Dynamic Spatial Task Generation for Collaborative Location-based Collecting Systems Coverage Objectives

María Dalponte Ayastuy[1,2][0000−0002−1412−5694], Diego Torres[1,2][0000−0001−7533−0133], and Bruno J. Lattanzio [1 [0000−0001−5380−0915]

[1] Depto CyT, Universidad Nacional de Quilmes Roque Saenz Peña 352, Bernal, Buenos Aires, Argentina.{mdalponte,brunoj.lattanzio}@unq.edu.ar
[2] LIFIA,CICPBA-Facultad de Informatica, Universidad Nacional de La Plata 50 y 120, La Plata, Buenos Aires, Argentina.
diego.torres@lifia.info.unlp.edu.ar

Abstract. Collaborative location-based collecting systems (CLCS) are a particular case of collaborative systems where a community of users collaboratively collect geo-referenced data. Each CLCS sets its territory coverage objectives, commonly defined as to guarantee that all the affected territory is surveyed with a particular coverage criterium. This paper presents a three-step pipeline to recommend the subareas that require observations dynamically. The first step generates a disjoint and adjacent set of areas -a mesh- covering the sampling territory. The second step sets a priority and coverage objective for each area. Finally, the third step considers the project's objectives and the area coverage situation to recommend the areas that need surveys. The output of this last step is an input for a user-task distribution process where the user's profile is taken into account. Moreover, an example of meshing strategy and task generation is proposed.

Keywords: Collaborative Location-based Collecting Systems · Meshing · Decision-making · Spatial Crowdsourcing

1 Introduction

Collaborative Location-based Collecting Systems (CLCS) are collaborative systems where the community of users collects data associated with their location normally by using a mobile application [3]. CLCS can be weighed as a supporting technology of some citizen science projects, such as the AppEar project[1], GeoVin [2] or iNaturalist [14]. Citizen science projects encourage and support the contributions of volunteers to the advancement of scientific research. Some location-based activities need to survey scientific data associated with a location and a timestamp, consolidating these contributions as tuples with the structure

$$< lat - long, timestamp, sampleData >$$

As an example, a *sampleData* value for AppEar project includes ecological information about rivers, lakes and estuaries; GeoVin *sampleData* describes the 'barber bug' insect (*Triatoma infestans*) presence; iNaturalist *sampleData* is about biodiversity observations around the globe.

This paper is partially supported by funding provided by the STIC AmSud program, Project 22STIC-01.

Received April 2023 Accepted July 2023; Published August 2023
https://doi.org/10.24215/15146774e018

Each CLCS sets its territory coverage objectives, commonly defined as to guarantee that all the affected territory is surveyed with particular coverage criteria [3]. For instance, the coverage criteria can be a sample every 100 meters. Setting coverage criteria implies dealing with the problem of organizing the division of observation tasks so that the project's objectives are met. Specifically, it requires (i) the segmentation of the territory into smaller areas and (ii) assigning coverage priorities to each area considering the project objectives. And so, if necessary, repeat (i) and (ii).

Even though the spatial modeling and segmentation problem is widely approached in geographical information and location-aware systems [9, 12, 8, 18, 6, 5], and the growing number of studies in spatial tasks assignment in a wide range of fields [10, 11, 15–17, 7] , the subject of relating tasks to spatial segments considering project's objectives is largely absent.

This article presents an approach for decision-making assistance to dynamically recommend the tasks for those subareas that require observations based on the coverage objectives and a geographical area of scope of the CLCS. Such a tool is much needed in CLCS as well in the industry. As the survey tasks are completed, the recommendation is dynamically executed to find those areas that have not been completed. Also, it is possible to trigger a new segmentation if a particular condition on the system's global status is achieved.

The recommendation is made through a three steps pipeline where the first step generates a disjoint and adjacent set of areas -a mesh- that covers the sampling territory. The second step sets a priority and objectives for each area, and it can be done manually or automatically through a rule set. Finally, the third step considers the project's objectives and the area coverage situation to recommend the areas that need to be surveyed. The output of this last step is an input for a user-task distribution process where the user's profile is taken into account. It is essential to notice that this approach generates a set of tasks considering, on the one hand, what is needed to achieve the project objectives and, on the other hand, the status of the project. However, nothing is addressed concerning the assignment of these tasks to the people who have to solve them. This assignment proposes a challenge in the adaptability research area to analyze the tailoring of tasks based on each user's preferences, characteristics, and behavior. Although this is an external process, the result of the task assignment in terms of how many were completed is essential feedback to update the CLCS's global state and be able to repeat the generation of tasks. In addition, this work presents a possible mesh computation strategy suitable for a cold start and a task generation strategy.

This article is organized as follows. In Section 2 the related work is presented. In Section 3 the pipeline steps are detailed. Finally, Section 4 shares discussions and future work.

Received April 2023 Accepted July 2023; Published August 2023

2 Related work

Spatial crowdsourcing is the process of crowdsourcing a set of spatial tasks (i.e., tasks related to a location) to a set of users, which requires the users to perform the spatial tasks by physically traveling to those locations[**?**]. Location-based task assignment needs to assess the available sensing resources to meet the objectives of the CLCS. The criteria for optimization of task assignment include sensing costs, coverage of areas of interest, quality, and redundancy of sampled data. The approach in [15] proposed a coverage-based task assessment that finds the least costly subset of participants to achieve the coverage goal. The work in [16] also proposed a coverage-based task assignment method for assigning viewpoints to a group of moving participants. Similarly, [17] focuses on one class of spatial crowdsourcing, in which the users send their locations to the server. After that, the server assigns the tasks in proximity to the user's location to every user, intending to maximize the overall number of assigned tasks. Notice that the tasks are defined by a geolocated point in all these approaches but not associated with a sampling geolocated area.

Regarding the segmentation into sampling areas, the discipline of geographic information systems (GIS) defines and uses different tessellated models to represent information about the earth[8]. Sometimes these models have the objective to build a hierarchical discretization of a high definition raster image or to support a vector representation of terrain surface, such as the Voronoi regions [9] or the Delaunay triangulation [18]. Articles that address the meshing problem by calculating the Voronoi regions from a set of centroids were found, as is the case of Fleischman et al. [6] and Du and Gunzberg [5]. These approaches can be helpful when, as was mentioned in Section 1, a new segmentation of the territory is needed considering the accumulated sampling activity.

3 Approach

CLCS spatial decision-making needs to optimize users' work by combining the defined objectives for the CLCS with the completed tasks in the territory. As was mentioned, one possible way to present the objectives is in terms of the number of samples per subarea and setting criteria for building a set of areas. Furthermore, these objectives can be complemented by prioritizing specific subareas.

This article proposes a pipeline to support a located-based task generation considering the project's objectives and the system's global status, as it is shown in Figure 1. The first step aims at generating a Mesh, given a list of geolocated geometries and some initial mesh configurations.

As a second step, the coverage and priority for each area must be defined, and it can be done manually or automatically through the application of a ruleset. These two configurations are essential at different times of the process: coverage setting is needed for the task generation step, and the area priority definition is a requirement for the location-based tasks recommendation system (see the gray box in Figure 1).

Received April 2023 Accepted July 2023; Published August 2023

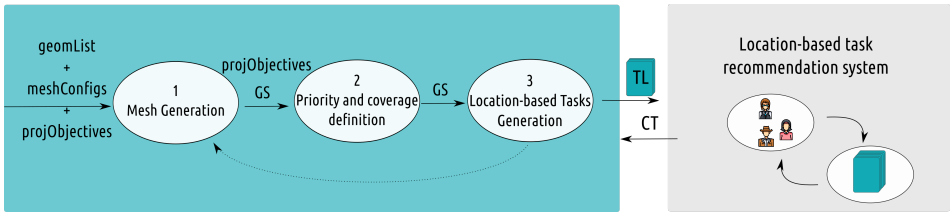


Fig. 1. Pipeline with feedback approach

The third step feeds on the project objectives and the system's global status to generate a list of spatial tasks to be distributed to users. After that, the recommendation of tasks (and their subsequent completion) offers an updated system's global status as feedback for this task generation (see CT -Completed Tasks- parameter in Figure 1).

Moreover, the mesh can be thought of as a dynamically adapted mesh that is recalculated based on the newly completed location-based tasks that represent the feedback to step 3 and can trigger feedback to step 1.

The following subsections introduce a few preliminary definitions and then give details about the steps of the pipeline, with an example strategy in each case.

3.1 Preliminary Definitions

Two methods are frequently used to represent geographic phenomena in ways that can be encoded in spatial databases, called raster and vector methods. Both can be used to encode continuous fields and discrete objects, but there is a strong association between raster and continuous fields and between vector and discrete objects in practice. One of the most common forms of raster data comes from remote-sensing satellites, which capture information as high-definition images. In a vector representation, all lines are represented as points connected by a straight line, and areas are captured as a series of points or vertices connected by straight lines, called polygons. Lines are represented in the same way, and the term polyline (or multiline) has been coined to describe a curved line represented by a series of straight segments connecting vertices [12].

Definition 1 (Point). *It is a tuple (latitude, longitude) of geographic coordinates. Negative values in latitude refer to points in the southern hemisphere, and negative values in longitude refer to points located west of the Greenwich meridian.*

For example, a point is pair with latitude: -24,5073190, and longitude: -68,3958578.

Definition 2 (Multiline). *A multiline object is a sequence of more than two points.*

Received April 2023 Accepted July 2023; Published August 2023

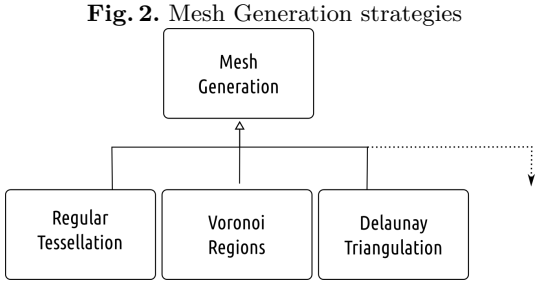
Definition 3 (Polygon). *A polygon is a particular case of a multiline element, where the first point is also the last one, making a closed geometry.*

3.2 Mesh Generation

Table 1. Mesh generation interface

Input Parameters			Output
`geomList`	Set of geometries representing the target territory	`gs`	initial system global status $GS = \{< a, rt, ct, w >\}$ where w is undefined
`meshConfig`	List of key/value tuples of configurations for mesh generation		
`projObjectives`	List of key/value tuples of project objectives definition		

Fig. 2. Mesh Generation strategies



As presented in the introduction, CLCS needs to relate each obtained sample with an area of the mesh. Depending on each project's particular characteristics, the area mesh must meet certain conditions on shape and granularity. However, in all cases, the mesh is a set of adjacent and disjoint cells organized following the shape of a given geometry.

For this reason, in this first step, it is necessary to delimit the territory and be able to configure the desired characteristics of the mesh. Table 1 details the input and output value types for the Mesh Generation step. The geolocated territory is defined by the `geomList` input parameter, while the characteristics are defined by the `meshConfig` input parameter. The `geomList` parameter can contain a single element that represents the total area of scope of the project or a set of geometries that spatially defines the territory. However, the details of this parameter depend on the particular meshing strategy.

The output of the Mesh Generation step is the initial system global status as is defined below:

Received April 2023 Accepted July 2023; Published August 2023

Definition 4 (System global status). *The system global status is a list of tuples:*

$$GS = \{< a, rt, ct, w >\}$$

where $\boldsymbol{a}$ *is a sampling area,* $\boldsymbol{rt}$ *is an integer number representing the required number of sampling tasks,* $\boldsymbol{ct}$ *is an integer number representing number of completed sampling tasks in area* $\boldsymbol{a}$*, and* $\boldsymbol{w}$ *is the assigned priority weight to area* $\boldsymbol{a}$ *.*

As is shown in Figure 2, in this work, several strategies are proposed, and they can be further extended. The Regular Mesh Generator (detailed in the following sub-section) is based only on a geometry (or a list of geometries) that defines the territory and the map of meshing parameters. This strategy is useful to tackle the cold-start meshing. Other strategies approach the mesh generation based on the territory spatial definition and the existence of other intermediate points located in the territory. Examples of these are the Voronoi regions [9], or Delaunay triangulations [18], among others.

Moreover, the granularity level of meshes can be thought of as dynamic and can be supported as an automatic process triggered in step 3. As an example, this can be used to divide areas that fulfill certain conditions, updating the system's status in the areas involved. Similarly, the areas can be merged to achieve larger areas.

Regular Mesh Generation (cold-start) Some scenarios require the sampling activity to be done in the proximity of a geographic element (a Point, multiline or polygon) without other points as a reference. This situation is associated with a cold-start mesh, and the `geomList` parameter is populated with these elements geographic, aiming at building a mesh following the shape of these geometries. For example, consider AppEar project, which needs to survey the ecological situation at the shores of rivers, lakes, and estuaries. In this case, `geomList` is made up of spatial geometries that represent the rivers and lakes.

This particular meshing strategy where the generated areas are organized in a regular shape grid following the shape of a given geometry is called *Regular Mesh*. In other words, the upper limit of the mosaic is the geometry's limit, the lines that separate the cells' rows are parallel to that limit, and the number of rows and the dimension of its cells are set through the configuration map.

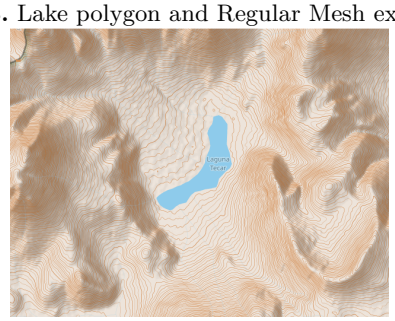
To detail the example scenario, suppose that sampling tasks need to be completed on the shore of a lake, that a ring parallel delimits the shore to the shoreline, spaced 100 meters apart, and that at least one sample is needed every 50 meters. This requirement would need an initial mesh with one row (or ring) of 50 meters width cells, like the one depicted in Figure 3 (b), and the `meshConfig` parameter is:

$$meshConfig = \{< cellWidth, 50 >, < nrows, 1 >, < gridHeigth, 50 >\}$$

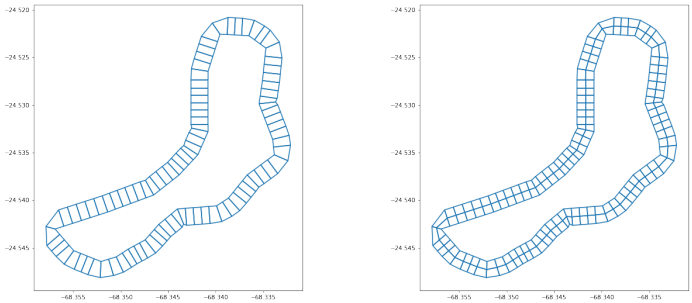
Another situation could need also a second sample of 70 meters from the shore, and in this case a different mesh would be needed, with two rings or rows

Received April 2023 Accepted July 2023; Published August 2023

Fig. 3. Lake polygon and Regular Mesh examples

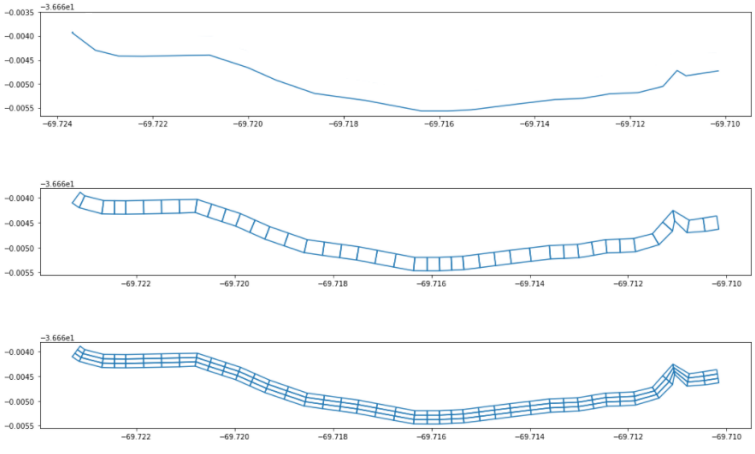


(a) Lake polygon



(b) Adaptive Mesh with one row (c) Adaptive Mesh with two rows

Fig. 4. Adaptive mesh for a river



(a) Original river geometry

(b) Mesh with one row

(c) Mesh with three rows

Received April 2023 Accepted July 2023; Published August 2023

around the lake. This is graphically explained in Figure 3 (c). In this case the `meshConfig` parameter is:

$$meshConfig = \{< cellWidth, 50 >, < nrows, 2 >, < gridHeigth, 100 >\}$$

Similarly, the different sampling requirements can be applied to the geometry of a river, as is depicted in Figure 4.

The regular mesh generation algorithm is presented in Listing 1.1 and builds a mesh for a given spatial geometry, a cell width, a total grid height, and a row number. Notice that this geometry can be a multiline (definition 2) or a polygon (definition 3), but in a general way, geometries are made up of sequences of segments, where each segment is a pair of connected points.

To generate the polygon's grid, this algorithm builds a grid for each segment, and after that, it connects the grids of consecutive segments. In each segment union, one of two possible situations is faced: either the segments form a convex angle (that is, there is an uncovered space between the grids to be connected), or they form a concave angle (that is, these grids overlap). Therefore, an additional process must be carried out that performs extrapolation or interpolation, respectively.

```
generate_regular_mesh(geom, cellWidth, gridHeight, nrows):
  polygons = []
  For i in geom.index:
    segment = LineString(geom[i:i+1])
    pSet = gridify_segment(segment, cellWidth, gridHeigth, nrows
    )
    polygons = fix_append(polygons,pSet, nrows)
  if ((geom.type == 'MultiPolygon') | (geom.type == 'Polygon')):
    polygons = fix_last_and_first_point(polygons, rows)
  res = gpd.GeoDataFrame({"geometry" : polygons})
  res['cid'] = res.index
  return res
```

Listing 1.1. GridConnector algorithm

Although this mesh generation is handy when the mesh must follow the shape of a geolocated geometry, it can be applied as a cold start strategy in other domains.

3.3 Area Priority and Coverage definition

A common trait in CLCS is to present the project's objectives in terms of coverage in a set of areas, as an integer value associated to each one. In this work, the coverage need is modeled with the `rt` value in the system global status (defined in 4). Furthermore, special areas can also be highlighted using polygons or points of interest (POIs), where another criterion is applied for setting the required number of tasks `rt`.

Received April 2023 Accepted July 2023; Published August 2023

Table 2. Area priority definition interface

Input Parameters		Output	
`GS`	System global status detailed by area	`GS`	System global status
`projObjectives`	List of key/value tuples of project objectives definition		

As was introduced, this step can be done manually or automatically employing a rule-based system and needs to be carried out after the subareas are defined (step 1 is executed). In any case, this is useful for generating tasks if the strategy applied for this considers their priorities. The first time this step is executed, the input system's global state has no defined weights, but after regeneration of the mesh, it may be necessary to update some of those weights.

Table 3. Example project objectives definition

key	value
Default samples per area	20
Default area weight	5
Special area weight	7
Points of interest (POIs)	$\{< -24,5073190, -68,3958578 >\}$

Considering a rule-based system, a possible `projObjectives` map is described in Table 3. This map determines that all areas must have a required sampling task of 20 samples ($rt = 20$), and the same weight value ($w = 5$), except for those areas that contain any of the points of interest defined in the `POIs` property, in which case $w = 7$.

3.4 Task Generation

In this step, the set of sampling tasks that are necessary to achieve the project objective is built. Specifically, is a set of tuples

$$TL = \{< a, n >\}$$

where **a** is a sampling area, **n** is an integer number representing the pending sample work.

Different strategies can be applied, considering that task completion is not directly related to the task list as users do not always complete the assigned task. For example, this step could estimate the number of redundant tasks based on the feedback through which completed tasks are reported.

In the different approaches to solving step 2, the first time this step is executed, it is based only on the system's global state. However, in the following

Received April 2023 Accepted July 2023; Published August 2023

Table 4. Task generation interface

Input Parameters		Output	
GS	System global status	TL	Sampling task list

iterations, feedback from step 3 is considered: how the community of users completed the required tasks. This feedback is modeled in a list of tuples

$$CT = < a, m >$$

where **a** is the surveyed area and **m** the completed task count that informs the finished sampling tasks in that area.

In the next section, a particular strategy for task generation is presented.

Offset task generation strategy The offset task generation approach is a particular task generation strategy that focuses on the pipeline feedback from an external task recommendation system. The value **m** is not necessarily less than or equal to the value **n** that was generated for that area in the previous cycle as an output of this step. Since the users community might perform fewer sampling tasks than those requested for that area, more tasks may be carried out since there are geographical locations that can attract more visitors.

The difference (offset) between m and n is calculated to measure the response to the lastly generated task list. If δ=**m-n** is positive, it is an over-sampled area, but if δ=**m-n** is negative, then an under-sampled area is present. In the first case, this area is already covered and is separated from the output. In the second case, redundancy is applied to the calculation of the offset, which is a scale factor (**f**) to take into account that the area in question is rarely visited. In addition, it is helpful to establish a threshold (**t**) that allows establishing which areas need more visibility.

Therefore, the task list update is done with the algorithm described in Listing 1.2.

```
updateArea(tl,areaId,n,m):
offset = m-n
if(offset>=0):
  tl.remove(areaId)
else:
  abs = absolute(offset)
  if (abs<t):
    tl.update(areaId,abs)
  else
    tl.update(areaId,abs*factor)
```

Listing 1.2. Task list update

Suppose the situation where for a certain area a_1 15 samples were requested but only 12 were solved, then δ is -3. On the other hand, if another area a_2

Received April 2023 Accepted July 2023; Published August 2023

had 15 requested samples but only 3 were solved, then $\delta = -12$. Also, if $t = 10$ and $f = 2$, meaning that when treshold is exceeded, double sample quantity is requested. Therefore in the first example the new value for area a_1 is $n' = 3$, and for a_2 is $n' = 12 \times 2 = 24$

Like the mesh generation step, several strategies could be defined for this step, and the pipeline could be configured according to the domain and objectives of each project.

3.5 User Location-based Task Recommendation System

An external system could consume the sampling task list generated in the former step to assign the tasks to particular users.

There are several approaches to distributing tasks taking into account other aspects such as user experience [4]. Although this step is introduced in the main description of the pipeline process, its details are out of the scope of this article. However, according to the nature of geolocated activities, some research lines are introduced in the following.

The use of health statutes, user behavior, the weather, and other context variables could be considered when recommending tasks to users. There are approaches related to physical exercise and the idea of avoiding users getting boring doing it [10]. In the context of this article, the sampling task could be assigned to people who complement their physical activities within the objective of the original projects: people do exercise and cooperate in collecting data of interest in an area.

In addition, several approaches centered the user preferences in order to recommend points of interest based on external resources like a knowledge base of restaurants, shopping malls, or any other social places[13]. However, the point of interest that feeds those approaches could be combined with the sampling task list, which can be considered as located areas of interest.

Finally, using user traveling behaviors is another approach to focus the distribution of sampling tasks. This means selecting those tasks that better adapt to the spatial and temporal behavior of users.

4 Conclusion and Future work

This work presented a pipeline approach for decision-making in generating location-based sampling tasks. This proposal considered the project's coverage objectives, area priority configuration, and global system status.

The generated task list may be a requirement for a recommendation system that assigns tasks to the users who participate in the project based on user profiles that can consider the person's characteristics, preferences, or historical behavior. Also, that system can recommend game items in an adaptive gamification approach.

The meshing approach developed in this work can be considered a cold start for an adaptation strategy based on the community's behavior. An initial set of

Received April 2023 Accepted July 2023; Published August 2023

areas can then be adjusted by calculating Voronoi regions taking as centroids the samples or from other hot spots that represent the busiest areas. This complemented approach allows having a dynamic set of sampling areas to propose a better distribution of the samples.

As further work, the inclusion of the proposed approach with the adaptive user-preferences-centered task distribution will be analyzed. Although there is a wide range of approaches, the first steps will be focused on using gamification strategies in the recommendation. For example, to generate specific gamification elements based on the generated task list to consider the project and user goals and preferences.

References

1. Cochero, J.: Appear: A citizen science mobile app to map the habitat quality of continental waterbodies. Ecologia Austral **28**, 467–479 (08 2018)
2. Cochero, J., Pattori, L., Balsalobre, A., Ceccarelli, S., Marti, G.: A convolutional neural network to recognize chagas disease vectors using mobile phone images. Ecological Informatics **68**, 101587 (2022)
3. Dalponte Ayastuy, M., Torres, D.: Relevance of non-activity representation in traveling user behavior profiling for adaptive gamification. In: Proceedings of the XXI International Conference on Human Computer Interaction. Interacción '21, Association for Computing Machinery, New York, NY, USA (2021). https://-doi.org/10.1145/3471391.3471431, https://doi.org/10.1145/3471391.3471431
4. Dalponte Ayastuy, M., Torres, D., Fernández, A.: A model of adaptive gamification in collaborative location-based collecting systems. In: Degen, H., Ntoa, S. (eds.) Artificial Intelligence in HCI. pp. 201–216. Springer International Publishing, Cham (2022)
5. Du, Q., Gunzburger, M.: Grid generation and optimization based on centroidal voronoi tessellations. Applied Mathematics and Computation **133**(2), 591–607 (2002). https://doi.org/https://doi.org/10.1016/S0096-3003(01)00260-0, https://www.sciencedirect.com/science/article/pii/S0096300301002600
6. Fleischmann, M., Feliciotti, A., Romice, O., Porta, S.: Morphological tessellation as a way of partitioning space: Improving consistency in urban morphology at the plot scale. Computers, Environment and Urban Systems **80**, 101441 (2020). https://doi.org/https://doi.org/10.1016/j.compenvurbsys.2019.101441, https://www.sciencedirect.com/science/article/pii/S0198971519302856
7. Franzoni, C., Sauermann, H.: Crowd science: The organization of scientific research in open collaborative projects. Research Policy **43**(1), 1–20 (2014). https://doi.org/https://doi.org/10.1016/j.respol.2013.07.005, https://www.sciencedirect.com/science/article/pii/S0048733313001212
8. Gold, C.: Tessellations in gis: Part i—putting it all together. Geo-spatial Information Science **19**(1), 9–25 (2016). https://doi.org/10.1080/10095020.2016.1146440, https://doi.org/10.1080/10095020.2016.1146440
9. Gold, C.M., Remmele, P.R., Roos, T.: Voronoi methods in GIS, pp. 21–35. Springer Berlin Heidelberg, Berlin, Heidelberg (1997). https://doi.org/10.1007/3-540-63818-0_2, https://doi.org/10.1007/3-540-63818-0_2
10. He, Q., Agu, E., Strong, D., Tulu, B.: RecFit: a context-aware system for recommending physical activities. In: Proceedings of the 1st Workshop

Received April 2023 Accepted July 2023; Published August 2023

on Mobile Medical Applications. pp. 34–39. ACM, Memphis Tennessee (Nov 2014). https://doi.org/10.1145/2676431.2676439, https://dl.acm.org/doi/10.1145/2676431.2676439

11. Kazemi, L., Shahabi, C.: Geocrowd: Enabling query answering with spatial crowdsourcing. In: Proceedings of the 20th International Conference on Advances in Geographic Information Systems. p. 189–198. SIGSPATIAL '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2424321.2424346, https://doi.org/10.1145/2424321.2424346
12. Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W.: Geographic information science and systems. John Wiley & Sons (2015)
13. Massai, L., Nesi, P., Pantaleo, G.: PAVAL: A location-aware virtual personal assistant for retrieving geolocated points of interest and location-based services. Engineering Applications of Artificial Intelligence **77**, 70–85 (2019). https://doi.org/https://doi.org/10.1016/j.engappai.2018.09.013, https://www.sciencedirect.com/science/article/pii/S0952197618301994
14. Nugent, J.: inaturalist, citizen science for 21st-century naturalists. Science Scope **41**(7), 12–13 (2018)
15. Reddy, S., Estrin, D., Srivastava, M.: Recruitment framework for participatory sensing data collections. In: Floréen, P., Krüger, A., Spasojevic, M. (eds.) Pervasive Computing. pp. 138–155. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
16. Shirani-Mehr, H., Banaei-Kashani, F., Shahabi, C.: Efficient viewpoint assignment for urban texture documentation. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. pp. 62–71 (2009)
17. To, H., Shahabi, C., Kazemi, L.: A server-assigned spatial crowdsourcing framework. ACM Trans. Spatial Algorithms Syst. **1**(1) (jul 2015). https://doi.org/10.1145/2729713, https://doi.org/10.1145/2729713
18. Tsai, V.J.D.: Delaunay triangulations in tin creation: an overview and a linear-time algorithm. International Journal of Geographical Information Systems **7**(6), 501–524 (1993). https://doi.org/10.1080/02693799308901979, https://doi.org/10.1080/02693799308901979

Received April 2023 Accepted July 2023; Published August 2023