Hot Spots Discovery on Large-scale Rough Taxi GPS Data with Clustering and Density Analysis

Dongchang Liu
Institute of Automation
Chinese Academy of Sciences
Beijing, China 100190
Email: dongchang.liu@ia.ac.cn

Shih-Fen Cheng
School of Information Systems
Singapore Management University
80 Stamford Road, Singapore 178902
Email: sfcheng@smu.edu.sg

Yi-ping Yang
Institute of Automation
Chinese Academy of Sciences
Beijing, China 100190
Email: yiping.yang@ia.ac.cn

***Abstract*—In this paper, we tried to locate hot spots of taxi jobs in a rough large-scale GPS data set from the urban taxi system of Singapore. To fulfill the task, a cluster method which uses features of density peaks, is employed to discover potential cluster centers. However, this method(can be called DPC in short) need to calculate all distance between each pair of data points, that means it is highly time and memory consuming. Therefore, DPC is not suitable for large scale data for time and memory limit. To concur this, we projected all points into a density image and combined DPC with image processes of density image. The experiment result showed that the method we proposed could get similar results with original DPC, but with much shorter time and lower memory. In the end, the clustering results revealed that there indeed a gap between current taxi stand configuration and the real need of the urban taxi system.**

I. Introduction

Taxi service is a significant means of public transportation. Typically,urban taxi service system is operated in three ways: prearranged pick-ups, street pick-ups and taxi stand pick-ups. Although prearranged pick-up is effective enough to meet the need, the entire operating efficiency is not quite satisfactory, because only a small part of passengers will book a taxi beforehand and the efficiency of whole system depends on the other two ways very much. If we take an overview of the urban area, there always a gap between service supply and demand. Our study showed that a taxi fleet would spend over 50% of time idling in a typical day [1]. The underlying reason is asymmetry of service system information. And taxi drivers mainly rely on their own experiences to decide whether searching potential passengers on the road or queuing in a taxi stand rather on real-time information from taxi system. Our research purpose is moving the first step to find hot spots of taxi requirement from GPS data provided by the taxi service system. Notably,hot-spots are not equal to taxi stands. The fact can be verified from a projection map of trips start points. In figure 1, we project all GPS points in one month and generate heat map according to density. The darker areas are regions contain more data points. Meanwhile we denoted locations of taxi stands and parking lines in the region on the same map. And there is a gap between taxi stands locations and density hot spots. There are several stands in low density areas and other hot spots out of list also could be explained. These hot-spots could either be taxi stands or busy parking lines along streets. Their locations are useful in the following three aspects:

1) Hot spots means more available jobs, then could help taxi drivers to find a reasonable routes which pass though one or several hot spots and then more likely to pick up a job quickly.
2) As we can see in conclusion, there's indeed a gap between hot spots and taxi stands in list. Thus it is a good reference to set up official taxi stands in the city.
3) Locations of these spots are the first step of our research, we could find further information about them, for example: range and queue of taxi stands or parking lines, which will offer more useful conclusion to increase the whole efficiency of the urban taxi system.

The taxi service system provides a few types of data, however, in this paper, we plan to focus on taxi trips that record every jobs of each taxi. Trips are generated by the device of taxis when a taxi pick up or drop off passenger(s) and sent back immediately to service centers. A typical trip includes vehicle ID, time and location and other information about both start and end point of a job. Our inputs are locations(GPS data) of trips which record the start points of taxi jobs. The precision of these GPS data are very low. Typically speaking, a taxi GPS device would generate one to two data points per minute and the accuracy of data is about 100 meters to 20 meters [1]. Data points may contain many errors because GPS signals could be disrupted or overshadowed by high buildings, trees and people in complex urban environment. So, it's difficult to locate hot-spots based on rough GPS data above. Though GPS points are not accurate enough to get info about taxi stands and parking lines, if we take a statistic view of the large scale GPS data within a certain period of time we could find hot spots in the target area. The basic steps of hot spots discovery are density analysis and clustering method. Density analysis filter noise of background, clustering methods assign filtered points into different clusters, namely, different hot spots. We transfer GPS locations into density image through points projection. And then many image processes are available, including Otsu binarization [2] and morphological operations [3]. Clustering algorithms are another key to identify different

[1]http://en.wikipedia.org/wiki/Global_Positioning_System Dec.07.2010

clusters with high densities in the data. There are many different kinds of clustering methods based on data density, for example:CLARANS [4], CURE [5] and GDBSCAN [6]. Kernel Density Estimation (KDE)is another branch of similar approaches. The popularity of the KDE approaches come from its easy of use and striking visualizations [7]. A very significant algorithm in these domain is mean shift [8]. Mean shift is a density based nonparametric clustering method and it has the following advantages:1)data driven, 2)self-proved, 3)only one parameter needed. However, the only super-parameter is very important and usually very hard to select. Different data sets need different bandwidths(window sizes). Inappropriate window size would cause modes to be merged or generate additional "shadow" modes. So an important improvement in mean shift research is to use adaptive window size. A related method called Variable Bandwidth Mean Shift [9] offer a way to cluster data set automatically. Since 2014, a new type of clustering method [10] has been emerged, which extended search of cluster centers from one dimensional to two by adding a new property of centers. In this paper we called it "Density Peaks Clustering" (DPC). This an approach based on the idea that cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. This idea forms the basis of a clustering procedure in which the number of clusters arises intuitively, outliers are automatically spotted and excluded from the analysis, in the mean while, clusters are recognized regardless of their shape and of the dimensionality of the space in which they are embedded. It has ability to select cluster centers on a graph named "decision graph", which is very suitable to identify hot spots. The DPC need calculate distance between each pair of data points. So it is highly time and memory consuming. We revise the algorithm and combined it with density image process which would largely speed up the original algorithm and decrease its memory use.

This paper is organized as followed. In Section III, we formally introduce basic steps of our method to detect hot spots and locate them. In Section IV, we provide two experiments which use different clustering method:the original DPC and improved with density images. In the last section, the locating result are analysis and possible improvements are discussed.

II. Related Work

Provide proper review and citation of past related work here: Some content from Introduction can be moved here.

III. Methodology

The method proposed here combined density peaks clustering and density image process. So we introduce the original DPC algorithm first and then describe how density image improved the algorithm. We focus on algorithm details, parameter selection of each method will be illustrated in the next section.

A. Density Peaks Clustering

The density peaks clustering is naturally for hot spots identification, cause it consider not only densities but also distances between different density peaks. Meanwhile hot spots are those clusters that spread across the noisy background. The details of this clustering process is showed in algorithm 1. The real strength of DPC lays on its cluster center selection method. The prior density clustering methods find potential centers merely based on densities. Therefore it employ a two dimensional criterions other than one dimension. Another key point of DPC is that it use relative densities in neighborhood system when calculate distance between density peaks. Therefore, it could find proper cluster centers and remove outliers in the neighborhood of each cluster centers automatically. However, there is no free lunch in the world. To get the density of each point they need all distances between each pair of data points to calculate projection radius for density estimation. This progress is very memory and time consuming. It will take a large amount of calculation and memory when computing all distances between each pair of points. For instance, to deal with a data set with 168668 data points, we need a double(8 bytes for each element) matrix of 168668×168668 to store all distance. It will need consume about 212 gigabytes memory. Furthermore, halo points removing that is also a time consuming step, may delete points that belong to the true clusters without considering the real shape of these hot spots, which in turn leads to loss some important properties of those clusters (contour for instance). There are our motive to improve this method and use it into taxi GPS data set.

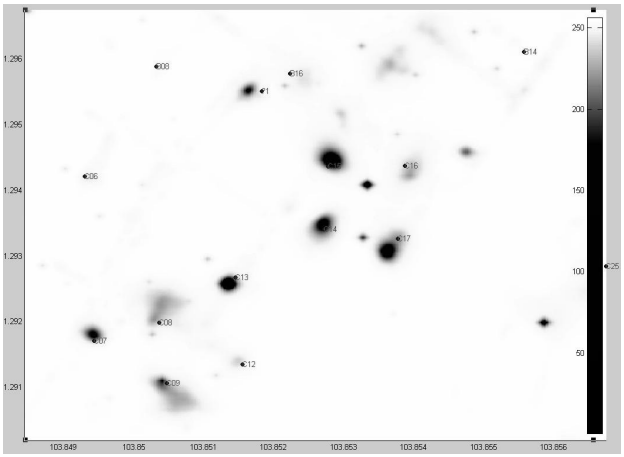

Fig. 1. Square Map Around City Hall

$$\rho_i = \sum_j exp\{-1.0 * (d_{ij}/d_{nb})\} \tag{1}$$

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}) \tag{2}$$

B. Improved Density Peaks Clustering

The kernel of DPC is cluster centers selection in a 2D design graph. It is a very creative idea. But its density peaks analysis process need a lot of time and memory. Can we estimate the density of each point without massive calculations and memory consuming and then extend this method to large-scaled data

Algorithm 1 Density Peaks Clustering

initiate input $P = \{p_1, p_2, \ldots, p_n\}$,output $C = \bigotimes$
super-parameter:M number of cluster centers, K the ratio to calculate neighborhood distance d_{nb}
calculate distance matrix between each pair of points $n \times n$
define neighborhood distance d_{nb} is the K-th smallest in all distance, $K = 2\% * n * (n-1)/2$
estimate local density of each point ρ_i, with formulate 1
compute distance of different density peaks δ_i, with formulate 2. For point with largest density, we assign the it's δ with the largest distance
select cluster centers based on $\gamma = \rho * \delta$, the points with top M large γ will be selected for cluster centers
assign points into different clusters based on ρ in descending order.
remove halo points (points in low density areas) if needed.
output cluster result C

Algorithm 2 Improved Density Peaks Clustering

initiate input $P = \{p_1, p_2, \ldots, p_n\}$,output $C = \bigotimes$
super-parameter:M number of cluster centers, H a fixed window size
project all GPS points by their locations with H
get rid of isolated pixels and smooth the density image I
binaries I with Otsu's threshold and find out points in high density areas or white area on binary image B
select deputy points $P' = \{p'_1, p'_2, \ldots, p'_m\}$ from each valid white pixels on B,
we also get their density of each point ρ_i from the pixel value of I
compute distance of different density peaks δ_i, with formulate 2. For point with largest density, we assign the it's δ with the largest distance
select cluster centers based on $\gamma = \rho * \delta$, the points with top M large γ will be selected for cluster centers
assign points into different clusters based on ρ in descending order.
skip the halo step, cause we've already removed the low density areas in Otsu binaries step .
output cluster result C

set. Can we remove the halo points and get contours of these clusters in the meantime? Both of the answers are yes. We could project these GPS points to form a density image and then employ some image processes on the image to speed up the algorithm. Details about improved density peaks clustering are described in algorithm 2. The density image from point projecting improved the DPC algorithm from the following aspects:

1) density image speed up the density estimation of each data point with small amount of memory. Because projection is very simple to carry out.
2) density image could help to select a representative point from each lattice of the image. This removes redundancy from the data and speed up the algorithm further.
3) a lot of excellent image processes may be used on this density image to filter background noise, which can replace halo removing process and remain the shape of these hot spots.

IV. Experiments

A. Data Preprocessing

Taxi trips of our research come from the taxi system in Singapore. Trips record every jobs that each taxi has ever done. They are generated by GPS device installed on each taxi. When a taxi pick up or drop off passengers, relevant information is recorded and after each job a trip record is sent back immediately to the schedule center. A typical trip includes vehicle ID, start time and location, end time and location and other information about the job. You can get a clear view of trips from Table I. Of course, there are many invalid trips in the data, we screened them with different filters. This is a huge data set. We could only use monthly (Sep. 2009) taxi trips in a limited square are in center of Singapore. We sample only one piece of data in the system, but there are still 168668 trips remain. Moreover, according to our research target (namely finding hot zones of potential passengers), we intercept longitude and latitude of start points in trips data set as input data. We project all input data and the projection map are showed in Figure 2.

TABLE I
STRUCTURE OF TAXI TRIPS

Name	Definition	Example
VID	Identity of vehicle	8
TID	Identity of trip	2164E1003
STime	Start time of a trip	1267374600
SLongitude	Longitude of start point in a trip	103.90583
SLatitude	Latitude of start point in a trip	1.39938
ETime	End time of a trip	1267374960
ELogitude	Longitude of end point in a trip	103.92881
ELatitude	Latitude of end point in a trip	1.38951
Distance	Distance of a trip	6.3
Fare	Fare of a trip	870

B. Parameter Selection

Both of algorithms introduced above need super parameters which are specified beforehand. For original DPC, M (number of cluster centers) and K (the ratio to calculate neighborhood distance d_{nb}) are needed. For improved one, M and H (fixed window size) are key parameters. In this paper, we suppose M in both algorithm is 20, which is equal the main blobs number in binary image B. K is K-th smallest in all distance, $K = 2\% * n * (n-1)/2$ is recommended formula to calculate K in original DPC reference. The last parameter is window size H also called resolution of data-image transformation, which is used to constitute lattices for GPS projection. It is highly relevant with the precision of GPS points. To prove this, we designed an experiment with simulate data set. We use simulate data because the real data has no labels at all. We

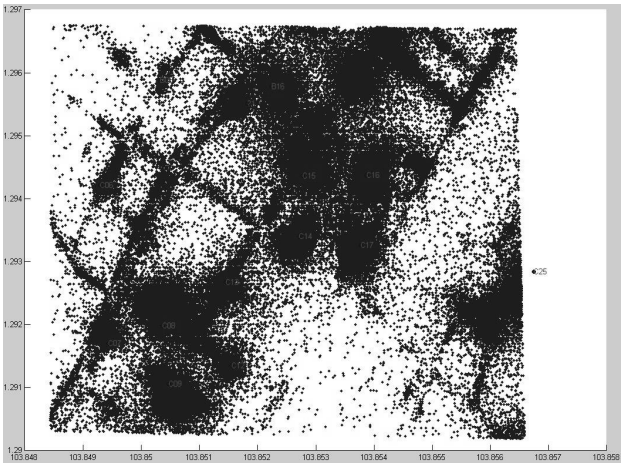

Fig. 2. Projection Map of Original Input

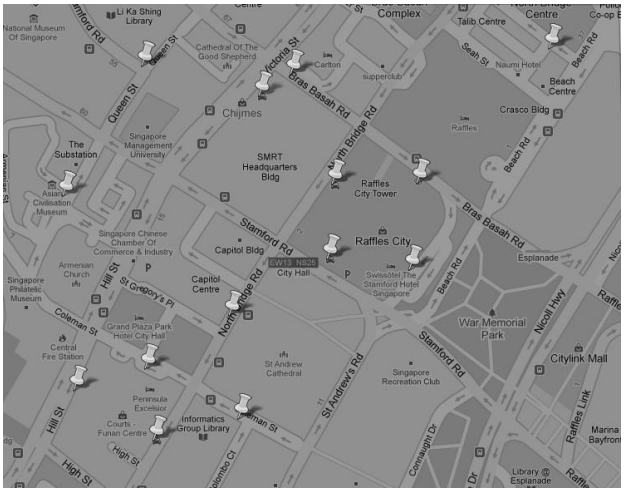

Fig. 3. Square Map Around City Hall

cannot accurately evaluate the result without any ground truth. We generate the ideal data set exactly based on information from the real world. The real instance we chose is a square district around City Hull MRT in Singapore. Its map [2] is showed in Figure 3. The map offers us locations of taxi stands and main roads in the region. We also obtain other information of taxi stands and use it to simulate taxi trips. To simplify the simulation, we suppose that all main roads in square are straight lines and trips in non-stand areas are uniform distribution and trips in stands obey normal distribution. We generate simulate data points have the same precision with real GPS data set. According to rules above, we could get taxi trips and their cluster labels in simulation data set. Figure 4 shows the projection of all simulation trips. Taxi trips and cluster labels are input of parameter optimization experiment. To find the optimum resolution, we introduce receiver operating characteristic(ROC curve) [3] here as selecting criterion. According to materials online, the precision of civilian GPS is from from 100 meters to 20 meters [4]. So, the upper bound of resolution is about 0.0001 degree (about 10 meters in real distance). We define the lower bound as 1% of the upper bound (if the value is not proper we can adjust it at any time). we tried different resolutions(H) in $[1-4e, 1-6e]$ and plot the ROC curve as Figure 5. we find that if the square size is in $[0.5-5e, 1.1-5e]$, the True Positive Rate (TPR) is high and False Positive Rate (FPR) is low. On the other hand, smaller resolution means larger amount of calculation in process. Therefore, $1.0-5e$ is the optimal unit size which is also the data precision of real GPS data points.

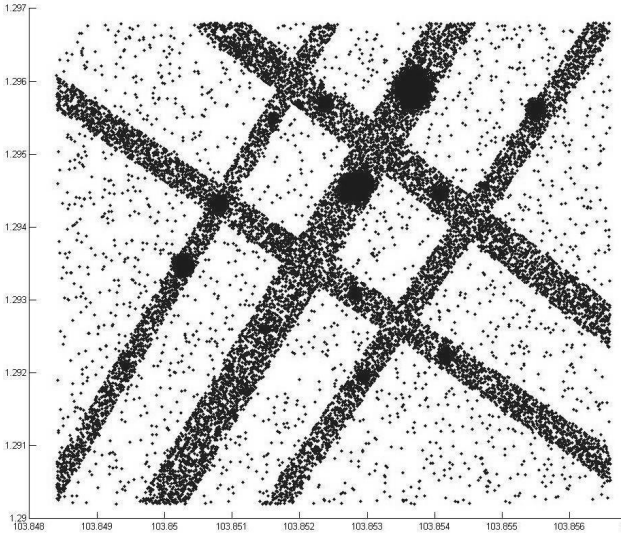

Fig. 4. Projection of Simulated Trips

C. Improved Density Peaks Clustering

The very first step to get density image is projecting all data points by their GPS locations. And the optimal unit size is fixed at $1.0-5e$. We could get densities of all different lattices whose length is the unit size. Till then, we get a

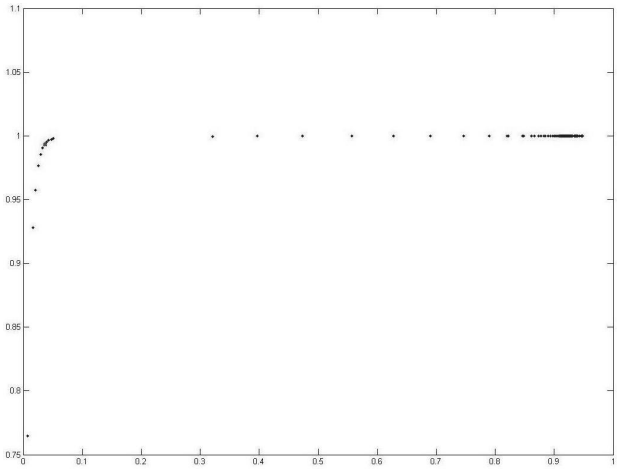

Fig. 5. ROC Curve of Different Resolutions

[2]http://maps.google.com.sg/maps

[3]http://en.wikipedia.org/wiki/ROC_Curve Dec.07.2010

[4]http://en.wikipedia.org/wiki/Global_Positioning_System Dec.07.2010

density image that each pixel of it is a density value in that lattice. Then we remove isolate peaks which are error trips in the square. Pixels that are five times more than sum of their 5×5 neighbors are set to zero. The last step of density image generation is array smooth with a 3×3 gaussian kernel. The final density image(showed in Figure 6) is binarized with Otsu's method which is one of the best parameter free methods as we know. Figure 7 showed the binary result in last step process connected component analysis. All components smaller than 200 pixels are removed, and small holes on binary image are filled. After that we get a mask image, as Figure 8, that can be used to filter input data. Because all points in a lattice(one pixel) have a same density, we select one data point in every lattice as the represent point for each location. This step is very important to speed up the original DPC, cause the data point are reduced from 168668 to 27656 points. In the mean time, we also get all density values of these remained points. We can use pixel values corresponding to their locations as estimation density which again saved a lot of time and memory. These 27656 points and their densities are input for density peaks calculation. And after this step, we would get a 2D array records density and peak distance of every point. According to the array, we get a projection map which are named decision graph. We could select cluster centers on decision graph manually, however, to choose them more objectively, we use the product of density and peak distance $\gamma = \rho * \delta$ as the only criteria. The points with top 18 large γ will be selected for cluster centers(showed in Figure 9). From these cluster centers, the program assign other points a label. We clustered those representation data points and then we assigned all points in a same lattice with same cluster ID. Figure 10 shows clustering result and their cluster centers. We do not need the following halo step in original DPC for image binarize has already removed background noise, which cuts down calculation and memory consuming in the third time. In final step of our algorithm, we find out contours around each hot spot to indicate range of them. Figure 11 shows these polygons and cluster centers. These ranges mean a lot for we could tell taxi drivers weather they are in a hot zone or not by compare their location with these polygons. To compare the result we also plot taxi stands and some parking lines in list.

D. Original Density Peaks Clustering

We input these 168668 points directly into the original density peaks clustering method. Unfortunately, we can not run matlab scripts on our PCs because of the error "out of memory". 168668 points apparently too large for original DPC. We have to run DPC clustering method on our compute servers that equipped with 1TB memory. It used a couple of hours to get clustering result. And the results including decision graph and cluster labels are plotted in in Figure 12 and Figure 13. The clustering result after halo removed is smooth than our algorithm. But these clusters are too smooth to offer contour information about hot spots. So we could not get correct boundaries with DPC. Next, we use the halo result to find information on hot spots. And the final result is showed

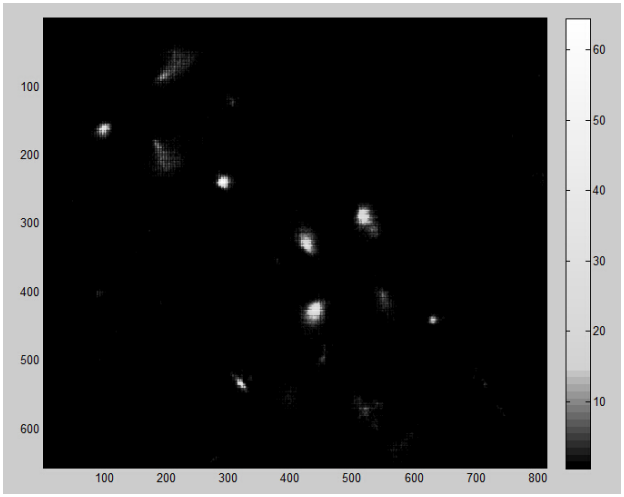

Fig. 6. Final Density Image

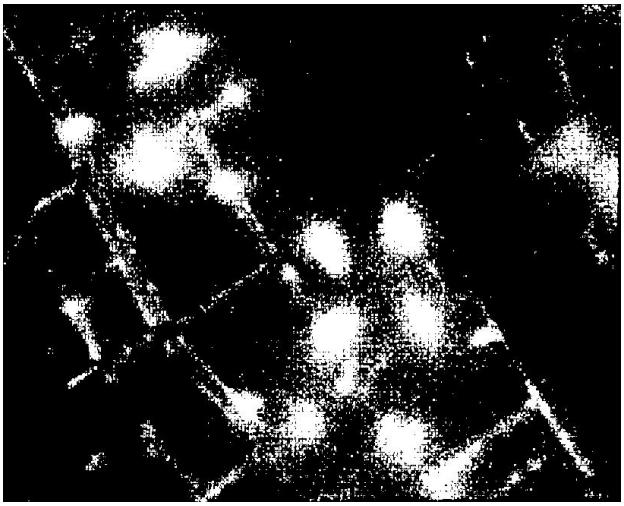

Fig. 7. Binary Image after Otsu's Method

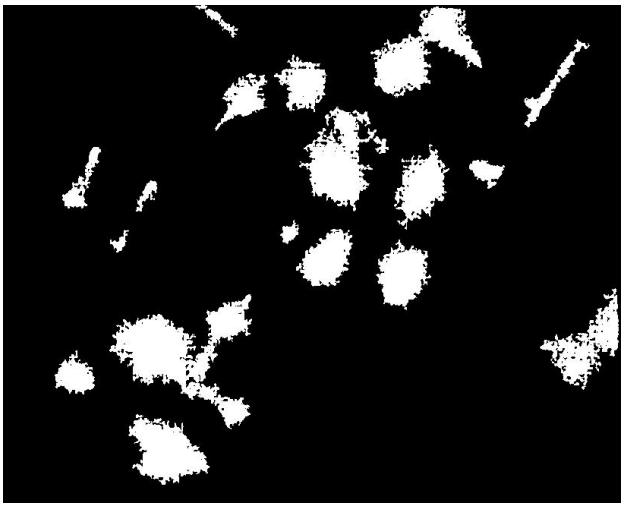

Fig. 8. Binary Mask to Filter Data

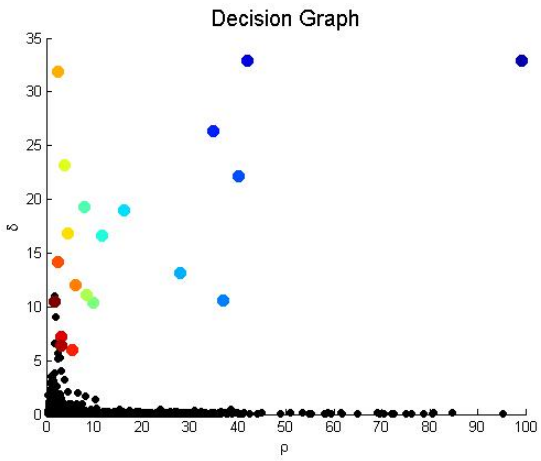

Fig. 9. Decision Graph for Cluster Center Selection

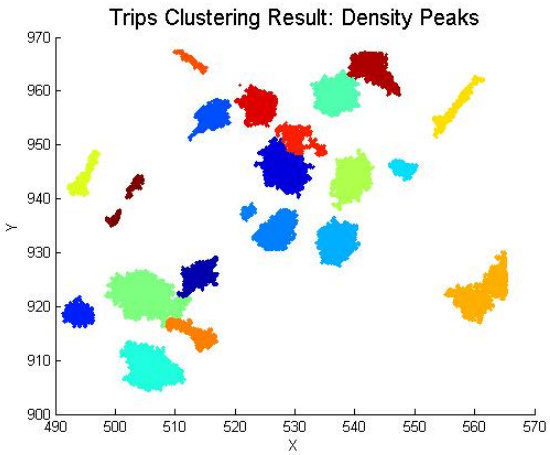

Fig. 10. Clustering Result of Improved DPC

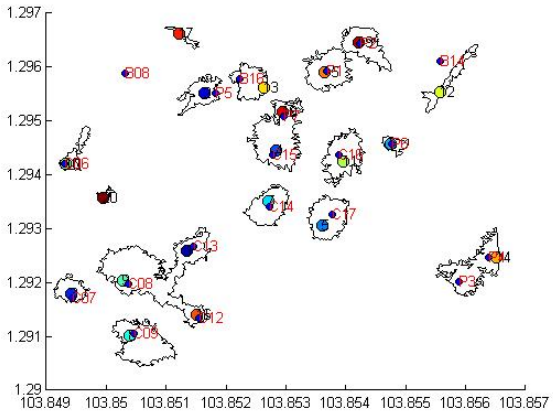

Fig. 11. Contours and Centers of Hot Spots

in Figure 14.

E. Performance Comparison

13005.227016 seconds 238GB 78.995863 seconds 7.75MB

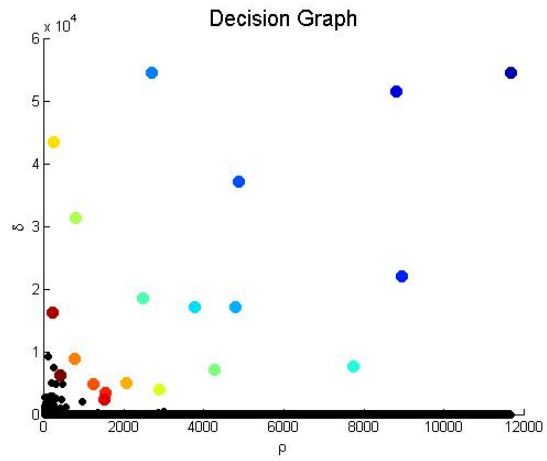

Fig. 12. Decision Graph for Cluster Center Selection

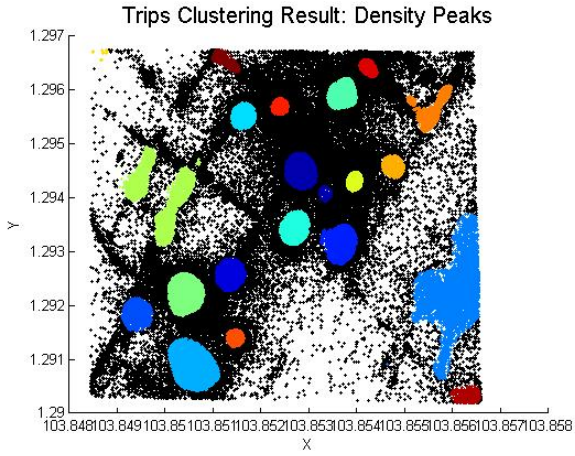

Fig. 13. Clustering Result of Improved DPC

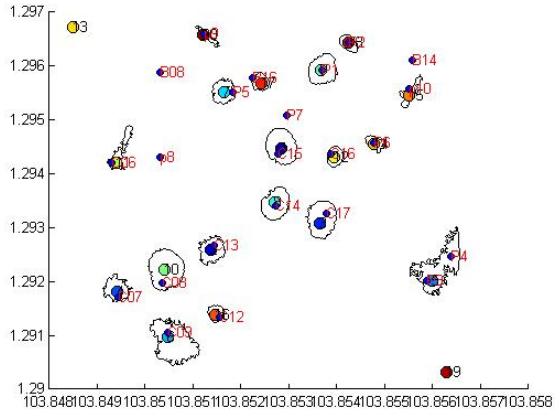

Fig. 14. Contours and Centers of Hot Spots

V. Conclusion

A. Performance Discussion

From experiments in last section, we prove that 1) combined DCP with density image analysis could largely improved DCP

TABLE II
TIME AND MEMORY CONSUMING

Method	Time	Memory
Original DPC	55640s	238GB
Improved DPC	79s	7.75MB

TABLE III
REAL DATA EXPERIMENT ANALYSIS

Ground Truth	Improved DPC	Original DPC
C17 taxi stand	05	03
C16 taxi stand	11	12
C15 taxi stand	04	01
C14 taxi stand	07	08
C13 taxi stand	01	02
C12 taxi stand	16	16
C09 taxi stand	08	06
C08 taxi stand	09	10
C07 taxi stand	03	04
C06 taxi stand	10	11
B16 taxi stand	13	17
P1 parking lines	15	09
P2 parking lines	19	18
P3 parking lines	miss	05
P4 parking lines	14	miss
P5 near hotels	02	07
P6 near hotels	06	14
P7 parking lines	18	miss
p8 parking lines	20	miss
p9 parking lines	17	20
p10 parking lines	12	15
Error Centers	–	13,19

in both time and memory consuming; 2) the method proposed is effective to find hot zones in target area whose model includes centers and contours; 3) compared with ground truth information, the result of our method is better than that of original. The first point is showed in Table II. Time consuming is reduced from 13005 seconds to 79 seconds. Memory is cut down from 238GB to 7.75MB. Therefore, the method we proposed has advantages in dealing with large scaled data sets. The lower time and memory consuming is very beneficial when deal with all data points in the whole CBD area or data accumulating more than one month (ex. couples of years).

To compare the result, we searched information on the web site [5] of Singapore Land Transport Authority. There is a list file including taxi stands locations and ID. Thirteen stands are in our target area and eleven of them are detected as hot spots. Although B08 and B14 are taxi stands in official list, they has very few trips every day. We could not detect them. For other spots, we tried to explain them out based on Google Street View. Interestingly, we find that two of them are near hotels which increase the number of potential passengers for taxis. The rest of unlist hot spots are parking lines (two parallel yellow lines along road-side) on main roads or near shopping malls. We summarize the comparison in Table III. We could find out that comparing with ground truth the method combined DPC with density image analysis got better result than original DPC.

[5]http://www.lta.gov.sg/ Feb.10.2011

B. Future Work

In the future, we still have much work to do. The following are some potential research directions:

- Expand the scope of target area to the whole CBD. There would be much more data to process. The original DPC may fail because of memory limit.
- Arrange trips and logs along each trajectories and analysis behaviors of taxis in a more general view.
- Employ other kinds of information in taxi data, find out different means to help taxi drivers finding a task as soon as possible.

ACKNOWLEDGMENT

. The authors would like to thank...

REFERENCES

[1] S.-F. Cheng and X. Qu, "A service choice model for optimizing taxi service delivery," *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pp. 1–6, oct. 2009.

[2] N. Otsu, "A threshold selection method from gray-level histograms," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, no. 1, pp. 62 –66, 1979.

[3] H. RM, S. SR, and Z. XH, "Image Analysis Using Mathematical Morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 532–550, Jul 1987.

[4] R. T. Ng and J. Han, "Clarans: A method for clustering objects for spatial data mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, pp. 1003–1016, 2002.

[5] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," *SIGMOD Rec.*, vol. 27, pp. 73–84, June 1998. [Online]. Available: http://doi.acm.org/10.1145/276305.276312

[6] M. K. Sander, Jorg Ester and Hans-Peter, "Density-based clustering in spatial databases: The algorithm gdbscan and its applications," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 169–194, April 1998.

[7] A. J. Brimicombe, "Cluster detection in point event data having tendency towards spatially repetitive events," 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.103.8424

[8] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 790–799, 1995.

[9] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," 2001.

[10] A. L. Alex Rodriguez, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, June 2014.