Implementation and Evaluation of Novel Buildstyles in Fused Deposition Modeling (FDM)

Tobias Wasser, Anshu Dhar Jayal, Christoph Pistor

University of Utah, Salt Lake City, 1999

1 Abstract

Previous investigations have shown that the optimization of extrusion dynamics in conjunction with the buildstyle pattern is of paramount importance to increase part quality in Fused Deposition Modeling (FDM). Recently domain decomposition and space filling curves have been introduced for slice generation in FDM [1]. The current work focuses on the implementations of fractal-like buildstyle patterns using Simulated Annealing [2, 3], Lin-Kernighan algorithms [4] and Construction Procedures based on Nearest Neighbor Heuristics [5]. These computational optimization procedures are able to generate filling patterns that allow the continuous deposition of a single road to fill arbitrary shaped domains. The necessary software modules to produce arbitrary three-dimensional artifacts have been developed and are evaluated with respect to part quality and build time.

2 Introduction

Today path generation and optimization problems found in manufacturing processes, such as FDM, typically aim for solutions that results in the shortest possible cycle time. On one hand this necessitates the determination of the shortest path. On the other hand, the manufacturing cycle time will be minimized the higher the speed which can be achieved on the individual path segments. As a result, variations in traveling speed are increased whenever sharp turns are taken, due to the deceleration and acceleration limitations of motion control and the manufacturing equipment. All manufacturing processes are more or less sensitive to these changes in traveling speed but usually require more advanced control capabilities as speed variations increase. Additional examples of processes particularly sensitive to tool traveling speed variations would be the robotic spray painting of car body parts or the automated fluid dispensing onto a surface. Both also benefit from a non-self-intersecting continuous path, which is a must for the FDM process. The simplest solution to the problem is therefore to apply hatch patterns in Meander-form, which consist of parallel lines only, that fill the desired domain. However, this approach results in a highly oriented final structure, which can only be by changing the Meander-orientation in different slices. Some of the problems caused by this orientation are:

- highly anisotropic parts (Figure 1),
- discontinuous nozzle- or toolpath,
- voids due to minimum curve radii and areas which cannot be reached by Meander-like patterns and
- patterns of the surface structure/roughness (surface- or microstructure).

A solution to this problem is a path with segments of random orientation, which can completely cover the domain to be processed and which does not self-intersect. Space Filling Curves (SFC) such as the Hilbert SFC have been used in the past [1] to build parts with the FDM process. But until now these processes were limited to highly regular shaped objects and not applicable for arbitrary boundaries.

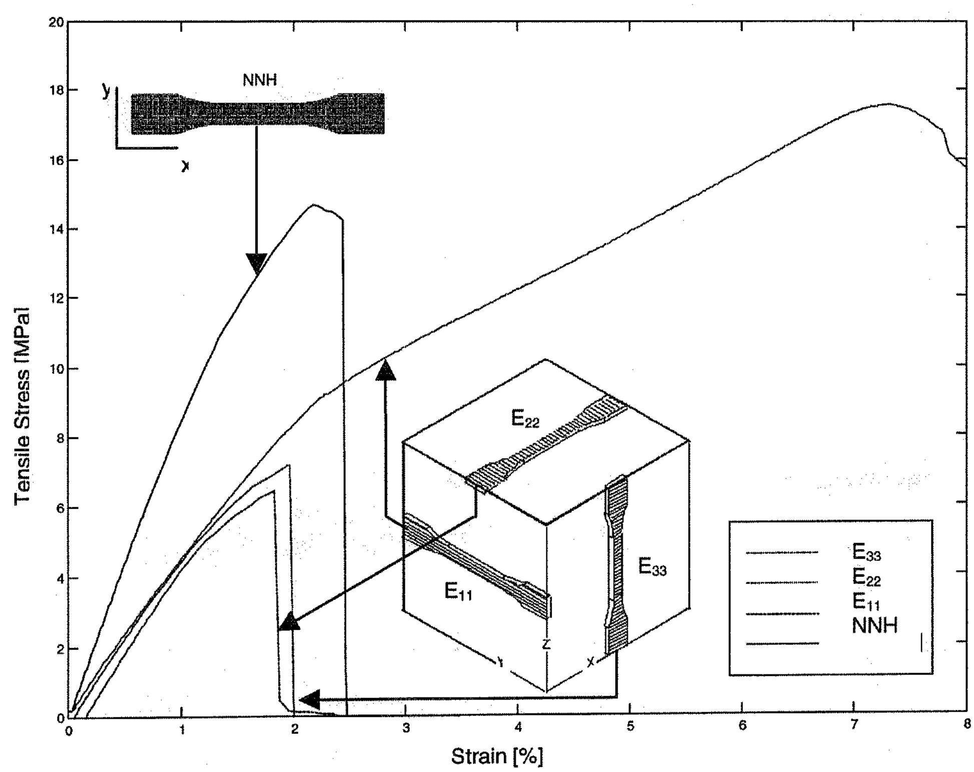

Figure 1: Stress-Strain Diagram for Uni-Directional (E_{11}, E_{22}, E_{33}) and Nearest Neighbor Heuristics (NNH) Tool-Path

With the introduction of underlying equidistant grids that are able to fill arbitrary boundaries, it is possible to get a number of nodes in each plane, which must be visited by the nozzle or tool. The spacing of the grid for the FDM-process is based on the roadwidth of the deposited material. For the Stratasys FDM 1650 a minimum roadwidth of 0.508 mm has been measured. The schedule of the visit must fulfill certain criteria, which are similar to the definitions for Eulerian and Hamiltonian tours in graph theory [5]:

1. all nodes have to be visited,
2. no node must be visited more then once,
3. minimal distance between nodes,
4. shortest possible total path length (sum over distances),
5. there may be no intersecting path elements,
6. only vertical and horizontal moves and
7. only in the nodes at the boundary diagonal moves are allowed.

The computational solutions for the problems with the aforementioned requirements are based on the Traveling Salesman Problem (TSP) [3,5]. The general TSP is not limited to points on equidistant grids but allows weight factors to discribe the "distances" between nodes or better criterias. Therefore the TSP is also adressing problems beyond Euclidean planes or even problems were the distance between AB isnot the same as the distance from BA (i.e. the distance matrix is not symetric).

3 Solutions for the Traveling Salesman Problem

State of the art solutions to the Traveling Salesman Problem are based on solutions to optimization problems. After a first random tour that visits all nodes only once and allows intersections, the total length (sum of distances) is optimized, using the fact that a shorter tour always exists when crossings are still present [5]. Very simple optimization algorithms are based on node or edge insertion algorithms. Better quality algorithms are based for example on Simulated Annealing or the Lin-Kernighan algorithm. Instead of using random tours at the beginning of optimization algorithms it is also possible to use heuristics to develop start up solutions. The described Maze algorithm is based on Nearest Neighbor Heuristics and makes use of Rotation Operations.

3.1 Simple Optimization Algorithms

Node insertion algorithms select nodes whose position in the tour schedule is changed. The nodes to be changed and their new position are chosen completely randomly and unguided. Edge insertion functions are similar to the simple node insertion function except, that two nodes get selected. Then the order of all points in between gets reversed. And the new edge is inserted in front of another selected node of the remaining tour. For small problems, this random procedure is able to achieve optimum solutions. But it also can happen that the algorithm becomes trapped in local minima. With an increase in problem size (number of nodes) the improvement in unguided random selection decreases asymptotically. Therefore all of the better algorithms focus on either guiding the solution finding process or on solving the problem of local minima.

3.2 Simulated Annealing

The algorithm starts at a high-energy (cost or initial tour) configuration. Then the simulation chooses random neighboring configurations called Careful Annealing in physical systems and Simulated Annealing in optimization problems. If the neighboring configurations result in an improvement in the Energy State (tour length), then the simulation is updated, in order to achieve the Ground State or the Optimal Solution [2,3]. Until now this procedure matches the simple optimization algorithm. However by introducing a probability factor which judges if a configuration is defined as the new state even if this neighbor is not better then his origin configuration, the algorithm solves the problem of being trapped in local minima. This probability factor is also continuously upgraded during iterations [2,3].

Problematic during the use of Simultaneous Annealing is the number of nodes; even with long running times there is the possibility that the Optimal Solutions intersect, since intersections are not optimized. Laarhoven and Aarts [2] recommended to run the whole algorithm as often as there are nodes, in order to secure an optimal tour safely. This increases the running time by the number of nodes. When the initial configuration is constructed by nearest neighbor heuristics, it is noticed that the annealing process first worsens the configuration and cannot guarantee an optimal solution.

3.3 Lin-Kernighan Algorithm

Lin and Kernighan introduced their improvement heuristic in 1971 [4]. Their objective was to improve the toolpath of a numerically controlled laser used for cutting holes. In order to improve the cycle time, their focus was on the positioning of the laser and not on the cutting process, since the laser was only burning holes with equal diameters. While Simulated Annealing focused on the problems of trapped situations, Lin and Kernighan started with guiding the acquisition of neighboring solutions. The main idea was that since the single distances caused in their sum the total cost or length of each configuration, an optimization algorithm should find better positions for nodes with very long single distances. Therefore Lin and Kernighan used guiding criteria in order to reach an optimal solution. Throughout the literature the Lin-Kernighan algorithm is recognized as the heuristic with the most promising results. In order to solve the problem of getting trapped in local minima and to achieve better running times, Lin and Kernighan found the best solutions by using five nodes simultaneously in each improvement loop. Similar algorithms are called k-optimization algorithms whereby k is the number of nodes in each loop. The improvement rate or time gain over all other processes is noticed throughout all problem instances, but similar to all discussed heuristics this solution rate is highly asymptotic (Figure 2).

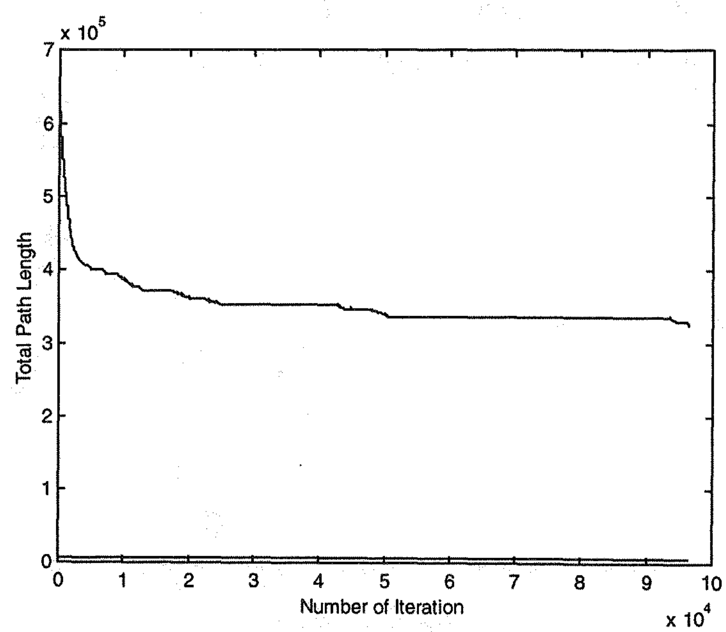

Figure 2: Total path length over the numbers of iteration for a problem with 10597 nodes (tensile-bar).

3.4 Heuristics for initial configurations

Instead of optimizing a complete random initial configuration, heuristics can be used to find better startup solutions. Both Insertion Heuristics and Nearest Neighbor Heuristics have been implemented and evaluated. Since the implementation of Nearest Neighbor Heuristics fulfill the requirements of non-self intersecting pathelements, an algorithm using rotation operations has been designed to create complete toolpaths.

3.4.1 Insertion Heuristics

Starting with an initial randomly chosen tour containing only a small amount of nodes, the final path is obtained by adding new nodes to this tour according to certain criteria. The starting tour consists of three nodes, which thus had as a Hamiltonian tour the minimal possible total path length. The new nodes were selected either randomly or by the criteria of maximal (Figure 3) or minimal distance to the rest of the already configured nodes. After their selection the nodes were included in the tour such that their position was a minimal addition in the total path length. These heuristics can create very good initial configurations in short time, since the running time is O(n), and they can create or introduce tour solutions that are similar to meanders (there are long quasi-parallel sections); however intersections still occur.

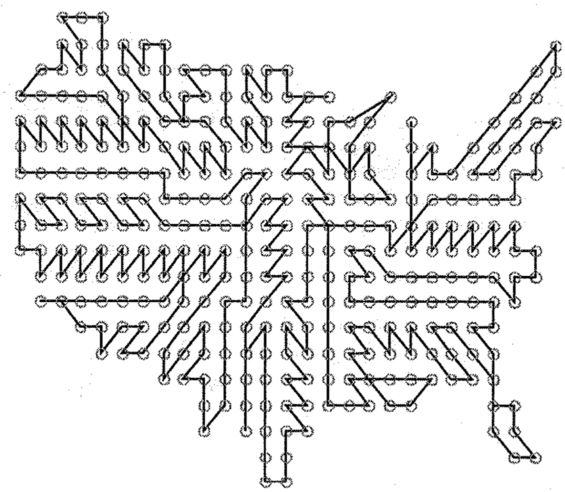

Figure 3: Tour consisting of 325 nodes achieved by Farthest Insertion Heuristic.

3.4.2 Nearest Neighbor Heuristics

In these algorithms a single random node is selected and from this starting node, the entire tour is constructed by adding one new node at a time. The objective for each new node is that it has a minimal distance to the current end point of the tour. For the equidistant grid this means that there are theoretically four possible connections But there are only up to three new possibilities, since the last connection reaching this point cannot be opened or lost. The main problem is that this algorithm gets trapped in areas, where all of the surrounding nodes are already part of the tour configuration.

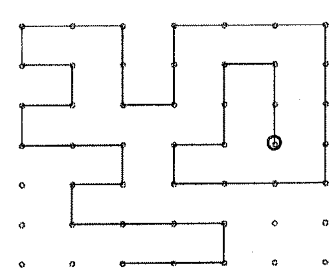

Figure 4: A dead end.

In other words the last node of the tour has reached a dead end (Figure 4).

The solution for this dead end is to allow the algorithm to connect randomly or guided to a free node. But the results are not acceptable as intersections occur in the generated toolpath. However, if this algorithm is to be used for finding a first initial tour that will be optimized by another algorithm, the selection of free nodes in this fashion is tolerable. In fact the quality of the tour configuration is already so good that the subsequently implemented optimization algorithms always lead to longer total path lengths.

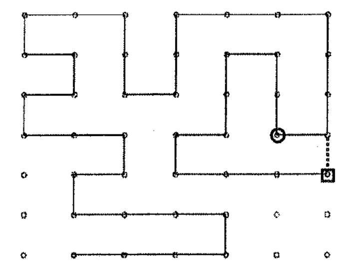

Figure 5: Rotation Operation

Instead of "jumping" in the tour to another free node, it is also possible to rotate the connections. The Operations that changed the configuration shown in Figure 4 to the configuration in Figure 5 were:

1. selecting a node which is already part of the tour configuration,
2. opening its connection with the next node in the tour schedule,
3. turning the scheduled order of points around for those nodes, which were freed from the tour schedule and
4. connecting the trapped node (circle) from before to the selected node.

With these changes a new tour end is achieved and the Nearest Neighbor Heuristic starts again. An example in Figure 6 shows the broad variation in a solution for a 1,300 nodes problem. A simple distance algorithm guides the selection of nodes where the connections are being broken up to the nearest free nodes. And in the final version, the algorithm allows diagonal moves at the boundary. On the Stratasys FDM 1650, several tensile bars were built, in which the toolpath was configured by the Nearest Neighbor Heuristic. One slice of a standard tensile bar includes 10,597 nodes. The subsequently

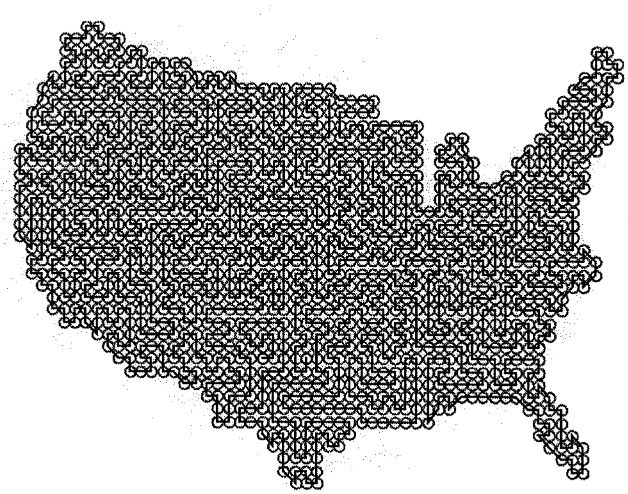

Figure 6: 1,300 node tour achieved through Nearest Neighbor Heuristic.

conducted Stress-Strain tests revealed a higher Elastic modulus (Figure 7). The built parts had a higher density and their fracture behavior was close to injection molted ABS parts. The noticeable corner in the graphs is caused by the start of delamination in the bars.

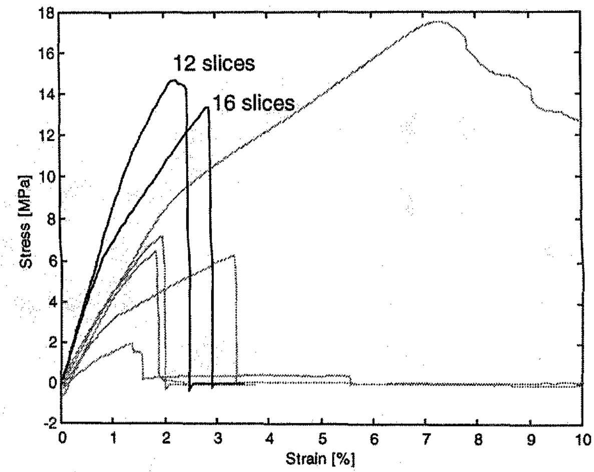

Figure 7: Stress-Strain Tests of bars using Nearest Neighbor Heuristic and unidirectional bars.

4 Conclusions

The quality achieved with the new tool-paths measurable in density (amount of voids) and Elastic modulus proves the importance of the implementation of the new buildstyle patterns. Figure 8, a representative 1,300 node task, shows the CPU Time involved for the solution finding. CPU Time has such an importance, since for instance in FDM for every slice the TSP has to be solved. This makes the Nearest Neighbor Heuristic approach with the use of rotation operations, as described by Reinelt [5] and Paragraph 3.4.2., the most feasible method for generating a toolpath in Solid Freeform Fabrication (SFF).

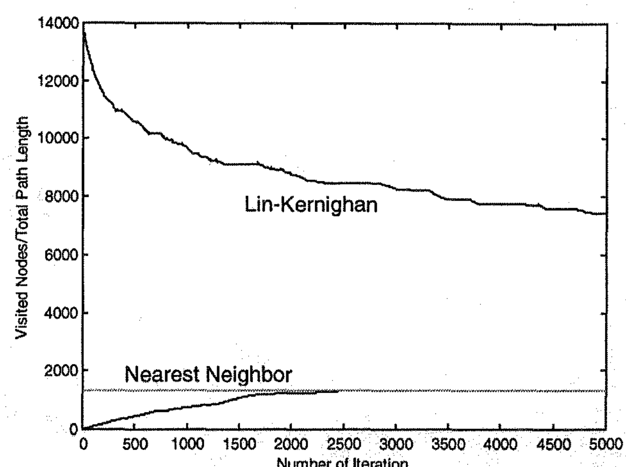

Figure 8: Progress of algorithms over time.

5 References

[1] M. Bertoldi, M. A. Yardimci, C. M. Pistor, S. I. Güceri, "Domain Decomposition and Space Filling Curves in Toolpath Planning and Generation," University of Illinois at Chicago, 1998.

[2] P. J. M. van Laarhoven, E. H. L. Aarts, "Simulated Annealing: Theory and Applications," D.Reidel Publisching Company: Dordrecht, 1987.

[3] Kunwoo Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley: Reading, Seoul, 1999.

[4] S. Lin, B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem," Bell Telephone Laboratories: Murray Hill, 1971.

[5] Gerhard Reinelt, "The Traveling Salesman: Computational Solutions for TSP Applications," Springer-Verlag: Berlin, Heidelberg, 1994.