MADALGO
CENTER FOR MASSIVE DATA ALGORITHMICS

madalgo
CENTER FOR MASSIVE DATA ALGORITHMICS
Summer School on Data Stream Algorithms
(Efficient) Robust Geometric Computing

ANNUAL REPORT 2007

CENTER FOR MASSIVE DATA ALGORITHMICS

MADALGO - a Center of the Danish National Research Foundation / Department of Computer Science / University of Aarhus

2007 Highlights

Research team

Center for Massive Data Algorithmics (MADALGO – see www.madalgo.au.dk) was established on March 1, 2007. At the end of 2007 the center research team consisted of 6 senior researchers (2 at AU), 4 post docs (2 at AU) and 13 PhD students (6 at AU). Additionally, one further Post Doc and two PhD students (who obtained their degrees in July) were part of the center in 2007. All center Post Docs are internationals and so is a good deal of the PhD students.

Research collaboration and results

Although a buildup year, MADALGO researchers have published 21 peer reviewed research papers within the center research areas in 2007. Several of these papers have appeared in highly ranked journals and conference proceedings. Some of the results in the papers have been obtained with the many international researchers that have visited MADALGO in 2007. The center has also had extensive multidisciplinary and industry collaboration, mainly on issues in connection with massive terrain data.

Modern airborne laser scanning technology is capable of acquiring very accurate terrain data at meter resolution. Such a scan of Denmark can easily occupy over a TeraByte (1000 Gigabytes).

One 2007 center paper describes the TerraSTREAM software package developed at MADALGO, which can e.g. be used to model water flow (and flooding) on truly massive terrain datasets.

Kort: DDH, Copyright: COWI A/S

Center events and publicity

Apart from a large number of smaller research seminars and workshops, as well as a retreat for center employees, MADALGO hosted two major events in 2007, namely an international summer school and an inauguration event. The center has also received quite a lot of media attention, appearing in more than 20 newspaper/magazine articles and radio/tv features (see www.madalgo.au.dk).

On Friday August 24 the inauguration of MADALGO was celebrated. The inauguration event included half-hour scientific talks by highly recognized international researchers in the core center research areas.

Just preceding the inauguration the center hosted a four day international Summer School on Data Stream Algorithms, where 5 leading international experts lectured for 70 participants (mainly PhD students) representing 21 nationalities.

Awards and acknowledgments

The senior center researchers have received a number of awards and acknowledgments in 2007. Demaine and Indyk received tenure at Massachusetts Institute of Technology and Meyer accepted a Chair (full professor position) in Algorithm engineering at Frankfurt University. Furthermore, Demaine received a honorary degree from Dalhousie University and Arge was elected chair of the steering committee for the European Symposium on Algorithms.

Content

This report describes the 2007 activities at the Danish National Research Foundation *Center for Massive Data Algorithmics*. The outline of the report more or less follows the agenda of the annual review meeting for the center held on March 14, 2008 (except that external funding and teaching is only covered in the appendices). The report is accompanied by a number of appendices (covering external relations; conference organization and participation; educational activities; academic service; external funding; awards and acknowledgments; public outreach; patents and applications; list of publications) as specified by the foundation; note that *some* of these appendices only cover the employees at University of Aarhus (and not the participants at Max Planck Institute for Informatics, Massachusetts Institute of Technology and Frankfurt University). Finally, note that the 2007 accounts for the center with appendices (especially the list of personnel) are also important in order to obtain a complete overview of the 2007 activities of the center.

Center director statement

By signature it is confirmed that the annual report and accounts/budget with accompanying notes and appendices contain all relevant information regarding the annual primary activities in the Danish National Research Foundation Center for Massive Data Algorithmics.

15 *April, 2008* Lars Arge

Lars Arge
Center Director

1 Center background

As discussed in the research plan for *Center for Massive Data Algorithmics* (MADALGO), the pervasive use of computers, as well as tremendous advances in the ability to acquire, store and process data, has resulted in a spectacular increase in the amount of data we collect. Our society is becoming increasingly ”data driven”; we increasingly expect to be able to access and process *massive datasets* anywhere at any time, and commercial and scientific applications are increasingly processing massive amounts of data.

As also discussed in the research plan, *algorithmics* – the design and analysis of algorithms – has always been a core area of computer science. Unfortunately, traditional algorithms theory is not adequate in many modern applications. One main reason for this is that in traditional theory, computation has been viewed as a simple process of transforming given input data into a desired output using a *well-defined and simple* machine consisting of a processor and an (infinite sized) memory. This scenario is not realistic in modern applications where computation is often being performed on very diverse devices that often have limited memory and computation power. In particular, the simplistic machine model does not take the hierarchical memory organization of modern machines into account. The memory system of a typical computer system is made up of several levels of cache, main memory and disk, where the access times of the different levels can vary by several orders of magnitude, and where data is transferred between the levels in large contiguous blocks. This means that often (especially when data reside on disk) it is much more important to minimize the number of block transfers than it is to minimize CPU computation (as is done in the simplistic models used in traditional algorithmics). Thus, the inadequacy of the simplistic machine models often translates into software inadequacies when processing massive data. Although the algorithms community has begun to address the inadequacies of traditional theory, and despite significant progress, current algorithmics theory is not adequate in many modern massive dataset applications.

MADALGO strives to become a world-leading center in algorithms for handling massive data; here massive is interpreted broadly to cover computations where the data is large compared to the resources of the computational devise. As outlined in the research plan, the high-level objectives of the center are:

- To significantly advance the fundamental algorithms knowledge in the area of efficient processing of massive datasets
- To train the next generation of researchers in a world-leading and international environment
- To be a catalyst for multidisciplinary collaborative research on massive dataset issues in commercial and scientific applications

To meet these objectives the center builds on existing research strength in the area of massive data algorithmics, existing and new international research collaboration, existing and new multidisciplinary and industry collaboration and in general on a vibrant international environment at the main center site. Another key to meeting the objectives is the focus on three related but also very different research focus areas (refer to the research plan for a thorough discussion of the areas):

I/O-efficient algorithms:
Algorithms designed in a two-level memory-disk *external memory* (or *I/O*-) model, where the memory hierarchy consists of a main memory of limited size *M* and an external memory (disk) of unlimited size; the goal is to minimize the number of times a block of *B* consecutive elements is read (or written) from (to) disk (an *I/O-operation*, or simply *I/O*). The model is motivated by the fact that transfers between main memory and disk, rather than e.g. CPU computation, is the bottleneck when processing massive datasets residing on disk.

Cache-oblivious algorithms:
Algorithms designed in the I/O-model – but without knowledge of *M* and *B* – and then analyzed as I/O-model algorithms; memory transfers are assumed to be performed by an off-line optimal replacement strategy. The beauty of the model is that since the I/O-model analysis holds for any block and memory size,

it holds simultaneously on *all* levels of *any* multi-level memory hierarchy. Thus the cache-oblivious model is effectively a way of modeling a complicated (maybe even unknown and/or changing) multi-level hierarchy using the simple two-level I/O-model.

Streaming algorithms:
Algorithms designed in a model where only one (or a small constant number of) *sequential pass(es)* over the data is (are) allowed. The goal is to solve a given problem while using significantly smaller space than the input data size (preferably less than the main memory size), and while processing each data object as fast (with as few CPU operations) as possible. The model is motivated by the fact that when processing truly massive datasets, solutions requiring more than one sequential pass over the data are often infeasible, since random access to disk blocks are much slower than sequential access. Moreover, in some applications (e.g. sensor network applications) data simply *has* to be processed sequentially as it is generated.

In recent years, the algorithms community has begun to seriously address *algorithm engineering* issues: The design and analysis of practical algorithms, efficient implementation of these algorithms, and experimentation providing insight into their applicability and further improvements. Algorithm engineering is naturally an integrated part of MADALO, both because a main motivation for the center is the inadequacy of traditional algorithms theory in providing practically efficient algorithms, and because engineering work naturally supports multidisciplinary and industry collaboration.

2 Center organization

Organizationally, 2007 has obviously been a build-up year, since the center was only initiated on March 1, 2007. The main center site is at University of Aarhus (with Brodal and Arge as core faculty) and initially the center also included researchers at the Max Planck Institute for Informatics (MPI) in Germany (Mehlhorn and Meyer) and at Massachusetts Institute of Technology (MIT) in the US (Demaine and Indyk). The sites were chosen not only because of the general algorithms strength of the three sites (in Denmark, Europe and the US), but mainly because of the complementary research strength of the core researchers within the focus areas, as well as their track record of collaboration. Initially, considerable effort had to be put into contract negotiations; especially the negotiations with MIT turned out to be much more complicated and time consuming than anticipated. As discussed with the foundation during the November 2007 budget revision, the center support for the MIT center research (summer support for Demaine and Indyk and a PhD student) outlined in the center contract also had to be slightly increased and changed (the number of supported PhD student months lowered) due to Demaine and Indyk receiving tenure (and thus a higher salary rate). Additionally, Meyer moved from MPI in order to take up a full professor position (chair in algorithm engineering) at Frankfurt University (FRA). In order to keep the research expertise of Meyer and his group in the center, and with the understanding of MPI (Mehlhorn) and the foundation, it was decided to add FRA to the center sites and move the center support for MPI (a PhD student) to FRA; thus a contract was also negotiated with FRA and the MPI contract was renegotiated.

Logistically the main center site – again after significant efforts – was more or less settled by the end of the summer. A center manager (Else Magård) and a center programmer (Kasper D. Larsen) had been hired, and a half time accountant (Ellen Lindstrøm) and secretarial support in general had been secured from the Computer Science Department (funded by AU). Center office facilities had also been secured. The center now occupies approximately 1½ (of 2) hallway(s) in the “Turing building” of the Computer Science Department, connected via a newly established internal spiral staircase ending in an also newly established lounge. Thus the center has nice and connected facilities, including meeting, conference and computer laboratory facilities, as well as adequate office space not only for research staff, administrative staff and PhD and master students, but also for visitors and collaborators from other departments. During the negotiations with the Computer Science Department, the center put considerable emphasis on securing such connected facilities with ample space for visitors, meeting, conference and lab facilities, as well as common areas, in order to support a dynamic and collaborative environment; such facilities also contribute to making the center attractive for visiting researchers.

On the research personnel side the center has developed more or less as planned. As it is obvious from the personnel overview, the center had somewhat of a “running start” since all center cites already had a number of Post Docs and PhD-students working in the center focus areas. At AU two new Post Docs (Rao and Abam) were hired in August as planned, but one (Abam) chose to defer his starting date to January 2008; two new PhD-students (Deleuran and Tsakalidis) were also hired in August as planned. Center Post Docs and PhD-students are further discussed in Section 6.

3 Center research activities

3.1 Research plan

The research plan discusses a number of main research challenges in each of the focus areas and lists a number of concrete open geometric algorithms and graph algorithms problems (that is, open problems in the areas of handling massive geometric data, such as e.g. a set of triangles in 3D representing a terrain, and of handling massive graph data, such as e.g. the graph – a set of vertices connected by edges – making up a large road network). However, it is also noted that the list of problems is non-exhaustive, and that the outcome of current and future research of course should influence the exact directions taken in the center. The research focus in the three areas is also somewhat different, since they are at different levels of development:

The *I/O-efficient algorithms* area is quite developed. Not only have a large number of algorithms and algorithm design techniques been developed, but the immense practical importance of I/O-efficient algorithms has also been established through experimental work. Still many important problems remain open. The research plan outlines a number of such problems, including a number of fundamental geometric data structure and graph traversal problems and some very practically motivated terrain data processing problems.

The *cache-oblivious algorithms* area is relatively new and even though efficient algorithms have been developed for a number of fundamental sorting and searching problems, as well as a few geometric and graph problems, many even very fundamental problems remain open. The practical importance of cache-oblivious algorithms also still needs to be investigated. The research plan outlines a number of fundamental problems to be considered, mainly on geometric data structure and batched problems.

Over the last few years the *streaming algorithms* area has flourished as the discovery of several novel algorithmic techniques has enabled the enlargement of the class of problems with efficient streaming algorithms. Nevertheless, fundamental gaps remain in the understanding of what problems can be solved in the streaming model. The research plan outlines a number of fundamental problems to be considered, including investigation of the general applicability of developed techniques, as well as of fundamental geometric problems and of graph problems in variants of the streaming model.

Finally, the research plan also outlines a number of *algorithm engineering* challenges in the I/O-efficient and cache-oblivious algorithms areas (the centers focus in streaming algorithms is on theory rather than engineering). In the cache-oblivious area focus is on engineering simple algorithms for very fundamental problems, whereas in the I/O-efficient algorithms area focus is on leveraging existing basic I/O-algorithms libraries, such at the TPIE library developed by Arge’s group, to further engineer algorithms for fundamental problems (e.g. graph problems) and to (further) develop software for efficient processing of massive terrain data.

3.2 Overview of selected 2007 focus area results

Despite 2007 being a build-up year for the center, a number of interesting and important results have been obtained in the focus areas, as well as in other related areas. In this section we briefly discuss a few of these results in each of the three focus areas. Note that due to space limitations we will not be able to give a complete overview of all the results obtained by center researchers (see Appendix I for a full list of papers published in 2007).

I/O-efficient algorithms

During 2007 we have obtained a number of interesting results in relation to the I/O-model problems outlined in the research plan. Below we discuss some of these results.

In terms of geometric problems, we have e.g. developed an improved point location data structure, that is, a data structure for maintaining a dynamically changing subdivision of the plane on disk such that the region containing a query point can be found I/O-efficiently. A number of structures have previously been developed for the problem, but using a number of new and existing techniques we managed to improve on the number of I/Os needed to update the subdivision. The paper will be presented at ACM Symposium on Computational Geometry in June 2008. At the same conference we will also present an I/O-efficient algorithm for another geometric problem, namely the problem of computing contour lines of a terrain stored as a triangulated surface (a triangulation of the plane where a height is associated with each vertex). We managed to develop the first I/O-optimal algorithm for the problem of computing all contours at a given height interval, as well as outputting these contours in clockwise order and with information about how they are nested. The techniques we developed can also be used to build a data structure on disk such that all contours at a given query height can be generated I/O-optimally.

While the contour line algorithm mentioned above is probably mostly of theoretical interest (i.e. it might be too complicated to be of practical interest), we have also (as outlined in the research plan) worked extensively on engineering I/O-efficient algorithms for massive terrain data processing, while leveraging our previous extensive implementation work. In fact, we have reimplemented and significantly extended the capabilities of our TerraFlow terrain processing software package. The resulting TerraSTREAM software package now e.g. has support for terrains represented as triangulated surfaces rather than only for terrains stored as a regular grid of height values [8].[1] We have released a (limited) version of the package to a number of research and industry collaborators, as well as to a number of other researchers and companies on request. We are continuing the development of TerraSTREAM and plan another major release this year.

In terms of graph algorithms, we have also worked both on theoretical and more practical engineering issues. We have e.g. considered the problem of computing the diameter of a massive graph, which is a key challenge in for example complex network analysis. Since exact diameter computation is very costly, one is typically satisfied with an approximation of the diameter, often computed using a Breadth First Search (BFS) traversal of the graph (a traversal that visits all vertices at distance one from the source, then all vertices at distance two, and so on). However, as discussed in the research plan, I/O-efficient algorithms for even the simple and fundamental BFS problem are not known other than for special classes of graphs. In 2007 we have made progress on the BFS problem, and thereby the diameter problem, in several ways: In connection with a (DIMACS) implementation challenge, we have engineered some of our previous theoretical algorithms to obtain an algorithm that in some cases runs up to three orders of magnitude faster than the previous best solutions, and recently we have devised an approach to efficiently re-compute BFS after graph modifications (this result was recently presented at Symposium on Theoretical Aspects of Computer Science). We have also investigated alternative approaches (than using BFS) for approximating the diameter of a graph, and developed both new heuristic algorithms [22] as well as algorithms with worst-case guarantees on both I/O-complexity and approximation factor (to be presented at Scandinavian Workshop on Algorithm Theory).

Cache-oblivious algorithms

In the cache-oblivious area we have obtained a couple of interesting results in 2007 along the lines outlined in the research plan.

First, a center PhD student has been involved in work exploring insert/search tradeoffs for cache-oblivious search structures [12]. More precisely, it was shown how to speed up insertions significantly at the expense of a slight increase in the search cost. Experiments were also performed, showing that tremendous insertion speedups can be obtained in the I/O-model setting compared to a normal (B-tree) search structure. Secondly, we managed to develop the first algorithm for the red-blue line segment intersection problem, that is, the problem of finding all intersections between a set of non-intersecting red segments and a set of non-

[1] Numbers in brackets are the number of the relevant papers in Appendix I.

intersecting blue segments in the plane. This problem has important applications in geographic information systems, where it arises in the context of map overlay operations. The new algorithm is optimal, and maybe even more important, the first cache-oblivious algorithm for an intersection problem involving non-axis-parallel objects.

Streaming algorithms

As discussed in the research plan, past research in data stream algorithms has led to the discovery of a variety of algorithms for specific problems. However, the problems are mostly solved on a case by case basis, and there are virtually no general characterizations of problems that are amenable to the streaming approach. In [2] we provided the first step towards such a general characterization. In particular, we considered a general class of "distance problems", where the goal is to estimate the distance between two vectors in a streaming fashion, and provided general characterizations of distance functions that are solvable in the streaming model. In a paper recently presented at the 2008 Symposium on Discrete Algorithms, we proved another general theorem characterizing streaming algorithms. We observed that almost all known algorithms for streaming problems in fact solve a harder problem of constructing efficient "sketches", and showed that this is not a coincidence. Specifically, we provided a general method for converting a streaming algorithm into a sketching scheme (for a wide class of functions that cover a large fraction of problems of interest).

Testing independence of two observed variables is an important problem in massive data processing: Given samples from the joint distribution, is it possible to estimate the degree of dependence between the distributions? In another 2008 Symposium on Discrete Algorithms paper we gave a positive answer to this question by providing two algorithms for this task. Both algorithms use only small amount of auxiliary storage to perform the estimation up to arbitrarily small precision.

Finally, we have e.g. also considered algorithms in streaming models where multiple passes over the data are allowed [1,17]. For example, in [1] we described a technique for turning a parallel graph algorithm into an efficient algorithm in the multi-pass W-Stream model, where in each pass one input stream is read and one output stream is written in a pipelined fashion (such that the output of one pass is the input to the next pass). The technique e.g. yields new (often even optimal) algorithms for the minimal spanning tree, biconnected component and maximal independent set problems.

3.3 Updated research plan – new directions and opportunities

As outlined in the previous section, center researchers have obtained a number of interesting and important results following the original research plan. In general, we will continue to follow the research plan, but the outcome of current and future research should of course influence the exact research directions taken by the center. Already in the research plan it is mentioned that as the center matures, we plan to consider other interesting or even more realistic methodologies for massive data processing than I/O-efficient, cache-oblivious and streaming algorithms; one of the external reviewers also commented that we should maybe consider methodologies that incorporate parallel computation, and that we could consider so-called succinct data structures. In fact, although one can hardly label the center as "mature", we have already considered other methodologies/models, such as parallel, fault tolerance and flash memory models, as well as succinct data structures, and obtained a number of results in these areas. Below we highlight some of these models and results. In parallel with the original research plan, we plan to continue to pursue most of these areas.

Parallel private-cache models

Chip manufactures are increasingly producing chips with several CPUs (or *cores*) on a single chip; current architectures have 2, 4 or 8 cores but it is predicted that this number will grow dramatically in the not too distant future. Thus there is a need for parallel algorithms that can utilize the many cores. Unfortunately, most of the algorithms described in the vast literature on parallel algorithms are developed in models that do not adequately describe the new multicore architectures. We have recently considered algorithms in a possibly more suitable model, which is basically a parallel extension of the I/O-model where P processors have a main memory of limited size M each and share an external memory of unlimited size. The goal is to minimize the number of *parallel I/Os*, in which blocks of B elements can simultaneously be transferred between the external memory and each of the main memories. In a paper to be presented at the ACM Symposium on Parallelism in Algorithms and Architectures this summer, we describe efficient algorithms (in fact, optimal algorithms under certain assumptions) for several fundamental problems such as selection and sorting. Recently, we have also managed to develop efficient algorithms for several graph problems in the model. This work was performed with UC Irvine Professor Mike Goodrich (and one of his students) when he visited the center as a Fulbright scholar for a longer period in the summer of 2007.

Models of flash memory

Another hardware trend is that flash memory devices are becoming increasingly large and cheap. Initially used in digital audio players, digital cameras, mobile phones, and USB memory sticks, flash memory may eventually replace (or at least supplement) disks as the external storage in mobile computing. Since flash memory appears to have very different characteristics than both internal memory (RAM) and disks, it is important to try to model such memory. In recent work to be presented at the Workshop on Experimental Algorithms 2008, we characterized the performance of various flash based storage devices (including several flash disks). Besides analyzing an expected huge difference in read and write speed, we also analyzed the effects of read/write patterns (misalignments, aging and historical patterns) on the performance. It turned out that despite the similarities between flash memory and internal memory (fast random reads) and between flash disk and normal disk (data block movement), algorithms designed in internal memory models or the I/O-model do not necessarily realize the full potential of flash memory devices. In fact, just replacing a hard disk by flash may result in degenerating performance. However, we also illustrate that a careful combination of internal memory and I/O-model algorithmic ideas can yield nice speedups. Based on this we suggested that a model for flash memory should at least include different block sizes for read- and write-accesses. More work is obviously needed in this area, something that has also been recognized by the broader algorithms community; for example, flash memory was one of the themes of the (biannual) 2008 Dagstuhl seminar (invitation only seminars held at the Dagstuhl castle in Germany) on data structures (co-organized by center director Arge).

Fault tolerance models (resilience)

Modern memory is not always fully reliable – sometimes the content of a memory location may be temporarily or permanently corrupted. This may depend on manufacturing defects, power failures, or environmental conditions such as cosmic radiation. Furthermore, error rates are expected to increase as memory is getting smaller and more complex, and working at lower voltage and higher frequencies.

We are interested in being able to handle memory errors because they can become a serious problem when processing massive data (since it commonly means performing a large number of memory accesses on many memory devices and over a long period of time). The handling of errors has been addressed in various ways, both at the hardware and software level. At the hardware level they are often tackled using various error detection mechanisms (such as redundancy, parity checking or Hamming codes). However, such mechanisms often involve non-negligible penalties with respect to performance, size, and cost. Thus memories implementing these mechanisms are rarely found in ordinary workstations. In the algorithmic community dealing with unreliable information has been addressed in a variety of different settings. Very recently, one particularly interesting model called the *faulty-memory RAM model* has been proposed for modeling and handling memory errors. In recent work [5,6] we managed to develop the first optimal dynamic data structures for this model (dynamic dictionaries and priority queues). We plan to continue to study algorithms and data structures in the faulty-memory RAM model and have e.g. recently considered combinations of the model and the I/O-model.

Succinct data structures

With the rapid proliferation of massive data it is becoming increasingly important to design really space-efficient data structures that also support fast queries on the data. For example, one might be interested in storing web/XML data or DNA sequence data with as little space as possible in order to be able to transport it efficiently or store it in the main memory of a relatively small machine.

Succinct data structures are highly space-efficient data structures that support efficient queries while occupying an amount of space that is provably close to the information-theoretic minimum. Since succinct data structures are obviously important when processing massive data, and following the recommendation of one of the center proposal reviewers, we have considered the area of succinct data structures and hired a Post Doc (Rao) that is an expert in the area. This has lead to new results on how to succinctly encode a so-called bit vector, a data structure that is fundamental to several other succinct data structures and therefore has been extensively studied [20]. To obtain the new results (including both improved upper bounds/algorithms and lower bounds), several new techniques of independent interest had to be introduced. In very recent work to be presented at the 2008 Scandinavian Workshop on Algorithm Theory, we further generalized some of the lower bounds and developed improved algorithms (upper bounds) for the bit vector problem, as well as for another well-studied problem (strings). We plan to further pursue the development of succinct data structures.

4 Collaboration

By design the center is highly collaborative and one of the main goals of the center is to maintain a vibrant, world-class and international environment at the main center site. Thus emphasis is e.g. on hosting international visitors (faculty as well as PhD students) at AU. Most of the core MIT, MPI and FRA faculty have visited AU during 2007; Indyk spent all of August at AU. Several MIT, MPI and FRA PhD-students and Post Docs have also visited. Additionally, the list of non-center faculty visiting AU for shorter or longer periods of time include Norbert Zeh (Dalhausie), Athanasios Tsakalidis (Patras), Peter Sanders (Karlsruhe), S. Muthu Muthukrishnan (Google and Ruthers), Charles Leiserson (MIT), Jeff Vitter (Purdue), Mike Goodrich (UC Irvine), Jonathan Shewchuk (Berkeley), Pino Italiano (Rome), Rajeev Raman (Leicester), Philip Bille (ITU) and Rolf Fagerberg (Southern Denmark), just as two PhD students Igor Nitto (Pisa) and Nodari Sitchinava (UC Irvine) visited for a longer period of time in 2007. In general, center researchers have extensive international research collaboration with other computer scientists (ongoing research collaborators – most with joint publications – are listed in Appendix A).

As mentioned, another key goal of the center is to be a catalyst for multidisciplinary and industry collaboration. Thus center researchers already collaborate extensively with environmental and agricultural researchers (at the Faculty of Agricultural Sciences, National Environmental Research Institute, Duke University and NC State University) on issues in connection with massive terrain data. This collaboration is centered around the TerraSTREAM software package, and also includes extensive collaboration (through a NABIIT grant from the Danish strategic research council) with the terrain data acquisition and processing (among many other things) company COWI A/S. Looser collaboration around terrain data and TerraSTREAM is also ongoing with several other companies (DHI, CARIS, Eiva A/S and JonesEdmunds). Recently, initial steps have also been taken in a multidisciplinary collaboration around spatial modeling of the current and future plant diversity of Denmark (where terrain data also plays an important role) between center researchers and researchers at the Faculty of Agricultural Sciences and the Department of Biology. The collaboration will include co-supervision of a Biology PhD student. A broader collaboration around climate change influence on ecosystems and on how to adapt to (and/or mitigate) such changes is also under way in collaboration with agricultural, environmental and biological researchers at the Faculty of Agricultural Sciences, the National Environmental Research Institute and the Department of Biology through participation in one of the five proposals the University of Aarhus has selected for submission to the new prestigious ”investment capital for university research” program of the Danish government (Arge is one of the 6 main applicants, which include two members of the UN IPCC panel that received the 2007 Nobel peace price). Center researchers also play a (minor) role in a technology platform project “A for Galileo based pervasive positioning” that includes a broad range of computer science researchers, as well as researchers from Aarhus School of Business and the Danish agricultural advisory service, and several companies, including Alexandra A/S, Terma A/S, and Systematic. Finally, center researchers also (through a PhD-student project) have collaboration with the company cofmam.com on internet search engine issues.

Overall, center researchers have extensive collaborations with other computer scientists, scientists in general (predominantly biological and environmental scientists), as well as with industry, and is actively seeking to extend collaborative efforts. Apart from the initiatives mentioned above, the center is also exploring other