Project number IST-25582

CGL

Computational Geometric Learning

D2.2: Work Package 2 [Period 1] Report

STREP

Information Society Technologies

Period covered:	November 1, 2010–October 31, 2011
Date of preparation:	October 30, 2011
Date of revision:	
Start date of project:	November 1, 2010
Duration:	3 years
Project coordinator name:	Joachim Giesen (FSU)
Project coordinator organisation:	Friedrich-Schiller-Universität Jena Jena, Germany

In the following we describe the work done within Work Package 2 within the first period. In the description we follow the structure imposed by the tasks for this work package and period. Wherever there is no deviation and all goals have been met, it is *not* mentioned specifically. We start by restating the objectives for Work Package 2 and conclude with a discussion of the milestones for Period 1.

Objectives

Computational Geometry has been very successful in the past 25 years in developing methods for problems in low dimensions, and CGAL has managed to provide a unified software framework for implementing such methods.

Within this work package, fundamental *high-dimensional* data structures and algorithms will be developed, in most cases with the goal of implementing them in CGAL. High-dimensional geometric data processing is now in a situation that we have encountered fifteen years ago in the low-dimensional setting: a number of useful theoretical concepts has been developed over the past years, but robust and efficient implementations are mostly lacking. In order to have an impact in practice, such implementations must scale with the dimension and exploit (hidden) structure of the data.

In analyzing high-dimensional data structures and algorithms, our focus is not on worst-case complexity, but on (provably) good performance under some given structural properties of the input. These properties may be of statistical nature (when we are dealing with noise, for example), or of geometric nature (when data is of low intrinsic dimension, say). Related to this, we are also aiming at output-sensitive algorithms.

Tasks

Task 1.a: Derivative free methods

We have proposed the *Random Pursuit* algorithm for derivative-free minimization of convex functions. In each step, the algorithm samples a direction at random, and then proceeds to the line search optimum in this direction. We conducted a theoretical convergence analysis of the algorithm under approximate line search, and we underpinned the theoretical results with ample experimental data. A journal version of the article [10] is currently under review. There is one distinctive feature of Random Pursuit that makes it relevant in the context of this project: unlike for example gradient descent, the Random Pursuit algorithm is invariant under strictly monotone transformations of the objective function, and this allows us to deal with certain types of non-convex functions as well.

We have successfully applied the adaptive stepsize method CMA-ES in a theoretical setting; the method was instrumental in refuting a conjecture of Häme et al. on properties of Euclidean traveling salesman tours [9]. This result further motivates us to find theoretical guarantees for CMA-ES and related methods in the next periods.

Task 1.b: Predicates and primitives

There exists currently a wide range of available algebraic tools for geometric predicates, such as Orientation. We considered the linear algebra libraries **LinBox** and **Eigen** in different geometric scenarios. LinBox is suitable for very high dimensions (typically > 100), targets state-of-the-art asymptotic complexity, and introduces a big overhead in medium dimensions. Eigen seems to be suitable for medium dimensions, whereas CGAL provides efficient predicates in lower dimensions. We considered basic geometric algorithms, e.g. for convex hull and triangulations, which reduce to a series of determinants, where the encountered matrices have many rows (or columns) in common. By hashing the computed minors, we reduced the overall runtime, especially in computing regular triangulations. The exponential memory consumption is overcome by cleaning up the hash table (with some time penalty). We successfully employed this technique in computing projections of resultant polytopes (Task 3.a). The work appears in [6], and the package may be submitted to CGAL.

No deviations but one comment. One deliverable of WP2 is the modification of the CGAL d-dimensional geometric kernel, in order to improve the computation of geometric predicates by implementing better linear algebra algorithms and interfacing with linear algebra libraries. Since the kernel is being redesigned by the CGAL community, the current version will soon be obsolete and our contribution to the kernel is being withheld. However, NKUA participates in the discussion within CGAL, with the aim of including our contributions in the final kernel.

Task 1.c: Adaptive data structures

We increase the query performance of data structures by exploiting nonrandom temporal patterns of the queries. We develop adaptive data structures for planar point location based on our recent work [1]. Furthermore, we study data structures for answering efficiently approximate nearest-neighbor (ANN) point queries over parallel segments in 3 dimensions. This can be extended to queries over points lying on a few lines.

Task 1.d: Machine learning techniques

Over the last ten years, Support Vector Machines (SVMs) have become one of the standard tools in Machine Learning for classification and regression. Most SVM solvers solve the optimization problem which is cast by an SVM by approximating the dual optimization problem. We have provided an implementation *fastSVM* that solves the original optimization problem in the primal. Doing so, we have more flexibility in modeling problems (not only kernels but also basis functions) as compared to the original SVM formulation, and we can still provide the same running time as state of the art SVM solvers.

Task 2.b: Compact representation of cell complexes

We have designed and implemented a parallel algorithm to compute the Hasse diagram underlying the flow complex. The Hasse diagram encodes the complete combinatorial information of the flow complex and can be augmented by some geometric information, especially information for topological multi-scale analysis.

It already turns out that the algorithm does not scale to dimensions substantially larger than ten even when using a massively parallel architecture like a compute cluster. Nevertheless, in combination with dimension reduction techniques this seems still an interesting range of dimensions also from an applications point of view. Another idea to scale the flow complex to dimensions larger than ten is to compute only partial information, i.e., only high index critical points which often provide enough information about the underlying data.

Instead of the planned twelve person months we spent only six months since we still had university funding available that we wanted to spent first. The underspending has not caused any delay in our goals for this task. Also, it is planned to compensate for the underspending in the next reporting periods.

Task 2.c: Delaunay type complexes

Witness complexes are related to Delaunay complexes but are much easier to compute in high dimensions. We studied witness complexes, established conditions under which witness complexes and Delaunay complexes are identical [2]. We also implemented a new data structure to represent witness complexes.

Task 3.a: Silhouettes of polytopes

We have designed an incremental, output-sensitive algorithm to compute the "resultant polytope", defined as Minkowski summand of the secondary polytope of a point set A, or to compute its silhouette along a given direction. Our algorithm extracts this substructure of the secondary polytope while computing a minimum number of secondary vertices. Formally, let A lie in a $(2n)$-dimensional space, and its points be partitioned in $n+1$ sets, used to define an equivalence relation on the secondary vertices. The resultant polytope's vertices represent the classes of this relation; we aim at computing projected resultant polytopes in dimension 4 to 7, while the secondary polytope lies in dimension > 10. A specific application is in implicitizing parametric (hyper)surfaces, where we obtain the Newton polytope of the implicit equation [7]. The algorithm is being implemented in C++ using CGAL, applies the software on extreme points (Task 3.b), and outputs both V- and H-representations as well as a triangulation of the resultant polytope. One bottleneck that we overcame was due to a sequence of Orientation predicates on similar matrices: we exploit geometric information to avoid redundant computation by hashing certain minors, thus reducing total runtime by more than half (see Task 1.b). Our results appear in [5].

Some overspending: seven funded person-months, three unfunded. There is an extra two funded person-months compared to Annex 1, because this task has been our main focus, and we happened to have experienced collaborators available. We have made progress beyond what was originally expected.

We shall reduce the Person-months in this task in the future, since the remaining work is less than what was initially anticipated for the next two years.

Task 3.b: Extreme Points

We have implemented the state-of-the-art extreme points algorithm by Dula and Helgason [4] in the form of a CGAL package (to be integrated in improved form into the library during a later period).

Task 3.c: Restricted Delaunay triangulations and variants

We have proposed and studied a new simplicial complex, the so-called tangential Delaunay complex. Using this data structure, we have proposed the first algorithm that can reconstruct a submanifold in time that is only linear in the ambient dimension (previous algorithms had an exponential dependence on the ambient dimension). The full version of the paper, including a proof that the complex output by the algorithm is a good approximation of the sampled manifold, is now available as [3].

Milestones

MS6: Tools needed to analyze CMA-ES

The tools currently available to us are not sufficient to compute convergence rates of CMA-ES on convex functions. This more generally holds for all known general methods that are trying to learn the "shape" of the function to be optimized. As a consequence, we are enhancing our arsenal of tools, taking into account very recent promising approaches from the optimization community [8]. Currently, we believe that CMA-ES might turn out to be too difficult to analyze, but that other similar adaptive approaches can successfully be handled.

MS7: Compact Representation of Delaunay and Rips Complexes

A CGAL package for Delaunay triangulations in any dimension is under preparation. The triangulation is represented by its simplices of full dimension and their adjacency graph. A more compact representation storing only the 1-skeleton (edges and vertices) will be released in a second step. For simplicial complexes of small dimensions embedded in high dimensional spaces, we decided to use a representation that stores all the faces exactly once in a prefix tree.

Bibliography

[1] S. Arya, T. Malamatos, D. Mount, and K. Wong. Optimal expected-case planar point location. *SIAM J. Comput.*, 37, 2007.

[2] Jean-Daniel Boissonnat, Ramsay Dyer, Arijit Ghosh, and Steve Oudot. Equating witness and restricted Delaunay complexes. Research Report CGL-TR-24, INRIA Sophia-Antipolis, 2011.

[3] Jean-Daniel Boissonnat and Arijit Ghosh. Manifold reconstruction using tangential Delaunay complexes. Research Report CGL-TR-15, INRIA Sophia-Antipolis, 2011. Submitted to Discrete and Computational Geometry.

[4] J.H. Dula and RV Helgason. A new procedure for identifying the frame of the convex hull of a finite collection of points in multidimensional space. *European Journal of Operational Research*, 92(2):352–367, 1996.

[5] I.Z. Emiris, V. Fisikopoulos, and C. Konaxis. An output-sensitive algorithm for computing projections of resultant polytopes. Technical Report CGL-TR-08, NKUA, 2011.

[6] I.Z. Emiris, V. Fisikopoulos, and L. Pearanda. Optimizing the computation of sequences of determinantal predicates. Technical Report CGL-TR-14, NKUA, 2011.

[7] I.Z. Emiris, T. Kalinka, and C. Konaxis. Implicitization using predicted support. In *Proc. Intern. Works. Symbolic-Numeric Computation*, San Jose, Calif., 2011.

[8] D. Leventhal and A. S. Lewis. Randomized hessian estimation and directional search. *Optimization*, 60:329–345, 2011.

[9] C. L. Müller. Finding maximizing euclidean TSP tours for the häme-hyytiä-hakula conjecture. Technical Report CGL-TR-13, ETHZ, 2011.

[10] S. U. Stich, C. L. Müller, and B. Gärtner. Optimization of convex functions with Random Pursuit. Technical Report CGL-TR-09, ETH Zürich, 2011.