

Chapter 2

Crowd+Cloud Machines

2.1 Combining Crowd and Cloud Computing

Computing today and the future could involve tens to thousands (to millions to trillions [23]) of (mobile/stationary) nodes that can cooperate in new ways, in order to provide new capabilities and applications, from massive context-awareness to new distributed computational platforms, forming the cloud or supported by large-scale (or data-centre scale) cloud computing resources (which we call the greater Cloud).

In this chapter, we review several examples how (machine and human) resources of a (mobile) *crowd* of people with separately owned devices can be pooled together and combined with a *cloud* computing mediating platform to form a type of crowd-powered system, or what we roughly call a *crowd+cloud machine*, to emphasise this combination between the two.

In the following sections, we first review types of mobile clouds, and then consider a range of examples of crowd+cloud machines:

- crowd+cloud machines that constitute supercomputers formed out of a loosely organised crowd of mobile devices;
- crowd+cloud machines that use the crowd for decentralised spatial computations;
- crowd+cloud machines that crowdsource to search for regions of certain properties;
- crowd+cloud machines that crowdsense and recognise group activities;
- crowd+cloud machines that bridge people with disabilities and workers who can provide real-time help; and
- crowd+cloud machines for annotating the physical world.

2.2 Types of Mobile Clouds

We first briefly look at a range of arrangements of computational resources or devices, which have been termed *mobile clouds*.

- *Personal clouds*: with multiple devices on a person (e.g., health sensors, smart-phone, smartwatch, smart shoes, smart jewellery and so on) and the widespread use of short range networking technologies like Bluetooth, one could pool together devices worn/carried by a user to form a cloud of personal resources that mobile apps can use. For example, mobile health apps can use specialised sensors for blood pressure, heart rate, and oxygen levels, and smart device based physical activity tracking as well as environmental sensing (e.g., for pollutants) and analyse the data in real-time to provide recommendations for the user, or upload data to the greater Cloud for storage and longer term analysis.
- *Vehicular clouds*: it has been proposed that vehicle-to-vehicle (v2v) networking or VANETS will link vehicles enabling sharing of vehicle sensor data and management of vehicle fleets for safety, content sharing (e.g., multimedia downloading and usage) and even virtualisation over the vehicular cloud to do data mining [11, 18]. Particular vehicles might play the role of a leader to coordinate the formation of a vehicular cloud with a (potentially changing) collection of vehicles. Cars at a long term parking lot such as an airport or a shopping centre might form a vehicular cloud but with dynamically varying resources as cars leave or come to the car park [2]. Longer range connectivity such as 4/5G can complement the shorter range v2v networks perhaps filling in gaps or to access remote resources.
- *Cloudlets*,¹ *Edge Clouds*, and *Fog Computing*: intermediary cloud servers (or cloudlets) between mobile users and the greater Cloud can be employed yielding reduced latency for mobile user applications (compared to data going to and from the remote cloud), improved security since data stays within particular geographical boundaries, and cloud outages can be masked via the edge cloudlet [32, 33]. Federation of such cloudlets can provide video analytics [34] for crowdsourced videos (e.g., using cameras on people or cars), useful in applications such as marketing and advertising, locating missing persons or property, and public safety survey (e.g., to monitor infrastructure such as damaged sidewalks, icy road surfaces and potholes)—a hierarchically structured system could pool multiple cloudlets together.
- *Mobile crowd clouds/mobile device clouds*: such clouds are formed mainly by pooling together devices from a nearby (say several metres or up to tens of metres away, via Bluetooth or WiFi-Direct) crowd of people. Fitzek and Katz [9] provided a range of ways in which a set of devices can combine resources. For example, a set of smartphones can share loudspeakers to create ‘social stereo and 3D social sound’, providing amplification when, for example, all the phones play

¹<http://elijah.cs.cmu.edu>.

the same sound files in a synchronised manner—while only a few devices are illustrated, the idea of scaling this to hundreds of devices at a rally could mean they act collectively as one powerful speaker. Or as mentioned earlier, imagine a crowd of people at a stadium watching a football match and imagine people taking pictures of the match from where they are and sharing them around—people can then have pictures and videos of the game from multiple perspectives, perhaps stitched together to form complex (almost) 3D footage of the game. Another idea is that the displays of a set of devices such as smartphones and tablets can be tiled together to form a large screen (though the inter-device boundaries cannot be removed).

- *Rethinking the Cloud for the IoT*: all the above are variations from the traditional model of cloud computing with huge building-sized array of computers and storage, towards more local pools of resources, still elastic and expandable to neighbouring clouds, and generalising on the notion of a resource that can be combined (going beyond CPU, memory and storage, to other kinds of capabilities, including sensing, connectivity, display, specialised processing functions such as video processing, analytics, positioning, media output, and so on). With the Internet of Things, we can consider multi-clouds of things, some of which are homogeneous, e.g., all the cameras in a building or along a street networked to work together, and some of which are heterogeneous, or different devices required to monitor and generate a comprehensive health report for a person. One could also think of specialised IoT clouds (1) for the elderly or the disabled (formed by the personal cloud on the individual and the local cloud in the home), (2) for streaming media such as video, (3) for agriculture, (4) for health/patient care, (5) for data mining, and (6) for a kid's room.

Smart things might themselves also crowdsource—e.g., if a device does not understand the user, it might ask the crowd to help it understand and answer the user, and smart things might also *cloudsource*, i.e., look for cloud resources when it is unable to perform a user specified task.

In the next section, we describe a system for integrating a (dynamically varying) crowd of mobile devices for machine and human processing.

2.3 Characteristics of Crowd+Cloud Machines: The Case of Honeybee and Multi-Layered Honeybee

We consider a *crowd+cloud machine* as a ‘computer’ formed by a crowd of people, each with their own (network of) devices (e.g., smartphone, smartwatch, smartcar, smart-*), inter-network-able to each other, providing human and machine computational capabilities, supported by the greater Cloud, and with varying boundaries and composition, characterised by

- *distributed ownership*: members (devices and their owners) of this crowd+cloud machine remain owned and largely administered separately by individuals, through the devices participate in a collective—the devices maintain a dual role: (1) as a participant in this crowd+cloud machine providing some of its resources in doing so, and (2) as a personal device used by its owner only;
- *heterogeneous devices*: members of the crowd-cloud machine are heterogeneous, some devices are more powerful than others and some have resources that others do not;
- *dynamic environment*: members of a crowd-cloud machine are coming and going, joining and leaving the machine, and so after a time, it could be that the set of devices that the machine consists of is an entirely or mostly different set from what it started with, even if the function the machine is performing has not changed;
- *ad hoc and opportunistic*: the crowd+cloud machine can be formed ad hoc for a specific application or opportunistically, e.g., when (the right number of) devices happen to be connect-able at the same time, the machine should then form and start working, or if a new device comes close enough with suitable resources, the machine could involve it, extending its capabilities;
- *context-aware*: the crowd-cloud machine needs to be sensitive to nearby devices and their state and to adapt accordingly;
- *localised*: closed proximity (or within the vicinity of each other) facilitates interaction, networking possibilities, high latency connectivity, and collaborations;
- *minimal assumed knowledge of devices/resources*: for devices to work together, they should not need to know too much information about each other, i.e., the less assumed knowledge of each other, the easier the interaction mechanism;
- *can combine human and machine computation capabilities*: while the idea is to pool together compute power and memory resources for machine computations, there are tasks that could be collaboratively performed not just with machine resources but with human input, e.g., in an application to search for someone or something (say a type of plant, bird, car, or some object), humans could use their cameras to point at particular crowds or areas and to walk around, or to find objects that fit a given description, but leave it to the image processing capabilities on the phone to pick out and identify potential required targets;
- *incentive-driven*: some incentive mechanism, be it monetary payment schemes, a point system, or a favour exchange mechanism is required for cooperation to happen, and incentives are required to maintain cooperation over time;
- *energy-aware (resource-aware)*: the task collaboratively being performed by a crowd-cloud machine needs to be aware of the resources it uses, not only in adapting or for metering, but so that optimisation to improve efficiency can take place;
- *elasticity*: as mentioned, the crowd+cloud machine should be opportunistic, but if there is a need to utilise more resources, e.g., it was found that more resources are needed than anticipated, the machine should be able to connect to the greater Cloud or other neighbouring crowd-cloud machines and collaborate.

There are different degrees to which one can possess the above characteristics.

As an example with an emphasis on computational resources, we briefly describe a framework called Honeybee [7, 8, 21] that maintains the above characteristics in a limited way. Honeybee is a mobile middleware, where a group of devices with Honeybee installed can work together on a heavy computation task, broken down into a series of small tasks or jobs. Honeybee uses a distributed work stealing algorithm for automatic load balancing so that heterogenous devices can effectively work together—more powerful devices can ‘steal’ non-started jobs from slower devices. Honeybee also uses device and resource discovery (as built into Bluetooth and WiFi-Direct) to look for new devices that could be integrated into the collective. Devices can leave and this is detected and its jobs can then be done by other devices. It was shown that speedups of up to four can be obtained with seven devices working together on a face detection problem over a sizeable collection of photos.

Also, perhaps somewhat surprisingly, a device that delegates its work to other (especially more powerful) devices can also save energy even if it has to transmit data to other devices for processing, especially when the short range device-to-device networking technology is adequately energy-efficient. Power savings can be had not just from offloading computations—other work [5] has noted that the power required to transfer data wirelessly from a wearable device to a smartphone (to be stored) can be less than that would be needed for storing the data in flash memory on the wearable device itself.

Not only can a group of workers work for a delegator, workers themselves can delegate tasks to other workers, hence, achieving elasticity when needed; when there are resources near-by, a tree of collaborating devices can be assembled ad hoc for a task and then disassembled when the task is completed. Figure 2.1 illustrates Honeybee with the two roles that devices can play. A task is divided into a set of jobs on the delegator, and the jobs are then taken up by workers according to how fast they can work, rather than the delegator assigning tasks to workers.

In an analysis done with devices in a library at a university as noted in [21], while sitting near the centre of the building, on a regular semester day, one can see that there is a crowd of 30 to over 100 devices detected per day via Bluetooth scanning over 5–15 h per day. Hence, there is virtually a supercomputer in the library surrounding a person at any time. Alternatively, consider the idea of the phenomenon of familiar strangers [36, 37, 39], including people who transit, using public transport daily on the same routes—even though they do not know each other, there is a relatively consistent group of devices surrounding users in urban environments; a perspective on this is that there is virtually a sizeable mobile ‘supercomputer’ comprising tens of nodes moving together with a user (or in this way, consistently stable relative to the user).

2.4 Decentralised Spatial Computing with the Crowd

A crowd+cloud machine can be formed ad hoc to compute, in a decentralised way, results which relate to the crowd itself. Duckham [6] presents a range of algorithms for nodes in a sensor network to compute spatial properties such as boundaries



Fig. 2.1 Honeybee with workers and delegators. Note that in this implementation of Honeybee, the delegator steals from slow workers, and faster workers then steal from the delegator—we could also have faster workers steal from slower workers directly, though trading-off some delegator control

and topological relationships, as well as establishing communication networks (e.g., trees for directed diffusion style communication). Similar algorithms with feedback from the crowd can be applied to find the boundaries of a crowd of people and whether more than a certain number of people are present, say at a rally or a large concert. To take an example from Duckham,² the following algorithm determines if there are more than 1000 people in the crowd.

An individual fan could start by placing a tally mark on a piece of paper. She can then pass this paper to a randomly selected neighbor, and ask them to add another tally. The paper then continues to be passed to neighbors subject to three rules:

1. please add a tally to the paper only if you have not already done so;
2. check the tally to see if it contains 1000 tally marks;
3. if it does, shout out ‘Crowd!’; if not, just pass the paper to another randomly selected neighbor.

Assuming the individuals in the crowd do as they are instructed, and if the crowd is large enough, sooner or later someone will shout ‘Crowd!’

The above algorithm can be viewed as basically a machine that computes the number of people in the crowd formed by the crowd itself, and the result perhaps uploaded to the Cloud. One can imagine adaptations of the algorithm above to compute the number of people in the crowd which satisfy a certain criteria C , e.g., just by changing rule 1 above to:

²<http://ambientspatial.net/book/?p=55>.

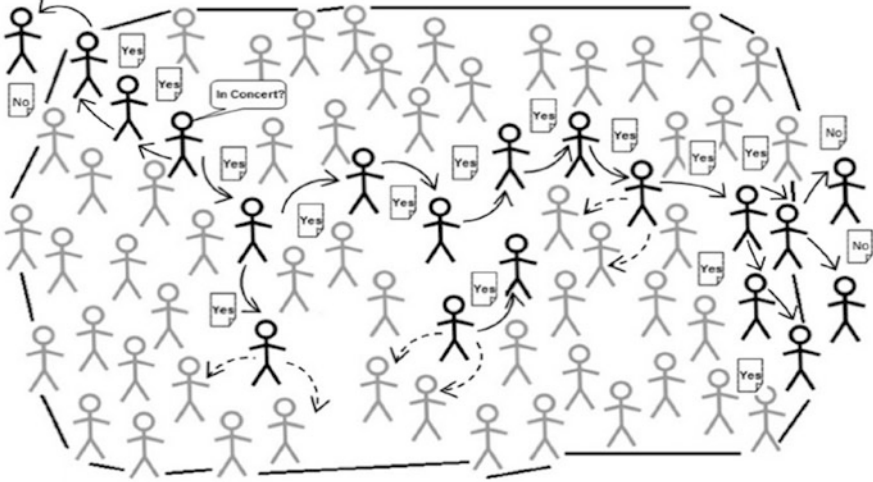


Fig. 2.2 Illustration of a decentralised peer-to-peer algorithm to ask the crowd, at a concert say, to determine the boundary. A query is passed around to ask if the person is in the concert or not, and eventually, the replies can be aggregated to determine where the boundary is

if you satisfy criteria C , please add a tally to the paper only if you have not already done so;

Figure 2.2 illustrates another algorithm to compute the boundaries of a crowd of people actually watching a concert [25]. As a query is passed around to ask if each person is watching the concert or not, and the replies can be aggregated to determine where the boundary is, people outside the concert crowd will reply ‘no’ and those inside reply ‘yes’, eventually revealing those at the boundary. Results could be returned along the same path that the queries were sent. Perhaps such a map of the crowd can then be used by someone to find the shortest direct path out of the crowd.

Also, those within the crowd could take a picture of themselves and upload it to a cloud server so that not only is there a rough count of the number of people at the concert, but also their identification (via a picture)—while not everyone might comply, the potential of such an algorithm exists. Instead of a picture of themselves, people could take pictures or videos of the concert and upload that to the cloud.

One can also provide such an algorithm to create more sophisticated versions of the ‘Mexican wave’ often seen at soccer matches. The decentralised algorithm for the Mexican wave is simple³:

You look to your right and see the wave approaching, accompanied by a crescendo. When it hits you, you jump up and throw your hands in the air, making whatever noise you feel apposite.

³<http://news.bbc.co.uk/2/hi/8742454.stm>.

Someone, of course, needs to initiate the wave. And if there are several points of initiation, multiple waves can be observed. A simple algorithm is in effect commonly used to orderly leave a building for example, after a movie or an indoor event: you look to your right or left, and when your neighbour leaves, you follow the person (follow either if both leaves) till you reach the exit. Another example is a Conway's game of life style crowd behaviour formation, e.g., you are at a concert, and suppose when at least the majority of your neighbours jump, you jump for at least 1 min after which you can choose to stop anytime—the overall effect, if the rule is followed, is a type of 'flock' or swarm behaviour emerging.

Interesting algorithms can be considered for crowds of people with mobile devices, e.g., leader election and computing the convex hull of a crowd of individuals, based on work in [29], to determine the boundaries of a place or the geographical range of crowd activities.

Another idea when the nodes or the people in the crowd can move or are moving is to use a decentralised algorithm via mobile-to-mobile communication to compute the best route to leave a building in case of an emergency, there are several algorithms to do this as given in [24].

2.5 Spatial Finding with the Crowd

We consider a crowd+cloud machine to compute, given bounded resources, a spatial map of a phenomenon, e.g., to compute the best possible, given the resource constraints, map of a where parking spaces are, where noisy areas are, real 3/4/5G bandwidth and coverage, as well as where the crowds in a city are currently. In [20] is a simple algorithm for crowdsourcing such urban maps while staying within a budget (since each question asked about an area could cost money, if money was paid for people to contribute information about an area). Briefly, the problem can be stated as follows.

Assume a large area R partitioned into n regions $\{r_1, \dots, r_n\}$. The problem is to find a set $S \subseteq R$ of at least $k \leq n$ regions, each of which evaluates to true for a given predicate F representing some criteria, i.e. $F(r) = \text{TRUE}$, for each $r \in S$. We also want to solve this problem with the lowest cost (assuming we need to pay to get a question about a region answered) and in a most efficient way (the number of rounds of questions required).

For example, we want to find at least k regions with available car parking spaces, and can divide a large area into a set of regions, about which we can then ask the crowd about, but each time we ask the crowd about a region, we assume that we incur a cost. Another example is to find a not-so-crowded cafe and can issue a query to find at least k regions with a not-so-crowded cafe, answers being given by people near or within the region. A third example is to find a high bandwidth (WiFi, 4G or otherwise) region.

Figure 2.3 illustrates five regions being queried (on the left) to reveal the results (on the right). After the five queries, and suppose there is enough in the budget to

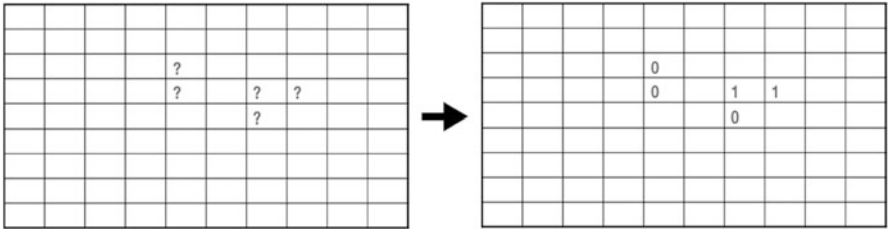


Fig. 2.3 Illustration of querying five regions to determine if the region satisfies a given property; the result is on the *right side*, where 0 means the region does not satisfy the property and 1 means it does

ask about more regions, a question is which other regions should one ask about? Note that if the phenomenon or property we are trying to find out about tends to be a clustering phenomenon so that, if one region satisfies the property, adjacent regions are likely also to satisfy the property, then a heuristic to determine which other regions to ask about would be to ask about regions which are surrounded by more 1s than 0s.

Figure 2.4 shows a coverage map and the corresponding fully disclosed idealised version. There are 7931 regions shown in the idealised version, i.e., to achieve full disclosure as in the figure, it would cost 7931 questions. This is likely too expensive and suppose there is budget for 100 questions, which of the 7931 regions would one ask about to maximise the chance of finding 1 (instead of 0) areas? For clustering phenomena such as bandwidth, a heuristic is to ask about areas adjacent to areas already found to be 1, but this could sacrifice finding new 1 clusters—hence, there is a classical exploration-exploitation trade-off. A range of heuristics can be examined for this purpose.

Other work in [38] used spatial regression techniques to learn spatial phenomena (e.g., radiation maps) represented as a continuous function from locations to values from possibly untrustworthy and noisy crowdsourced inputs, assuming the phenomenon can be modelled as such.

2.6 CAROMM and GroupSense: Crowdsensing and Crowd Activity Recognition

Crowdsensing has been explored for many years [13] and has a range of applications including crowdsourcing to detect and identify traffic regulators, traffic lights, and stop signs [14], traffic densities,⁴ and in emergency evacuation scenarios [15] as well as crowd management [10] and understanding crowd behaviour in events

⁴See <http://www.hh.se/download/18.c3a9c5b12ba24af5f580001878/1341267487930/WWVCJai erGPart1okt10.pdf>, <https://www.ncbi.nlm.nih.gov/pubmed/26761013>.

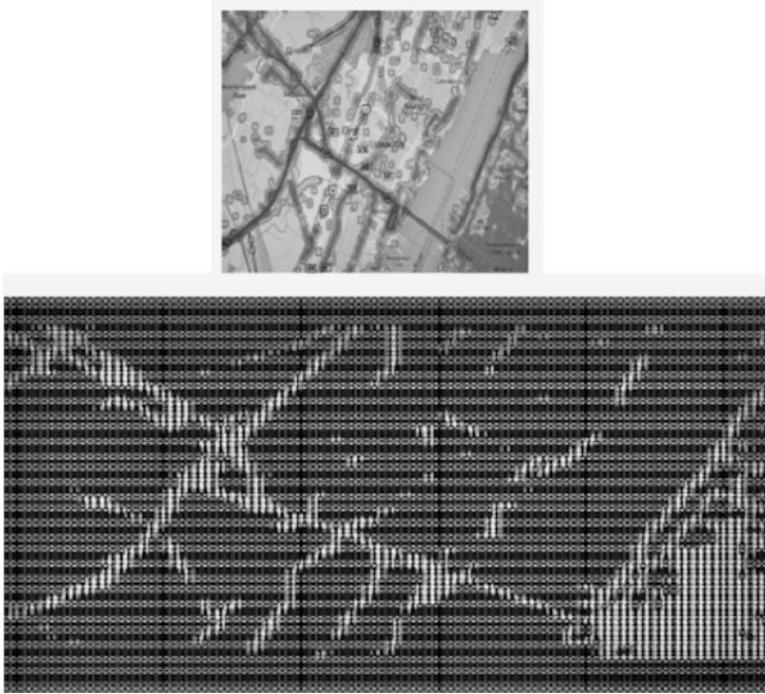


Fig. 2.4 Illustration of a 3G/4G coverage map—in the *top figure*, the *darker regions* show the coverage; the *figure below* is a discretised version where the entire area is partitioned into $103 \times 77 (=7931)$ regions—the *lighter regions* are the 1s and the *darker regions* are the 0s, corresponding to the bandwidth map above. The figure shows the fully disclosed map but the idea is that it is too expensive to ask about all 7931 regions so that some heuristic is needed for selecting the small number of regions to ask about

such as a music festival [16]. As mentioned, especially for transportation based applications, it is not only user’s personal mobile devices that can be used for crowdsensing but also vehicles (e.g., [19, 40]). Such crowdsourcing can provide insightful situation awareness but issues of privacy should be taken into account [26]. A number of frameworks have emerged (e.g., [28, 30, 35]) to allow efficient and effective crowdsensing, integrating the use of crowd mobile devices as well as cloud platforms.

Not just simple sensing of the behaviour of individuals in the crowd, one can also attempt to sense and infer the activity of the crowd as a whole. There have been tremendous work in activity recognition and group activity recognition using computer vision techniques, but recent work on group activity recognition attempts to use non-image processing techniques and rely only on sensors on mobile devices and objects [3, 12, 30].

A crowd-cloud machine can be built which tracks the activity of a group or a crowd of individuals (perhaps opt-in by the individuals) and uses that to provide new

mobile services to the individuals or to the group as a whole—e.g., one is providing more information beyond that of individual activity recognition, i.e., the system does not only know if a person is walking but that the person is walking together with some others (who are also walking). A tour group, a fitness group, a sports group, an expedition, or a bush-walking group can have their activities tracked via a cloud assisted platform, using mobile sensing readily available from the individuals' mobile devices, and reasoning with and aggregating the mobile sensor data from the group via cloud analytics.

2.7 Crowd+Cloud Machines to Assist People with Disabilities

While many of the above applications use mobile computational resources and sensing, there is opportunity for mobile crowdsourcing to involve users, not only on the worker or service side as in the spatial crowdsourcing mentioned earlier but also on the client side for whom the workers are doing their tasks. An example of this is the system called VizWiz⁵ for blind users to crowdsource and receive quick answers to questions about their surroundings, and others reviewed in [4]. Chorus:View [17] is a system that assists users with workers getting into a continuous conversation with the user about a video stream from the user's mobile device. Exploring crowd help to aid the visually impaired navigate along paths was also explored in [22], where a video stream from a client's mobile device is sent to a cloud platform and then forwarded to workers to be viewed so that they can then advise the user on his/her navigation. Apart from human workers, image processing algorithms can also be used to aid visually impaired clients.

A crowd-cloud machine can be built to help the disabled navigate places but issues of responsibility and risks remain open as to the extent and quality of help that can be given by remote workers.

2.8 Physical Annotation Systems

Physical annotation systems are systems (typically involving a mobile app for users as in [1]) for users to leave notes (which can be videos, photos, or audio, and not just text) or information that can be associated with particular objects or places, similar to how one might mark up a piece of text and link it with notes. The idea is that a layer (or multiple layers) of annotations can be added to the physical world, and stored in the cloud. The association can be done by associating an RFID tag of an object or place with the information in a database, but other forms of object or place identification can be used such as GPS coordinates and even just pictures of

⁵<http://vizwiz.org>.

objects. This enables not only other users to retrieve and read information (e.g., via an augmented reality style app) about a particular place or object that other users left for that place or object, but also allows robots and drones to obtain additional information about objects and places in the real world.

Humans (via crowdsourcing) and machines with aggregation and filtering algorithms might be needed to moderate the annotations. The system of mobile apps (for users to write annotations and link them to objects and places or to read annotations) and client software (e.g., on robots and drones to manipulate annotations) together with the cloud server (for storage and management of such information) constitutes the crowd+cloud machine for physical annotation.

2.9 Summary

We have reviewed a range of crowd+cloud machines, which are essentially distributed systems formed by mobile resources (often with human involvement) and cloud services. The different crowd+cloud machines can be combined. For example, the physical annotation systems can be combined with a machine for helping the disabled, so that the annotations can be read to clients who are visually impaired to provide more information about a place or to help people with memory loss [27]. When processing images or videos, resources from a Honeybee like pool of devices might be used, e.g., to identify faces in video streams used in a crowd-cloud machine to help the disabled when workers are not available or reliable. In the absence of human workers, crowd activity recognition can be used to automatically annotate what a group of people in front of a blind person is doing, and then this is read to the person. Crowdsensing can be used to obtain additional contextual information about the surroundings to predict available resources for upcoming jobs to be done via Honeybee-like computations. There are many interesting systems that could be considered crowd+cloud machines which we did not review here—an interesting example is *crowd physics* [31] which considers using people to help deliver packages. Uber delivery⁶ employs people to deliver food and goods for relatively small payments. Such a notion, if viewed as a system, albeit an open one, can be viewed as a crowd+cloud machine for deliveries.

References

1. Ahmad A. Alzahrani, Seng W. Loke, and Hongen Lu. An advanced location-aware physical annotation system: From models to implementation. *J. Ambient Intell. Smart Environ.*, 6(1): 71–91, January 2014.

⁶<https://www.uber.com/deliver>

2. S. Arif, S. Olariu, J. Wang, G. Yan, W. Yang, and I. Khalil. Datacenter at the airport: Reasoning about time-dependent parking lot occupancy. *IEEE Transactions on Parallel and Distributed Systems*, 23(11):2067–2080, Nov 2012.
3. Amin Bakhshandehabkenar, Seng W. Loke, and J. Wenny Rahayu. A framework for continuous group activity recognition using mobile devices: Concept and experimentation. In *IEEE 15th International Conference on Mobile Data Management, MDM 2014, Brisbane, Australia, July 14-18, 2014 - Volume 2*, pages 23–26, 2014.
4. Erin Brady and Jeffrey P. Bigham. Crowdsourcing accessibility: Human-powered access technologies. *Found. Trends Hum.-Comput. Interact.*, 8(4):273–372, November 2015.
5. R. Chandra, S. Hodges, A. Badam, and J. Huang. Offloading to improve the battery life of mobile devices. *IEEE Pervasive Computing*, 15(4):5–9, Oct 2016.
6. Matt Duckham. *Decentralized Spatial Computing: Foundations of Geosensor Networks*. Springer Publishing Company, Incorporated, 2012.
7. N. Fernando, Seng W. Loke, and W. Rahayu. Computing with nearby mobile devices: a work sharing algorithm for mobile edge-clouds. *IEEE Transactions on Cloud Computing*, PP(99): 1–1, 2016.
8. Niroshinie Fernando, Seng W. Loke, and Wenny Rahayu. Honeybee: A Programming Framework for Mobile Crowd Computing. In *Proc. of the 9th Int. Conf. on Mobile and Ubiquitous Systems: Comp., Netw. and Serv. (MobiQuitous)*, pages 224–236, 2012.
9. Frank H.P. Fitzek and Marcos D. Katz. *Mobile Clouds: Exploiting Distributed Resources in Wireless, Mobile and Social Networks*. Wiley Publishing, 1st edition, 2014.
10. Tobias Franke, Paul Lukowicz, Martin Wirz, and Eve Mitleton-Kelly. Participatory sensing and crowd management in public spaces. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys '13*, pages 485–486, New York, NY, USA, 2013. ACM.
11. M. Gerla, E. K. Lee, G. Pau, and U. Lee. Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 241–246, March 2014.
12. Dawud Gordon. *Group Activity Recognition Using Wearable Sensing Devices*. PhD thesis, Karlsruhe Institute of Technology, 2014.
13. Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y. Yen, Runhe Huang, and Xingshe Zhou. Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM Comput. Surv.*, 48(1):7:1–7:31, August 2015.
14. Shaohan Hu, Lu Su, Hengchang Liu, Hongyan Wang, and Tarek F. Abdelzaher. Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification. *ACM Trans. Sen. Netw.*, 11(4):55:1–55:27, July 2015.
15. Azhar Mohd Ibrahim, Ibrahim Venkat, K. G. Subramanian, Ahamad Tajudin Khader, and Philippe De Wilde. Intelligent evacuation management systems: A review. *ACM Trans. Intell. Syst. Technol.*, 7(3):36:1–36:27, February 2016.
16. Jakob Eg Larsen, Piotr Sapiezynski, Arkadiusz Stopczynski, Morten Mørup, and Rasmus Theodorsen. Crowds, bluetooth, and rock’n’roll: Understanding music festival participant behavior. In *Proceedings of the 1st ACM International Workshop on Personal Data Meets Distributed Multimedia, PDM '13*, pages 11–18, New York, NY, USA, 2013. ACM.
17. Walter S. Lasecki, Phyo Thiha, Yu Zhong, Erin Brady, and Jeffrey P. Bigham. Answering visual questions with conversational crowd assistants. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS '13*, pages 18:1–18:8, New York, NY, USA, 2013. ACM.
18. E. Lee, E. K. Lee, M. Gerla, and S. Y. Oh. Vehicular cloud networking: architecture and design principles. *IEEE Communications Magazine*, 52(2):148–155, February 2014.
19. Yazhi Liu, Jianwei Niu, and Xiting Liu. Comprehensive tempo-spatial data collection in crowd sensing using a heterogeneous sensing vehicle selection method. *Personal Ubiquitous Comput.*, 20(3):397–411, June 2016.
20. Seng W. Loke. Heuristics for spatial finding using iterative mobile crowdsourcing. *Hum.-centric Comput. Inf. Sci.*, 6(1):61:1–61:31, December 2016.

21. Seng W. Loke, Keegan Napier, Abdulaziz Alali, Niroshinie Fernando, and Wenny Rahayu. Mobile computations with surrounding devices: Proximity sensing and multilayered work stealing. *ACM Trans. Embed. Comput. Syst.*, 14(2):22:1–22:25, February 2015.
22. Seng W. Loke and Batni Prabhanjan. Guidemate: a crowd-powered system to assist the disabled. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers, UbiComp/ISWC Adjunct 2015, Osaka, Japan, September 7-11, 2015*, pages 1379–1384, 2015.
23. Peter Lucas, Joe Ballay, and Mickey McManus. *Trillions: Thriving in the Emerging Information Ecology*. Wiley Publishing, 1st edition, 2012.
24. Sabrina Merkel. *Building Evacuation with Mobile Devices*. KIT Scientific Publishing, 2014. Available at <http://www.ksp.kit.edu/9783731502074>.
25. J. Phuttharak and S. W. Loke. Towards declarative programming for mobile crowdsourcing: P2p aspects. In *2014 IEEE 15th International Conference on Mobile Data Management*, volume 2, pages 61–66, July 2014.
26. Layla Pournajaf, Daniel A. Garcia-Ulloa, Li Xiong, and Vaidy Sunderam. Participant privacy in mobile crowd sensing task management: A survey of methods and challenges. *SIGMOD Rec.*, 44(4):23–34, May 2016.
27. Eduardo Quintana and Jesus Favela. Augmented reality annotations to assist persons with alzheimers and their caregivers. *Personal and Ubiquitous Computing*, 17(6):1105–1116, 2013.
28. Moo-Ryong Ra, Bin Liu, Tom F. La Porta, and Ramesh Govindan. Medusa: A programming framework for crowd-sensing applications. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, MobiSys '12*, pages 337–350, New York, NY, USA, 2012. ACM.
29. Sergio Rajsbaum and Jorge Urrutia. Some problems in distributed computational geometry. *Theoretical Computer Science*, 412(41):5760–5770, 2011.
30. Haggaï Roitman, Jonathan Mamou, Sameep Mehta, Aharon Satt, and L.V. Subramaniam. Harnessing the crowds for smart city sensing. In *Proceedings of the 1st International Workshop on Multimodal Crowd Sensing, CrowdSens '12*, pages 17–18, New York, NY, USA, 2012. ACM.
31. Adam Sadilek, John Krumm, and Eric Horvitz. Crowdphysics: Planned and opportunistic crowdsourcing for physical tasks. In *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013.*, 2013.
32. M. Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, Jan 2017.
33. M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8(4):14–23, Oct 2009.
34. M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, and B. Amos. Edge analytics in the internet of things. *IEEE Pervasive Computing*, 14(2):24–31, Apr 2015.
35. W. Sherchan, P. P. Jayaraman, S. Krishnaswamy, A. Zaslavsky, S. Loke, and A. Sinha. Using on-the-move mining for mobile crowdsensing. In *2012 IEEE 13th International Conference on Mobile Data Management*, pages 115–124, July 2012.
36. Lijun Sun, Kay W. Axhausen, Der-Horng Lee, and Xianfeng Huang. Understanding metropolitan patterns of daily encounters. *Proceedings of the National Academy of Sciences*, 110(34):13774–13779, 2013.
37. Jameson L. Toole, Yves-Alexandre de Montjoye, Marta C. González, and Alex (Sandy) Pentland. *Modeling and Understanding Intrinsic Characteristics of Human Mobility*, pages 15–35. Springer International Publishing, Cham, 2015.
38. Matteo Venanzi, Alex Rogers, and Nicholas R. Jennings. Crowdsourcing spatial phenomena using trust-based heteroskedastic gaussian processes. In *Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2013, November 7-9, 2013, Palm Springs, CA, USA, 2013*.

39. Fusang Zhang, Beihong Jin, Tingjian Ge, Qiang Ji, and Yanling Cui. Who are my familiar strangers? revealing hidden friend relations and common interests from smart card data. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 619–628, New York, NY, USA, 2016. ACM.
40. Wangsheng Zhang, Guande Qi, Gang Pan, Hua Lu, Shijian Li, and Zhaohui Wu. City-scale social event detection and evaluation with taxi traces. *ACM Trans. Intell. Syst. Technol.*, 6(3):40:1–40:20, May 2015.

Crowd-Powered Mobile Computing and Smart Things

Loke, S.

2017, X, 65 p. 13 illus., 6 illus. in color., Softcover

ISBN: 978-3-319-54435-9