Theory and Applied Computing: Observations and Anecdotes

M. Brand, S. Frisken, N. Lesh, J. Marks, D. Nikovski, R. Perry, J. Yedidia

Mitsubishi Electric Research Labs (MERL), Cambridge, MA 02139, USA
E-mail: {brand,frisken,lesh,marks,nikovski,perry,yedidia}@merl.com

Abstract. While the kind of theoretical computer science being studied in academe is still highly relevant to systems-oriented research, it is less relevant to applications-oriented research. In applied computing, theoretical elements are used only when strictly relevant to the practical problem at hand. Theory is often combined judiciously with empiricism. And increasingly, theory is most useful when cross-pollinated with ideas and methods from other fields. We will illustrate these points by describing several recent projects at Mitsubishi Electric Research Labs that have heavy mathematical and algorithmic underpinnings. These projects include new algorithms for: traffic analysis; geometric layout; belief propagation in graphical models; dimensionality reduction; and shape representation. Practical applications of this work include elevator dispatch, stock cutting, error-correcting codes, data mining, and digital typography. In all cases theoretical concepts and results are used effectively to solve practical problems of commercial import.

1 Introduction

Many of the classical topics of theoretical computer science (e.g., algebra, finite automata, geometry, graph algorithms, logic, numerical methods, queuing theory, string processing) are still studied to good effect in the industrial labs of systems-oriented companies like IBM and Microsoft. However, as ubiquitous computing becomes a reality, many companies (e.g., Mitsubishi Electric, Philips, Siemens, and Sony) are focusing on *applied* computing. Computer-science theory has still to establish itself in applied computing. We argue that theory has a useful role in this context, but only when the following precepts are borne in mind:

- *Theory is a tool to understand and solve practical problems.* In other words, the nail should take precedence over the hammer: in applied computing the problem is paramount and the means of solution is secondary. This mindset leads to better, more eclectic problem selection and ultimately to more-relevant research.
- *Theory and empiricism complement each other.* Many real-world problems involve incomplete or uncertain data; many are NP-hard. Completeness, optimality, and asymptotic complexity are theoretical concepts that are rarely

useful on their own for such problems. However, in combination with experimentation and statistical analysis—the tools of empirical analysis—these theoretical notions can be very useful.

– *Theoretical computer science can be informed by insights from other fields.* Cognitive science, economics, electrical engineering, statistics, theoretical physics: concepts from these fields and others have proven useful for practical problems when used in tandem with computer-science theory.

We illustrate these points by describing several recent projects at Mitsubishi Electric Research Labs (www.merl.com) that have strong mathematical and algorithmic underpinnings. These projects include new algorithms for: traffic analysis; geometric layout; belief propagation in graphical models; dimensionality reduction; and shape representation. Practical applications of this work include vehicular-traffic prediction, elevator dispatch, stock cutting, error-correcting codes, image processing, data mining, and digital typography. In all cases theoretical concepts and results are being used in accord with the precepts above to solve practical problems of commercial import.

2 Traffic Analysis

Traffic of goods, vehicles, and passengers is a very complex phenomenon characterized by significant stochasticity, non-stationarity, incomplete observability, and huge problem sizes. Two transportation problems of large economic significance are optimal routing of vehicular traffic from origin to destination, and optimal elevator service for passengers in large buildings. Recent progress on both of these problems has resulted from a synergistic combination of theoretical and empirical concepts and methods.

Early successes in the field of vehicle routing were the result of important theoretical insights, most notably the formulation of the principle of optimality by the mathematician Richard Bellman, and the subsequent widespread use of dynamic programming [3]. In particular, efficient algorithms for finding shortest paths in static graphs have been available for a long time and now run on the relatively weak computers found in car-navigation systems. Current models offer spectacular performance, planning routes between two points in a whole country in less than one second, and navigation-system vendors are looking for novel and more advanced applications [24].

One such application is dynamic route guidance, or car navigation in dynamic stochastic networks. Nowadays, heavy congestion plagues the roads of most cities in the developed industrial world, and travel times can vary significantly depending on the time of day, week, year, etc. Finding optimal routes and optimal departure times under such conditions opens a number of novel problems such as sensing the state of the transportation network, predicting travel time on short-term and long-term horizons, and finding the shortest-routing policies in time-varying stochastic networks.

Predicting travel times from past observations is decidedly on the practical side of scientific research, and is currently an active area of investigation in industry. Recent results include a fast and efficient linear method for travel-time prediction with surprisingly good accuracy [14]. However, the very foundations of the traffic-prediction enterprise depend on answering some very theoretical questions that concern the limits of predictability in congested transportation networks. Nagel and Rasmussen have put forward the hypothesis that a transportation network would exhibit chaotic behavior when its load is pushed to its capacity, and hence its long-term prediction would be impossible [12]. An alternative and simpler hypothesis explains the high variance of travel times in heavily congested regimes from the point of view of queuing theory, without adverse implications to expected predictability. While finding the correct explanation is ultimately a highly theoretical question, its answer is likely to affect significantly all fielded systems.

Planning routes with dynamic stochastic travel times and scheduling elevators under dynamic stochastic passenger flows are two related problems that are also accompanied by partial observability. Modern frameworks such as decision-theoretic planning, considered theoretical and abstract only until quite recently, are slowly starting to bear fruit and find their way into practical applications. For example, Figure 1 shows a seven-floor building with four hall calls and one car call assigned to an elevator car. The uncertainty in passenger destinations can lead to an exponential number of possible car trajectories, illustrated here by a partial tree. However, dynamic programming can be employed to marginalize out this uncertainty in linear time [13]. Although explicit decision-theoretic methods have provable performance, they have yet to be embraced by industry: all current commercial elevator-scheduling systems use heuristic AI methods. To be accepted by engineers, new elevator-scheduling algorithms must prove themselves empirically, through thorough simulation and field tests.

3 Human-Guided Search

The Human-Guided Search (HuGS) project is an ongoing investigation into designing interactive human-in-the-loop optimization systems. This work illustrates the value of combining theory with empiricism, and of combining theory with techniques from other fields, in this case human-computer interaction (HCI). Interactive optimization produces more usable solutions than automatic optimization because users can steer an interactive algorithm based on their knowledge of real-world constraints. People are better able to trust, justify, and modify solutions when they help construct those solutions. Additionally, interactive optimization leverages people's skills in areas in which people currently outperform computers, such as visual perception, strategic thinking, and the ability to learn.

A major component in HuGS research is designing algorithms that are amenable to human guidance. We have developed a human-guidable version of tabu search for jobshop scheduling, edge-crossing minimization, the selective traveling sales-

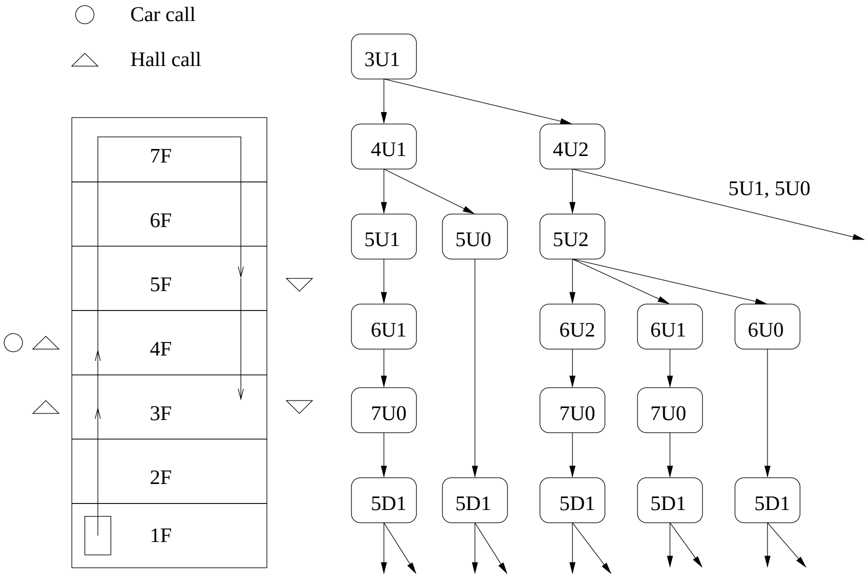


Fig. 1. An elevator car (one of several) in a seven-floor building can have an exponentially large number of possible trajectories when serving all calls assigned to it, due to the uncertainty in passengers' destinations. (Each node in the tree denotes the stopping floor, direction, and number of passengers inside the car.) However, the resulting probabilistic tree has sufficient structure that allows these trajectories to be folded into an efficient Markov model that can be evaluated in linear time.

man problem, and simplified protein folding [8, 11]. More recently, we have built a HuGS application for a 2D rectangular strip packing, which has many industrial applications, such as glass and steel cutting [9].

Our interactive packing application consists of several components. One component is an interface, shown in Figure 2, that is common to many of our applications. The interface allows the user to manually modify solutions, backtrack to previous solutions, and invoke, monitor, and halt a variety of optimization algorithms on the whole problem or a subset of the problem.

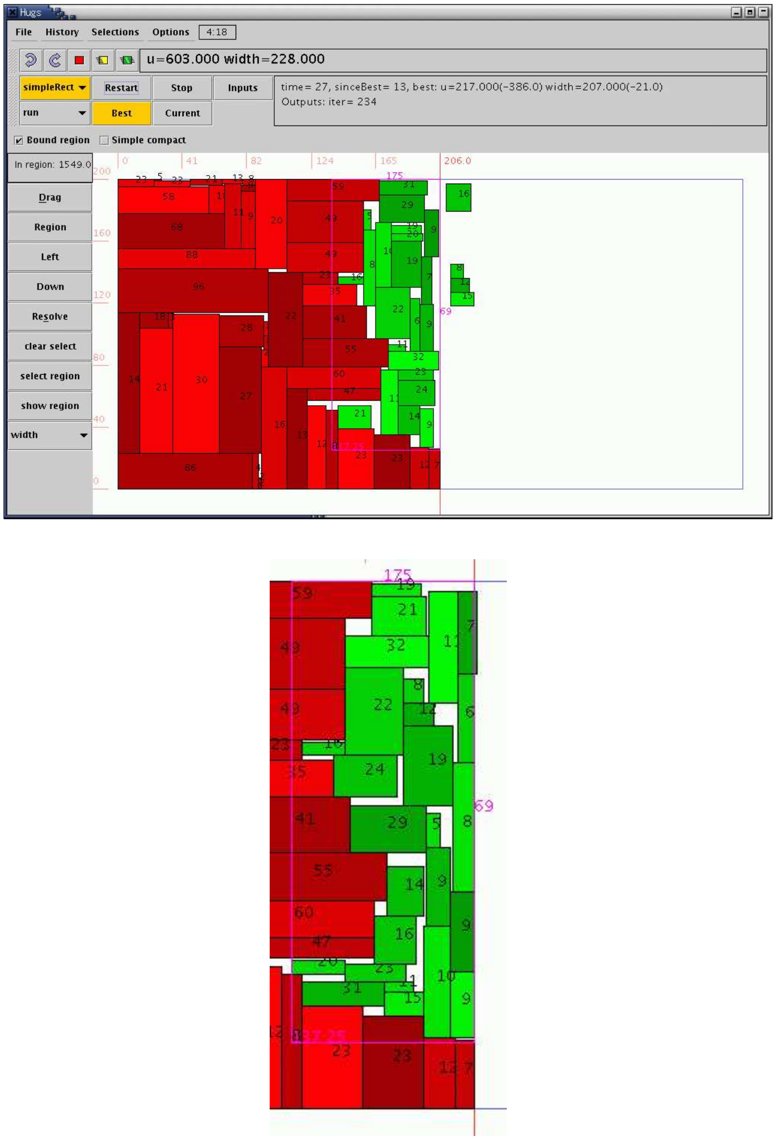
We provide two packing algorithms to the user. The first is a branch-and-bound algorithm for producing perfect packings, i.e., packings in which there is no unused space [10]. One can think of this special case as a jigsaw puzzle with rectangular pieces. We developed several powerful bounding methods that enable our algorithm to produce exceptionally good results on artificially constructed benchmarks in the literature that were designed to have solutions that are perfect packings. The real value of our algorithm, however, is as a subroutine that the user can invoke on a portion of the target packing area. Our algorithm fills as much of the user-defined region as it can with the user-selected rectangles without leaving any unused space between rectangles. Even though this algorithm cannot solve realistic problems by itself, it is a very useful tool or subroutine for realistic problems.

The second algorithm we provide is an extension of a priority-based greedy heuristic that was shown to be a 3-approximation [2]. We produced an anytime algorithm that stochastically searches for solutions near the single solution produced by these heuristic. We often find substantially better solutions after only a small number of iterations. The results produced by this algorithm are better than previously published results on these benchmarks.

However, the most distinctive aspect of our system is the way in which it leverages our algorithmic novelties by incorporating the innate geometric-reasoning abilities of humans into the process. We have found that people can identify particularly well-packed subregions of solutions, and focus our algorithms on improving the other parts. Furthermore, people can readily envision multi-step repairs to a packing problem to reduce unused space. Our experiments on large benchmarks show that interactive use of our system can produce solutions 1% closer to optimal in about 15 minutes than our algorithms can produce automatically in 2 hours [9]. For typical industrial applications, this represents a commercially significant advantage.

4 Belief Propagation

The "belief propagation" (BP) algorithm is used to solve "inference" problems, at least approximately. Inference problems are important in many different scientific and industrial fields. Essentially any time you receive a noisy signal and need to *infer* what is really out there, you are dealing with an inference problem. Some fields that are dominated by the issue of inference are computer vision, speech recognition, and digital communications. In recent work at MERL, re-



```
u=603.000 width=228.000
```

```
time= 27, sinceBest= 13, best: u=217.000(-386.0) width=207.000(-21.0)
Outputs: iter= 234
```

Fig. 2. Interactive system for an industrial cutting-stock problem, 2D Rectangular Strip Packing. In the first image, the user has selected a region to which an optimization algorithm can be applied and has "frozen" most of the already-packed rectangles (those shown in red) in their place. The second image shows a blowup of the selected portion of the packing after the optimization algorithm has run for a few seconds. By allowing the human user to focus the search on a small region and subset of the rectangles, a better packing for the problematic region is found quickly, thereby improving the overall solution. The combination of user interaction and automatic placement does better than either approach on its own.

sults from statistical mechanics have been combined with results from theoretical computer science to shed new light on the BP algorithm.

It is therefore perhaps not so surprising that a good algorithm to solve such problems has been repeatedly re-discovered in different scientific communities. In fact, one can show that such apparently different methods as the "forward-backward algorithm," the "Viterbi algorithm," Gallager's "probabilistic decoding" algorithm for low-density parity check codes, the "turbo-decoding" algorithm, the Kalman filter, Pearl's belief propagation algorithm for Bayesian networks, and the "transfer-matrix" approach in statistical physics are all special cases of the BP algorithm.

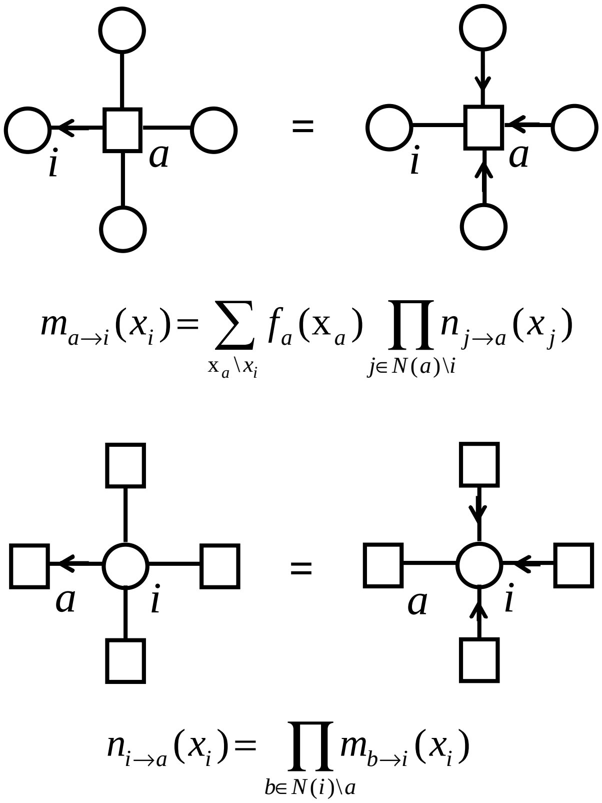
BP algorithms come in many flavors. The goal of the version that we consider here is to compute marginal probabilities for variables defined in a graphical model. These graphical models are referred to in various communities as Bayesian networks, Markov random fields, factor graphs, or statistical mechanical spin systems. Computing marginal probabilities thus corresponds to computing magnetizations for a spin system, or making a diagnosis in a Bayesian network, or computing a bit value for an error-correcting code.

In a BP algorithm, variable nodes in a graphical model iteratively send each other "messages" (see Figure 3). These messages are estimates that each variable node has of its own state, given what it is told by all of its neighboring nodes except for the node to which it is sending a message. If and when the iterative algorithm converges, the desired marginal probabilities can be read off from the converged messages.

It is known that BP is exact when the graphical model has a tree-like topology; that is, when it has no cycles. However, the graphical models used in computer vision, or those for error-correcting codes, are infested with cycles, and BP still seems to give excellent approximate answers in these cases. The goal of the MERL project described here was to understand why BP worked so well even for cases when it seemed to have no justification [22, 23].

At MERL, we showed that the fixed points of the BP algorithm are the same as the stationary points of the "Bethe free energy," which is an approximate free energy for the graphical model. This deep connection between a classical approximation in statistical mechanics and a classical algorithm in computer science has important implications. For example, it means that by minimizing the Bethe free energy, one can invent algorithms that have the same fixed points as BP, but for which convergence is *guaranteed* [19, 25]. Moreover, it means that one can *improve* upon the approximation made by BP, by improving upon the Bethe-free-energy approximation. We have developed a theory of such generalized belief propagation (GBP) algorithms [22, 23].

Algorithms that minimize the Bethe free energy directly have been shown to improve upon standard BP decoding algorithms for state-of-the-art error-correcting codes, by eliminating the failure mode of lack of convergence [16]. GBP algorithms have given improved results over BP algorithms for such disparate problems as decoding error-correcting codes [21, 20], and the computer vision problem of recovering shading and reflectance information from a single



$$m_{a\to i}(x_i)=\sum_{\mathrm{x}_a\setminus x_i} f_a(\mathrm{x}_a)\prod_{j\in N(a)\setminus i} n_{j\to a}(x_j)$$

$$n_{i\to a}(x_i)=\prod_{b\in N(i)\setminus a} m_{b\to i}(x_i)$$

Fig. 3. This figure illustrates BP message-update rules, operating on a "factor graph." A factor graph has two kinds of nodes: variable nodes, indicated by a circle, and function nodes, indicated by a square. Variable nodes are connected to function nodes if they are arguments of that function. Messages are sent from variable nodes to function nodes and vice-versa according to a set of rules that can be derived by minimizing the Bethe free energy, a measure from the field of statistical mechanics.

image [18]. GBP has inspired closely-related algorithms like “structured summary propagation,” which has been applied with good results to the problem of synchronization using linear feedback shift registers [6]. Finally, GBP has been combined with fast Fourier transforms to give an exciting new algorithm for reconstructing missing data [17].

5 Dimensionality Reduction

Cheap sensing and storage devices have produced massive data streams, and new challenges to researchers in data mining, machine learning, and machine perception. Because data processing has not kept pace with data production, it is often necessary to reduce data sizes by ”squeezing” out redundancies before any expensive processing begins. This challenge of dimensionality reduction is being met successfully by hybrid approaches that combine theory from many fields with empirical methods such as simulation and visualization.

Where data can be interpreted as points in a vector space with a Euclidean metric, squeezing out redundancy is usually synonymous with reducing the dimensionality of that space via subspace projection, e.g., principal components analysis. The orthogonal basis of this subspace is computed via singular-value or eigen-value decomposition. These decompositions are usually computed in quadratic time. We have developed a linear-time online approximation for the principal singular vectors and values of streaming data that is exact for data having true low rank and provably convergent to the optimal vectors when high-rank data arrives in a random order [5]. It is an enabling technology for compression and correlational analysis of massive data sets and streams. For example, correlated tastes between movie-goers in a 14-dimensional subspace of movie ratings are remarkably accurate predictors of how well 1000s of people will like 1000s of different movies. Since consumer tastes are non-stationary and not sampled at random, there are interesting questions as to convergence rate and stability over time, and thence sample and computational complexity.

Many data sets do not comfortably fit in low-dimensional linear subspaces. Instead, the data lies on some low-dimensional manifold embedded with curvature in the high-dimensional measurement space. Recently there has been great ferment in nonlinear dimensionality reduction, which aims to unfurl the manifold in a low-dimensional space so that the distribution of the data can be studied in a linear space. This problem area is a rich interface between graph theory, differential geometry, and statistics. Nearly all current methods have (unacknowledged) ancestry in Tutte’s theorems on graph embeddings from the 1960s, which view data points as vertices in a locally connected graph that is to be embedded in a Euclidean space with minimal distortion. We have shown how to estimate smooth maps that relate the original data space to a coordinate system intrinsic to the manifold, so that, for example, high-resolution 3D scans of human faces can be assigned low-dimensional coordinates, and novel faces can be synthesized by varying those coordinates as if they were a linear system [4]. Although this depends on a locally linear approximation of the manifold, we have shown that a slight elaboration of the scheme is capable of exact isometric embeddings of a significant class of curved manifolds that includes developable surfaces. A sample reconstruction is shown in Figure 4. Although these methods have already seen extensive practical use, the relationship between topology, geometry, and sample complexity remains largely unexplored.

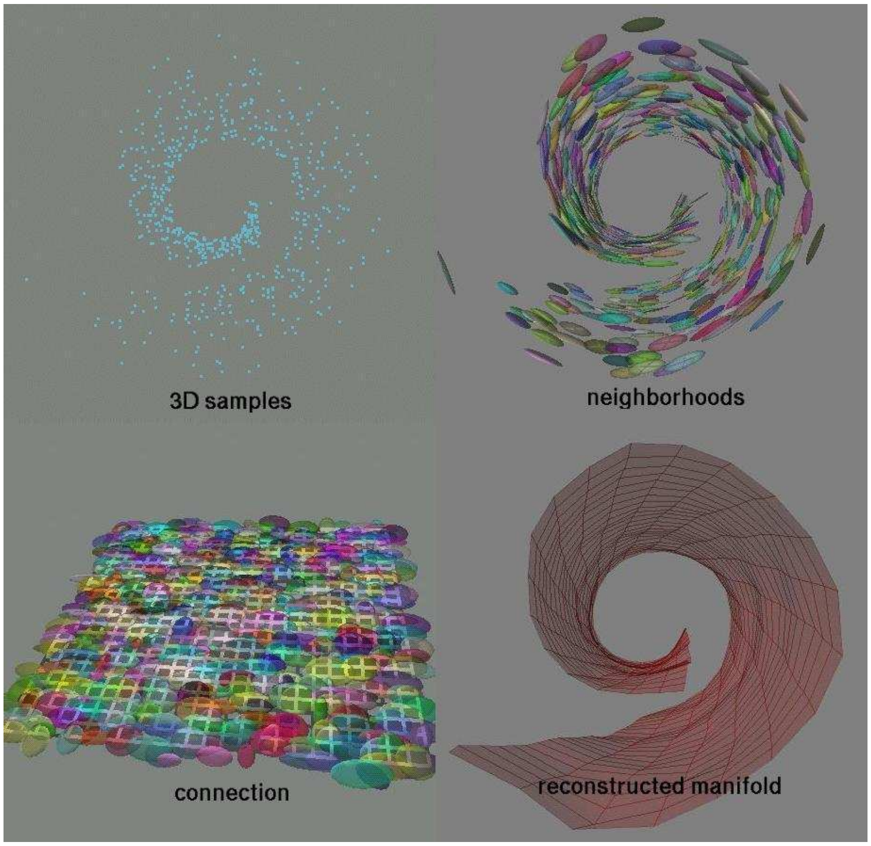


Fig. 4. A kernel-based reconstruction of a non-linear manifold.

6 Shape Representation

The representation of geometry is a fundamental topic in computational geometry. Different representations are usually compared in terms of the computational efficiency of different computations on those representations. However, efficiency of computation is but one criterion: for digital typography several other measures are relevant. These additional criteria include perceptual measures, aesthetic design considerations, and ease of hardware implementation: these criteria are inherently empirical.

The dominant paradigm for representing high-quality, antialiased, scalable type is hinted outline fonts [1]. Outline fonts render horribly as they scale unless hints—arbitrary procedures—are provided that perform grid fitting and geometric adjustment for a given scale (see Figure 5). Hints have to be crafted by hand for each typeface by skilled typographers. The arbitrary nature of hints makes hardware implementation problematic.

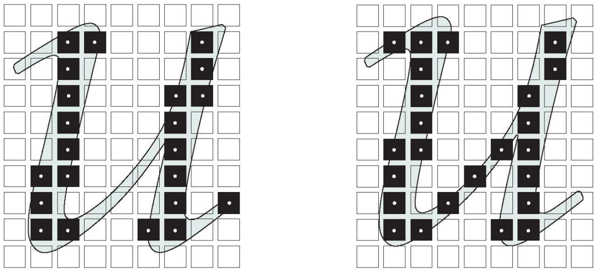


Fig. 5. Hints for grid fitting and other geometric adjustments radically improve the quality of rendered type for conventional outline fonts. An unhinted rendering is shown on the left; a hinted rendering of the same character is shown on the right.

Saffron is a new digital font technology that represents 2D shape with adaptively sampled distance fields [7, 15]. A distance field is an implicit representation of shape. A continuous distance field can be represented by regular samples (see Figure 6). However, regularly sampled distance fields are too big and inefficient and do not provide enough detail in some critical regions, such as corners.

Adaptive sampling preserves geometric detail where needed. An adaptively sampled distance field can be stored in an efficient spatial data structure (see Figure 7). The original shape can be reconstructed using various methods: biquadratic interpolation is one good technique. Antialiased images can be computed directly from the distance field, in contrast to the approximate area-coverage computations used with other representations (see Figure 8). Thus Saffron type can be rendered without hints. The simple rendering algorithm is amenable to hardware implementation.

7 Conclusions

It is likely that the projects described here are very different from those described in the other papers in this proceedings. Theoretical computer science is useful for applied computing, but usually only when combined with other techniques

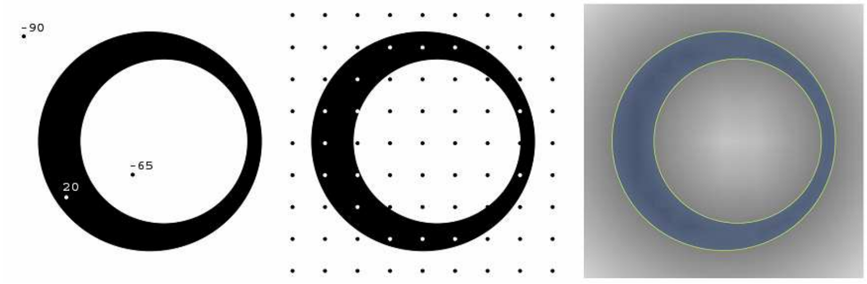


Fig. 6. A distance field gives signed distance values at each point in the plane: these values correspond to the shortest distance to an outline edge. The continuous field depicted on the right can be represented inefficiently by the discrete regular samples depicted in the center.

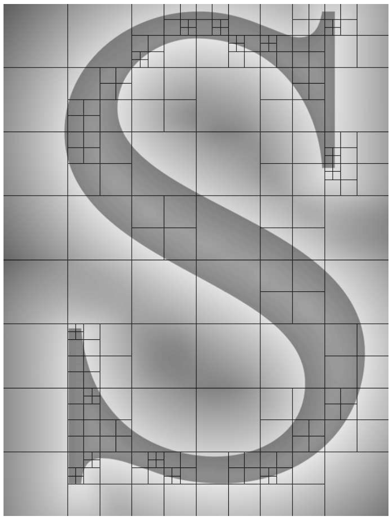


Fig. 7. A depiction of an adaptively sampled distance field for the character 'S'.

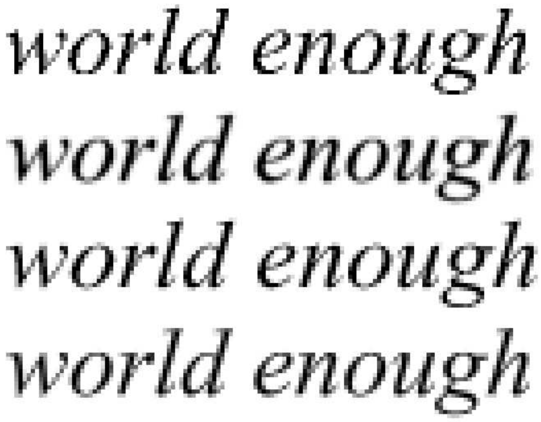


Fig. 8. A comparison of different representations and rendering algorithms, from top to bottom: unhinted outlines, box filter, four samples per pixel; unhinted outlines, Gaussian filter, sixteen samples per pixel; hinted outlines with sophisticated filtering and multiple samples per pixel (a proprietary algorithm); unhinted adaptive distance fields (Saffron), one sample per pixel.

and when evaluated and refined empirically. A broader view of what constitutes computer-science theory can make it more useful for the many varied problems that arise in applied computing.

References

1. Adode Systems, Inc. *Adobe Type 1 Font Format.* Addison Wesley, 1990.
2. B. S. Baker, J. E. G. Coffman, , and R. L. Rivest. Orthogonal packings in two dimensions. *SIAM Journal on Computing*, 9:846–855, 1980.
3. D. P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, Belmont, Massachusetts, 2000. Volumes 1 and 2.
4. M. Brand. Continuous nonlinear dimensionality reduction by kernel eigenmaps. In G. Gottlob and T. Walsh, editors, *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 547–552, San Mateo, CA, 2003. Morgan Kaufmann.
5. M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Proceedings of the European Conference on Computer Vision*, volume 2350, pages 707–720, Berlin, 2003. Springer.
6. J. Dauwels, H.-A. Loeliger, P. Merkli, and M. Ostojic. Structured-summary propagation, LFSR synchronization, and low-complexity trellis decoding. In *Proceedings of the 41st Allerton Conference on Communication, Control, and Computing*, pages 459–467, 2003.
7. S. Frisken, R. Perry, A. Rockwood, and T. Jones. Adaptively sampled distance fields: A general representation of shape for computer graphics. In *Proceedings of SIGGRAPH 2000*, pages 249–254, July 2000.

8. G. Klau, N. Lesh, J. Marks, and M. Mitzenmacher. Human-guided tabu search. In *Proceedings of AAAI 2002*, pages 41–47, July 2002.
9. N. Lesh, J. Marks, A. McMahon, and M. Mitzenmacher. New exhaustive, heuristic, and interactive approaches to 2D rectangular strip packing. Technical Report TR2003-05, Mitsubishi Electric Research Laboratories (MERL), 2003.
10. N. Lesh, J. Marks, A. McMahon, and M. Mitzenmacher. Exhaustive approaches to 2D rectangular perfect packings. *Information Processing Letters*, 90(1):7–14, April 2004.
11. N. Lesh, M. Mitzenmacher, and S. Whitesides. A complete and effective move set for simplified protein folding. In *Proceedings of the 7th Intl. Conf. on Research in Computational Molecular Biology*, pages 188–195, April 2003.
12. K. Nagel and S. Rasmussen. Traffic at the edge of chaos. In *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, Cambridge, MA, 1994. MIT Press.
13. D. Nikovski and M. Brand. Decision-theoretic group elevator scheduling. In *13th International Conference on Automated Planning and Scheduling*, pages 133–142, Trento, Italy, June 2003. AAAI.
14. N. Nishiuma, H. Kumazawa, Y. Goto, D. Nikovski, and M. Brand. Traffic prediction using singular value decomposition. In *Proceedings of the 11th ITS World Congress (to appear)*, Nagoya, Japan, October 2004.
15. R. Perry and S. Frisken. A new framework for representing, rendering, editing, and animating type. In preparation.
16. T. Shibuya, K. Harada, R. Tohyama, and K. Sakaniwa. Iterative decoding based on concave-convex procedure. In review, 2004.
17. A. Storkey. Generalized propagation for fast fourier transforms with partial or missing data. In *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.
18. M. F. Tappen, W. T. Freeman, and E. H. Adelson. Recovering intrinsic images from a single image. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2003.
19. M. Welling and Y. W. Teh. Belief optimization for binary networks: A stable alternative to belief propagation. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 554–561, 2001.
20. J. S. Yedidia, J. Chen, and M. Fossorier. Generating code representations suitable for belief propagation decoding. In *Proceedings of the 40th Allerton Conference on Commmunication, Control, and Computing*, 2002.
21. J. S. Yedidia, W. T. Freeman, and Y. Weiss. Characterizing belief propagation and its generalizations. Technical Report TR2001-15, Mitsubishi Electric Research Laboratories (MERL), 2001.
22. J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*, chapter 8, pages 239–269. Morgan Kaufmann, 2003.
23. J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical Report TR2004-040, Mitsubishi Electric Research Laboratories (MERL), 2004.
24. K. Yokouchi, H. Ideno, and M. Ota. Car-navigation systems. *Mitsubishi Electric Advance*, 91:10–14, 2000.
25. A. L. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14(7):1691–1722, 2002.