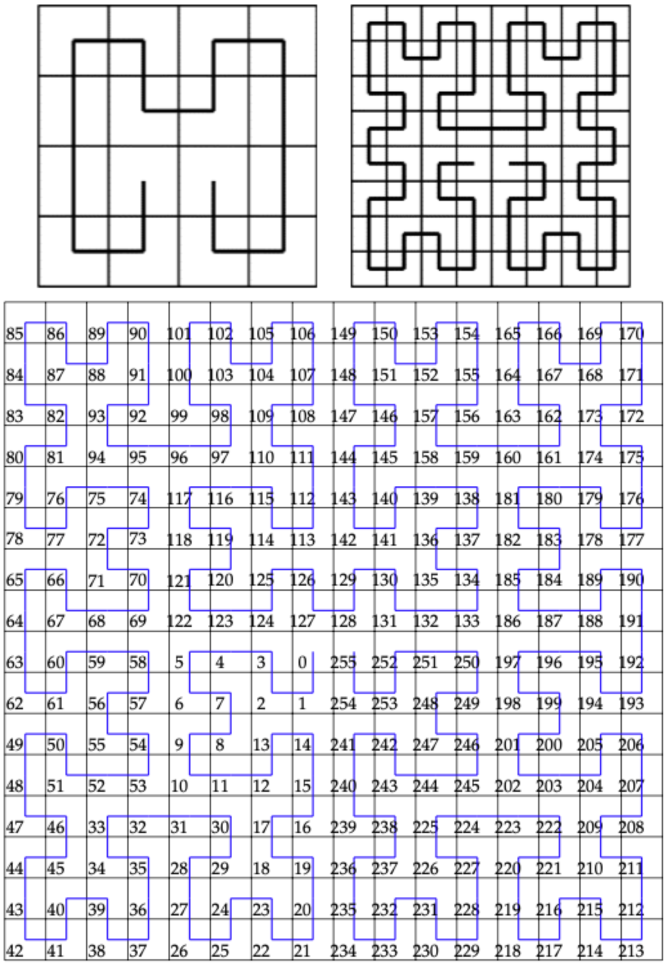

Figure 10. Orders 2, 3, and 4 of the $L1$ variant of the Hilbert curve.

a fake POI will never be returned to a real user. As mentioned earlier, the TS must generate fake user queries for fake POIs.

Balancing the category tree for the Socorro data set (Figure 6) required over twice as many fake POIs as real POIs. This is feasible for our small town but may not be useful for large cities. In summary, balancing the category tree is possible and crucial to stop the LBS from identifying POIs based on the category distribution.

VI. The $L1$ Variant

We also considered the scheme of *offsets*, in which the TS applies an offset to all encoded locations of both POIs and users to confuse a rogue LBS, i.e., it sends Hilbert cell $h(x, y)+k$ instead of $h(x, y)$, where k, a constant, is the offset. Such offsets and wraparounds have been discussed in [2].

We assume that the LBS *knows* that an offset has been applied and has identified Hilbert cells c_1 and c_2, i.e., found the physical locations of the two corresponding POIs, and therefore knows their spatial (geographical) relationship (angle of the vector joining them). For the degree of uncertainty of the LBS, we choose the number of Hilbert cell pairs (c'_1, c'_2) with a similar spatial relationship that could have yielded (c_1, c_2) through the offset scheme (considering rotation too). If the number is low, then the LBS can guess the amount of offset that was applied and which of the curve rotations was selected.

Taking a curve of order 4, we considered every possible gap g and every pair of cells (c_1, c_2), where $g = |c_1 - c_2|$. We define $n_p(g)$ as the number of such *possible* pairs for a given gap g. For example, allowing only $c_2 > c_1$, $n_p(9) = 247$ but $n_p(254) = 2$, i.e., there are many more pairs for a small gap than for a large one. The total number $\sum_{g=1}^{255} n_p(g)$ = 32,640.

Next, we considered every possible offset and rotation, while ruling out disruptive offsets which occur when a cell is moved off the grid; when that happens, wraparound is the only practical option, i.e., $h(x, y) + k$ gets replaced by $(h(x, y) + k) \bmod 2^{2N}$. Since cell $2^{2N} - 1$ is geographically distant from cell 0, these cause extreme discontinuities, and for a pair of cells, a sudden change in the angle of the vector.

We define (c'_1, c'_2) a *feasible* pair for (c_1, c_2) if (i) (c'_1, c'_2) can be transformed into (c_1, c_2) through offset and rotation; and (ii) the angles of the two vectors (one from c_1 to c_2 and the other from c'_1 to c'_2) are acceptably similar, i.e., are less than or equal to an upper bound α.

Even α = 90° can imply similarity. For example, in Figure 1, suppose the LBS has identified POIs in 8 and 9. It must consider the possibility that $(8, 9)$ was transformed from $(2, 3)$ via an offset of 6 (with no rotation) even though the vectors $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$, joining the centers of $(8, 9)$ and $(2, 3)$ respectively, are at 90° to each other. This is because if the POIs are *physically* in the bottom left corner of cell 2 and top right corner of cell 3, then the vector joining *them* would be almost indistinguishable from $\overrightarrow{v_1}$. We choose α = 45° for our analysis. Our results do not change significantly when α is increased to 90°.

We define $n_f(g)$, the number of feasible pairs for a gap g, as the count of feasible pairs for all possible identified pairs having a gap g. This number is a measure of total uncertainty regarding offset and rotations that the LBS faces (in its attempt to break the offset scheme) for a given gap.

Finally, to get the uncertainty per cell pair for a given gap, we averaged over the possible pairs for each gap. We plotted $n_f(g)/n_p(g)$, the average amount of uncertainty for each gap g; we found that it declined steadily with increasing gap g. Low uncertainty, even if it is only for larger gaps, is a weakness of the offset scheme that the LBS could well exploit.

To address this weakness, we repeated the earlier computation on the $L1$-variant of the traditional Hilbert curve [8], which begins and ends in adjacent cells in the middle of the grid, i.e., cell $(2^{2N} - 1)$ is next to cell 0 (Figure 10). Unlike the traditional curve, there are no sudden disruptions, i.e., a cell does not move off the grid when $(h(x, y) + k) > 2^{2N}$. So, we allowed all offsets $k \in [0 \cdots 2^{2N}]$ (the total number of cells is 2^{2N}) while transforming $h(x, y)$ to $(h(x, y) + k) \bmod 2^{2N}$. We found that the $L1$-variant provides a steady amount of uncertainty even for large cell gaps, thus remedying the weakness of the normal curve. The results are shown in Figure 11.

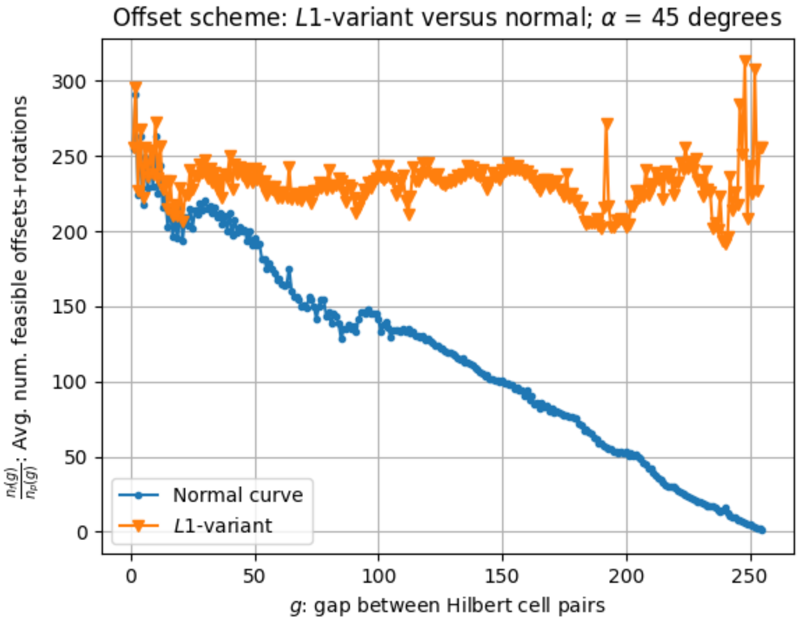

Figure 11. $\frac{n_f(g)}{n_p(g)}$, a measure of strength of the offset scheme, is plotted against g, the gap (i.e., difference) between the Hilbert numbers of a pair of identified POIs. The numerator is the number of *possible* cell pairs with a given gap g; the denominator is the number of *feasible* pairs, i.e, pairs that could have been transformed by offset and rotation into the identified pair such that the angle between the two pairs is less than or equal to α. The ratio gives the average number of *feasible* POI pairs per *possible* cell pair for a given g. Unlike the normal curve, the $L1$-variant shows steady resilience against large gaps.

VII. Conclusion

Spatial transformation via Hilbert curves has been shown to be useful in reclaiming privacy for mobile users when their location-based queries are answered. However, the method shows vulnerabilities when applied to small towns. In this paper, we have analyzed one such town — a university town in New Mexico, USA — and found that a rogue LBS can attack the privacy protection quite effectively, as predicted by earlier researchers, by exploiting the sparsity of POIs and the imbalance in category trees.

We have proposed and tested effective countermeasures, all of which either limit or eliminate potential weaknesses. The *Replication and Rotation* method limits attacks based on sparsity; balancing category distribution effectively counters semantics-based attacks. The $L1$ variant protects the *offset* scheme against compromise of numerically distant cells.

One limitation of our approach is that both the *Replication and Rotation* method and the normalization of category distribution require the additional overhead of posting fake queries. However, the $L1$-method requires little computational overhead because it neither needs fake POIs nor periodic fake queries.

In the future, we hope to better quantify the improvement to location based privacy provided by this method. We also hope to quantify the fraction of POIs required for a general data set, rather than just for Socorro. A full-fledged implementation would allow us to get a quantitative estimate of the response times and unforeseen computational challenges.

Acknowledgment

This work was supported by the National Science Foundation under grants CNS-1757945 and CNS-2150145. We gratefully acknowledge the contribution of Eric Majidian who participated in the research but could not be involved in the writing of this paper owing to contractual obligations with his current employer.

References

[1] A. Paturi and S. Mazumdar, “Can spatial transformation-based privacy preservation compromise location privacy?” in *Proc. 15th Intl. Conf on Trust, Privacy and Security in Digital Business (TrustBus ’18), part of DEXA 2018*, 2018, pp. 69–84, ISBN:978-3-319-98384-4.

[2] A. Paturi and S. Mazumdar, “Exploring origin and rotation parameters while using Hilbert curves in mobile environments,” in *Proc. 8th Intl. Conf on Mobile Services, Resources, and Users (IARIA Mobility 2018), part of DataSys 2018*, 2018, pp. 8–13, ISBN: 978-1-61208-656-9.

[3] A. Khoshgozaran and C. Shahabi, “Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy,” in *Proc. 10th International Conference on Advances in Spatial and Temporal Databases*, 2007, pp. 239–257, ISBN: 978-3-540-73539-7.

[4] D. J. Abel and D. M. Mark, “A comparative analysis of some two-dimensional orderings,” *International Journal of Geographical Information Systems*, vol. 4, no. 1, pp. 21–31, 1990, ISSN: 1365-8816.

[5] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, “Analysis of the clustering properties of the Hilbert space-filling curve,” *IEEE Trans. Knowledge & Data Engineering*, vol. 13, no. 1, pp. 124–141, Jan 2001, ISSN: 1041-4347.

[6] J. Dai, “Efficient range query using multiple Hilbert curves,” in *Current Trends and Challenges in RFID*, C. Turcu, Ed. InTech, 2011, pp. 375–390, ISBN: 978-953-307-356-9.

[7] S. Liao, M. A. Lopez, and S. T. Leutenegger, “High dimensional similarity search with space filling curves,” in *Proceedings 17th International Conference on Data Engineering*, 2001, pp. 615–622, ISSN: 1063-6382.

[8] X. Liu, “Four alternative patterns of the Hilbert curve,” *Applied Mathematics and Computation*, vol. 147, no. 3, pp. 741–752, 2004, ISSN: 0096-3003.