Algorithmically-Guided User Interaction (Vision Paper)

Thomas C. van Dijk
Chair for Computer Science I
University of Würzburg
thomas.van.dijk@uni-wuerzburg.de

Alexander Wolff
Chair for Computer Science I
University of Würzburg
ORCID: orcid.org/0000-0001-5872-718X

ABSTRACT

There are many practical problems in GIS that currently cannot be solved automatically, not because our algorithms are too slow but because we have no satisfactory algorithm at all. This can occur when semantics are involved, such as when extracting information or designing visualizations.

A computer currently cannot be expected to solve such problems in a completely unsupervised manner. To achieve high productivity anyway, we explicitly consider human effort as a resource. Clearly the algorithm should do as much of the work as is possible, at high quality – but crucially the algorithm should also be smart enough to see where it needs help, what it should ask the user to do, and how it takes those answers into account. This concept relates to emerging fields such as human(-based) computation and active learning, but we put the focus on the proper design and analysis of algorithms, and on the resulting dialogue between algorithm and human, which we call *algorithmically-guided user interaction*. As a showcase, we argue that this approach should be applied to information extraction from historical maps.

KEYWORDS

Algorithmically-guided User Interaction, Human Computation, Historical Maps, GeoHumanities

ACM Reference format:
Thomas C. van Dijk and Alexander Wolff. 2017. Algorithmically-Guided User Interaction (Vision Paper). In *Proceedings of 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Los Angeles Area, CA, USA, November 7–10, 2017 (ACM SIGSPATIAL'17),* 4 pages.
https://doi.org/10.1145/3139958.3140032

1 INTRODUCTION

The *algorithmic lens* (or: *computational lens*) is a computer scientist's point of view that "provides new insights and ways of thinking" [14] and in doing so "transforms the sciences" – not in the straightforward sense of being able to crunch the same numbers faster, but by providing a "powerful perspective that leads to unforeseen insights" [18]. In this paper we argue that this algorithmic lens should be applied to user interaction with geographic information. The starting point for our claim is that many tasks in GIS involve semantics or design and cannot currently be solved fully automatically.

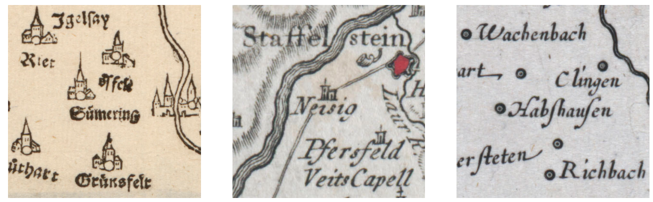

Figure 1: Place markers and labels on several historical maps from the Franconica collection of the Würzburg University Library. Note the variety of visual styles, both in the pictographs and the lettering.

The general template for designing algorithms for GIS applications is (or at least should be) the following: first develop a formal problem statement, then find an algorithmic solution to this problem. It may not be immediately clear, or be subjective, what the actual objective and constraints are. Good research practice is then to perform the following feedback loop:

(1) formalise a tentative problem statement (objective function, constraints),
(2) develop algorithms and/or heuristics for this problem,
(3) evaluate the computed solutions, and
(4) repeat from Step 1, adjusting the problem statement until satisfied in Step 3.

This is a powerful methodology for solving practical problems. However, it can be fundamentally unclear what the correct objective and constraints are when semantics are involved. To illustrate this point, we will discuss information extraction from historical maps.

It has been repeatedly argued in recent years (for example by Chiang [7]) that there is great potential in historical spatiotemporal datasets and historical Geographic Information Systems. Such data is of significant scientific interest, both in the humanities and the sciences. An important step, then, is the automatic extraction of information from historical maps. However, the problem of "understanding everything on this map" is so vague, and the data so diverse, that it is hard to specify clean problem statements. Often it is even unclear what the input to an algorithm is supposed to be. The classical sense of a *problem statement* to be solved by an algorithm breaks down when we are unable to sufficiently formalise the objective or even the input. But this does not mean that the algorithmic approach must be abandoned: instead, this means that we should apply the algorithmic lens to these classically ill-defined problems and advance the state of algorithmics. We propose the

ACM SIGSPATIAL'17, November 7–10, 2017, Los Angeles Area, CA, USA

ACM ISBN 978-1-4503-5490-5/17/11…$15.00
https://doi.org/10.1145/3139958.3140032

development of algorithms with user interaction on the level of *individual instances*. This introduces a second feedback loop where the algorithm and the user (or a group of users) cooperate to turn an efficiently-computed first attempt into a reliable solution as follows.

(1) Algorithm computes an initial solution.
(2) User evaluates the current solution, guided by the algorithm.
(3) User adds (removes, modifies) 'hints' or 'corrections'.
(4) Algorithm incorporates these in a new solution, providing feedback about how the built-up user input affects the current solution and possibly asking for help on certain parts of the instance.
(5) Repeat from Step 2 until satisfied.

This loop is somewhat reminiscent of the concept of active learning, but in a more general setting, and can be seen in some of our previous work [3–5]. Our overarching research question is this.

Q: *In what ways can an algorithm efficiently assist a user in a mathematically ill-defined task?*

Taking a more technical view leads to a slightly different question.

Q: *What are the algorithmic techniques that enable algorithms to take meaningful hints and to give effective feedback?*

Based on the state of the art, we propose that the answer to the latter question includes sensitivity analysis and active (machine) learning (Section 2). Next we look at a fruitful application area for this approach: information extraction from historical maps (Section 3). We argue that crowdsourcing should also be considered in this light (Section 4). These questions have far-reaching implications for the area of *volunteered geographic information* (VGI), which is currently investigated in a priority programme[1] of the German Research Foundation (DFG). Our contribution to this programme focuses on algorithmically-guided user interaction for information extraction from historical maps.

2 STATE OF THE ART

There is an expansive field of work on VGI, with the successful OpenStreetMap project [11] as its poster child. See for example Goodchild [9] for a general review of the concept, placing it within a context of more traditional citizen science and the role of the general public in geographic observation. For a comprehensive overview of VGI as geographic information science, see the recent book on the topic edited by Arsarjani et al. [1]. However, the algorithmic lens is sorely lacking in this area – often, even the first step (the development of a formal problem statement) is missing.

Much of the literature is concerned with the case where users directly input the features of interest: they contribute for example GPS-measured geometric data (such as a building footprint or the course of a road). We need not limit ourselves to users providing raw data. There can be an algorithmic step where the system generates tasks for the user, adaptively distributes those tasks, and integrates the human answers. For example, Arteaga [2] develops a system where the data of interest (a polygonal representation of building footprints extracted from a raster map) is generated algorithmically, and the users perform quality control. A further algorithm integrates the (possibly ambiguous) user answers into a reliable consensus answer [5].

Coming from theoretical computer science, Shahaf and Amir [21] have introduced the notion of *Human-assisted Turing machines*. In their framework, an algorithm can not just compute in the usual sense but can also make calls to a human oracle that can solve problems that are hard for computers. They measure the time complexity of an algorithm by a pair of functions $\langle H, M \rangle$, where, for an input of size n, $H(n)$ counts the number of calls to the human oracle and $M(n)$ the number of computation steps of the machine. They say that a human-assisted algorithm A is more efficient than another such algorithm A' if A needs fewer oracle calls, but does not increase the machine complexity to exponential.

While overall the definition of Shahaf and Amir is convincing from a theoretical point of view, it is probably not fine-grained enough for analysing and comparing algorithms for practical problems. Moreover, a notion of quality is missing: how much human interaction do we need in order to solve a problem *sufficiently well*? Can we formulate trade-offs between human interaction and quality of the solution?

Zhang et al. [24] consider crowdsourcing for general computation, but their techniques are general and unlikely to lead directly to practical applications. Von Ahn gives an overview of this field and calls it *human computation* [23]. A more recent overview is given by Quinn and Bederson [19] and parts of our vision are also visible in their "openings for growth."

Another relevant research area (considered a subfield of machine learning) is called *active learning*. Settles [20] provides an excellent survey of this area. Quinn and Bederson also identify it as being relevant to advancing human-assisted computation. Particularly interesting to us is batch-mode active learning [6, 10, 13]. This means that instead of adaptively querying the human "oracle" one tiny question at a time, the algorithm has to commit to a set of questions before it can gather the responses. This can improve the user experience for single-user interfaces [4, 17] and is clearly relevant for crowdsourcing systems.

A related aspect of active learning (and the common technique of *uncertainty sampling*) is that it may tend to find genuinely difficult questions, in the sense that the answer may be unclear even to humans. These samples may not serve well to train a learning algorithm [15], but this is a way to find exceptional situations in the input, which is important for quality assurance and knowledge discovery.

3 APPLICATION: HISTORICAL MAPS

In a recent SIGSPATIAL vision paper, Chiang explicitly asks for "semi-automatic map processing services" to "efficiently extract map features with uncertainty measures" and for these services to be trainable using crowdsourcing [7]. Our proposal of algorithmically-guided user interaction and smart crowdsourcing fits squarely within this programme.

Many (university) libraries have an extensive collection of historical maps. Besides their value as historical objects, these maps are an important source of information for researchers in various scientific disciplines. With the progressing digitisation of libraries, these maps become more easily available to a large number of scholars.

[1] www.vgiscience.org

In order to make the maps meaningfully searchable, information describing their contents is required. Particularly useful would be a *georeferenced* index of the contained geographical features (such as labeled cities and rivers) and geopolitical features (such as political or administrative borders). This enables relevant queries such as "all 17$^{\text{th}}$ century maps that include the surroundings of modern-day Würzburg." It also enables analyses of the accuracy or distortion of the map, which is of historical and cartographic interest.

Unfortunately, analysing the contents of historical maps is a complex and time-consuming process. For the most part, this information extraction task is performed manually by experts – if at all. Automated tools are scarce, for a variety of reasons. For one, there is a large variety of drawing styles in historical maps. This makes it hard for a single algorithm or software tool to automatically perform well on a large set of maps: see Figure 1 for some examples. Systems that assist with the digitisation of maps are still mostly manual (for example [8]). *Recogito* has a particularly nice web interface and includes automatic tools for text mining [22], but does not meaningfully assist with maps. These tools all make inefficient use of human effort.

Some research has gone into image processing specifically for historical maps. These systems are usually rather sensitive to their parameters, requiring careful tweaking in order to perform well. Höhn et al. [12] specifically raise this as an area for improvement: their experiments work well, but do not necessarily generalise to a large variety of maps. Some fully-automatic approaches exist, but only for restricted inputs – that is, developed specifically to digitise a particular corpus. For example, Leyk et al. [16] describe a method to find forest cover in a specific set of 19$^{\text{th}}$ century topographic maps. The effectiveness of such approaches is due to the homogeneity of these relatively recent maps. A general solution must handle a wider range of maps (cf. Fig. 1).

Additionally, there is the question of input. When a historian georeferences a map, he or she brings a wealth of background information and the ability to do additional research when required. Lastly, we address the issue of correctness: in general, algorithms for extracting semantic information from bitmap images are far from perfect. This is to be expected since these problems are truly difficult for computers. Algorithmically-guided user interaction can help in the following ways.

Provide judgement. Firstly, user interaction can be used to set (or adjust, or tweak) the parameters of an algorithm, such as thresholds and templates. If this interaction is well designed, it can allow existing algorithms to be applied more effectively to a wider range of inputs. We propose to take this further: the concept of user interaction should influence the design of the algorithms from the start. What information can the algorithm get from the user that will be particularly helpful? The user's time is a resource to be optimised, involved in a trade-off with computer runtime and quality. This should be considered during algorithm design.

Possibilities for user input include semantic interpretation such as "This is what a place marker looks like on this map!" (cf. Fig. 1) and answering questions like "This label looks like it says *Neunbrun*: could that be modern-day *Neubrunn?*" (cf. Fig. 2, rightmost label).

Provide domain knowledge. An (expert) user can provide domain knowledge that no algorithm would realistically have. Consider Figure 2 and assume the labels have been found and read automatically: which of the three proposed assignments is correct, based solely on information that an algorithm would have? Note that *Rinderfelt* and *Neunbrun* do not appear on modern-day maps. As suggested above, it should be possible to automatically relate "Neunbrun" to the modern-day town of Neubrunn. (But is this correct?) Historical records show that the Rinderfelt meadows belonged to the Gerlachsheim monastery, which is in modern day Lauda-Königshofen, to the south of Neubrunn. This suggests that the rightmost assignment in Figure 2 is correct, once we know that south is up in this map. We do not expect to develop algorithms that can do this kind of inference automatically. Even ignoring algorithmic difficulties: where does all this information come from? Instead, we note that this *conclusion* can easily be communicated to an algorithm, which can then take it into account while reasoning about the rest of the map. Indeed, that is precisely what our work on label–marker matching is able to do [4].

Provide quality assurance. Finally, the issue of quality can also be addressed interactively and adaptively. When extracting semantic information from unstructured input such as bitmap images, we must rely on a human for ground truth. Our task as algorithm designers is then to facilitate *quality control*: the user and algorithm together should efficiently arrive at confidence that the end result is correct. One way to achieve this is using sensitivity analysis: the user's attention can then be focused on parts of the solution that are most uncertain. When those have been fixed – and the consequences have been propagated – the more certain parts of the instance can ideally be assumed to be correct.

Q: *In what ways can algorithmically-guided user interaction help to efficiently extract information from historical maps? How can user effort and quality control be sensibly balanced?*

4 SMART CROWDSOURCING

Most crowdsourcing applications rely on the crowd's most obvious power: its multitude. If we throw enough users at a problem, we may solve it by brute force. But just like brute-force algorithms are often grossly inefficient, so is the indiscriminate application of human effort. A theory and practice of efficient crowdsourcing is sorely missing. This will involve developing appropriate notions of efficiency, as well as the algorithmic techniques to optimise them. As mentioned, there is the emerging theory of human-assisted Turing machines [21], but it is rather theoretical for now. On the applied side, there are many ad-hoc crowdsourcing projects. What is needed is a practical toolbox of algorithmic techniques and evaluation criteria for efficient crowdsourcing.

Q: *How can algorithms be used to increase the value of crowdsourcing, by intelligently selecting tasks and aggregating the results? How does this influence the design of effective user tasks?*

The raw data gathered in a crowdsourcing process is often of questionable quality, making quality assurance is a central concept. An obvious approach is to get multiple responses for each individual task and take a majority vote. This is possibly inefficient and while a majority vote works for simple multiple-choice questions, it might

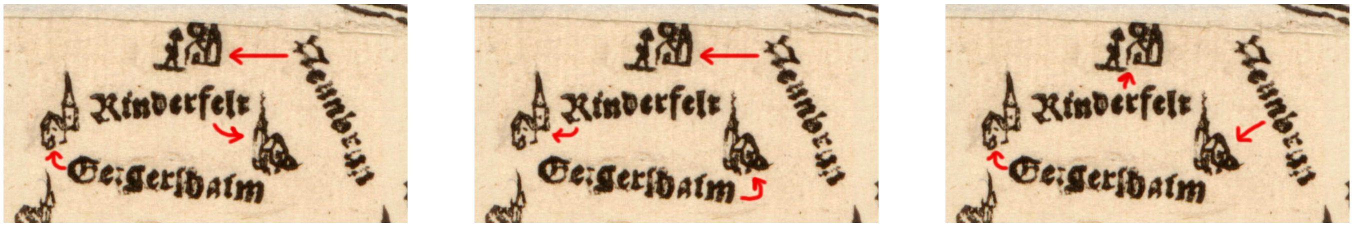

Figure 2: A difficult case on a map from the Franconica collection: without historical context it is hard to tell which label belongs to which place marker.

not be clear what the "consensus answer" is in more complex user tasks. This leads to interesting algorithmic questions, where good solutions will have to be based on proper algorithmic modeling of the underlying computational problem being solved. This gives a well-founded criterion for analysing how much the resulting data are influenced by missing or incorrect answers and allows us to direct users to those parts of the data where their effort will be most effective – either by deciding what new task to give to a certain user, or which existing data is most important to check. When errors are caught and fixed, the consequences can be propagated algorithmically. This leads to algorithmic quality assurance: the users and algorithm *together* arrive at confidence that the end result is correct.

Q: *How can algorithms guide a crowdsourcing task and help with quality control of crowdsourced data?*

5 SUMMARY

In this paper we have described a vision of algorithmically-guided user interaction. This is the application of the algorithmic lens to tasks that resist fully-automatic solutions. A central aspect is the explicit consideration of human effort as a resource to be optimised.

We have illustrated this approach with possible applications to information extraction from historical maps. This is far from the only application that can be addressed in this way. Other areas within GIS that face challenging problems due to semantics and difficult-to-interpret data include: land-use detection and change monitoring from remote-sensing data, spatio-temporal sentiment analysis (such as from microblogs and geotagged photos), and advanced analyses of trajectory data. The latter touches on visual analytics, which is already explicitly concerned with user interaction and would be well served by the algorithmic lens.

The SIGSPATIAL community operates on the frontiers of algorithms research for spatially-related information. Combined with the multitude of hard problems it faces, this community is ideally suited to answer the research questions posed in this paper and, in doing so, advance the state of algorithmics.

ACKNOWLEDGMENTS

The authors acknowledge support from the German Research Foundation (DFG) within the priority programme 1894 "Volunteered Geographic Information" www.vgiscience.org (grant Di 2161/2-1).

REFERENCES

[1] J. J. Arsanjani, A. Zipf, P. Mooney, and M. Helbich. 2015. *OpenStreetMap in GIScience: Experiences, Research, and Applications.* Springer.
[2] M. G. Arteaga. 2013. Historical Map Polygon and Feature Extractor. In *Proc. 1st ACM SIGSPATIAL Intl. Workshop on MapInteraction.* 66–71.
[3] B. Budig, Th. C Van Dijk, and A. Wolff. 2016. Matching labels and markers in historical maps: an algorithm with interactive postprocessing. *ACM Trans. Spatial Algorit. Syst. (TSAS)* 2, 4 (2016), 13.
[4] B. Budig and Th. C. Van Dijk. 2015. Active learning for classifying template matches in historical maps. In *Int. Conf. Discovery Sci.* Springer, 33–47.
[5] B. Budig, Th. C van Dijk, F. Feitsch, and M. G. Arteaga. 2016. Polygon consensus: smart crowdsourcing for extracting building footprints from historical maps. In *Proc. 24th ACM SIGSPATIAL Int. Conf. Advances GIS.* ACM, 66.
[6] Y. Chen and A. Krause. 2013. Near-Optimal Batch Mode Active Learning and Adaptive Submodular Optimization. In *Proc. of the 30th Intl. Conf. on Machine Learning.* 160–168.
[7] Y.-Y. Chiang. 2015. Querying Historical Maps as a Unified, Structured, and Linked Spatiotemporal Source. In *Proc. 23rd ACM SIGSPATIAL Int. Conf. Advances in GIS (ACM SIGSPATIAL '15).* ACM, 270–273.
[8] C. Fleet, K. C. Kowal, and P. Pridal. 2012. Georeferencer: Crowdsourced Georeferencing for Map Library Collections. *D-Lib Magazine* 18, 11/12 (2012).
[9] M. F. Goodchild. 2007. Citizens as sensors: the world of volunteered geography. *GeoJournal* 69, 4 (2007), 211–221. https://doi.org/10.1007/s10708-007-9111-y
[10] Y. Guo and D. Schuurmans. 2007. Discriminative Batch Mode Active Learning. In *Proc. 21st Ann. Conf. Neural Information Proc. Syst.* 593–600.
[11] M. Haklay and P. Weber. 2008. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE* 7, 4 (2008), 12–18.
[12] W. Höhn, H.-G. Schmidt, and H. Schöneberg. 2013. Semiautomatic Recognition and Georeferencing of Places in Early Maps. In *Proc. 13th ACM/IEEE Joint Conf. Digital Libraries (JCDL'13).* 335–338.
[13] S. Hoi, R. Jin, J. Zhu, and M. Lyu. 2006. Batch Mode Active Learning and Its Application to Medical Image Classification. In *Proc. of the 23rd Intl. Conf. on Machine Learning.* 417–424.
[14] R. M. Karp. 2011. Understanding Science Through the Computational Lens. *J. Comp. Sci. Technol.* 26, 4 (2011), 569–577. https://doi.org/10.1007/s11390-011-1157-0
[15] G. Krempl, D. Kottke, and V. Lemaire. 2015. Optimised probabilistic active learning (OPAL). *Machine Learning* 100, 2-3 (2015), 449–476.
[16] S. Leyk, R. Boesch, and R. Weibel. 2006. Saliency and semantic processing: Extracting forest cover from historical topographic maps. *Pattern Recognition* 39, 5 (2006), 953–968.
[17] C. Lintott and J. Reed. 2013. Human Computation in Citizen Science. In *Handbook of Human Computation.* 153–162. https://doi.org/10.1007/978-1-4614-8806-4_14
[18] Ch. Papadimitriu. 2007. The Algorithmic Lens: How the Computational Perspective is Transforming the Sciences. In *Proc. Federated Computing Research Conf. CCC.*
[19] A. J. Quinn and B. B. Bederson. 2011. Human computation: a survey and taxonomy of a growing field. In *Proc. SIGCHI Conf. Human Factors Comput. Syst. (CHI'11).* ACM, 1403–1412.
[20] B. Settles. 2010. *Active Learning Literature Survey.* Computer Sciences Technical Report 1648. University of Wisconsin-Madison.
[21] D. Shahaf and E. Amir. 2007. Towards a Theory of AI Completeness. In *Proc. AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning.* 150–155.
[22] R. Simon, E. Barker, L. Isaksen, and P. de Soto Cañamares. 2015. Linking Early Geospatial Documents, One Place at a Time: Annotation of Geographic Documents with Recogito. *e-Perimetron* 10, 2 (2015), 49–59.
[23] L. Von Ahn. 2009. Human computation. In *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE.* IEEE, 418–419.
[24] H. Zhang, E. Horvitz, R. C. Miller, and D. C. Parkes. 2011. Crowdsourcing general computation. In *Proc. ACM Conf. Human Factors Comput. Syst. (CHI'11), Workshop on Crowdsourcing and Human Computation: Systems, Studies, and Platforms.*