2015 IEEE 15th International Conference on Data Mining Workshops

Geographically Robust Hotspot Detection: A Summary of Results

Emre Eftelioglu, Xun Tang, Shashi Shekhar
Department of Computer Science
University of Minnesota
Email:{emre, xuntang, shekhar}@cs.umn.edu

***Abstract*—Given a set of points in two dimensional space, a minimum radius, a minimum log likelihood ratio and a significance threshold, Geographically Robust Hotspot Detection (GRHD) finds hotspot areas where the concentration of points inside is significantly high. The GRHD problem is societally important for many applications including environmental criminology, epidemiology, etc. GRHD is computationally challenging due to the difficulty of enumerating all possible candidate hotspots and the lack of monotonicity property for the interest measure, namely the log likelihood ratio test. Related work may miss hotspots when hotspots are divided by geographic barriers (the road network, rivers etc.) or when hotspot centers are close to parks, lakes, mountains, etc. To address these limitations, a novel approach is proposed based on two ideas: cubic grid circle enumeration and a grid log likelihood ratio upper bound. A case study on real crime data shows that the proposed approach finds hotspots which cannot be discovered by the related work. Experimental results show that the proposed algorithm yields substantial computational savings compared to the related work.**

I. Introduction

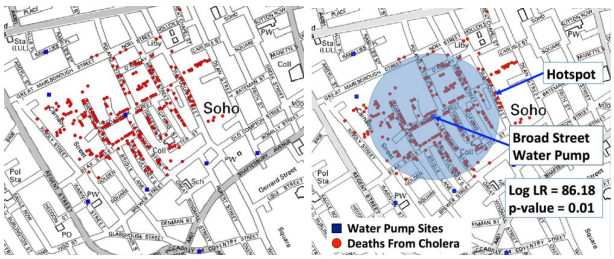
Given a set of geolocated points (e.g. locations of a disease/crime), a minimum radius, a minimum likelihood ratio and a significance threshold, Geographically Robust Hotspot Detection (GRHD) finds hotspot areas where the concentration of points inside is significantly higher than the number of points outside. This paper refers geographically robustness as the ability of being insensitive to minor gaps (e.g, contiguous hotspots separated by road segments, rivers, etc.) in the contiguity of the spatial pattern.

An example of a hotspot is shown in Figure 1. Example shows the London Cholera Outbreak in 1854 including 491 deaths aggregated on 250 house locations (in red) and 8 water pump locations (in blue). The detected hotspot (in blue) has a high test statistic value (i.e. log likelihood ratio) and a high statistical significance (i.e. p-value = 0.01). As can be seen in Figure 1(b), the detected hotspot points out the infected water pump, which may have caused the Cholera deaths.

A. Application Domain:

GRHD is important for application domains such as epidemiology, environmental criminology, ecology, medical imaging, biology, etc. where detection of circular hotspots may reveal important information for domain experts. There are three important concepts related to the detection of hotspots in these domains: Elimination of chance hotspots (prevent false positives), ability to detect a hotspot if it exists (geographically robustness), modeling hotspots with respect to diffusion model.

Elimination of chance hotspots (i.e. false positive) is important since false positive hotspots may result in poor allocation of resources and inefficient management in critical situations (e.g. crime, epidemic). Moreover, chance hotspots may result in stigmatizing as such locations will not be visited by people. Thus in these domains (i.e., epidemiology, environmental criminology, etc.), in order to eliminate chance hotspots, statistical significance test is done.



(a) Input Cholera Cases Dataset (b) Output of GRHD

Fig. 1. 1854 London Cholera Outbreak [1]. The blue hotspot points out the location of the infected water pump.(best in color).

Another important concept in hotspot detection is the ability to detect a hotspot if it really exists (geographically robustness). In some application domains, the cost of failing to detect a hotspot although it really exists has important consequences (e.g. unnecessary deaths). Thus a hotspot detection technique should manage to detect all existing hotspots.

Spread of infectious diseases and crime is tend to be similar to the diffusion model in physics and chemistry. Diffusion means that molecules or heat will move away from their sources once they are discharged. Similarly, diffusion model provides a natural way to describe the "circular diffusion" of cases (i.e. diseases, crimes). For example, most infectious diseases move from their source to physically nearest neighbors and these transmit the disease to their nearest neighbors and so on, causing hotspots of cases around the sources of diseases [2]. This also gives rise to circular footprints of hotspots in isotropic geographies.

Next, two example application domains will be introduced to illustrate these three important concepts.

Epidemiology is the study of distribution and determinants of disease spread across human populations and the applications to prevent and control the spread of a disease [3]. In epidemiology, infectious disease cases are known to follow diffusion models. For example, malaria transmission tends to

978-1-4673-8493-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDMW.2015.159

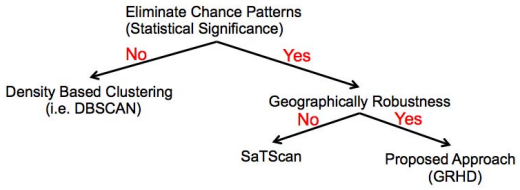


Fig. 2. Related work of hotspot detection.

be spatially heterogeneous around a radius [4]. The ability to detect (geographically robustness) these hotspots will help officials to take the required precautions to prevent the diffusion of a disease and elimination of chance hotspots (prevent false positives) will prevent waste of resources. For example, Ebola outbreak in 2014 raised concerns about the poor allocation of resources and the late reaction of the officials [5].

Environmental Criminology is the study of criminal patterns and how crimes are affected by the physical environment of the criminal [6]. An important theory in environmental criminology namely "Crime Pattern Theory" states that most of the serial crimes diffuse within a radius around an anchor point of a criminal [6]. The ability to detect (geographically robustness) these hotspots of crimes will help officials to decide where to deploy new units to prevent new crimes and will point out the "anchor point" to determine the criminal's location [7], [8]. Similarly, elimination of chance hotspots (prevent false positives) will help security officials to focus their efforts to a specific location and will prevent stigmatizing of neighborhoods. For example, a recent study shows that property values are affected by crime hotspots [9].

B. Challenges:

GRHD is challenging since it is hard to enumerate all possible candidate hotspots in a study area. Additionally, the location and the radius of the hotspots are not known beforehand which makes it difficult to select an enumeration technique. Once candidate hotspot enumeration is done, those are evaluated by their test statistic with a log likelihood ratio test. There are two challenges associated with the log likelihood ratio test: First, it requires a count of the points inside the enumerated candidate hotspot, causing the whole point set to be scanned for each candidate. Second, it is anti-monotonic, meaning a smaller candidate hotspot may have a higher log likelihood ratio than a bigger one or vice versa. Thus, monotonicity based filtering techniques are inapplicable. Furthermore, the test for statistical significance using randomization (Monte Carlo simulation) multiplies the cost.

C. Related Work:

Figure 2 shows the related work for hotspot detection. There are many techniques to find dense regions (i.e. hotspots) in a study area [10], [12]–[14]. For example, density based clustering techniques (i.e. DBSCAN [10], CLIQUE [13], etc.) are capable of finding arbitrarily shaped clusters and their ability to detect a hotspot is high. While these techniques are inexpensive, they tend to generate many false positive hotspots since they lack a statistical significance test. For example, in Figure 3 given an input point set with 800 points with two circular hotspots of 200 points each, DBSCAN produced 8 different clusters as shown in Figure 3(b).

SaTScan [11], is widely used for the detection and evaluation of circular hotspots of diseases in epidemiology. It uses spatial scan statistics, which is known as the most powerful statistical test, to eliminate chance patterns (reduce false positives) [15]. However, its reliance on point-centered hotspots is bad under some geographic conditions. For example, some geographic features such as road segments, rivers, etc. may cause gaps or discontiguity on hotspots (see Figure 3(a)). Also, some geographic features (i.e. lakes, mountains, parks) close to the center of a hotspot may cause not to observe points and therefore cause a hotspot with a sparse center. Therefore, SaTScan risks failing to detect a hotspot (is not geographically robust) in those cases. Also, in some cases SaTScan may return very small hotspots that occurred by chance (suppose two or more points are exactly at the same location) since it lacks a minimum radius r_{min} threshold.

In contrast, proposed approach for GRHD does not rely on points to enumerate candidate hotspot centers and thus it is not affected by the sparseness around the center when detecting a hotspot. Also it is not affected by the gaps or discontiguity caused by the geographic features (road network, rivers, etc.). Therefore its ability to detect a hotspot (if it exists) is better (geographically robustness) in those cases. It also uses the spatial scan statistics to eliminate chance patterns (false positives). Moreover, it uses a minimum radius r_{min} threshold, which eliminates very small hotspots that occurred by chance.

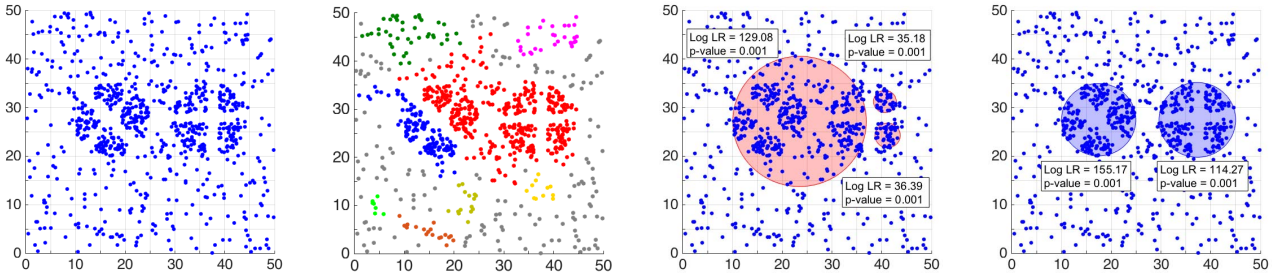
D. Contributions:

This paper formally defines the problem of detecting geographically robust hotspots. To solve this problem, a novel approach namely cubic grid circle algorithm (CGC) is presented. CGC filter phase uses a cubic grid to filter points which will not contribute to a hotspot and refine phase enumerates hotspots using a smallest enclosing circle algorithm. A case study shows that CGC finds hotspots that are not discovered by related work (i.e. SaTScan). Computational analysis and experimental results show that CGC yields substantial computational savings compared to the related work.

E. Scope and Outline:

This paper focuses on geographically robust diffusion hotspots modeled as circles in a two dimensional isotropic space. The underlying population and other variables associated with the point set are not considered (these will be considered in future work). There may be multiple hotspots in the study area and those are assumed to be non-overlapping as described in the SaTScan user guide [11]. In addition, since this paper is focused on circular hotspots as defined in diffusion model, rectangular [16] and/or snake-shaped hotspots as well as the predefined locations [17] are out of the scope.

Section II presents the basic concepts and problem statement of GRHD. Section III reviews the related work (i.e. SaTScan [11]) solution and describes the proposed Cubic Grid Circle (CGC) algorithm. Theoretical evaluation of the proposed approach is covered in Section IV. Section V presents a case study comparing CGC with SaTScan on a real crime data. The experimental evaluation is covered in Section VI. A discussion on the shape of significant hotspot detection is presented in Section VII. Conclusions and future work are covered in Section VIII.



(a) Input (b) Output of DBSCAN ($Eps = 2.5$, $k = 6$). Clusters in different colors. (c) Output of SaTScan (d) Output of GRHD

Fig. 3. Example output of Geographically Robust Hotspot Detection (GRHD) compared to DBSCAN [10] and SaTScan [11] (best in color).

II. Basic Concepts and Problem Formulation

This section introduces basic concepts and defines the Geographically Robust Hotspot Detection (GRHD) problem.

A. Basic Concepts

Definition 1: **A point set** P is a collection of geolocated points (i.e. crime, disease locations). A **point** $p \in P$ is associated with a pair of coordinates (x, y) representing its spatial location in the study area S.

Definition 2: **Study area** S is the minimum orthogonal bounding rectangle of P in the two dimensional Euclidean space. The area of S is denoted as $area_S$.

Definition 3: Given center coordinates (x, y), **Circle** C is the closed curve where any point on its perimeter is equidistant with a radius r to its center. Each C has three parameters: x, y coordinates of its center and its radius r. The area of C is denoted by $area_C$.

Definition 4: **Log Likelihood Ratio** ($Log\ LR_C$) is the interest measure that is used as the test statistic for a candidate circle C [15], [18], [19]. The equation can be shown as;

$$Log\ LR_C = Log\left(\left(\frac{c}{B}\right)^c \times \left(\frac{|P|-c}{|P|-B}\right)^{|P|-c} \times I()\right) \quad (1)$$

$$B = \frac{|P| \times area(C)}{area(S)} \quad and\ I() = \begin{cases} 1, \text{ if } c > B \\ 0, \text{otherwise,} \end{cases}$$

B is the expected and c is the observed number of points in a particular area, $|P|$ is the cardinality of P and $I()$ is an indicator function. $I() = 1$ when the candidate hotspot has more points than expected ($c > B$) and $I() = 0$ otherwise [11].

For example, the area of the circle on the right in Figure 3(d) is $\pi * r^2 = 3.14 * 7.71^2 = 186.62$ and $area(S) = 50 \times 50 = 2500$. Thus $B = \frac{800 \times 186.62}{2500} = 59.72$. In this circle C, there are $c = 199$ points. Finally, $I = 1$ since $199 > 59.72$.

Using Equation 1, $Log\ LR_C = Log\left(\left(\frac{199}{59.72}\right)^{199} \times \left(\frac{800-199}{800-59.72}\right)^{800-199} \times 1\right) = 114.27$

Definition 5: **Monte Carlo Simulation (MCS)** is a randomization test which is used to get the distribution of the test statistic ($Log\ LR_C$). MCS is done as follows; first, m random point sets ($P_{random_{1\ldots m}}$) are created in the study area S. For each $P_{random_{1\ldots m}}$, new circles are enumerated and the maximum $Log\ LR_C$ of each $P_{random_{1\ldots m}}$ is stored in decreasing order in a list, namely $Log\ LR_C^{MCS}$.

Hypothesis Test: In GRHD, the null hypothesis (H_0) states that the points are distributed randomly according to a homogeneous Poisson process over the study area S. The alternative hypothesis (H_1) states that the inside of a circle C has a higher number of points than outside [15]. Using the test statistic ($Log\ LR_C$) of a circle C and the distribution of the test statistic (acquired by MCS), the statistical significance of C is determined. The statistical significance (p-value) of a circle C is computed by finding the position (order) of its $Log\ LR_C$ in the distribution of the test statistic ($Log\ LR_C^{MCS}$) and dividing that position by $m+1$. Given a desired significance level (α_p), if $p\text{-}value \leq \alpha_p$, then H_1 can not be rejected.

B. Problem Formulation

The Geographically Robust Hotspot Detection (GRHD) problem is formulated as follows:

Given:

1) A set of points P where each $p \in P$ has x and y coordinate in a two dimensional Euclidean space,
2) A minimum radius r_{min}
3) A log likelihood ratio threshold (θ),
4) A $p\text{-}value$ threshold(α_p) and a number of Monte Carlo simulation trials (m)

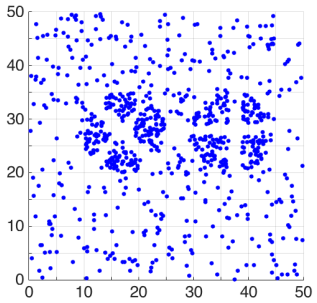
Find: Circular Hotspots ($C(x, y, r)$) in the study area S with $Log\ LR_C \geq \theta$ and $p\text{-}value \leq \alpha_p$.

Objective: Computational efficiency and scalability

Constraints:

1) Correctness of the result set,
2) Detected circular hotspots do not overlap

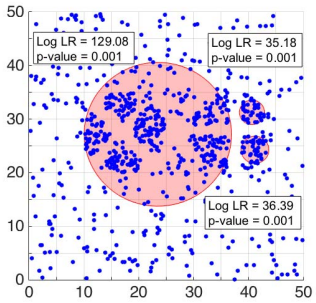
The minimum radius r_{min} input is domain specific and is intended to eliminate very small hotspots (several disease cases in a very small area -a house- may not be interesting). θ indicates the minimum desired $Log\ LR_C$ for a circle C. and α_p is the desired level of statistical significance for a circle C. Depending on the domain, a good practice is selecting the α_p as either 0.01 or 0.001. m indicates the number of Monte Carlo simulation trials and should be selected compatible to the desired level of statistical significance (α_p) [20]. The output of GRHD is non-overlapping circular hotspots with $r \geq r_{min}$ meeting the desired significance and log likelihood ratio levels. Non-overlapping constraint allows to get a single hotspot instead of multiple hotspots given a subset of P [11].



(a) Input

Step 1				Step 2			Step 3
Center x	Center y	r	area(C)	B	c	Log LR	p-value
23.48	27.16	13.42	566.1	181.15	389	129.08	0.001
23.52	27.74	13.42	565.9	181.09	388	128.01	0.001
23.34	27.05	13.28	553.76	177.2	382	126.75	0.001
23.34	27.84	13.52	574.5	183.84	390	126.3	0.001
41.18	24.27	2.54	20.21	6.47	38	36.4	0.001
41.05	24.05	2.42	18.37	5.88	36	35.69	0.001
40.63	31.26	2.29	16.47	5.27	34	35.18	0.001
40.78	31.13	2.24	15.72	5.03	32	32.7	0.001
44.14	43.94	0.99	3.09	0.99	7	7.71	0.76
43.47	43.86	1	3.13	1	6	5.76	0.89

(b) Execution Trace



(c) Output

Fig. 4. Execution trace of SaTScan algorithm with points as centers (in color). Red rows correspond the red circles in Figure 4(c) (best in color).

Example: Suppose, given the point set in Figure 3(a), the aim is to find hotspots with a minimum radius of $r_{min} = 2.5$, minimum log likelihood ratio $\theta = 100$ and minimum level of significance $\alpha_p = 0.001$, even if a geographic barrier (river, road network, etc.) divides hotspots. Figure 3(d) shows the output of GRHD with two hotspots with log likelihood ratio 155.17 and 114.27 respectively. These hotspots have p-value= 0.001 indicating statistical significance at 99.9% confidence.

III. CGC Algorithm

In this paper, SaTScan serves as the baseline algorithm. First, SaTScan is reviewed in detail, an illustrative execution trace is provided and its limitations are summarized. Then a cubic grid circle algorithm (CGC) which is geographically robust (is not affected by gaps or discontiguities) is introduced.

A. Baseline Approach: SaTScan Algorithm

In order to review SaTScan in detail, it is implemented according to its user guide in [11]. The pseudocode of SaTScan, as shown in Algorithm 1, has three steps:

Step 1-Enumeration of Circles: The algorithm takes each $p \in P$ and makes it the center of a circle $C(x, y, r)$. Next, the radius r of $C(x, y, r)$ is determined by the distance of the rest of the points to its center. This step is done for all possible pairs of points in P (line 1-4).

Step 2-Log Likelihood Ratio Test: For each circle C, $Log\ LR_C$ is computed by Equation 1 using $area_C$, $area_S$, the count of the number of points inside (c), and the total number of points in the point set ($|P|$). Among the overlapping circles generated, only the ones which have the highest log likelihood ratio are stored as the $candidateCircles$ (line 5-8).

Step 3-Monte Carlo Simulation & Hypothesis Test: For the circles $C \in candidateCircles$, a $p\text{-}value$ is computed by Monte Carlo simulation. First, m random datasets with Poisson distribution are generated. For each random dataset, new circles are enumerated and the maximum $Log\ LR_C$ of each random dataset is stored in $Log\ LR_C^{MCS}$ in decreasing order. To find the significance of a C, the position of the $Log\ LR_C$ associated with C is determined within the $Log\ LR_C^{MCS}$ list. This position is divided by $(m + 1)$ to determine the $p\text{-}value$ (line 9-12). Finally, all non-overlapping circles with $p\text{-}value \leq \alpha_p$ are returned by the algorithm (line 13).

SaTScan Execution Trace: Figure 4 shows a sample execution trace of SaTScan. There are $|P| = 800$ points in the set P and S is $50 \times 50 = 2500$. The thresholds are set to $\theta = 100$ and $\alpha_p = 0.001$. Figure 4(a) shows the input dataset.

Algorithm 1 SaTScan Algorithm

Input:
1) A point set P with points $p(x, y)$,
2) A $p\text{-}value$ threshold α_p, and A number of Monte Carlo Simulation trials (m)

Output:
Non-overlapping circular hotspots C with $p\text{-}value \leq \alpha_p$.

Algorithm:
Step 1: Enumeration of Circles
1: **for each** point $p_i \in P$ **do**
2: $C \leftarrow$ coordinates of p_i as center x, y
3: **for each** point $p_j \in \{P - p_i\}$ **do**
4: $C \leftarrow$ distance d_{p_i,p_j} as r
Step 2: Log Likelihood Ratio Test
5: **for each** C created in Step 1 **do**
6: Compute B, c, and $area_C$
7: $Log\ LR_C \leftarrow$ Log Likelihood Ratio using Equation 1
8: Add C and $Log\ LR$ to $candidateCircles$
Step 3: Monte Carlo simulation
9: **for each** Monte Carlo simulation $trial_1 ... trial_m$ **do**
10: Create P_{rand} and determine the $max(Log\ LR_C^{trial})$
11: Insert $max(Log\ LR_C^{trial})$ into the ordered $Log\ LR_C^{MCS}$ list
12: Determine the p-value of the circles
13: Return $candidateCircles$ with $p\text{-}value \leq \alpha_p$ as $significantCircles$

Step 1 generates all possible circles by assigning each $p \in P$ to be the center of a circle and then calculating the circle's radius. For illustration purposes, only 10 out of $800 \times (800 - 1) = 639200$ circles are shown in Figure 4(b).

Step 2 computes the $Log\ LR_C$ of each generated circle. First, the count (c) of the points inside a circle C is determined by computing the distance of every point to the circle center (a point whose distance is less than or equal to the radius r of a circle is determined to be inside that circle). For each enumerated circle, expected number of points B is computed using $area_C$, $|P|$, and $area_S$. Then, $Log\ LR_C$ of each circle is computed using B, c, and $|P|$. These values are shown in Figure 4(b). The circles in the white rows are overlapped with the circles with higher $Log\ LR_C$. Thus they are removed from the $candidateCircles$.

Step 3 determines the statistical significance of candidate circles using Monte Carlo simulation. The p-values of 10 circles are shown in Figure 4(b). Red rows in Figure 4(b) correspond to the output (i.e. red circles) in Figure 4(c).

As demonstrated, SaTScan relies on points as the centers which causes it to miss hotspots with sparse centers. In Figure 4(c), two hotspots are divided into multiple portions.

x_{min}	x_{max}	y_{min}	y_{max}	c
1	5	1	5	6
6	10	6	10	5
11	15	11	15	5
16	20	6	10	5
16	20	16	20	14
16	20	26	30	37
31	35	21	25	30
46	50	46	50	7

(a) Count Grid

x_{min}	x_{max}	y_{min}	y_{max}	r_{min}	r_{max}	$\widehat{LR_{Grid}}$
16	20	26	30	6	10	760.43
36	40	26	30	6	10	665.93
21	25	26	30	11	15	635.79
21	25	21	25	11	15	614.55
...	...	...	...	...	...	...
46	50	1	5	11	15	12.42
46	50	46	50	11	15	12.42
6	10	1	5	21	25	10.26

(b) Cubic Circle Grid

Fig. 5. Count and Circle Grid cells for $l_{cell} = 5$ (best in color).

Although these can be spotted visually, the output of SaTScan does not align with them and not satisfy $\theta = 100$ threshold (although shown in Figure 4(c)). In addition, SaTScan does not have a minimum radius. This may cause small hotspots (if two points have the same location, detected hotspot will have $area_C = 0$ causing $LR_C = \infty$) in the output that may occurred by chance. In addition, due to the costly circle enumeration and $Log\ LR_C$ computation, execution time of SaTScan becomes exorbitant for large datasets.

B. Proposed Approach

In this paper, a cubic grid circle algorithm (CGC) is proposed in order to address the following issues: (1) eliminate chance patterns, (2) detect non-contiguous (divided by rivers, road segments, etc.) or sparse center hotspots, (3) eliminate very small hotspots, (4) improve the scalability that is affected by the cardinality of P.

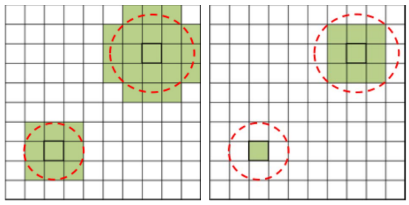
CGC consists of three phases. Filter phase enumerates hotspots in a parametric space, filters those which do not survive an upper bound on likelihood ratio ($Log\ \widehat{LR_{grid}}$) and return $filteredSets \in P$. Refine phase enumerates actual hotspots using $filteredSets$ and returns a hotspot with the highest $Log\ LR_C$ for each $filteredSet$. Finally, Monte Carlo simulation phase assesses the statistical significance of the enumerated hotspots. It should be noted that although the problem formulation states a non-overlapping constraint and refine phase returns only one hotspot for each $filteredSet$, the refine phase can be tweaked to return all generated hotspots.

Basic Concepts:

Definition 6: **A count grid** with cell length l_{cell} is a partitioning of the study area S into a 2-dimensional grid where each cell is a square with an area of $l_{cell} \times l_{cell}$. The number of count grid cells is denoted by $N \times N$, where $N = side\_length(S)/l_{cell}$. Each count grid cell ($cell_{count}$) is defined by its coordinate intervals $([x_{min}, x_{max}], [y_{min}, y_{max}])$ and the count (c_{cell}) of the points inside.

Figure 3(a) shows a point set (P) with 800 points in $S = 50 \times 50$ units. Suppose $l_{cell} = 5$, then $N = 50/5 = 10$ and the total number of the count grid cells is $10 \times 10 = 100$. Figure 5(a) shows 8 $cell_{count}$ with their point counts c_{cell}. For example, $cell_{count} = ([16, 20], [16, 20]))$ has $c_{cell} = 14$ as shown in the fifth row.

Definition 7: **A cubic circle grid** is a three dimensional grid which represents sets of circles in parametric space that are defined with two dimensional center coordinate intervals $([x_{min}, x_{max}], [y_{min}, y_{max}])$ and a radius interval $[r_{min}, r_{max}]$. Given an $N \times N$ count grid, a cubic circle grid is a $N \times N \times N$ grid which parameterize the space into cells with $([x_{min}, x_{max}], [y_{min}, y_{max}], [r_{min}, r_{max}])$. Cubic circle grid cells are denoted by $cell_{circle}$. A $cell_{circle}$ represents a collection of circles with $C(x, y, r)$ where $(x_{min} \leq x \leq x_{max})$, $(y_{min} \leq y \leq y_{max})$ and $(r_{min} \leq r \leq r_{max})$.



(a) MECC (b) MFCC

Fig. 6. Illustration of minimum enclosing (MECC) and maximum fit cell collections (MFCC) shown in green which represent a circle (in red)

Figure 5(b) lists several cubic circle grid cells for the dataset in Figure 3(a) when $l_{cell} = 5$. For example, the top row shows a $cell_{circle} = ([16, 20], [26, 30], [6, 10])$. Count grid and cubic circle grid can be matched by the first two columns.

Definition 8: Given a count grid, a **Minimum enclosing cell collection (MECC)** is a collection of $cell_{count}$ which encloses a set of circles with radius $r \leq r_{MECC}$ where r_{MECC} is the radius of the MECC. The green cells in Figure 6(a) represent an MECC for any actual circle with r less than that of C shown in dashed red lines.

Definition 9: Given a count grid, a **Maximum fit cell collection (MFCC)** is a collection of $cell_{count}$ which can fit completely inside a circle with radius r where $r_{MFCC} = \lfloor \frac{r}{l_{cell}} \rfloor$ and thus $r_{MFCC} \leq r$. The green cells in Figure 6(b) represent an MFCC for a C shown in dashed red lines.

Definition 10: **The grid upper bound likelihood ratio** ($Log\ \widehat{LR_{grid}}$) is an upper bound of the log likelihood ratio of the collection of circles which are defined by the cubic circle grid cells. Given a $cell_{circle}$ with $([x_{min}, x_{max}], [y_{min}, y_{max}], [r_{min}, r_{max}])$, $Log\ \widehat{LR_{grid}}$ equation is [21]:

$$
\begin{aligned}
&Log\ \widehat{LR_{grid}} = Log\left(\widehat{LR_{int}} \times \widehat{LR_{ext}} \times \widehat{I()}\right), \text{ where} \\
&\widehat{LR_{int}} = \left(\frac{U(c)}{L(B)}\right)^{U(c)}, \text{ and} \\
&\widehat{LR_{ext}} = \begin{cases} \left(\frac{|P|-L(c)}{|P|-U(B)}\right)^{(|P|-U(c))}, & \text{if } L(c) \geq U(B) \\ 1, \text{otherwise} \end{cases} \\
&\widehat{I()} = \begin{cases} 1, & \text{if } U(c) > L(B) \\ 0, \text{otherwise} \end{cases}
\end{aligned}
\tag{2}
$$

$U(c)$ is an upperbound of c, $L(c)$ is a lowerbound of c, $U(B)$ is an upperbound of B and $L(B)$ is a lowerbound of B.
$U(c)$ = Number of points in MECC and $L(c)$ = Number of points in MFCC
$U(B) = \frac{area(MECC) \times |P|}{area(S)}$ and $L(B) = \frac{area(MFCC) \times |P|}{area(S)}$

Note that the grid upper bound log likelihood ratio can be applied when $MFCC$ has at least one count grid cell and $MECC$ has less than $N \times N$ count grid cells.

$\widehat{LR_{int}}$ and $\widehat{LR_{ext}}$ are the upper bounds of the two parts of the multiplication in the Equation 1, representing the interior

and the exterior of the circle, respectively. In order to make the grid upper bound log likelihood ratio greater than the actual log likelihood ratio of the hotspots represented by $cell_{circle}$, $\widehat{LR_{int}}$ and $\widehat{LR_{ext}}$ are defined. $\widehat{LR_{int}}$ is always greater than $(\frac{c}{B})^c$ in Equation 1. $\widehat{LR_{ext}}$ is conditioned on whether $L(c) \geq U(B)$ or not. The indicator function $\widehat{I()}$ is similar to the indicator function in Equation 1 and is set to $\widehat{I()} = 1$ when the $U(c)$ is higher than the $L(B)$ under the hypothesis H_0 [21]. Finally, it is worth mentioning that the cell length l_{cell} makes the upperbound tight/loose and $\lim_{l_{cell} \to 0}(Log\ \widehat{LR_{grid}} = Log\ LR_C)$.

For example, in the second row of the table in Figure 5(b), $Log\ \widehat{LR_{grid}}$ is computed by the following; $U(B) = 72$, $L(B) = 8$, $L(c) = 15$ and $U(c) = 205$. $Log\ \widehat{LR_{grid}} = Log\left(\left(\frac{205}{8}\right)^{205} \times \left(\frac{800-15}{800-72}\right)^{800-205} \times 1\right) = 665.93$.

Cubic Grid Circle Algorithm (CGC): Algorithm 2 shows the three phases of CGC. Next, these phases will be explained in detail.

Algorithm 2 Cubic Grid Circle Algorithm

Input:
1) An point set P with points $p(x, y)$, 2) A minimum circle radius r_{min} 3) A log likelihood ratio threshold (θ), 4) A p-$value$ threshold α_p and 5) A number of trials in Monte Carlo Simulation (m)

Output:
Circles C with $r \geq r_{min}$, $Log\ LR_C \geq \theta$ and $p\text{-}value \leq \alpha_p$

Algorithm:
1: **Filter Phase:** $filteredSet \leftarrow$ all $p \in MECC$ for a $cell_{circle}$ with $Log\ \widehat{LR_{grid}} \geq \theta$
2: **Refine Phase:** $candidateCircles \leftarrow C$ with $Log\ LR_C \geq \theta$.
3: **Monte Carlo Simulation Phase:** $resultSet \leftarrow candidateCircles$ with $p\text{-}value \leq \alpha_p$
4: return $resultSet$

Filter Phase: The pseudocode for the filter phase is given in algorithm 3. First, S is discretized into count grid and cubic circle grid using $l_{cell} = \frac{r_{min}}{2}$ (lines 1-2). The cell length l_{cell} is selected by the given input r_{min}, because in order to detect all the hotspots with $r \geq r_{min}$ at least one cell should fit completely inside the hotspot. Next, the number of expected points inside a single cell is determined by $B_{cell} \leftarrow |P|/(N \times N)$ (line 3). B_{cell} is used when computing the $U(B)$ and $L(B)$, since MECC and MFCC are constituted of cells. For every $cell_{circle}$, $Log\ \widehat{LR_{grid}}$ are computed (lines 5-7). To prevent the overlapping $filteredSet$s, once $Log\ \widehat{LR_{grid}}$s are computed, the $cell_{circle}$ with the highest $Log\ \widehat{LR_{grid}}$ is stored in $cell^{top}_{circle}$ if $Log\ \widehat{LR_{grid}}^{top} \geq \theta$ (line 8) and points, which are associated with $cell^{top}_{circle}$, are stored as a $filteredSet$ (line 9-10) and are removed from P. This process is repeated until none of $Log\ \widehat{LR_{cell}} \geq \theta$ or $P = \emptyset$ (lines 4-11). Finally, $filteredSet$s and $cell_{count}$s are sent to the refine phase.

Lemma 1: Filter phase of the CGC Algorithm can detect a circular hotspot C with $r \geq r_{min}$, if $l_{cell} \leq (r_{min}/2)$ and thus a count grid cell is completely inside C.

Proof: In the Equation 2, $L(B)$ and $L(c)$ are defined by the cells in MFCC. If at least one $cell_{count}$ is inside a hotspot, then $Log\ \widehat{LR_{grid}} > Log\ LR_C$. The proof lies in $U(B)$, $U(c)$, $L(B)$ and $L(c)$. Suppose a $cell_{count}$ is completely inside a

Algorithm 3 Filter Phase Algorithm of CGC

Input:
1) A log likelihood ratio threshold (θ)
2) $l_{cell} = \frac{r_{min}}{2}$ for the cell size of count grid and circle grid

Output:
All $filteredSet$s for each non-overlapping $cell_{circle}$ with $Log\ \widehat{LR_{grid}} \geq \theta$

Algorithm:
1: Create count grid with $N \times N$ cells using l_{cell}
2: Create cubic circle grid with $N \times N \times N$ cells
3: Compute the expectation for a single $cell_{count}$ by $B_{cell} \leftarrow |P|/(N \times N)$
4: **while** $P \neq \emptyset$ || $Log\ \widehat{LR_{grid}}^{top} \geq \theta$ **do**
5: **for each** $cell_{count}$ in count grid **do**
6: **for each** $r \leftarrow 1$ to $N/2$ **do**
7: $cell_{count} \leftarrow Log\ \widehat{LR_{grid}}$
8: $cell^{top}_{count} \leftarrow cell_{count}$ with the highest $Log\ \widehat{LR_{grid}}$
9: **if** $Log\ \widehat{LR_{grid}}^{top} \geq \theta$ **then**
10: $filteredSet_{1...k} \leftarrow$ all $p \in MECC$ of the $cell^{top}_{count}$
11: $P \leftarrow [P - filteredSet]$
12: return $cell^{1...k}_{count}$ and $filteredSet_{1...k}$

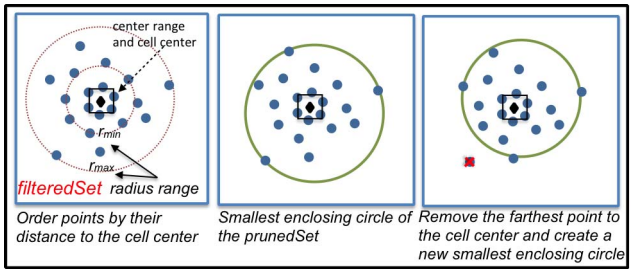


Fig. 7. Illustration of the refine phase in CGC algorithm

hotspot. Then $U(B) \geq B$, since the area of MECC will be larger than $area_C$. Also, $L(B) \leq B$, since MFCC will be inside the circle and thus its area will be smaller than that of $area_{circle}$. In the worst case, suppose $U(c) = c$ and $L(c) = c$, then $Log\ \widehat{LR_{grid}}$ will still be $\geq Log\ LR_C$ since $L(B) \leq B$ and $U(B) \geq B$. Therefore, if $l_{cell} \leq (r_{min}/2)$, then $Log\ \widehat{LR_{grid}} \geq Log\ LR_C$. ■

Refine Phase: The pseudocode of the refine phase in algorithm 4 starts by ordering the points in the $filteredSet$ by their distance to the center of the $cell_{count}$ (line 2). Next, the minimum enclosing circle (C) of the $filteredSet$ is determined [22](line 4). If the center of C is inside the center interval of $cell_{count}$, the $Log\ LR_C$ is computed using the count of the points inside (which is equal to the cardinality of $filteredSet$) and the area of C (line 5-7). If the $Log\ LR_C$ is greater than the one computed previously as $Log\ LR^{previous}_C$, it is saved (line 8-9). Next, the point farthest from the center of the $cell_{count}$ is removed from the $filteredSet$ (line 10) and the algorithm repeats the process of finding a new minimum enclosing circle for the new $[filteredSet - p_{farthest}]$ until $|filteredSet| = \emptyset$ or the radius of the generated C is $r \leq r_{min}$ (line 3-10). Finally, the C with the highest $Log\ LR_C$ is saved in $candidateCircles$ (line 11). This process is done for all $filteredSet$s returned by the filter phase (line 1-12).

An execution trace of the refine phase can be seen in Figure 7. Given the $filteredSet$ and $cell_{circle}$, the points are ordered by their distance and their smallest enclosing circle is determined (second box). Once the $Log\ LR_C$ of this circle (shown in green) is computed, the point farthest from the center of $cell_{count}$ is removed from the set (shown in red) and a new

smallest enclosing circle is created using the rest of the points. This process is repeated until the $filteredSet = \emptyset$.

Algorithm 4 Refine Algorithm of CGC

Input:
1) A log likelihood ratio threshold (θ)
2) $filteredSet_{1...k}$ and $cell^{1...k}_{count}$ returned by the filter phase

Output:
A circular hotspot C for each $filteredSet$ if $Log\ LR_C \geq \theta$

Algorithm:
1: **for** each $filteredSet_i \in filteredSet_{1...k}$ **do**
2: order $p \in filteredSet_i$ by distance to the center of $cell^i_{count}$
3: **while** $filteredSet_i \neq \emptyset$ or $r_i \geq r_{min}$ **do**
4: $C_i \leftarrow SEC(filteredSet_i, tmpSet)$ where $tmpSet = \emptyset$
5: **if** $center^i_C$ is inside the center interval of $cell^i_{count}$ and $r_i \geq r_{min}$ **then**
6: compute B and c
7: $Log\ LR^i_C \leftarrow$ Log Likelihood Ratio
8: **if** $Log\ LR^i_C \geq Log\ LR^{previous}_C$ **then**
9: $Log\ LR^{previous}_C = Log\ LR^i_C$ and $C_{previous} = C_i$
10: $filteredSet_i = filteredSet_i - p_{farthest}$
11: $candidateCircles \leftarrow C_{previous}$ and $Log\ LR_{previous}$
12: return $candidateCircles$

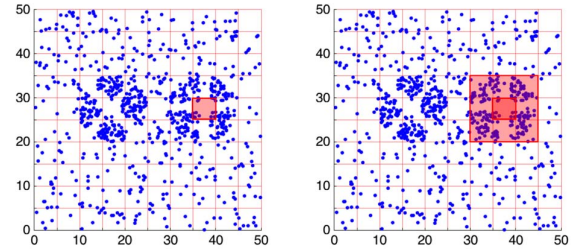
13: **procedure** SEC($filteredSet, tmpSet$) [22]
14: **if** $filteredSet == \emptyset$ || $|tmpSet| == 3$ **then**
15: $C \leftarrow computeCircle(tmpSet)$
16: **else**
17: select a random $p \in filteredSet$
18: $C \leftarrow SEC(filteredSet - p, tmpSet)$
19: **if** p is not in C **then**
20: $C \leftarrow SEC(filteredSet - p, tmpSet \cup p)$
21: **end procedure**

Monte Carlo Simulation Phase: During Monte Carlo simulation, the filter phase and refine phase of CGC algorithm is run for each individual random datasets created for the Monte Carlo simulation and the highest $Log\ LR_C$ are stored and the circles' p-values are determined using the ordered list of these highest $Log\ LR_C$.

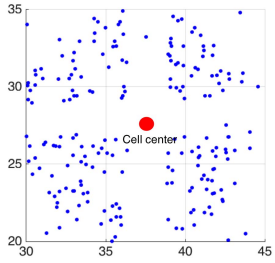
CGC Execution Trace: In Figure 8, the execution trace of CGC is illustrated using the point set in Figure 3(a) with $|P| = 800$ points in an area $S = 50 \times 50 = 2500$. The log likelihood ratio threshold $\theta = 100$ and the $p\text{-}value$ threshold $\alpha_p = 0.001$. In order to make the execution trace be easy to follow, the cell length is selected $l_{cell} = 5$ for this example.

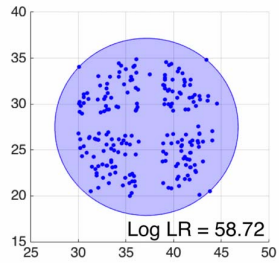
In the filter phase, the study area is partitioned into a count grid and a cubic circle grid. For example, the $cell_{count}$ $([36, 40], [36, 40]))$ has $c_{cell} = 8$ (shown in red in Figure 8(a). Suppose a $cell_{circle}$ is created for coordinates $([36, 40], [36, 40], [6, 10])$. The MECC for this cell is shown in Figure 8(b) and the MFCC will be the red cell in Figure 8(a). Using these, $U(B) = 72$, $L(B) = 8$, $L(c) = 15$ and $U(c) = 205$. and $Log\ \widehat{LR_{grid}} = Log\left(\left(\frac{205}{8}\right)^{205} \times \left(\frac{800-15}{800-72}\right)^{800-205} \times 1\right) = 665.93$. Once all the $\widehat{LR_{grid}}$ are computed, the $cell_{circle}$ with the highest $\widehat{LR_{grid}}$ and the associated points are sent to the refine phase and the filter algorithm repeats for the rest of the points in P.

In the refine phase, as shown in Figure 8(c), $filteredSet$ is ordered distance wise to the center of the $cell_{circle}$ which was returned by the filter phase. Next, the minimum enclosing circle of the $filteredSet$ is determined and the $Log\ LR_C$ is computed as shown in Figure 8(d). Then, the farthest point to

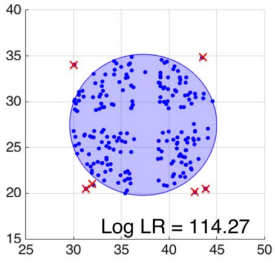


(a) Count grid cells are shown with red lines (b) A sample $cell_{circle}$ shown as red cells.





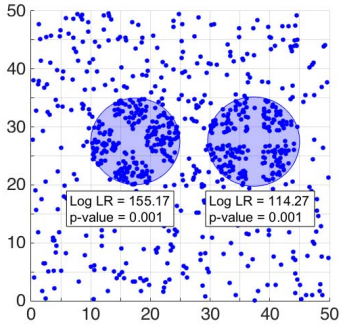
(c) Points in $filteredSet$ are ordered distance-wise to the center of $cell_{circle}$ (d) Minimum enclosing circle of the $filteredSet$



(e) Minimum enclosing circle for a new $filteredSet$

Center x	Center y	r	area(C)	B	c	Log LR	p-value
37.16	27.51	9.64	292.21	93.51	205	58.72	0.01
37.54	27.52	9.39	277.17	88.69	204	64.51	0.01
37.54	26.45	8.62	233.47	74.71	203	86.71	0.01
37.22	26.53	8.45	224.26	71.76	202	91.22	0.01
37.62	26.75	8.37	220.11	70.44	201	92.65	0.01
37.19	27.01	7.97	199.55	63.86	200	105.64	0.01
37.3	27.48	7.71	186.62	59.72	199	114.26	0.01
37.32	27.46	7.69	186	59.52	198	113.35	0.01
37.34	27.44	7.68	185.27	59.29	197	112.52	0.01
37.34	27.43	7.68	185.24	59.28	196	111.14	0.01
37.34	27.43	7.68	185.24	59.28	195	109.75	0.01
...	...	...	...	...	...	...	...

(f) Minimum enclosing circles and their $Log\ LR_C$ for the $filteredSet$



(g) Output of CGC

Fig. 8. Execution trace of the CGC algorithm (best in color).

the center of $cell_{circle}$ is removed from the $filteredSet$ and a new minimum enclosing circle and its $Log\ LR_C$ is determined. This process is repeated until $filteredSet$ is empty or the minimum enclosing circle has a radius $r \leq r_{min}$. Now a list of $Log\ LR_C$ and circles enumerated using the $filteredSet$ are acquired as shown in Figure 8(f). Finally, the circle C with the highest $Log\ LR_C$ is returned as a $candidateCircle$ for Monte

Carlo simulation. Monte Carlo simulation then determines the p-value of each candidate circle. This process is similar to the filter and refine phases except that random datasets are used as input. The final output of CGC can be seen in Figure 8(g).

IV. Theoretical Evaluation

A. Correctness of the CGC Algorithm

Lemma 2: CGC is functionally correct. Functional correctness means that CGC returns only the circular hotspots with $r \geq r_{min}$, $Log\ LR_C \geq \theta$ and $p\text{-}value \leq \alpha_p$.

Proof: Circles generated by CGC are evaluated against the $r \geq r_{min}$ in Algorithm 4 Line 5 and against the $Log\ LR_C \geq \theta$ in Algorithm 4 Line 8 and only if they satisfy the thresholds they are saved in $candidateCircles$. In the Algorithm 2 in Monte Carlo simulation phase each $candidateCircle$ is evaluated against the p-value threshold in Line 3 and only the ones which satisfy α_p are returned as the result set. Since CGC returns the circles with $r \geq r_{min}$, $Log\ LR_C \geq \theta$ and $p\text{-}value \leq \alpha_p$, it is functionally correct. ∎

Lemma 3: Given a finite point set P, CGC terminates in finite time.

Proof: In algorithm 3, the iteration in Line 4 is bounded by the $Log\ \widehat{LR_{grid}}$ and $|P|$. Since in every iteration the associated points with cells which exceed the θ threshold are removed from P in Line 11, the $\lim_{iterations \to \infty} P \to \emptyset$. In the case where none of the cells exceed θ threshold loop stops again in Line 4. Thus algorithm 3 will terminate in a finite time. In the refine phase, on every iteration of the loop in Line 3, the farthest points are removed from the $filteredSet$ in Line 10. Thus the loop will terminate when the $filteredSet = \emptyset$. Since filter and refine phase terminate in finite time, CGC terminates in finite time. ∎

Theorem 1: CGC is a correct approach to detect statistically significant circular hotspots. An algorithm is correct if it is functionally correct and it terminates in finite time.

Proof: Using Lemma 2, CGC is a functionally correct approach. Also using Lemma 3, CGC will terminate in finite time. Thus, CGC is a correct approach to detect statistically significant circular hotspots. ∎

B. Computational Analysis of the CGC Algorithm

The complexity of SaTScan is $O(m \times |P|^3)$ ($|P|^2$ to enumerate circles, $|P|$ to count points for each circle), where $|P|$ is the cardinality of the point set and m is the number of Monte Carlo simulation trials.

In the filter phase of CGC, circles are enumerated using the count grid cells. Since all count grid cells (N^2) are traversed for all possible radii, the total cost of the filter phase is $O(N^3)$ which is equal to the size of the cubic circle grid. Note that, grids are created using l_{cell} which is determined by the minimum radius r_{min} defined in the problem statement. Therefore CGC filter algorithm execution time depends on the careful selection of these inputs.

In the refine phase of CGC, circles are enumerated for each $filteredSet$ using a smallest enclosing circle algorithm of a linear cost $O(|filteredSet|$, then points are removed one by one from the $filteredSet$ and new circles are enumerated until $filteredSet = \emptyset$ or $r \leq r_{min}$. Therefore the total cost of removal and recompute is $O(|filteredSet|log\ |filteredSet|)$. In the worst case, $|filteredSet| = |P|$ and the refine phase cost is $|P|log\ |P|$ and in the best case $filteredSet = \emptyset$ and the refine phase cost is $O(1)$.

Thus, the worst case cost of CGC is $O(m \times (N^3 + |P|log\ |P|))$, if no filtering occurs and all points are returned to the refine phase. In the best case, the cost will be $\Omega(m \times N^3)$ as the filter phase will not return any $filteredSet$ and the algorithm will terminate. Note that since the datasets in the Monte Carlo simulation are created randomly, depending on the θ, these do not survive filter phase and the algorithm performance gets closer to the best case scenario.

V. Case Study

The proposed approach is evaluated qualitatively by comparing the CGC algorithm output with SaTScan using continuous Poisson process [11] on a crime dataset shown in Figure 9. The input point set in Figure 9(a) includes 64 unarmed robbery cases in San Diego between March 2013 - 2014 [23]. The inputs are selected as $r_{min} = 0.009$ degrees, $\theta = 5$ and $\alpha_p = 0.01$. Maps were prepared using QGIS' OLP [24].

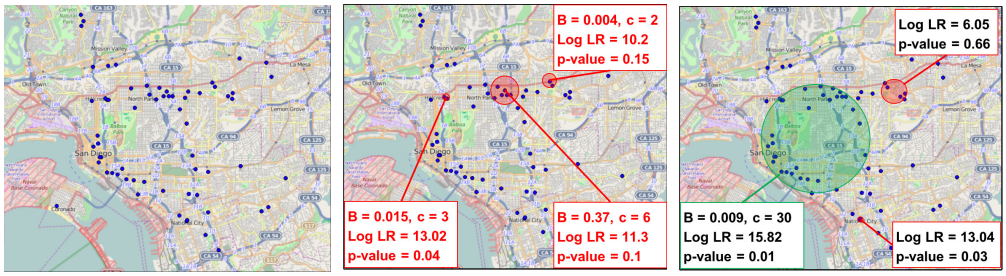
The two algorithms generated quite different results. It appears that SaTScan's reliance on point centered circles caused it to miss a significant hotspot and its output did not satisfy the thresholds (Figure 9(b)). Also the output of the SaTScan includes small hotspots consists of two/three points which couldn't be filtered out. CGC algorithm handles this issue by using r_{min} threshold.

GRHD with the CGC algorithm discovered a hotspot (Figure 9(c)- shown in green) which satisfies the input thresholds. Although, domain experts may interpret these better, it can be stated that these crimes occurred along a road and this road surrounds a residential area where the crimes are sparse. This type of crime pattern can be seen in environmental criminology [6]. It should be noted that these crimes' sources are anonymized (for privacy issues), meaning that we can't compare the results with ground truth labels.

VI. Experimental Evaluation

The goal of the experiments was twofold: to evaluate the performance of the CGC algorithm under different parameters and to compare its performance with SaTScan. To achieve these goals, the following questions are asked: (1) How is the scalability and the result quality of the proposed algorithm compared to its rivals (e.g. SaTScan)? (2) How effective is the filter step in reducing the cost of CGC algorithm?

Experimental Design: Experiments are performed on synthetic datasets which is created with varying number of points (default 5000) in a 1000×1000 study area. In these datasets 20% of the points were generated to form a hotspot and the rest of the points were created using complete spatial randomness (CSR). Inputs were; log likelihood ratio threshold $\theta = 1000$, p-value threshold $\alpha_p = 0.01$ (99% confidence level) and minimum circle radius $r_{min} = 100$. Performance of both algorithms was measured in terms of CPU time. All experiments were performed on a MacBook Pro with a Intel



(a) Input crime dataset (b) Output of SaTScan (c) Sample Output of GRHD

Fig. 9. Figure 9(a) shows 64 unarmed robbery crimes committed in San Diego in 2013 (blue dots) [23]. Figure 9(b) and Figure 9(c) compare the output of SaTScan and GRHD with the CGC algorithm(red/green circles - best in color).

Core i7 2.2 GHz CPU and 4GB memory. To compare CGC and SaTScan algorithms, both algorithms were implemented on Java platform and were executed 10 times for each experiment. When implementing SaTScan, continuous Poisson process is selected as defined in SaTScan user guide [11].

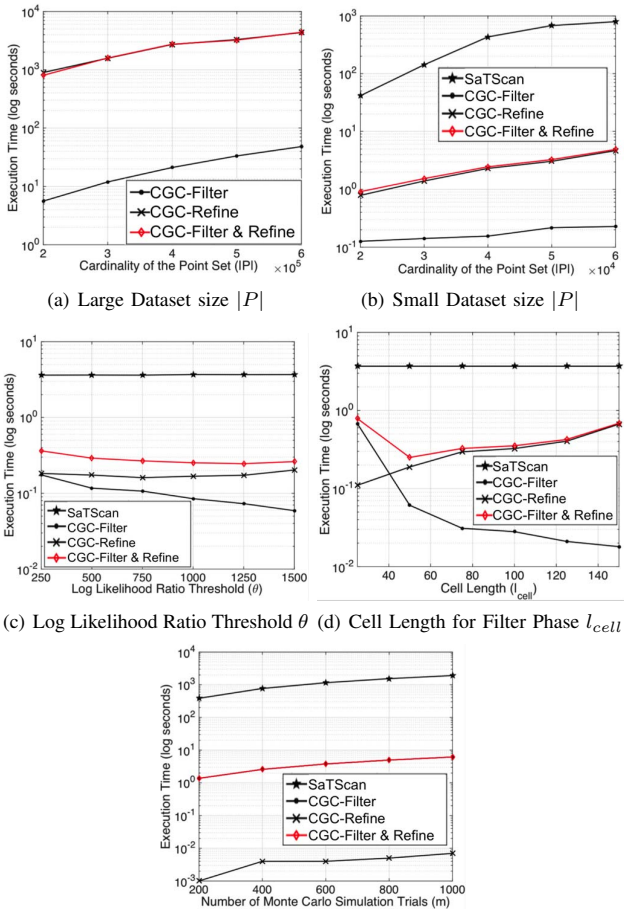
A. Experimental Results

Effect of the Number of Points in P: In this experiment, two sets of synthetic points sets were created. The first set of point sets included points with cardinality ranging from 20K to 60K which used to compare SaTScan with CGC. With the second set of point sets, the scalability of CGC is tested with its Filter and Refine phases and does not include SaTScan. For the point sets ranging from 20K, 30K, 40K, 50K, 60K, θ is selected 10^4. For the point sets ranging from 200K, 300K, 400K, 500K, 600K, θ is selected 10^6. Figure 10(a) shows that there is at least two orders of magnitude difference between SaTScan and CGC execution times. Also in Figure 10(a), it can be seen that CGC filter phase performs faster and most of the execution time is spent on the refine phase. Overall CGC algorithm performs faster than SaTScan and the savings increase when the point set size increases.

Effect of the Log Likelihood Ratio Threshold (θ): In this experiment, the log likelihood ratio threshold (θ) is varied by $250, 500, 750, 1000, 1250$ and 1500. In Figure 10(c), there is no change on the execution times of SaTScan, since SaTScan does not use any filter depending on θ. However CGC filter phase benefits from the θ and savings increase with θ .

Effect of the Cell Length (l_{cell}): In order to observe the effect of the cell length l_{cell}, cell length is changed by $25, 50, 75, 100, 125, 150$. In Figure 10(d), since SaTScan does not use any filtering method based on grid generation, cell length did not affect SaTScan. For CGC algorithm, when the cell length is small, filter phase takes more time and refine phase takes less time since the number of points in the $filteredSet$ is close to the actual points in the hotspot. On the other hand, when the cell length is large, filter phase takes less time but this time refine phase takes more time since the $filteredSet$ includes more points than the actual hotspot. It can be concluded that a careful selection of cell length is important for the performance of the CGC algorithm.

Effect of the Number of Monte Carlo simulation trials (m): In this experiment, $m = 200, 400, 600, 800, 1000$ random synthetic datasets with 5000 points in a 1000×1000 study area are generated. Other inputs were kept the same per experimental design. Figure 10(e) shows the execution



(a) Large Dataset size $|P|$ (b) Small Dataset size $|P|$

(c) Log Likelihood Ratio Threshold θ (d) Cell Length for Filter Phase l_{cell}

(e) Number of Monte Carlo simulation trials m

Fig. 10. Scalability of CGC with increasing (a), (b) number of points, (c) log likelihood ratio threshold, (d) cell length and (e) number of Monte Carlo simulation trials.

times for the Monte Carlo simulation trials. As can be seen CGC performed at least two orders of magnitude faster than SaTScan and in many trials CGC algorithm did not need to run the refine phase since most of the random datasets did not have any circular hotspot which exceed the specified $\theta = 100$ and therefore filter phase did not generate any $filteredSet$. Therefore it can be concluded that the grid circle upper bound log likelihood ratio filtering in the proposed CGC algorithm

improves its scalability. In summary, the experiments show that CGC is more scalable than SaTScan for large point sets.

VII. Discussion

It is worth mentioning that there are other techniques for the detection of statistically significant hotspots which use predefined grids (e.g. overlap-multires partitioning [25], predefined locations -counties/zip codes and aggregated number of points- [17] approximation approaches [26]). These techniques are fast when the cardinality of the input point set is higher than the number of grids. However, they are designed for rectangular/square hotspot detection. Moreover, these approaches use grids as the shape of the hotspot and their output is a "single" hotspot with the highest test statistic in the study area. Therefore, these techniques were not considered in this paper. However, it should be noted that CGC filter approach is not limited to circles and any shape that can be defined by parameters can be evaluated with this approach. For example, proposed CGC filter phase uses a cubic circle grid to enumerate circles using three parameters that define a circle, namely center coordinates and radius. Similarly, CGC approach may be generalized to be used with other shape-specific (rectangle, square, circle, ellipse, etc.) state-of-the-art methods to filter the candidate hotspots that do not contribute to an actual hotspot.

VIII. Conclusions and Future Work

This work explored the Geographically Robust Hotspot Detection (GRHD) problem, which is important for societal applications e.g. epidemiology, environmental criminology, etc. GRHD is challenging due to the difficulty of enumerating all possible hotspots and the high computational cost of the statistical significance test. This paper proposed a CGC algorithm which discovers statistically significant hotspots which couldn't be discovered previously by SaTScan. A case study demonstrated CGC's superior performance over SaTScan on a real crime dataset. Experiments also showed that the proposed algorithm (CGC) is highly scalable.

In the future, we envision to improve geographical robustness by adding enhancements to detect imperfect hotspots (i.e., half/quarter circles) where the points occur along the coast lines, jurisdiction boundaries, roads etc. which may cause these hotspots to be missed. In addition, we plan to add the effect of unhomogeneous point distributions caused by the characteristics of the geographic location (population, etc.). Also we plan compare GRHD with other state-of-the-art methods [16], [17], [26]–[28] that are designed to detect rectangular, elliptical and irregular shaped hotspots. Finally, we plan to add expectation based Poisson statistics to improve the result quality and scalability of our proposed approach.

IX. Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 1029711, IIS-1320580, 0940818 and IIS-1218168, USDOD under Grant No. HM1582-08-1-0017, HM0210-13-1-0005, and University of Minnesota via U-Spatial. We would like to thank Kim Koffolt and the members of the University of Minnesota Spatial Computing Research Group for their comments.

References

[1] J. Snow, *On the mode of communication of cholera*. John Churchill, 1855.

[2] G. F. Pyle, *Applied medical geography*. VH Winstond and Sons, 1979.

[3] R. M. Merrill, *Introduction to epidemiology*. Jones & Bartlett Publishers, 2013.

[4] J. F. Mosha *et al.*, "Epidemiology of subpatent plasmodium falciparum infection: implications for detection of hotspots with imperfect diagnostics," *Malar J*, vol. 12, no. 221, pp. 10–1186, 2013.

[5] A. Sifferlin, "The 5 Biggest Mistakes in the Ebola Outbreak," *Time Magazine, http://time.com/3426642/the-5-biggest-mistakes-in-the-ebola-outbreak-so-far/*, Sept 25 2014.

[6] P. J. Brantingham and P. L. Brantingham, *Environmental criminology*. Sage Publications Beverly Hills, CA, 1981.

[7] L. W. Kennedy and D. R. Forde, "Routine activities and crime: An analysis of victimization in canada*," *Criminology*, vol. 28, no. 1, pp. 137–152, 1990.

[8] J. Eck *et al.*, "Mapping crime: Understanding hotspots," 2005.

[9] L. N. Boggess, R. T. Greenbaum, and G. E. Tita, "Does crime drive housing sales? evidence from los angeles," *Journal of Crime and Justice*, vol. 36, no. 3, pp. 299–318, 2013.

[10] M. Ester *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." AAAI Press, 1996, pp. 226–231.

[11] M. Kulldorff, "Satscan user guide for version 9.0," 2011.

[12] D.-W. Choi *et al.*, "A scalable algorithm for maximizing range sum in spatial databases," *Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1088–1099, 2012.

[13] R. Agrawal *et al.*, *Automatic subspace clustering of high dimensional data for data mining applications*. ACM, 1998, vol. 27, no. 2.

[14] S. Shekhar *et al.*, "Identifying patterns in spatial information: A survey of methods," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, p. 193, 2011.

[15] M. Kulldorff, "A spatial scan statistic," *Communications in Statistics-Theory and methods*, vol. 26, pp. 1481–1496, 1997.

[16] D. B. Neill, "Expectation-based scan statistics for monitoring spatial time series data," *International Journal of Forecasting*, vol. 25, no. 3, pp. 498–517, 2009.

[17] ——, "Fast subset scan for spatial pattern detection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 74, no. 2, pp. 337–360, 2012.

[18] "Quantum gis software web site," http://www.qgis.org/en/site/, accessed: 2013-12-12.

[19] M. Kulldorff, *Spatial scan statistics: models, calculations, and applications*. Springer, 1999.

[20] K. Binder, *Introduction: Theory and technical aspects of Monte Carlo simulations*. Springer, 1986.

[21] E. Eftelioglu *et al.*, "Ring-shaped hotspot detection: A summary of results," in *2014 IEEE International Conference on Data Mining*, 2014, pp. 815–820.

[22] E. Welzl, *Smallest enclosing disks (balls and ellipsoids)*. Springer, 1991.

[23] S. D. City, "Robbery (w/o weapon) cases in san diego in 2013," http://www.sandiego.gov/police/services/ewatch.shtml, Accessed: 2015-03.

[24] OpenStreetMap, http://www.openstreetmap.org/,Retrieved Jan. 2015.

[25] D. B. Neill and A. W. Moore, "Rapid detection of significant spatial clusters," in *Proceedings of the 10th ACM SIGKDD*. ACM, 2004, pp. 256–265.

[26] D. Agarwal *et al.*, "Spatial scan statistics: approximations and performance study," in *Proceedings of the 12th ACM SIGKDD*. ACM, 2006, pp. 24–33.

[27] D. Neill, "An empirical comparison of spatial scan statistics for outbreak detection," *International Journal of Health Geographics*, vol. 8, no. 1, p. 20, 2009.

[28] D. B. Neill, "Detection of spatial and spatio-temporal clusters," Ph.D. dissertation, University of South Carolina, 2006.