

Improving the evaluation performance of space-time trellis code through STTC visualization tool

H. Harun¹, U. A. I. Ungku Chulan², K. Khazani²

¹Aerospace Engineering Department, Engineering Faculty, University Putra Malaysia, 43400 Serdang, Selangor, Malaysia

²Developing Reality, 22A-1, Jalan 7/7, Seksyen 7, 43650 Bandar Baru Bangi, Selangor, Malaysia

Email address:

harlisya@eng.upm.edu.my(H. Harun), proactive.azmi@gmail.com(U. A. I. U. Chulan), proactive.khamizon@gmail.com(K. Khazani)

To cite this article:

H. Harun, U. A. I. Ungku Chulan, K. Khazani. Improving the Evaluation Performance of Space-Time Trellis Code through STTC Visualisation Tool. *International Journal of Science, Technology and Society*. Vol. 1, No. 1, 2013, pp. 19-23.

doi: 10.11648/j.ijsts.20130101.13

Abstract: In this paper we present a new visualisation approach in the effort of improving the evaluation strategy of space-time trellis code (STTC) generator matrix G . To our knowledge, although visualisation is widely used to handle a variety of problems, it has never been employed specifically to solve complexity problems that are related to generator matrix G evaluation. Most approaches are either mathematically or algorithmically inclined. As such, they tend to offer a series of refinement that enhances the current available method, but do not provide fresh insight on the problem at hand. By comparing it with the enhancement strategy that was discovered via the normal approach (i.e., by analysing algorithm) it was discovered that visualisation had inspired an entirely different pruning technique that outperformed the common approach by 20%.

Keywords: Space-Time Trellis Code, Generator Matrix, Heuristic, Visualisation Tool

1. Introduction

Great effort has been directed toward the search for optimal STTC code and the reduction of computational burden in calculating the minimum determinant. Fukuda *et al.* [1] and Lisya *et al.* [2] have developed high-speed and improved algorithm which is based on tree structure in calculating the minimum determinant effectively in obtaining optimal generator matrix G . As for now the improvement of the algorithm is done via the analysis of the pseudocode. This is not a productive way of understanding the actual impact of the design that is heuristically inclined. In the attempt of finding an effective solution of improving the performance of the current approach, the potential of visualisation is explored.

2. Visualisation

Visualisation is not a new approach in engineering. It has been employed widely in the process of analysing algorithms [3]. To our knowledge however, the effort to enhance the evaluation of generator matrix G in STTC has never been approached with visualisation.

To improve the design of evaluation, most approaches [4] rely on the analysis of algorithm, equation, matrix etc. As such they are mathematically or algorithmically inclined. This is only expected given the nature of the problem.

The analysis of heuristic algorithms could face the challenge of visibility. As such, analysing the behaviour via the pseudocode per se, may not reveal the actual behaviour of the algorithm. This is due to the fact that heuristic algorithms are not completely deterministic, where the algorithm can progress in a direction totally unforeseen by analysis.

Visualisation can help in resolving this visibility issue. In fact, it can contribute to the solution finding process in three aspects [5]:

2.1. Measuring the Impact of Change

Knowing the impact of changing a certain parameter is not an easy task. Visualisation can offer a concrete and controllable display of impact, in which it can interact at any desired level-of detail. There are a number of existing visual representations, which include scatterplots, parallel coordinates and starplots, tree visualizations, node-link diagrams and to name a few [6]. Although it is possible to do research by relying entirely on the theoretical foundation

and the algorithm per se, a quick and effective way of seeing the impact of making a particular innovation to a heuristic search algorithm is by visualizing the tree generated during the search process. Thus, in this paper tree visualizations representation has been used to measure the impact of change of the parameters.

2.2. Testing of What-if Scenarios

Once the impact of change has been understood, it is possible to test varying what-if scenarios without much complication and get the result immediately. The key of devising better solutions to search depends on how well it is understood. Research has shown the potential of visualization tool in learning the behavior of search under varying situations [7].

The motivation behind the development of a visualization tool goes beyond aesthetic functionality. Heuristic search for example, is useful for complex problem space but it cannot be analyzed theoretically. Instead, the heuristic search can only be observed empirically via visualization tools [8].

2.3. Simplification of Processing

Visualization tools can assist in the extraction of novel knowledge for a particular discipline that is not achievable previously [9]. Reason being, the complexity of information is simplified with visualisation. This way, it is easier for us to process it.

As some may argue, certain problems are just not innately visual. It must be remembered however that a design problem that is mathematical or algorithmic by nature need not remain that way. It is possible to re-frame [10] the mathematically-algorithmically inclined problem (MIP) into a visually inclined problem (VIP) that is more stimulating perceptually, as shown in Figure 1.

Visualising the problem instead of just analysing the equation can promote an entirely different perspective on it. For instance, it could make research more sensitive to patterns that are readily apparent through visual means [11] but completely cryptic when analysed algorithmically.

3. Dimension of Analysis

The essential reason of using a visualization tool when dealing with heuristic search lies in its capacity to act as a bridge that connects our cognitive perception of the heuristic algorithm with the actual computation that occurs during the search process. This is highly imperative given the nature of heuristic search that is not entirely deterministic through our conception.

To analyse the heuristic search based on the visualisation tool, these dimensions or features are given the emphasis [8]:

3.1. Size of Search Space

The most fundamental way of analysing the efficiency of

search is by studying the size of the search space. Search space corresponds to the amount of effort performed by an algorithm to search for the solution given a particular problem. The space expands in proportion to the number of trials made by the algorithm, as well as the mistakes. In effect, the higher the efficiency, the smaller the space would be.

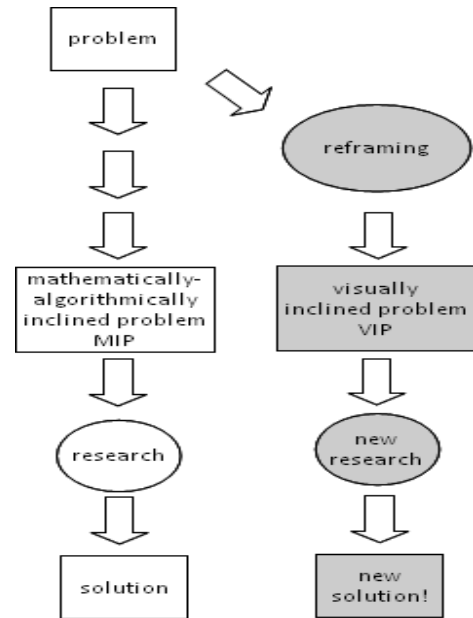


Figure 1. Effect of Re-framing.

3.2. Space Structure

The space structure is defined by its branching factor and depth. Branching factor is the number of branches made by a particular node during the search process. Depth on the other hand, refers to how deep the search must progress to find a solution or cease from trying. Two other measures are also important. They are:

3.2.1. Depth-node Count

Depth-node count is the total number of nodes expanded for each level of the search tree. It reflects when the search is most active in term of tree level. The count is considered good when it is high at earlier level of the tree. This implies that the algorithm discriminates potential paths from trivial ones at early stages of the search instead of later.

3.2.2. Time Trace Representation of Node Count

Time trace shows the progression of search. Depth first search shows higher progression comparatively. Breadth first search shows slower progression.

3.3. Frequency Domain Analysis

Another aspect of search that is crucial is the extent of its backtracking. Frequency domain analysis shows how far search backtracks when reaching a dead end. Depth first search has higher frequency than breadth first search when it comes to backtracking. It implies that depth first search

is more focused in the search process.

3.4. Problem – Search Space Evolution

The space evolution is the change undergone by the search space when the input is changed. In order to analyse the evolution of a tree in comparison to their input, a series of similar trees can be analysed. Here, the concept of tree similarity or tree congruence is really imperative. The congruence of a tree can be measured via the external nodes shared by subtrees [12]. It is also possible to analyse congruence by studying the structural differences between trees [13, 14, 15]. Other approaches include the usage of an index to make the comparison process more economic [16].

The aforementioned methods can be materialised by visualising the search with a two dimensional tree that is highly interactive.

4. STTC Visualisation Tool (SVT)

STTC visualisation tool (SVT) is an application that translates the evaluation of the generator matrix G in STTC into a visual structure that allows researchers to interact directly with the elements that determine the minimal determinant of the tree. SVT was developed with JAVA and an open source 3rd party library JUNG 2 [22].

SVT promotes a variety of interaction to occur [17], which includes zooming, abstracting, highlighting, separating etc. Users can immediately see the impact of a particular generator matrix G on the complexity of search without being bogged down by details.

The tool is used in a cyclic manner, as shown in Figure 2, where different generator matrix G is repeatedly generated and processed to uncover interesting patterns within the tree. This is performed using a simplified version of the P-Set Model [18] where variations of the generator matrix G are iteratively explored and the corresponding visualisations recorded.

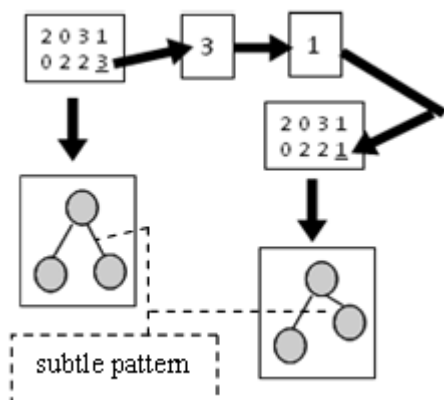


Figure 2. Iterative Exploration.

The patterns of different constructs for the generator matrix G can be gathered (as in Figure 2) by studying the

evolution of the search tree from the visual aspect. For instance, the search tree is first generated for the generator matrix G [2 0 3 1 ; 0 2 2 3]. To gain insight on how the pattern could change, an element within the initial generator matrix G is changed from 3 to 1. The search tree for the new generator matrix G [2 0 3 1 ; 0 2 2 1] is then compared to the previous one to capture any subtle patterns that are inherently difficult to discover through analysis.

In using SVT, the main objective is to identify the patterns [19] that can assist the development of a high performance algorithm. That is, to become aware of insights that were previously elusive [20].

Knowing what to focus is a challenge in using any visualisation tools. In our context, emphasis is given to the patterns of interest that practically lies within the regions of the tree where:

4.1. Solutions are Found Most Frequently

The region where solutions are frequently found is rich with potential insights. This is where most observation should focus on, to unveil the possible causality patterns that can direct the search more effectively towards the solution without unnecessarily traversal.

4.2. Solutions are Found Earliest

Fast solutions are highly useful in providing insights on the estimation of the initial upper bound for the search tree.

4.3. Pruning is Employed Most Frequently

Frequently pruned regions can suggest a way of refining the pruning strategy such that pruning can be performed earlier to a small number of shallow branches instead of a high number of deep branches.

An important fact to note is that visualisation is partially experimental. There is no guarantee that it can improve the process of discovering better solutions for improving the evaluation approach of STTC generator matrix G . However, it posits a compelling alternative to the current way of perceiving analysis as being a completely sequential process.

5. Case Study

In the following, the impact of approaching the problem with visualisation is illustrated. The first tree is generated via the original algorithm (Figure 3) that evaluates the minimal determinant of a generator matrix G [1]. It is quite apparent that the tree spans a large search space.

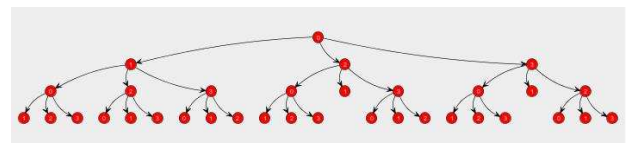


Figure 3. Original Algorithm.

To improve the original approach, the common method of research (i.e., mathematical-algorithmic analysis) was employed. The algorithm of the original approach was analysed to identify the areas of which it can be further enhanced.

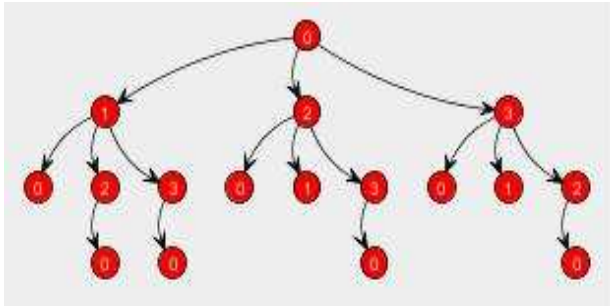


Figure 4. Improved Inspired by Analysis of Algorithm.

As shown in Figure 4, with the improved approach, the search space was reduced by approximately 50%. By analysing the original algorithm, the last layer pruning was introduced [2, 21].

The improvement inspired by analysing the algorithm was initially rather impressive. However, with visualisation, the best enhancement was discovered. By analysing the visual growth of the tree i.e., comparing the structural differences [13, 14, 15] between trees of different constructs of generator matrix G , an entirely new pruning strategy, called the optimal substructure pruning was developed. It can reduce a huge portion of the search space (approximately 70%) with only 3.7% of risk. This is shown in Figure 5.

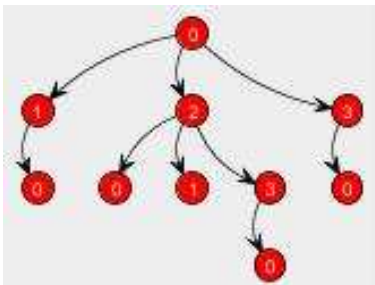


Figure 5. Improvement Inspired by Visualisation.

5. Conclusions

The discovery made by researching the original algorithm through visualisation is superior to the one done with algorithm analysis. For instance, the improved algorithm that was developed through algorithm analysis reduced the search space by approximately 50% as compared to the $\approx 70\%$ reduction achieved by the enhancement found through visualisation. It is quite intriguing to acknowledge the fact that when the same problem was analysed visually, the research was suddenly provided with a set of new and empowering insights that were not previously visible with the mathematically or algorithmically inclined analysis. This had given rise to

the formulation of an entirely different pruning strategy that enabled a more efficient reduction.

Acknowledgements

The authors would like to acknowledge UPM for supporting this research through grant number 91955 entitled “Investigation and Implementation of the Viterbi - MLSE STTC Decoder for Aeronautical Applications”.

References

- [1] T. Fukuda, S. Otsu, Y. Tokunaga and H. Zhao, “A realization of determinant criterion for STTC design,” Proc. of the 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2008), July 2008, pp. 61–64.
- [2] H. Harun, “An improved algorithm for fast evaluation of space-time trellis code (STTC) generator matrix”, Ph.D Thesis, Faculty of Engineering, University Malaya, Kuala Lumpur, Malaysia, June 2010, pp.1–204.
- [3] C. Demetrescu, I. Finocchi, G. F. Italiano, and S. Naeher, “Visualization in algorithm engineering: tools and Techniques,” in Dagstuhl Seminar on Experimental Algorithmics, Springer Verlag, 2002.
- [4] S. Nitinawarat, and N. A. Boston, “Complete analysis of space-time group codes,” appeared in Proceedings of the 43rd Annual Allerton Conference on Communication, Control, and Computing, 2005.
- [5] S. Carpendale, “Evaluating information visualizations,” in Information Visualisation: Human-Centered Issues and Perspective, Springer, 2008, pp. 19–45.
- [6] N. Elmqvist and J-D. Fekete, “Hierarchical Aggregation for Information Visualization: Overview, Techniques and Design Guidelines”, IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 3, 2010, pp. 439–454.
- [7] T. L. Naps, “JHAV'E – Supporting Algorithm Visualization”, IEEE Computer Graphics and Applications, vol. 25, no. 5, 2005, pp. 49-55.
- [8] Y. Kuwata and P. R. Cohen, “Visualization tools for real-time search algorithms,” Computer Science Technical Report 93-57, 1993.
- [9] D. J. Johann et al., “Novel approaches to visualization and data mining reveals diagnostic information in the low amplitude region of serum mass spectra from ovarian cancer patients”, Disease Markers, vol. 19, no. 4-5/2003,2004, pp. 197-207.
- [10] P. Linkow, “What gifted strategic thinkers do,” Training & Development”, vol. 53, no. 7,1999, pp. 34–37.
- [11] M. Tory and T. Moller, “Human factors in visualization research,” IEEE Transactions on Visualization and Computer Graphics, vol. 10, no. 1, 2004, pp. 72–84.
- [12] H. V. D. Parunak, “Don't link me in: set based hypermedia for taxonomic reasoning,” in Proc. of the Third ACM Conference on Hypertext (Hypertext'91), ACM Press, Dec. 1991, pp. 233-242.

- [13] Y. Zhong, C. A. Meacham and S. Pramanik, "A general method for tree-comparison based on subtree similarity and its use in a taxonomic database," *Biosystems*, vol. 42, no. 1, 1997, pp. 1–8.
- [14] T. N. W. Nye, P. Liò and W. R. Gilks, "A novel algorithm and web-based tool for comparing two alternative phylogenetic trees," *Bioinformatics*, vol. 22, no. 1, 2006, pp. 117–119.
- [15] W. N. Wan Zainon and P. Calder, "Visualising phylogenetic trees," in *Proc. Seventh Australasian User Interface Conference (AUIC 2006)*, pp. 145-152.
- [16] D. M. De Vienne, T. Giraud and O. C. Martin, "A congruence index for testing topological similarity between trees," *Bioinformatics*, vol. 23, no. 23, 2007, pp. 3119–3124.
- [17] J. S. Yi, Y. Ah Kang, J. T. Stasko and J. A. Jacko, "Toward a deeper understanding of the role of interaction in information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, 2007, pp. 1224–1231.
- [18] T. J. Jankun-Kelly, K. L. Ma, and M. Gertz, "A model and framework for visualization exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 2, 2007, pp. 357–369.
- [19] R. A. Amar and J. T. Stasko, "Knowledge precepts for design and evaluation of information visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 4, 2005, pp. 432–442.
- [20] C. Plaisant, "The challenge of information visualization evaluation," In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, ACM Press, 2004, pp. 109-116.
- [21] H. Harun, K. Dimiyati and U. A. I. Ungku Chulan, "Optimal Generator Matrix G", *Elsevier Aerospace Science and Technology*, vol. 24, no. 1, 2013, pp. 136-140.
- [22] "Java Universal Network/Graph Framework", obtained via URL: <http://jung.sourceforge.net/>.