

Experiences on Using TRAKLA2 to Teach Spatial Data Algorithms

Jussi Nikander¹, Juha Helminen² and Ari Korhonen³

*Department of Computer Science and Engineering
Helsinki University of Technology
Espoo, Finland*

Abstract

This paper reports on the results of a two year project in which visual algorithm simulation exercises were developed for a spatial data algorithms course. The success of the project is studied from several point of views, i.e., from developer's, teachers's, and student's perspective. The amount of work, learning outcomes, and feasibility of the system has been estimated based on the data gathered during the project. The results are encouraging, which motivates to extend the concept also for other courses in the future.

Keywords: Algorithms, algorithm visualization, TRAKLA2, spatial data algorithms, geoinformatics, teaching

1 Introduction

Spatial data algorithms (SDA) are algorithms that work on location data, such as geographic data. These algorithms are an integral part of geoinformatics, a branch of science where information technology is applied to cartography and geosciences. Since geoinformatics often uses illustrations, such as maps and other diagrams, the students are well acquainted with visualization. Thus, software visualization is a natural tool for teaching SDA and its applications to geoscientists.

There are many software systems that visualize spatial data algorithms or other concepts important for geoinformatics. For example GASP [19,20], GAWAIN [5] and Vega [6] illustrate several geometric algorithms while Hull2VD [3] and VoroGlide [7] provide visualizations for Voronoi diagrams [1]. Also, the LEDA algorithm library⁴ contains its own visualization tool GeoWin [2] that can be used for the visualization

¹ Email: jtn@cs.hut.fi

² Email: jhhelmi2@cs.hut.fi

³ Email: archie@cs.hut.fi

⁴ <http://www.algorithmic-solutions.com/leda/about/index.htm>

of spatial data algorithms. However, it seems that the visualization of algorithms important for geoinformatics is currently not a very active area of research. A recent paper on the MAVIS algorithm visualization tool as one spatial algorithm as a case example [8], but we could not find other recent progress in the field.

The existing systems typically offered the users two types of interaction. First, the users could view the animations with different input. Sometimes the input could be freely defined by the user. Furthermore, the systems supported the creation of new algorithm animations by implementing your own algorithms and visualizing them with the system. These interaction methods correspond to the viewing, changing and constructing levels of the engagement taxonomy [14].

Tracing exercises are a teaching method employed by many instructors in which the students trace an algorithm by keeping track of the changes in data structures while stepping through the algorithm. In *visual algorithm simulation exercises*, this procedure is supported by a graphical learning environment that provides visualizations, which eliminates the burden of drawing the same data structure over and over again. TRAKLA2 is a learning environment that utilizes visual algorithm simulation to deliver tracing exercises to students [13]. The system can automatically assess the solutions and give feedback on the correctness of the simulation.

There are two major differences between the visualizations included in TRAKLA2 and the previous systems. First, the TRAKLA2 exercises are designed for being used in teaching geoinformatics. Other systems, even if they contain for example geometric algorithms, seem to be designed for teaching computer science. Second, the tracing exercises offer a different type of interaction than provided by the other systems. The students construct their own algorithm animation sequences through visual interaction when they solve the exercises. This is in the constructing level of the engagement taxonomy [14]. However, unlike in the other systems, there is no need to write any algorithm code in order to create a new animation.

Previous studies [10,11] have shown that as long as the assignments are the same, there are no differences in the learning results between a test group that solves visual algorithm simulation exercises on the web and a control group that solves tracing exercises in a classroom. This is an encouraging result that motivated us to apply visual algorithm simulation exercises for spatial data algorithms. The challenge was to extend the TRAKLA2 system to cover a new area of algorithmics. The main research question was whether applying the visual algorithm simulation concept to new application areas is worth the effort spent on extending the framework. This question has three separate aspects: the developer's, teacher's and student's point of views. From the developer's perspective, we are trying to find out under what circumstances this kind of project pays off. In particular, how much time and effort does it take to extend the system to cover a new application area. And, what kind of challenges we expect to encounter during implementation and in the design of new visualizations? From the teacher's point of view, we are interested in the learning results: the level of learner engagement (i.e., how much work they did) and the overall performance in the final examination (i.e., the correlation between

the exercises and the achieved learning results). Finally, from the student's point of view, by interviewing them, we seek to find out how this new technology affects the learning process.

After the implementation (i.e., extending the application framework and implementing the exercises) the use of the system requires only minimal effort. Thus, this research aims not to answer whether somebody should repeat our experiment, but to answer whether the concept of visual algorithm simulation exercises is mature enough to be applied to other disciplines than data structures and algorithms. The main challenge is that it requires the instructor to be *proactive* rather than *reactive*, that is, the workload is much higher before the course than in traditional teaching in which the work (grading the exercises) is done during or after the course. This might explain the slow adoption of such systems in every day teaching.

In this paper, we report the results on our experiences from a two year project in which we implemented spatial TRAKLA2 exercises. Overall, taking into account that this was partially a research project and partially a course development project, it was a success. However, from a single instructor's point of view, two years appears to be too short a time to reap the benefits of automatic assessment, unless the system is used on a large course with hundreds of students. Furthermore, student feedback suggests that they want more visual algorithm simulation exercises on the course.

2 Spatial Extension to TRAKLA2

TRAKLA2 is a framework for automatically assessed visual algorithm simulation exercises [13]. The system is built on the concept of visual algorithm simulation. The user can construct animations of algorithm execution via GUI actions such as dragging and dropping data items. In the exercises, students can freely step backward and forward in these animations. They can also reset the exercise and get new randomly generated input. A model answer is also available as an animation. The exercises are deployed as Java applets within a web environment. Figure 1 shows a screenshot of the TRAKLA2 environment.

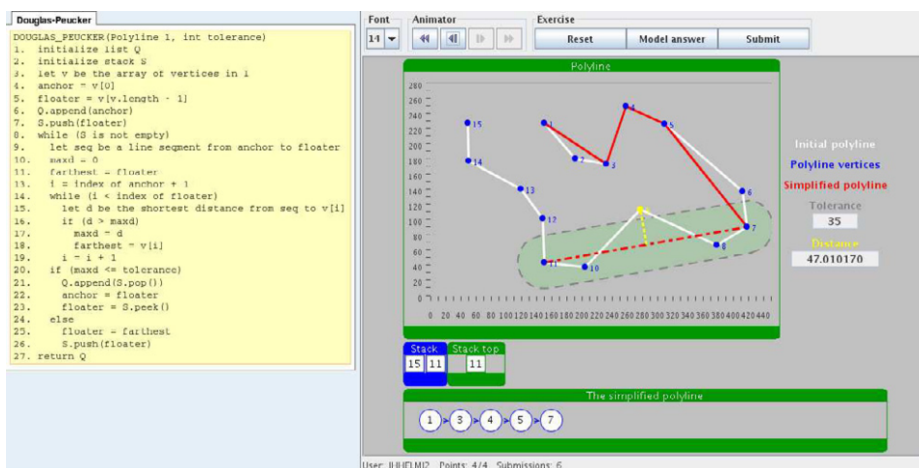


Fig. 1. TRAKLA2 exercise for the Douglas-Peucker line simplification.

The SDA extension to TRAKLA2 consists of three major components: spatial primitives, spatial data structures and spatial data visualization. In addition, implementations of geometric functions and algorithms for processing spatial data and generating random input for the algorithms are required. For now, the extension is limited to two-dimensional data and exercises.

The primitives on which the spatial data structures and algorithms operate are geometric entities, such as points, lines, and polygons. This multidimensional data can be stored in the form of key values derived from their geometric properties. In this respect, a spatial data element is an ordered list of values, a tuple, which in the spatial context represents a specific geometric entity. For example, a polygon can be stored as an ordered list of the coordinates of its vertices.

In basic TRAKLA2 exercises, the data is 1-dimensional, single-letter characters and integers. Their relationships are simple, and obvious without any additional visualization. The SDA extension introduces two new visualizations for use with spatial data. First, the *tuple representation* of a spatial primitive can be visualized in simple and exact tabular form. Despite being precise, this data-intensive approach does a poor job of conveying the geometric nature of and the relationships between the data elements. Thus, a different visualization is needed to illustrate the spatial attributes. This is the *area visualization*, which is fundamentally a two-dimensional coordinate plane, onto which the geometric entities of the spatial primitives are drawn [15].

Furthermore, spatial data is represented at three different levels of visualization. First, at the exact data item level, the data is shown as tuples of values based on their geometric properties. Second, at the data structure level, we have canonical visualizations specific to the data structures, where spatial data items are labeled with unique identifiers or shown as tuples. At the highest level of abstraction, the conceptual relationships and spatial attributes are illustrated with the area visualization showing the data items and the structure with possibly additional visual cues to represent algorithm constructs, such as a sweep line or an in-circle test. Figure 2 depicts two visualizations of the same R-tree containing polygons. The area visualization shows the areas covered by the polygons and tree nodes. The tree visualization shows how the tree is organized.

A particular challenge with the SDA exercises is the generation of spatial data for use as input to the algorithms. For the visualization to contribute to learning, the generated data sets must be clear and visually pleasing. There has to be enough spacing between the geometric entities to distinguish and select them. Also, labels must not overlap too much. In addition, each exercise has very specific constraints for the data, which makes it hard to create a generic data source. Therefore, most exercises have their own mechanism for generating data such as simple polygons.

Currently, the SDA extension comprises 12 visual algorithm simulation exercises. All of the exercises can be found on our web site⁵. The exercises fall into two categories: *tracing exercises* and *open tracing exercises* [9]. In tracing exercises, an

⁵ <http://www.cs.hut.fi/Research/TRAKLA2/exercises/index.shtml>

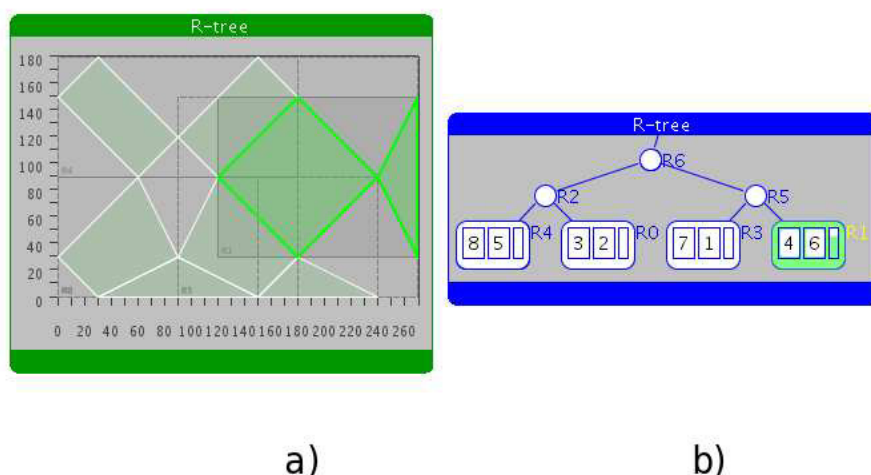


Fig. 2. R-tree a) drawn as an area and b) drawn as a tree.

algorithm and its input is given, and the student's task is to work out the output and construct an animation of the algorithm's progress by simulating it step-by-step. The simulation is done by emulating the algorithm's operations by dragging, dropping and selecting data items, as well as invoking operations via other GUI components, such as buttons. The user-created animation is then compared against a correct sequence of states created by an actual execution of the algorithm, and graded based on the number of matching steps. The goal is to give a conceptual understanding of the algorithm.

In open tracing exercises, the algorithm is not strictly specified and therefore the exercises are more exploratory in nature. The student is given a goal, such as creating the Delaunay triangulation [16] of a point set, and the means to achieve it, *e.g.*, adding edges between vertices and to carry out the in-circle test. The student can then interactively explore the structures by making modifications to them and observing the changes. Finally, the correctness is assessed by comparing the final state with the expected outcome. The overall goal is not to teach some specific algorithm but a concept, such as the min-max-angle criterion related to Delaunay triangulations.

3 Results

The spatial TRAKLA2 exercises were first adopted in the spring 2007 spatial data algorithms course and were used again on the spring 2008 course. The course is aimed at third year geoinformatics students. In addition to spatial exercises, some other TRAKLA2 exercises were also included on the course. These exercises covered data structures important for the understanding of spatial algorithms. The details of the course are shown in Table 1. The table tells how many students started the course, how many participated in the final examination, how many TRAKLA2

exercises there were on the course (and how many of those were spatial exercises), the total number of TRAKLA2 submissions, and average score gained (compared to maximum).

Table 1
Basic course data for the Spatial Data Algorithm course

year	# students	# in exam	# exer. (SDA)	# subs	avg. score
2007	16	9	15 (9 SDA)	723	67%
2008	20	16	16 (10 SDA)	1036	83%

Before TRAKLA2 was introduced the spatial data algorithms course consisted of combined lecture and studio sessions, a programming project and an exam. In the studio sessions the students worked in groups and studied spatial algorithms on a conceptual level. In the programming project the students implemented one of the algorithms discussed on the course. The exam was held after the last lecture. TRAKLA2 was added without reducing any other requirements on the course.

In the course, each student got personalized input for all TRAKLA2 exercises. The exercises were divided into rounds with 1–3 exercises in each. In order to pass the course, the students needed to gain at least 50% of the points from each round. Students were not penalized for returning exercises late. The TRAKLA2 exercises did not affect the students’ final course grade.

3.1 Developer’s Point of View

The project for creating the TRAKLA2 spatial extension was started in February 2006. The first exercises were introduced in January 2007. At that time, 9 spatial exercises were used in the course. In the project, a total of 12 spatial data exercises have been implemented, and 11 of them have been used in practice. The one untested exercise was finished so late that it could not be added to the spring 2008 course. Several people participated in the project, but most of the time there were two people working on it. A crude approximation of the amount of work put into the project is 10 person–months.

The implementation work itself can be divided into two separate tasks: extending the exercise framework and implementing the exercises. Most of the work in the project went into the design and implementation of the exercises. Less than 20% of the total effort consisted of extending the framework. Based on our experience, the implementation of a spatial exercise typically was more time–consuming than basic data structure or algorithm exercise.

3.2 Teacher’s Point of View

Data on the students’ learning results was collected from TRAKLA2 exercises and the course exam. TRAKLA2 kept record of each student’s final points and number of submissions to each exercise. The data was collected the same way in both years.

Linear regression was used to see if the students’ TRAKLA2 performance were a good indicator of their exam results. The analysis was made both between

TRAKLA2 results and exam results as a whole, as well as between TRAKLA2 results and a single exam question that covered R-trees [4]. In both years, there were two TRAKLA2 exercises and an exam question about R-trees. Summary of the results can be found in Table 2.

The results in Table 2 are divided into three categories. First, Course info indicates the number of students who passed the TRAKLA2 exercises and participated in the first final exam. The second category contains characteristics of linear regression analysis between TRAKLA2 results and exam results as whole, and the third category regression results for the R-tree exercises. For the linear regression ρ (correlation), adjusted R^2 , and its statistical significance are reported.

Table 2
Learning results on the spatial data algorithms course

Course info		Whole exam			R-trees		
Year	N	ρ	adj. R^2	p	ρ	adj. R^2	p
2007	9	0,83	0,65	0,005	0,85	0,69	0,003
2008	16	0,48	0,18	0,058	0,60	0,31	0,015

There was a significant change in the TRAKLA2 results between the years 2007 and 2008. The change, however, can be explained by the modifications made to TRAKLA2 exercises between the two courses. One exercise was removed, and two new ones were added. The exercise removed after 2007 course was one of the hardest exercises (average score 54%), while the exercises added to 2008 course were among the easiest (average score 98%). The exam results were similar in both years.

As can be seen in Table 2, in both years, there was a strong correlation between students' performance in TRAKLA2 and in the course exam, especially for the R-tree exercises. All results were statistically significant ($p < 0,05$) except the correlation between TRAKLA2 results and exam results in 2008, which was almost significant ($p = 0,058$). Similar results have been observed in data structures and algorithms courses [10].

3.3 Student's Point of View

In spring 2008, we carried out interviews to learn about student experiences with the spatial exercises. We used the interview guide approach [17], where the interviewer has an outline of topics to be covered, but may vary the wording and order of the questions to some extent. We interviewed a total of 4 students (two males, and two females) with two different nationalities, thus two of the interviews were in Finnish and two in English. The age (22 to 28 years) and background of the interviewees varied as they had had their education in different countries and universities/high schools.

Two main paths of questioning were explored: what was the student's subjective opinion of the system and what did they think about it compared to other teaching methods and learning materials. The main questions are listed in Table 3. For each question, we also had a set of follow-up questions that expanded on the subject to

help us in getting more informative responses.

Table 3
Main interview questions.

Opinion
How useful do you think the system is?
How well do you think the system contributes to learning?
How useful did you find the feedback given by the system?
How did you find the usability of the system?
Should we in the future continue to use the system on the course?
Comparison
How would you compare TRAKLA2 spatial algorithm exercises to other teaching and learning methods and materials, such as lectures, classroom exercises and individually studying from a textbook?

All interviewees found the system to be beneficial and also thought it was an important learning tool that should continue to be utilized on the course. Furthermore, they felt that compared with the lectures, they had learned more details about the algorithms from the TRAKLA2 exercises, but lectures were still considered to be the most important learning method. Compared with reading algorithms from articles, they felt that it was faster to grasp the idea from TRAKLA2 exercises. Yet, before attempting the exercises, students thought they should have some basic knowledge of the algorithm first.

The main benefits mentioned were the visual appearance and interactivity. Students felt that they were able to better make sense of an algorithm’s principles by observing animations of them. Indeed, all said that the model answer animation was very helpful. However, it was unclear from the responses whether they truly thought that it aided in learning the algorithm or that it simply helped them to successfully solve the exercise. Similarly, the students found that the simulation aspect of the system, which allowed them to actually practice the algorithm, makes it easier to memorize the algorithm’s principles. One student pointed out that by observing and manipulating a visualization, you can actually see how the algorithm progresses, unlike in a programming exercise in which you need to implement the algorithm, but typically cannot observe its execution very well.

The issues with the system were related to the automatic feedback and exercise-specific simulation interfaces. When an incorrect solution is submitted, the system replies with the number of correct steps from the beginning of the animation. All felt that while you may this way find the first error, searching for it by stepping through the long model answer animation is cumbersome. In addition, students thought it was unfair that the system does not give any points for the correct execution of the algorithm after the first mistake, and that you cannot continue

to solve the exercise from where you made the mistake, leaving you to trace the algorithm over and over from the beginning. Moreover, they complained about the mappings between algorithm operations and simulation interface actions such as pushing buttons and dragging data items. According to the students, this exercise-specific behaviour was not documented well enough and as one student phrased it, it took some effort to learn how to get the things move in the way they are supposed to. The pseudocode included in every TRAKLA2 exercise was considered either completely useless or very useful depending on the interviewee's familiarity with programming. A student having strong programming skills found the pseudocode useful, while less skilled students did not pay attention to it.

4 Discussion

4.1 Developer's Point of View

In this project, the design and implementation of spatial data exercises was found to take significantly more time and effort than most of the data structure exercises done prior to this experiment. The most important factor was that spatial algorithms were mostly unfamiliar to the development team in the beginning of the project. Thus, for each exercise, the algorithm in question needed to be studied in order to comprehend it on a level required to implement, visualize and teach it. This is a time-consuming task. In addition, the use of more complex visualizations increased the amount of required effort. For example, the use of area visualizations is not as straight-forward as using basic data structure level visualizations. The area view is flexible, but specifying how to visualize data using it is more time-consuming than when using canonical data structure views.

Two years and 10 person-months for creating just 12 exercises may seem to be a lot effort for quite a little gain, at least from a single instructor's point of view. However, once the exercises have been implemented, using them on a course requires only a very small, constant amount of effort, regardless of the size of the course or the number of exercises used on it. This proactive approach is in direct opposition to the traditional reactive approach of manually assessed classroom exercises. In manual assessment, most effort goes into assessing the students' answers, and it is proportional to the number of exercises and the size of the course. In addition, the comparison is not straight-forward as this number would be even greater if resubmissions could be allowed (that is typically not the case, due to the fact that it increases logistical problems and work load too much). Furthermore, manual assessment needs to be done on each iteration of the course. Thus, the longer the automated exercises are in use, the more benefit they offer. Eventually, automated exercises will require less overall effort than traditional classroom exercises, since the job needs to be done only once.

The time it takes for this to happen depends on the number of exercises and students on the course. In large courses (many students), this time limit is reached very soon (in a couple of years). For example, in a data structures and algorithms course, some 500 students make some 50.000 submissions with 40 exercises (ap-

proximately 2.5 resubmissions/exercise). Thus, it is easy to see that the system pays off very soon in this case (it is hard to find enough personnel to grade this amount of submissions within feasible time limits, not to mention that the work is not very pleasant). However, it is more difficult to define a precise time limit for smaller courses such as spatial data algorithms. One thousand submissions (in 2008) is quite an easy task to handle even by a single instructor and takes probably only a couple of days to grade manually. Even then, we believe the investment is worthwhile if we take also the teacher's and student's point of views into account.

4.2 Teacher's Point of View

The learning results show that the correlation between TRAKLA2 exercises and exams on the spatial data algorithms course is strong and statistically significant. It is even stronger than the correlation in the basic data structures course [10]. However, this is likely to be an artefact of the small sample size on the SDA course. With the smallest sample being only 9 persons, the results of a single person are likely to affect the overall correlation quite a lot. Despite this, it seems that TRAKLA2 exercise results are a good indicator of exam results.

One interesting aspect of the TRAKLA2 results is how much effort students put into them. On the data structures and algorithms course, where TRAKLA2 exercises directly affect the course grade, a large portion of the students get maximum points, even when no further benefit is gained after getting 90% of the points [12]. On the spatial data course, TRAKLA2 points have no effect on the course grade. Nonetheless, students seem to want to do as many TRAKLA2 exercises as they can. Even if this is nowhere near the amount of effort students use on the data structures and algorithms course, this indicates that most students are willing to do more work than required. This supports the opinion that this kind of exercises are not only well accepted by the students but also motivate students to do more work than in traditional teaching setups.

4.3 Student's Point of View

The results from the interviews indicate that the students feel that they benefit from the system, even if this is not obvious from the quantitative results. In fact, the overall attitude towards the system was very positive and when asked to give it a grade from 0 to 5, all interviewees rated the system at around 4. During the course of the interviews, most interviewees also asked for more exercises that could cover other algorithms discussed in the course. Nonetheless, lectures were still considered the most important teaching method while the exercises should be in a supporting role.

In contrast to the majority opinion one student, who had a more advanced background in computer science than the others, felt that the exercises were not a terribly important learning tool for him. Yet, even he said that one or two of the exercises had actually straightened out existing misconceptions he had about the algorithms in question. That is, via the visual algorithm simulation, he was able

to learn some details of the algorithms and adjust his incorrect pre-existing mental model of them without delving into implementation.

The interviewees also brought up the notion that this kind of more involved learning with interactive visualizations would lead to a more lasting effect than simply reading. Essentially, they felt that the exercises engaged them more effectively, and they would remember the lessons longer. Thus, the student opinions agree with the findings of [14].

5 Conclusions

In this paper we have described the experiences we had implementing and using the TRAKLA2 system on a spatial data algorithm course. The implementation of the system was a more challenging task than originally anticipated. One significant contributor to the large amount of effort required was that we were not very familiar with the topic. The effect of the SDA exercises on student learning is similar to the effect of TRAKLA2 exercises on a basic data structures course. Furthermore, the students' attitude to the system seems to be generally positive and they believe that the system helps them to learn.

From the student's point of view, the weaknesses of the system were the quite minimal feedback on incorrect solutions and complex exercise-specific simulation interfaces. Improving the quality of feedback is currently a topic of ongoing research [18]. The difficulty of designing intuitive interfaces for visual algorithm simulation is something that turned out to be more challenging than expected based on previous experience. The mapping of complex mathematical operations to simulation interface actions for the purpose of creating automatically assessed algorithm exercises is also open to more research.

Despite the amount of effort required for creating the exercises, we consider the project a success. Visual algorithm simulation can be used for teaching topics besides basic data structures and algorithms. Furthermore, after the initial work put into the implementation, the system can be used effortlessly. The longer it is in use, the more effort it saves.

References

- [1] Aurenhammer, F., *Voronoi diagrams - a survey of a fundamental geometric data structure*, ACM Comput. Surv. **23** (1991), pp. 345–405.
- [2] Bäsken, M. and S. Näher, *Geowin a generic tool for interactive visualization of geometric algorithms*, in: S. Diel, editor, *Software Visualization*, Lecture Notes in Computer Science **2269**, Springer, 2002 pp. 637–639.
- [3] Fisher, J., *Visualizing the connection among convex hull, voronoi diagram and delaunay triangulation*, in: *37th Midwest Instruction and Computing Symposium*, 2004.
- [4] Guttman, A., *R-trees: a dynamic index structure for spatial searching*, in: *SIGMOD '84: Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (1984), pp. 47–57.
- [5] Hausner, A. and D. P. Dobkin, *Gawain: visualizing geometric algorithms with web-based animation*, in: *SCG '98: Proceedings of the fourteenth annual symposium on Computational geometry* (1998), pp. 411–412.

- [6] Hipke, C. A. and S. Schuierer, *Vega – a user-centered approach to the distributed visualization of geometric algorithms*, Technical report, University of Freiburg (1998).
- [7] Icking, C., R. Klein, P. Kollner and L. Ma, *Java applets for the dynamic visualization of voronoi diagrams*, <http://www.pi6.fernuni-hagen.de/publ/ikkm-jadvv-03.pdf>.
- [8] Koifman, I., I. Shimshoni and A. Tal, *Mavis: A multi-level algorithm visualization system within a collaborative distance learning environment*, Journal of Visual Languages & Computing **19** (2008), pp. 182–202.
- [9] Korhonen, A. and L. Malmi, *Taxonomy of visual algorithm simulation exercises*, in: A. Korhonen, editor, *Proceedings of the Third Program Visualization Workshop*, The University of Warwick, UK, 2004, pp. 118–125.
- [10] Korhonen, A., L. Malmi, P. Myllyselkä and P. Scheinin, *Does it make a difference if students exercise on the web or in the classroom?*, in: *Proceedings of The 7th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, ITiCSE'02* (2002), pp. 121–124.
- [11] Laakso, M.-J., T. Salakoski and A. Korhonen, *The feasibility of automatic assessment and feedback*, in: *Proceedings of Cognition and Exploratory Learning in Digital Age (CELDA 2005)*, IEEE Technical Committee on Learning Technology and Japanese Society of Information and Systems in Education, Porto, Portugal, 2005, pp. 113–122.
- [12] Malmi, L., V. Karavirta, A. Korhonen and J. Nikander, *Experiences on automatically assessed algorithm simulation exercises with different resubmission policies*, Journal of Educational Resources in Computing **5** (2005).
- [13] Malmi, L., V. Karavirta, A. Korhonen, J. Nikander, O. Seppälä and P. Silvasti, *Visual algorithm simulation exercise system with automatic assessment: TRAKLA2*, Informatics in Education **3** (2004), pp. 267–288.
- [14] Naps, T. L., G. Rößling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodgers and J. Ángel Velázquez-Iturbide, *Exploring the role of visualization and engagement in computer science education*, SIGCSE Bulletin **35** (2003), pp. 131–152.
- [15] Nikander, J. and J. Helminen, *Algorithm visualization in teaching spatial data algorithms*, in: *11th International Conference Information Visualization IV2007* (2007), pp. 505–510.
URL <http://www.graphicslink.co.uk/IV07/>
- [16] Okabe, A., B. Boots, K. Sugihara and S. N. Chiu, “Spatial Tessellations: Concepts and Applications of Voronoi Diagrams,” John Wiley & Sons, 2000.
- [17] Patton, M., “Qualitative Research and Evaluation Methods,” Sage Publications, 2002.
- [18] Seppälä, O., L. Malmi and A. Korhonen, *Observations on student errors in algorithm simulation exercises*, in: *Proceedings of the 5th Annual Finnish / Baltic Sea Conference on Computer Science Education*, University of Joensuu, 2005, pp. 81–86.
- [19] Shneerson, M. and A. Tal, *Interactive collaborative visualization environment for geometric computing*, Journal of Visual Languages & Computing **11** (2000), pp. 615–637.
- [20] Tal, A. and D. Dobkin, *Gasp-a system for visualizing geometric algorithms*, Visualization, 1994., Visualization '94, Proceedings., IEEE Conference on (1994), pp. 149–155, CP16.