Pervasive Human Computing

Joel Ross

Humans Computing Everywhere

Humans perform informal computations throughout their daily lives across a variety of localized situations: from the arithmetic of estimating the cost of a purchase at a grocery store, to the calculus of regulating vehicle speed to match surrounding traffic, to executing synchronous scheduling algorithms to make sure that someone picks up the kids from school on time. In this sense, human computation is already a pervasive phenomenon—a process that is performed by a vast number of people in a variety of contexts.

Most prominent human computation systems rely on this pervasiveness in order to enable human-driven problem solving and information processing on a large scale. Human computation is frequently crowdsourced through systems such as Amazon's Mechanical Turk (AMT 2013) in order to either harness the vast quantities of human processing required to make human computation more effective than machine systems, or to enable the benefits of collective intelligence and crowd wisdom (e.g., Lévy 2001; Surowiecki 2005) in solving computational problems. Indeed, the "remote person call" or "human-as-a-service" view of human computation (see Irani and Silberman 2013) relies on such computation to be available at all times: on its home page, AMT describes itself as offering "a global, on-demand, 24 ×7 workforce" AMT (2013). Human computation systems require a near-constant connection between human computers and the mechanical systems that direct Quinn and Bederson (2011) their computing.

J. Ross (✉)
University of Puget Sound, Tacoma, WA, USA
e-mail: jross@pugetsound.edu

P. Michelucci (ed.), *Handbook of Human Computation*,
DOI 10.1007/978-1-4614-8806-4_27, © Springer Science+Business Media New York 2013

This requirement for constant access to computations performed by humans—a process that already occurs pervasively in a variety of locations—suggests that *pervasive computing* may offer a suitable interaction paradigm for supporting human computation-based systems. Pervasive computing[1] is a model of human-computer interaction (that is, interaction between a human and a computer) that involves moving away from traditional desktop interaction to focus on computing-in-context, embedding digital computer systems into the everyday physical world. Such computing systems may be passively embedded in the environment so that users are only peripherally aware of them (such as with ambient displays Ishii and Ullmer (1997)), or may represent computing systems with which users actively engage. One of the most common examples of a shift away from the desktop can be found in the increasing ubiquity of mobile devices and smart phones specifically—the mobility and constant network access afforded by such devices allow them to be integrated into everyday interactions, so that their use becomes "pervasive" in everyday experience. Research in pervasive computing often focuses on the ideas of "computing everywhere" and "everything can be a computer." Indeed, emerging research and even consumer products that make use of mobile augmented reality (AR) systems and "wearable computing" Mann (1997) continue to support embedding computers into people's everyday lives.

Pervasive computing thus offers an intriguing interaction paradigm for human computation. Just as pervasive technologies move digital computation away from the desktop machine into the everyday physical environments, pervasive *human* computation emphasizes moving the human computing into a variety of localized contexts. Indeed, pervasive computing as a form of interaction is highly interested in the context in which computation is used (e.g., Dourish 2004)—how computation can be embedded into the everyday lives of users. Such concerns remain valid even when the computation is performed by humans on the other end of a persistent network, rather than machines. Yet when considering pervasive human computation, we also need to perform a kind of inversion of this focus, since the human computers are the "users" of interest. Pervasive computing considers how computation may be used by humans in an everyday context; pervasive human computing introduces the question of how computation may be *performed* by humans in an everyday context.

In this chapter, I explore some of the uses of pervasive systems as platforms for performing human computation: porting current microtask-based interaction forms to mobile devices, and having humans act as computational controllers for mobile sensors. I discuss how these forms of human computation utilize or respond to the situatedness of the pervasive context in which they are performed. I follow this analysis with a reflection on some of the implications of considering human computation through the lens and goals of pervasive computing, particularly in terms of the visibility of the humans performing computation.

[1] Also known as *ubiquitous computing*, or "ubicomp" for short. Although "pervasive computing" and "ubiquitous computing" have been used to imply different emphases, in this article I will be using them interchangeably.

Mobile Human Computation

As mentioned above, mobile devices such as mobile phones are one of the most common platforms for moving computation into an everyday context and making it pervasive. Indeed, at a simple level, human computation can be made pervasive by porting existing systems and interaction patterns such as AMT for use on mobile devices. As an example, consider Harvard professor Jonathan Zittrain's vision of crowdsourced human computation combined with pervasive technologies:

> One can visualize in the near future a subway car packed with people, each far less attuned to the local environment and to each other than even with today's distractions of newspapers and iPods. Instead, they will stare into screens even for just a few minutes and earn as much money [via systems such as AMT] in that time as their respective skills and stations allow. (Zittrain 2008)

In this scenario, any extra minutes (extra mental "cycles," to use a mechanical metaphor) are devoted towards human computation rather than alternative activities such as media consumption.[2] While Zittrain problematizes this behavior (particularly contrasting for-pay activity with human contact or conversation), such mobile-based human computation need not be entirely profit driven. As a more positively framed alternative, those subway riders could be using their mobile devices to play *Foldit* (Khatib et al. 2011) on their mobile devices instead of *Angry Birds*—performing socially beneficial human computation in a mobile context.

In this way, human computation can be made pervasive by making the context in which it is formed more pervasive, such as through mobile technologies. This strategy has been refined through a number of research projects (e.g., Eagle 2009; Gupta et al. 2012; Narula et al. 2011), enabling human computation particularly in the context of developing countries. A second common strategy for making human computation pervasive applies crowdsourcing techniques for data gathering to pervasive contexts, creating what Zittrain goes on to describe as "distributed human sensors" (Zittrain 2008). These systems have humans act as computer sensors and record information about their localized environment (e.g., Paulos et al. 2009; Tuite et al. 2011). I discuss these projects and methods in more detail in the following sections.

In both of these methods, humans perform computation pervasively in the contexts of their everyday lives—yet such methods may or may not fully utilize the pervasive context in which they occur. Pervasive computing gives computing *situatedness*: the computation occurs within a specific local and social situation, allowing that situation to serve as input to and shape the interaction with the computational system. In pervasive human computation, this situatedness may allow human computers to access localized and contextualized knowledge, actions, or behaviors, thereby influencing the computation they perform. In exploring pervasive human computation systems, it is important to consider the impacts and use of this situatedness: what makes pervasive human computation different from non-pervasive human computation?

[2] In his novel *Rainbow's End*, Vernor Vinge expands this vision to include cognitive labor performed through mobile, wearable AR systems.

Again, note that in this chapter I am interested in how human computers perform their computation pervasively, not in how human computation as a replacement for mechanical computation (computation performed by machines) may be used pervasively. There has been significant and admirable work in the latter context: for example, *VizWiz* (Bigham et al. 2010) uses human computation harnessed through AMT to perform pervasive image recognition to support blind people in interacting with their environments. Yet in such systems, the human computation is still performed non-pervasively—the humans doing the image recognition are likely still using the desktop model of interaction, working through AMT using a web browser. Such systems address problems in pervasive computing using human computation, rather than making the human computation itself pervasive, which is the topic of interest here.

Human Computation Tasks on the Go

One of the simplest and earliest ways to make human computation pervasive is to have human computers report the results of their computation through mobile devices. This enables people to perform human computation during their everyday life, in a variety of different contexts and environments. Such systems must be enabled by existing infrastructures for pervasive technologies (i.e., ubiquitous network connections,[3] energy for powering mobile devices, etc.)—pervasive human computing "piggy-backs" off of mechanical pervasive computing systems.

Yet despite these requirements, systems for enabling such pervasive human computation have primarily been explored in the context of developing regions. For example, *txtEagle* Eagle (2009) built on the ubiquity of mobile devices and GSM reception in East Africa to deliver AMT-style human computation tasks to the mobile phones of workers in Kenya and Rwanda. These tasks—like those in AMT—were performed for pay, and offered as a way to supplement the low-income populations. Indeed, because of infrastructure in place for transferring mobile airtime (and the popularity of using airtime as a kind of currency), payments in either cash or airtime could easily be delivered to workers. The system's use is described with the following hypothetical scenario:

> *David, Maasai Herdsman, Kisumu, Kenya*. While David had been unable to complete formalized education, he, along with many of his Maasi peers, does own a mobile phone. David completes voice-tasks, helping Nokia train a speech recognition engine on his native Maasai dialect. When David wishes to complete a task, he 'flashes' the txteagle Asterisk box that calls him back, asking him to repeat specific key words and phrases. After 30 minutes of work, David has earned enough airtime to last him a week… (Eagle 2009)

Due to the limitations of available mobile phones (e.g., relying on numeric text entry), human computation tasks supported by *txtEagle* were primarily text- and

[3] Though even the computation of transmitting network data could be performed by humans, in what is informally called a "sneakernet".

audio-based: for example, human computers would perform transcription (of English words for those who were fluent) or translating text between their local languages to support software localization. Other systems have been developed to overcome these mechanical limitations. For example, *mClerk* (Gupta et al. 2012) uses proprietary protocols that predate MMS to send images to low-end phones in semi-urban India. This enables human computers in the region to perform optical character recognition (OCR) on scanned images.[4] Similar to *txtEagle*, *mClerk* pays human computers with mobile airline, administered manually through a "recharge shop."

Interestingly, in deploying the system, the researchers developing *mClerk* found that potential workers were skeptical of the system (perceiving it as a possible scam rather than a potential source of income). Yet once they overcame their skepticism, most users reported such human computation tasks were good for killing time. This study highlights some of the complications of developing mobile-based human computation systems: *computation activities need to be able to fit into existing activity structures*. For a human computation system to be operated pervasively, it needs to fill the same interaction gaps addressed by other mobile usage (see e.g., O'Hara et al. 2007)—for example, tasks that computers are able to complete in short bursts of time, or that can be performed while engaged in other activities. The micro-tasks common to systems such as AMT are usually suitable for such situations; nevertheless, such a restriction may influence the development of future pervasive computing systems.

The projects sampled here are all systems deployed within developing regions, raising the question of what factors may make such contexts amenable to pervasive human computing. I suggest that the main factor may be the "for pay" nature of crowdsourced human computation systems (such as AMT) that provide an interaction model for use of these systems. Although the economics of such systems are still being researched (see e.g., Horton and Chilton 2010; Silberman et al. 2010; Toomim 2011), in practice AMT-style tasks are performed for a relatively small wage.[5] As payment is the primary motivator in these markets, a low wage may restrict usage to those computers for whom the wage is still "worth the time": those in developing regions. Even non-pervasive human computation markets such as AMT see more work from lower-income regions such as India than higher-income countries such as the U.S. (Ross et al. 2010).

Thus designing pervasive human computation systems that are deployable in developed regions may require designs beyond "AMT on a cell phone", offering non-monetary motivations for performing computation. For example Heimerl et al. (2012) describes integrating human computation into a vending machine, using non-vital snacks as a reward instead of monetary payment. This design is exemplary of pervasive human computation, as the human computing is integrated

---

[4] *MobileWorks* (Narula et al. 2011) also supports human-performed OCR via mobile phones, but delivers images over a web application that requires a more powerful (and expensive) mobile phone.

[5] In 2009 (Ross et al. 2010) report workers from India make about USD 2.00/h on AMT, while in 2012 (Gupta et al. 2012) report the *mClerk* system payed around USD 2.84/h.

into the everyday environment. Other motivation structures may avoid extrinsic rewards all together, such as by "gamifying" human computation (e.g., von Ahn and Dabbish 2008; Carranza and Krause 2012). Such efforts can build on research into pervasive games (Montola et al. 2009) and games for harnessing collective intelligence (e.g., McGonigal 2007) to design interactions in which utilizable human computation pervades a game activity, which itself can pervade everyday life.

Whatever the motivation, while deploying AMT-style human tasks to mobile human computers does move the computation into a pervasive context, this form of interaction may not fully utilize the situatedness enabled by pervasive computing. Classical human computation tasks such as image identification rarely depend on or consider the context in which the computation is performed: indeed, identifying images on a mobile phone may even be made more difficult because of differing environmental lighting conditions! Systems such as *txtEagle* and *mClerk* do consider the social and cultural context of the computers to a small extent (e.g., when asking for translations between local languages), but these systems fail to consider the human computer's *specific* environment. Further research is needed into how the specific context in which human computation is performed may influence either the distribution or evaluation of AMT-style tasks in order to more effectively develop pervasive human computation systems.

In sum, the ubiquity of mobile devices offers a suitable platform for developing pervasive human computation systems—whether they simply provide a method for participating in existing crowdsourcing markets while on the go, or if they build on new forms of interaction for motivating contributions during short moments of free time. Yet motivating adoption of human computation platforms may require moving beyond the mobile device as a platform, embedding avenues for performing human computation in the artifacts that fill peoples' environments. Such embedding may help systems to better utilize the situatedness of the pervasive human computing, taking advantage of the computer's specific local and social context.

Human Sensing of Local Environments

While many existing human computation systems utilize the AMT-style "receive a task; complete a task; receive a reward" model of interaction, such systems do not fully utilize the mobile, pervasive nature of the interaction. Other forms of pervasive human computation work to expand the idea of what it means for humans to perform computation in order to take advantage of the localized context afforded by pervasive computing. These systems move beyond asking humans to act as just information processors, to asking them to emulate other aspects of mechanical computation.

The most prevalent of these other aspects is *sensing* the surrounding environment: in particular, having humans control and direct the use of embedded sensors. Also known as *participatory sensing*, this mode of interaction emphasizes crowdsourcing the use of sensors embedded in mobile devices, thereby enabling large groups of people to "gather, analyze and share local knowledge" (Burke et al. 2006).

Such interaction can be used to enable citizen science (e.g., Paulos et al. 2009), having humans direct the collection of pollution or noise data to better inform scientific research. Similarly, other systems such as *PhotoCity* (Tuite et al. 2011) have humans direct the use of an even more common type of sensor: the visual sensors that form the cameras found in most smart phones. In this system (framed as a game to motivate participation), humans use the cameras to intelligently provide photos that can be combined to successfully create a 3D reconstruction of a location. Thus rather than performing computation to *process* data, these human computers use their decision making skills to *produce* data that can then be processed.

As Reeves and Sherwood (2010) point out, the decision-making performed by humans in choosing how to direct the sensors is still a valid form of human computation. Such decisions "draw upon human agency and local practices" (Reeves and Sherwood 2010) to produce data more efficiently than may be produced by a fully automated sensor network (à la, Chong and Kumar 2003), much as human computation can be more efficient at the prototypical task of image identification. By putting humans in the loop in these pervasive sensing systems—turning them into pervasive human sensing systems—the computational efforts exerted by humans can outperform the computational efforts of the machines. Thus such human-directed sensing is a form of pervasive human computation: one that effectively utilizes the situated, localized nature of the computation being performed

Beyond simply directing mechanical sensors, pervasive human computation can even involve humans performing the sensing themselves! In this model of interaction, a system may query people for information that they can sense (e.g., "is there traffic?" "how's the weather?"), and then aggregate that data in order to produce computational models. To ease participation (and make such participation truly pervasive), the aggregating system can rely on reports that humans already produce, such as through social media. For example, people's reports of earthquakes on Twitter can be used to send alerts and notifications faster than traditional reporting systems (Sakaki et al. 2010), or provide situational awareness to support disaster response (Vieweg et al. 2010) *because* sensed data result from very specific contexts. These applications thus demonstrate how the situatedness of pervasive human computation can enable novel and effective systems.

This view that humans-as-sensors perform computation stretches the traditional understanding of what "human computation" entails (though Reeves and Sherwood (2010) note that even some tasks on the traditional human computation platform of AMT, such as writing product reviews, might not be considered "computation"). Zittrain's paper *Ubiquitous Human Computing* (Zittrain 2008) even suggests that systems that report biological vital signs from humans can be conceived as a form of human computation—computation that involves humans directly. Indeed, Zittrain suggests that such sensing could be used to support epidemiology–building on existing data mining systems such as Google Flu Trends (goo 2013). Quinn and Bederson (2011) argue that data mining systems are not human computation systems in themselves, but may they not be systems that involve or rely upon human computation?

In order to consider how human computation can best take advantage of contextual information available in pervasive systems, we may need to expand our understanding of what it means for a task to be computational. For example, social interactions are not normally considered to be computation, yet there may be identifiable "algorithms" which apply in these situations (such as the scheduling algorithm of how to plan one's day). If we want to make human computation pervasive, we may need to apply technomorphisms[6] to the wide range of actors and artifacts that exist within pervasive environments—using the lens of computer science and computation to look at traditionally non-computational systems. Such considerations can help us to take full advantage of the situated pervasive contexts in which pervasive human computation is performed.

Situating Pervasive Human Computation

Pervasive computing is computing that occurs in a variety of contexts: computation in the everyday world in which we live. Similarly, pervasive human computing moves human computers away from the desktop and "into the wild," allowing that computation to occur within a particular localized and social context—where and how the computation occurs matters! But how can we best utilize the contextualization afforded by pervasive human computing? How can performing human computation out in the world benefit existing forms of interaction (beyond simply increasing the availability of human workers), or otherwise enable the development of new systems? Future research is needed to further study the impacts of pervasive computing's situatedness on human computation, and how to best harness local contexts in human computation systems. Thus the significant open question is: *in what ways does the situatedness enabled by pervasive systems influence human computation?*

For one, research needs to explore how location influences computations performed: do humans tend to perform different types of computation (or perform computation in different ways) depending on their location? Are there problems that are dependent on localized computation but may be amenable to completion by human computers? Are there forms of human computation that could be immediately applied to problems in a local environment?

Second, research might consider how the presence of other nearby actors and artifacts—human or mechanical—can shape human computation performed pervasively. For example, research might consider the effectiveness of encouraging impromptu face-to-face collaborations, either between existing social groups or between co-located human computers. Other systems might use the pervasive presence of computers in order to help organize or control devices embedded in the environment. The question of how human computers may interact with their environments when computing in a pervasive context—how to best harness the potential benefits of this interaction—requires further study.

[6] A play on "anthropomorphism," referring to the attribution of technological characteristics to non-machines; see e.g., Vertesi (2008).

Finally, what are the influences of different cultural or social contexts? Cultural context is already a factor that needs to be considered when using existing human computation systems: translation tasks may require a certain fluency, or identification tasks may rely on knowledge of particular cultural touchstones. These issues may be further complicated when human computation occurs in a potentially more heterogeneous pervasive context. Similarly, the value or acceptability of systems may be influenced when presented within a social context that is not traditionally understood as computational—such as how the *mClerk* system was viewed as a potential scam (Gupta et al. 2012). The relationship between human computation, the connectivity and attention it requires, its framing of human labor, and other such factors need to be carefully considered in the development of pervasive human computation systems.

These are just some example questions that are ripe for future research; indeed, all these questions will need to be addressed in order to effectively utilize the situational context in which pervasive human computation is performed.

Invisible Human Computation

The research domain of pervasive computing is significantly based on the vision presented in Mark Weiser's foundational article, *The Computer for the twenty first Century* (Weiser 1995). In this paper, Weiser highlights the "seamlessness" of computer interaction enabled by pervasive computing—computers are so integrated with everyday artifacts and actions, that the computers "vanish into the background" and become invisible. The drive for computing technologies to become invisible, which has motivated large swaths of pervasive computing research, is clearly established from the article's first sentence: "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until their are indistinguishable from it." Tolmie et al. (2002) refer to this idea as "unremarkable computing," suggesting that such seamless interaction results not from the design of the technology, but rather from how the technology is utilized in practice. Although there has been some criticism of invisible computing as a model of interaction (especially work on "seamful design" Chalmers and Galani (2004); see also Bell and Dourish (2006)), it has remained the dominant vision of pervasive computing for decades.

But what happens when pervasive computing's idea of invisibility is applied to human computers? What happens when the humans that are doing the work "vanish into the background?" Such vanishing already occurs in non-pervasive human computation systems, such as how AMT obscures worker identities and renders them invisible by framing them as a form of infrastructure (Irani and Silberman 2013; Ross et al. 2010)—a part of the system's API. This obscuring leads to issues such as wage disparity (Silberman et al. 2010) in existing human computation systems—issues that likely would continue with pervasive human computation systems. Moreover, Weiser's vision of invisible computing suggests the idea of "scrap computers" (disposable computers, analogous to scrap paper); could making human computers invisible also cast them as disposable? We need to make sure that such obscuring does not become even more prominent when developing human computation systems for a pervasive context in which seamless interaction is the norm. While machines and technology can vanish into the background, we as developers and researchers have a moral obligation not to let our technomorphism of human computers cause the same to happen to them.

Notably, Weiser's goal in making computers invisible was to "make individuals more aware of the people on the other ends of their computer links" (Weiser 1995): users would be more cognizant of the others they are interacting with than the technology. Yet human computation systems—in addition to obscuring the computer (who happens to be a human)—often work to obscure the "user" of that human computation. Zittrain argues that obscuring the user (the requester or employer in for-pay systems) denies the human computers the moral choice about what they do or how their computational labor is used (Zittrain 2008). Indeed, legitimate human computation platforms such as AMT have been used for illicit purposes (such as allowing spammers to break CAPTCHAs), likely without the human computers being aware (Harris 2011). The problem of computation being decontextualized may be more significant in pervasive systems, particularly if the computation involves actions taken within a localized context—a human computer may be asked to act as a sensor and take a picture of a particular location without knowing the purpose of that surveillance.[7]

In these ways, considering human computation through the lens of pervasive computing highlights issues in how human computation systems often render the computer invisible, whether or not that computation is performed pervasively. In developing pervasive human computation systems, we should adopt a design stance that acknowledges—even emphasizes—the "seams" in the system. We should support awareness of the connections between the mobile human computers and the users of their computation, as well as limitations of the system that may be introduced by a particular localized context. Research should focus on revealing and harnessing the details of the human computation's context, and not let the computing fade invisibly into the background.

Conclusion

In this chapter, I have discussed the concept of pervasive human computation: a mode of interaction in which human computation is performed by people during their everyday lives in a variety of localized contexts. This form of human computation can range from current microtask-based interaction forms ported to mobile devices, to having humans control or act as mobile sensors to provide

[7] The dangers of crowdsourcing activity without context are effectively dramatized in Bruce Sterling's short story, *Maneki-Neko*.

human-gathered data to computational systems. Pervasive human computation has the potential to allow human computers to harness localized or contextualized information from their environment, thereby supporting a greater variety of systems and problem solving based on large-scale human-driven information processing.

When making human computing pervasive, the differentiating factor is the *context* in which the computation is performed: rather than sitting at a desk, human computers can be out in the world. It is this situatedness that makes pervasive computing significant—the computation occurs in a particular context. What is important is not that pervasive human computing occurs everywhere, but that it can occur *any*where—in a variety of specific locations and contexts. In developing systems, we need to be careful to not lose track of the particulars of the computation's context. Instead, we need to harness these specific contexts through systems that respect and make apparent the participating human actors (whether the computers or the users of the computation), in order to develop the most effective uses of pervasive human computation.

References

Amazon mechanical turk (2013). http://www.mturk.com/

Bell G, Dourish P (2006) Yesterday's tomorrows: notes on ubiquitous computing's dominant vision. Pers Ubiquitous Comput 11(2):133–143

Bigham JP, Jayant C, Ji H, Little G, Miller A, Miller RC, Miller R, Tatarowicz A, White B, White S, Yeh T (2010) VizWiz: nearly real-time answers to visual questions. In: Proceedings of the 23nd annual ACM symposium on user interface software and technology, UIST '10. ACM, New York, pp 333–342. http://doi.acm.org/10.1145/1866029.1866080

Burke J, Estrin D, Hansen M, Parker A, Ramanathan N, Reddy S, Srivastava MB (2006) Participatory sensing. In: ACM sensys world sensor web workshop, Boulder [AU1]

Carranza J, Krause M (2012) Evaluation of game designs for human computation. In: Eighth artificial intelligence and interactive digital entertainment conference, Stanford. http://www.aaai.org/ocs/index.php/AIIDE/AIIDE12/paper/viewFile/5535/5756

Chalmers M, Galani A (2004) Seamful interweaving: heterogeneity in the theory and design of interactive systems. In: Proceedings of the 5th conference on designing interactive systems: processes, practices, methods, and techniques, DIS'04, Cambridge. ACM, New York, pp 243–252. http://doi.acm.org/10.1145/1013115.1013149

Chong CY, Kumar S (2003) Sensor networks: evolution, opportunities, and challenges. Proc IEEE 91(8):1247–1256

Dourish P (2004) What we talk about when we talk about context. Pers Ubiquit Comput 8(1):19–30. http://dx.doi.org/10.1007/s00779-003-0253-8

Eagle N (2009) Txteagle: mobile crowdsourcing. In: Internationalization, design and global development. Springer, Berlin/New York, pp 447–456 [AU2]

Google flu trends (2013). http://www.google.org/flutrends/

Gupta A, Thies W, Cutrell E, Balakrishnan R (2012) mClerk: enabling mobile crowdsourcing in developing regions. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI'12, Austin. ACM, New York, pp 1843–1852. http://doi.acm.org/10.1145/2207676.2208320

Harris CG (2011) Dirty deeds done dirt cheap: a darker side to crowdsourcing. IEEE, pp 1314–1317 [AU3]

Heimerl K, Gawalt B, Chen K, Parikh T, Hartmann B (2012) CommunitySourcing: engaging local crowds to perform expert work via physical kiosks. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI'12, Austin. ACM, New York, pp 1539–1548. http://doi.acm.org/10.1145/2207676.2208619

Horton JJ, Chilton LB (2010) The labor economics of paid crowdsourcing. In: Proceedings of the 11th ACM conference on electronic commerce, EC'10, Cambridge. ACM, New York, pp 209–218. http://doi.acm.org/10.1145/1807342.1807376

Irani LC, Silberman MS (2013) Turkopticon: interrupting worker invisibility in amazon mechanical turk. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI'13, Paris. ACM, New York, pp 611–620. http://doi.acm.org/10.1145/2470654.2470742

Ishii H, Ullmer B (1997) Tangible bits: towards seamless interfaces between people, bits and atoms. In: Proceedings of the SIGCHI conference on Human factors in computing systems, Atlanta. ACM, New York, pp 234–241

Khatib F, Cooper S, Tyka MD, Xu K, Makedon I, Popović Z, Baker D, Players F (2011) Algorithm discovery by protein folding game players. Proc Natl Acad Sci 108(47):18949–18953. http://www.pnas.org/content/108/47/18949

Lévy P (2001) Collective intelligence. Read Digit Cult 4:253

Mann S (1997) Wearable computing: a first step toward personal imaging. Computer 30(2):25–32

McGonigal J (2007) Why i love bees: a case study in collective intelligence gaming. In: John D, Catherine T (eds) MacArthur foundation series on digital media and learning, pp 199–227 [AU4]

Montola M, Stenros J, Waern A (2009) Pervasive games: theory and design. Morgan Kaufmann, Burlington

Narula P, Gutheim P, Rolnitzky D, Kulkarni A, Hartmann B (2011) MobileWorks: a mobile crowdsourcing platform for workers at the bottom of the pyramid. In: Association for the advancement of artificial intelligence. http://www.aaai.org/ocs/index.php/WS/AAAIW11/paper/viewPDFInterstitial/3962/4263

O'Hara K, Mitchell AS, Vorbau A (2007) Consuming video on mobile devices. In: Proceedings of the SIGCHI conference on human factors in computing systems, San Jose. ACM, pp 857–866. http://portal.acm.org/citation.cfm?id=1240754

Paulos E, Honicky RJ, Hooker B (2009) Citizen science: enabling participatory urbanism. In: Handbook of research on urban informatics: the practice and promise of the real-time city, pp 414–436

Quinn AJ, Bederson BB (2011) Human computation: a survey and taxonomy of a growing field. In: Proceedings of the 2011 annual conference on human factors in computing systems, CHI'11, Vancouver. ACM, New York, pp 1403–1412. http://doi.acm.org/10.1145/1978942.1979148

Reeves S, Sherwood S (2010) Five design challenges for human computation. In: Proceedings of the 6th nordic conference on human-computer interaction: extending boundaries, NordiCHI'10, Reykjavik. ACM, New York, pp 383–392. http://doi.acm.org/10.1145/1868914.1868959

Ross J, Irani L, Silberman MS, Zaldivar A, Tomlinson B (2010) Who are the crowdworkers?: shifting demographics in mechanical turk. In: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems, Atlanta. ACM, pp 2863–2872. http://portal.acm.org/citation.cfm?doid=1753846.1753873

Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on world wide web, WWW'10, Hong Kong. ACM, New York, pp 851–860. http://doi.acm.org/10.1145/1772690.1772777

Silberman MS, Ross J, Irani L, Tomlinson B (2010) Sellers' problems in human computation markets. In: Proceedings of the ACM SIGKDD workshop on human computation, HCOMP'10, New York. ACM, pp 18–21. http://doi.acm.org/10.1145/1837885.1837891

Surowiecki J (2005) The wisdom of crowds. Anchor, New York. http://portal.acm.org/citation.cfm?id=1095645

Tolmie P, Pycock J, Diggins T, MacLean A, Karsenty A (2002) Unremarkable computing. In: Proceedings of the SIGCHI conference on human factors in computing systems: changing our world, changing ourselves, Austin. ACM, Minneapolis, pp 399–406. http://portal.acm.org/citation.cfm?doid=503376.503448

Toomim M (2011) Economic utility of interaction in crowdsourcing. In: Workshop on crowdsourcing and human computation at CHI 2011, Vancouver. http://crowdresearch.org/chi2011-workshop/papers/toomim.pdf

Tuite K, Snavely N, Hsiao DY, Tabing N, Popovic Z (2011) PhotoCity: training experts at large-scale image acquisition through a competitive game. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI'11, Vancouver. ACM, New York, pp 1383–1392. http://doi.acm.org/10.1145/1978942.1979146

Vertesi J (2008) "Seeing like a rover": embodied experience on the mars exploration rover mission. In: CHI'08 extended abstracts on human factors in computing systems, Florence. ACM, pp 2523–2532

Vieweg S, Hughes AL, Starbird K, Palen L (2010) Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI'10, Atlanta. ACM, New York, pp 1079–1088. http://doi.acm.org/10.1145/1753326.1753486

von Ahn L, Dabbish L (2008) Designing games with a purpose. Commun ACM 51(8):58–67

Weiser M (1995) The computer for the 21st century. Sci Am 272(3):78–89

Zittrain J (2008) Ubiquitous human computing. Phil Trans R Soc A 366(1881):3813–3821