Putting User Reputation on the Map: Unsupervised Quality Control for Crowdsourced Historical Data

Björn Barz
Lehrstuhl für Digitale Bildverarbeitung
Friedrich Schiller University Jena
bjoern.barz@uni-jena.de

Thomas C. van Dijk
Lehrstuhl für Informatik I
Würzburg University
thomas.van.dijk@uni-wuerzburg.de

Bert Spaan
Independent Map and Data Engineer
Amsterdam
bertspaan.nl

Joachim Denzler
Lehrstuhl für Digitale Bildverarbeitung
Friedrich Schiller University Jena
joachim.denzler@uni-jena.de

ABSTRACT

In this paper we propose a novel method for quality assessment of crowdsourced data. It computes user reputation scores without requiring ground truth; instead, it is based on the consistency among users. In this pilot study, we perform some explorative data analysis on two real crowdsourcing projects by the New York Public Library: extracting building footprints as polygons from historical insurance atlases, and geolocating historical photographs. We show that the computed reputation scores are plausible and furthermore provide insight into user behavior.

CCS CONCEPTS

• **Information systems → Geographic information systems**; **Crowdsourcing**; **Reputation systems**;

KEYWORDS

historical data, crowdsourcing, quality control, user reputation

ACM Reference Format:
Björn Barz, Thomas C. van Dijk, Bert Spaan, and Joachim Denzler. 2018. Putting User Reputation on the Map: Unsupervised Quality Control for Crowdsourced Historical Data. In *2nd ACM SIGSPATIAL Workshop on Geospatial Humanities (GeoHumanities'18), November 6, 2018, Seattle, WA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3282933.3282937

1 INTRODUCTION AND RELATED WORK

The use of crowdsourcing provides a big opportunity for the digital humanities in general, and the mapping of historical documents specifically. However, it comes with major concerns about data quality—see for example a recent survey by Daniel et al. [4]. In this paper, we propose a novel method for quality assessment, based on a notion of *consistency*: good users are likely to give answers that are similar to the answers of other good users. This apparently circular definition is resolved by finding a stationary vector of reputations, similar to the popular PageRank algorithm [11].

This approach allows us to determine a user reputation based on work items for which a sufficient number of different users have submitted an answer. We can subsequently use this knowledge about a user's performance to more accurately assess the quality of answers for work items for which only a few users have provided an answer. This enables a kind of *smart crowdsourcing*, where algorithms are used to increase the value of volunteered information.

In the Daniel et al. taxonomy, our method is a computation-based assessment. Our conceptual starting point is close to their concept of "output agreement," but we take it in a different direction. For example, various authors have considered the game-theoretic aspects of two crowd workers answering the same question [8, 14]. Instead, we take a global view of the whole dataset and do a post-hoc analysis.

Rajasekharan et al. [12] and Faisal et al. [5] use similar ideas in the context of open-source programming forums. However, they rely on direct interactions between users such as comments or up- and down-votes. This is a property of many existing user reputation systems: they employ some kind of control instance evaluating the crowd workers. This control instance could either come from external experts or the workers themselves who judge the work of others (*e.g.*, [1]). However, finding and remunerating experts can be difficult in settings requiring highly specific domain-knowledge. On the other hand, allowing the crowd to evaluate itself opens up a variety of possibilities for manipulation [13].

In contrast, we follow a content-driven approach for evaluating the work of each user without any additional input about its performance. Instead, it is based solely on the agreement between users. As a result, our method is more generally suited to the many kinds of crowdsourced historical data.

Similar in spirit but methodically entirely different is the STAPLE algorithm for crowdsourced image segmentation [15]. It computes consensus and user performance ratings simultaneously using the expectation-maximization algorithm: First, a probabilistic consensus is approximated, and then the performance of the annotators is assessed by comparing them to this consensus. A new consensus is then estimated based on the user performance computed in the previous step. These two alternating steps are iterated until convergence. This approach does not only provide user ratings, but also a consensus. However, it also requires the definition of a suitable performance metric and a probabilistic model for the consensus between multiple annotations, which can be hard to define given the

GeoHumanities'18, November 6, 2018, Seattle, WA, USA

ACM ISBN 978-1-4503-6032-6/18/11…$15.00
https://doi.org/10.1145/3282933.3282937

Figure 1: The user interface of the NYPL Building Inspector "Check Footprints" task. This polygon should be voted Fix, since it is semantically quite close to an actual building footprint, yet in terms of its vertices does not describe it cleanly: the number of vertices is wrong and the dark patch in the bottom right is wrongly excluded.

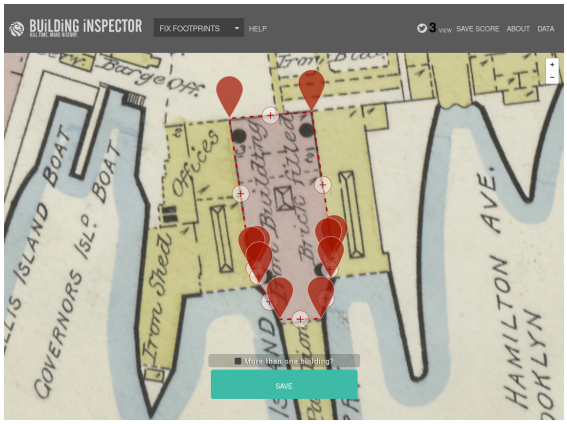

Figure 2: The user interface of the NYPL Building Inspector "Fix Footprints" task. This the task is performed on polygons that were voted Fix, the polygon is already close to a building footprint on the map; the user can edit the polygon by moving, inserting and deleting vertices.

complexity of some tasks. Our method, in contrast, only requires a measure of similarity between two annotations, which is much easier to obtain in most cases.

2 DATASETS

In this paper we perform a pilot study for our reputation-score method on two real-world crowdsourced datasets. Both are created by the New York Public Library within the *NYC Space/Time Directory* project, with support from the Knight Foundation.[1] This project aims to create a "digital time-travel service for New York City with historical maps, collections rich in geospatial data, and the public's help." It focuses on urban history and approaches this subject from a linked-data perspective: providing a searchable atlas of the city's past, with a historical location directory and geocoder, a set of APIs and datasets, and a discovery tool linking NYPL collections together in a historical and geographic context. This includes items such as old maps and photographs, and even historical restaurant menus. For more information, see

`http://spacetime.nypl.org`.

The source code of the NYC Space/Time Directory crowdsourcing applications is freely accessible at

`https://github.com/nypl-spacetime`.

2.1 The Building Inspector

The New York Public Library (NYPL) has a an extensive collection of insurance atlases from the 19th and early 20th century. The collection includes are large number of map sheets from 1853 to 1930 organized in 200 atlases. One of the goals of the NYC Space/Time Directory is to extract vector polygon and attribute data from these maps. Previous effort was based on staff and volunteer work to manually trace polygons in a custom web-based GIS. Using this manual process, it took three years to extract about 179,000 polygons across three atlases. At that pace, it would be impossible to extract the bulk of the data in any reasonable amount of time (and/or budget). In 2013, NYPL Labs started development of a semi-automatic pipeline for this polygon extraction task, which includes a crowdsourcing website called *Building Inspector*, which is still live at

`http://buildinginspector.nypl.org`.

Scans of the map sheets are processed by a pipeline of mostly off-the-shelf computer-vision algorithms in an attempt to automatically identify and extract building footprints; the process is described by Giraldo Arteaga [6]. (See Chiang et al. [3] for a comprehensive survey of digital raster-map processing techniques for historical maps.) This information extraction task is challenging and the extracted polygons—though often of reasonable quality—contain a significant number of errors. Therefore, these polygons are forwarded to the Building Inspector website for two stages of quality assurance and improvement in the crowd.

The first step is a straightforward application of crowdsourcing: in the "Check Footprints" task, each of the algorithmically detected polygons is presented to multiple users, who vote on whether this polygon correctly describes a building footprint or not. In addition to the obvious Yes and No options, the Building Inspector includes the option for users to vote Fix. See Figure 1 for an example of "fix" polygons. Polygons for which the majority vote is indeed Fix are forwarded to the next crowdsourcing step.

The subsequent "Fix Footprints" task (for the Fix polygons) is what we analyze in this paper. Here users can move, add and remove vertices to the polygon in order to make it match the underlying map image: see Figure 2. At the time of our analysis, over 128,000 polygons have been contributed by more than 3,900 users.

Since this is not a multiple-choice task, it is not immediately clear how majority voting could be used decide among multiple fixes of the same polygon. The Building Inspector tackles this using a "polygon consensus" algorithm [2]; our reputation-based approach

[1] `https://www.knightfoundation.org`

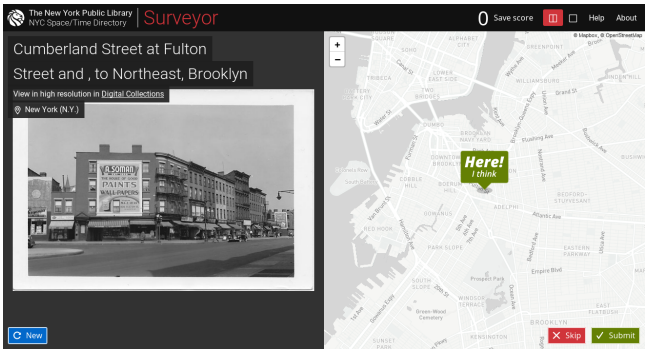

Figure 3: The NYPL Surveyor Website. Left: historical photograph, possibly including some metadata. Right: web map interface to locate the photographer's position.

can be used complementarily, or possibly the two approaches could be integrated.

2.2 The NYPL Surveyor

The second crowdsourcing project we evaluate within the NYC Space/Time Directory is the *Surveyor* at

`http://spacetime.nypl.org/surveyor`.

This is a geotagging tool designed generically to enhance the metadata of items within NYPL Digital Collections.[2] It allows users to view images and place them on a modern web map of New York City. We specifically look at a dataset for historical photographs and illustrations. At the time of our analysis, it contained 5,533 contributions by 2,064 users.

Figure 3 shows a screenshot of the Surveyor interface. The user is provided with a picture, sometimes annotated with metadata if available in the catalog. (For example, some historical photographs have writing on the back, possibly an address, and in some cases this has been transcribed already.) The crowd task is then: from which vantage point was this photograph taken (or: illustration drawn), and in which direction? Such information is of obvious relevance to the NYC Space/Time Directory goal of visualizing the spatiotemporal context of the NYPL collections. In this paper, we only consider the location data, that is, points given as latitude-longitude.

In contrast to the Building Inpsector tasks – which can be performed by lay users without outside assistance – users of the Surveyor are likely to consult outside material such as Google StreetView and Wikipedia. Since there is no gold standard "ground truth" data available for this task, it is not clear a priori that crowd users can even perform this task accurately. Here our unsupervised reputation method serves an additional purpose: if uncoordinated[3] crowd users generally agree on how to georeference a photograph, we can conjecture that they do so correctly since the only information they share is the picture.

[2] `https://digitalcollections.nypl.org`

[3] Manual inspection of the data confirms that there is no obvious collusion. See also the literature on agreement games [10].

3 METHODOLOGY

In the following, we first describe our general framework for computing user reputation scores and then provide the implementation details regarding the datasets mentioned in the previous section.

3.1 Building the User Agreement Graph

Instead of direct feedback about a user's performance, we employ the average *agreement* of a user with trustworthy users as an implicit quality measure. To this end, let $\mathcal{G} = (\mathcal{U}, W)$ be an undirected graph of n users $\mathcal{U} = \{u_1, \ldots, u_n\}$, connected by edges with non-negative weights $W \in \mathbb{R}^{n \times n}$. Furthermore, let $\mathcal{E} = \{e_1, \ldots, e_m\}$ be a set of samples that are available for annotation and $E : \mathcal{U} \to \mathcal{P}(\mathcal{E})$ be a function mapping a user to a set of all samples annotated by this user. The set of all annotations provided by the crowd workers is denoted as $\mathcal{A}$ and the function $A : \mathcal{U} \times \mathcal{E} \to \mathcal{A}$ gives the annotation by a certain user for a particular sample.

In our framework, the weight W_{ij} of any edge in the user graph equals the *average agreement* between the users u_i and u_j over all cases where they annotated the same sample:

$$W_{ij} = \frac{\sum_{e \in E(u_i) \cap E(u_j)} s\big(A(u_i, e), A(u_j, e)\big)}{|E(u_i) \cap E(u_j)|}\,, \tag{1}$$

where $s : \mathcal{A} \times \mathcal{A} \to \mathbb{R}^+$ is some measure of *agreement* or *similarity* between two annotations and depends on the type of data. We will describe the similarity measures used in our experiments for the two NYPL datasets in Section 3.3.

Edges between users whose sets of annotated samples are disjoint are assigned a weight of 0.

3.2 Computing User Reputations

Given this user-agreement graph defined by the weight matrix W, we can apply the PageRank algorithm [9, 11] to obtain a vector of user reputation scores $r \in \mathbb{R}^n$.

The PageRank algorithm was originally designed to determine the importance of web pages based on the hyperlinks between them. The foundation of this algorithm is the so-called *random surfer* model: a user is assumed to surf through the web by following a random link on the current web page with probability $\alpha \in [0, 1]$. With probability $1 - \alpha$, however, the user just visits a random web page. This is important for allowing the surfer to visit web pages without any in-going links. The PageRank of a particular web page is then defined as the stationary probability of the random surfer ending up on that page.

In analogy to this, we define the *random crowd worker*: at step t, this worker imitates user u_i with probability $r_i^{(t)}$, $i = 1, \ldots, n$. With probability α, the user imitates another user at the next step, chosen randomly with weights proportional to the agreement between the next and the current user, so that the random crowd worker exhibits a consistent behavior. Formally:

$$r^{(t+1)} = (D^{-1}W)^\top r^{(t)}\,, \tag{2}$$

where $D = \operatorname{diag}\left(\sum_{j=1}^n W_{1j}, \ldots, \sum_{j=1}^n W_{nj}\right)$.

With probability $1 - \alpha$, however, the random crowd worker contributes annotations similar to those of any other worker, chosen uniformly at random, that is, $r^{(t+1)} = \mathbb{1}/n$, where $\mathbb{1}$ is a length-n vector of ones.

To determine the user reputation scores, we compute the stationary distribution $r \in \mathbb{R}^n$, which fulfills

$$r = \left(\alpha \cdot (D^{-1}W)^\top + \frac{1-\alpha}{n} \mathbb{1}\mathbb{1}^\top \right) r \ . \tag{3}$$

This equation can be solved using eigendecomposition, for example, since r is the eigenvector of the combined transition matrix corresponding to eigenvalue 1.

The so-called "damping" hyper-parameter α (the probability of "jumping" to a random worker) can be seen as controlling the amount of regularization. We used $\alpha = 0.8$ for the Surveyor dataset and $\alpha = 0.1$ for the Building Inspector, but did not find it to be very sensitive. We hence defer a more detailed analysis of the effect of this hyper-parameter to future work.

In contrast to the original PageRank algorithm, which is applied to the directed graph of web pages, we apply this method to the undirected but weighted graph of users. In such a scenario, the PageRank often exposes some similarity to the weighted degrees of the nodes in the graph. It is, however, not identical to the degree distribution but only bounded by it [7].

3.3 Agreement Measures

Since the measure of agreement $s(a, b)$ between two annotations a, b depends on the type of annotations, we will explain the measures used in our experiments in separate subsections.

NYPL Surveyor. In the case of the NYPL Surveyor data, an annotation consists in the geographical position of a photo. It is hence intuitive to derive a similarity measure between two locations p, p' from their geographical distance $d(p, p')$, given in meters according to the great circle distance.

We employ an RBF kernel function to convert this distance measure into a similarity:

$$s(p, p') = \exp\left(-\gamma \cdot d(p, p')\right) \ . \tag{4}$$

The kernel hyper-parameter γ is set to 0.5 in our experiments.

Building Inspector. In the simplest case, an annotation consists in a single polygon depicting the building footprint. We measure the similarity between two polygons P and Q by means of their *intersection over union* (also known as the *Jaccard index*):

$$\mathrm{IoU}(P, Q) = \frac{|P \cap Q|}{|P \cup Q|} \ . \tag{5}$$

In some cases, however, samples from the Building Inspector dataset could also be annotated with a set of multiple polygons if the sample actually comprised more than one building. In such a case, we greedily assign each polygon from the first annotation to the most similar one from the second annotation. However, if that most similar polygon has already been assigned to another one, the second-best will be chosen if not already assigned, so that each polygon can have exactly one or no matches at all. The similarity between the two annotations is then given by the average similarity of all polygons. Unassigned polygons contribute with a similarity of 0 in order to penalize annotations with different numbers of polygons.

4 EXPLORATIVE DATA ANALYSIS

The following analysis of the user reputation scores computed by our method opens up some interesting insights about the behavior of crowd workers, which can be found in both datasets.

4.1 Reputation vs. Number of Contributions

A fairly obvious question regarding the performance of crowd workers is: do power-users perform better than casual users, in general? To answer this question, the scatter plot in Figure 4 shows the reputation score of each user depending on their total number of contributions.

There clearly is a strong correlation between these two variables. The Pearson correlation coefficient on the Surveyor data is 0.91 and 0.84 on the Building Inspector dataset. We attribute this to the intuitive explanation that the more active users who invest a lot of time into annotating also care more about quality and get more expertise over time.

Still, the most active user is not necessarily the one with the best reputation. In the case of the Surveyor data, the best score is obtained by the user with the fourth-most number of contributions and there also is a user with just 25 contributions but a better reputation than another user with more than 80 contributions. As regards the Building Inspector data, there even is a user with as few as 83 contributions but a better reputation than most other users with up to 1,000 annotations.

4.2 Evolution of Reputation over Time

We have analyzed how the quality of the contributions made by the four most reputable users on each dataset evolves over time. For this purpose, we move a sliding window with a fixed size over the list of a user's contributions, ordered by time. All contributions of that user outside of the window are then ignored during the construction of the inter-user agreement graph and the reputation scores are recomputed. We do this for each of the top 4 selected users individually. Since our reputation scores are calculated globally, the removal of these contributions also effects the reputation of other users. Therefore we normalize reputations by dividing between the maximum reputation of any user at each time step.

For the NYPL Surveyor data, we used a sliding window with a size of 20 contributions. A slightly different approach was taken for the Building Inspector data, since the total number of contributions by the top 4 users varies more strongly there: Instead of using a fixed number of contributions per window, we set its size to 20% of the user's contributions. Note that in both cases the windows at consecutive time steps overlap.

Several types of plausible user behaviors can be identified in the resulting graphs (see Figure 5). We discuss some of our observations and conjectured explanations here, with the caveat that this is a preliminary, explorative analysis.

Decreasing motivation over time. The users S2, S3, B1, and B4 start quite well, but apparently lose motivation or become tired after some time, resulting in a decreasing quality of their contributions.

Learning effect. User S1 and B3, on the other hand, seem to have needed some initial training to get used to the task, indicated by a steep initial increase in reputation. Later on,

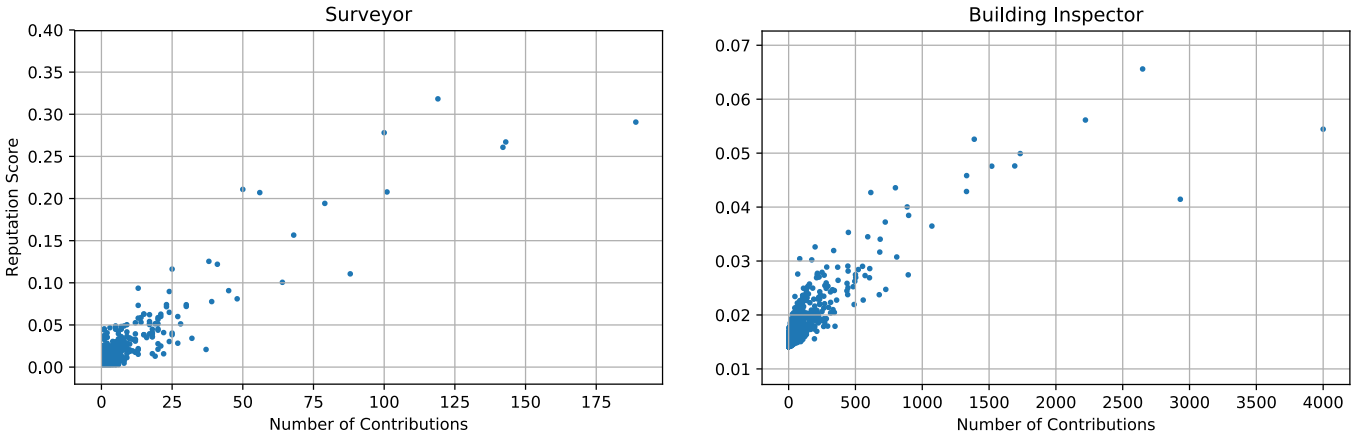

Figure 4: User Reputation Score vs. Number of Contributions.

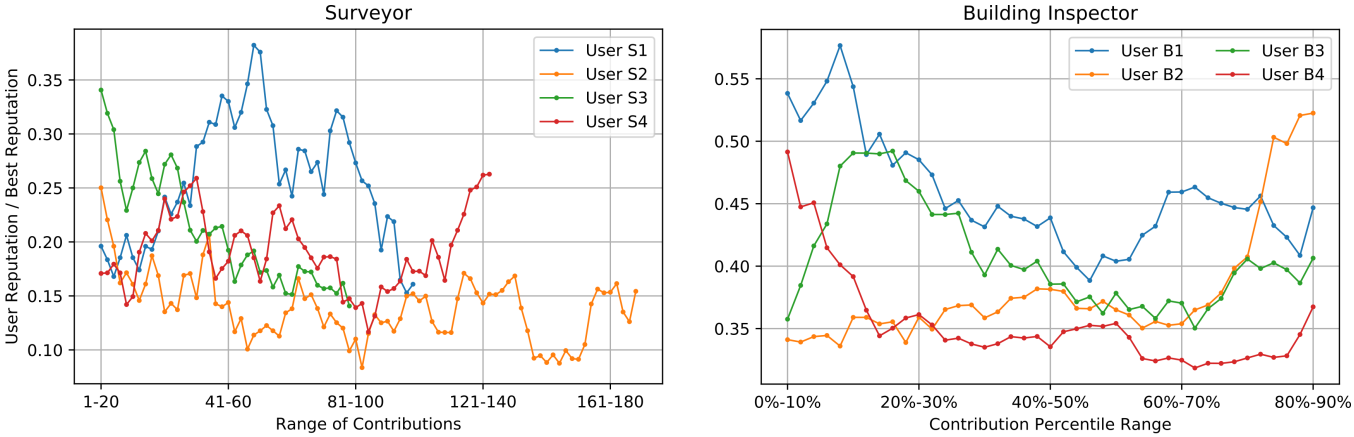

Figure 5: Evolution of User Reputation over Time.

however, these users drop in reputation, which suggests that their answers got worse again.

Finally, we provide a more detailed interpretation of the evolving reputation of user S4. The performance of this user is quite unsteady, but a closer look at the data reveals a plausible explanation:

- It starts with a steep learning curve, which drops off after a while. This matches the two behaviors mentioned above.
- Then there is a second, sudden increase of reputation after the 70^{th} contribution—again followed by a gradual decrease. To explain this phenomenon, we inspected the time stamps of this user's contributions and found that the first 70 contributions were made in one long session. After that, the user apparently went to bed and continued with the task the next morning. This rest period may explain the sudden increase in annotation quality. After 117 contributions, the user starts to take more and longer breaks, which might explain the notable increase of annotation quality towards the end.

Unfortunately, we do not have this exact timing information for the Building Inspector data. Thus, the sharp performance increase of user B2 remains open to speculation.

5 CONCLUSION

We have presented a widely applicable method for assessing individual crowd worker performance without any external input or evaluation. Our method is based on average inter-user agreement and an adaptation of the PageRank algorithm. The only requirement is a suitable measure of similarity between two annotations.

Experiments on crowdsourced data from the NYC Space/Time Directory exposed intuitively reasonable insights about the behavior of crowd workers—and power users in particular—as well as the effect of recreation on the annotation quality.

In future work, we will apply our method to more datasets and evaluate it quantitatively. One way to do this is using external "gold standard" judgments about the correctness of some annotations. This will tell us how well the reputations correlate to actual user quality. In addition, a small number of such (high quality, but expensive) judgments could be integrated into the reputation model, so that our specific knowledge about some users could propagate over the network. In this regard, it will also be interesting to investigate the use of active learning methods for selecting those annotations whose judgment would improve the reputation model the most.

ACKNOWLEDGMENTS

We thank Mauricio Giraldo Arteaga, formerly of NYPL Labs, for helpful discussions about the NYPL crowdsourcing projects. This research took place within the Volunteered Geographic Information Research Programme of the German Research Foundation (DFG). Björn Barz is supported by grant number DE 735/11-1; Thomas van Dijk is supported by grant number Di 2161/2-1.

REFERENCES

[1] Mohammad Allahbakhsh, Aleksandar Ignjatovic, Boualem Benatallah, Elisa Bertino, Norman Foo, et al. 2012. Reputation management in crowdsourcing systems. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*. IEEE, 664–671.

[2] Benedikt Budig, Thomas C van Dijk, Fabian Feitsch, and Mauricio Giraldo Arteaga. 2016. Polygon consensus: smart crowdsourcing for extracting building footprints from historical maps. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 66.

[3] Yao-Yi Chiang, Stefan Leyk, and Craig A. Knoblock. 2014. A Survey of Digital Map Processing Techniques. *Comput. Surveys* 47, 1, Article 1 (2014), 1:1–1:44 pages.

[4] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. 2018. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys (CSUR)* 51, 1 (2018), 7.

[5] Muhammad Faisal, Ali Daud, and Abubakr Akram. 2017. Expert Ranking using Reputation and Answer Quality of Co-existing Users. *International Arab Journal of Information Technology (IAJIT)* 14, 1 (2017).

[6] Mauricio Giraldo Arteaga. 2013. Historical Map Polygon and Feature Extractor. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on MapInteraction (MapInteract '13)*. 66–71.

[7] Vince Grolmusz. 2012. A note on the pagerank of undirected graphs. *arXiv preprint arXiv:1205.1960* (2012).

[8] Shih-Wen Huang and Wai-Tat Fu. 2013. Enhancing Reliability Using Peer Consistency Evaluation in Human Computation. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW '13)*. ACM, New York, NY, USA, 639–648. https://doi.org/10.1145/2441776.2441847

[9] Amy N Langville and Carl D Meyer. 2004. Deeper inside pagerank. *Internet Mathematics* 1, 3 (2004), 335–380.

[10] Edith Law and Luis Von Ahn. 2009. Input-agreement: a new mechanism for collecting data using human computation games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1197–1206.

[11] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd, et al. 1998. *The pagerank citation ranking: Bringing order to the web*. Technical Report. Computer Science Department, Stanford University.

[12] Karthikeyan Rajasekharan, Aditya P Mathur, and See-Kiong Ng. 2013. Effective Crowdsourcing for Software Feature Ideation in Online Co-Creation Forums.. In *SEKE*. 119–124.

[13] Yan Sun and Yuhong Liu. 2012. Security of online reputation systems: The evolution of attacks and defenses. *IEEE Signal Processing Magazine* 29, 2 (2012), 87–97.

[14] Bo Waggoner and Yiling Chen. 2014. Output agreement mechanisms and common knowledge. In *Second AAAI Conference on Human Computation and Crowdsourcing*. 220–226.

[15] S. K. Warfield, K. H. Zou, and W. M. Wells. 2004. Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation. *IEEE Transactions on Medical Imaging* 23, 7 (July 2004), 903–921. https://doi.org/10.1109/TMI.2004.828354