Perspex Machine IV: Spatial Properties of Computation

AI has its many debates about the nature of computation, about the role of symbols and analog signals, about the relationship between the physical world and computational representations. The perspex machine aims to cut through the Gordian knot of these debates by combining geometry with computation, and relating these directly to the world. This provides, at least, a virtual machine that can exploit the geometrical properties of any computation, and which gives quantitative predictions about how the geometry of space limits any form of symbolic computation. This is a radical departure for AI in which all computations are spatial. It throws up new ways of computing things, and challenges accepted ideas about computation.

The perspex machine arose from the unification of the Turing machine with projective geometry.{{{2}}} In essence, certain geometrical objects were identified with the program tape and the states of a finite state machine, and certain geometrical transformations were identified with the operations of the Turing machine. This gave a constructive proof of how to make a Turing machine out of geometrical stuff.

The constructive proof guarantees that any Turing program can be compiled into a neural network. A C source to perspex compiler has been implemented in Pop11. The compiler does a lexical analysis of the C source, performs a recursive descent parse, then generates perspexes that are the data and operations specified by the C source. Initially, the compiler templates for data and operations were exactly the templates provided by the constructive proof, but these were soon adapted to provide a more convenient implementation of C's arithmetic operations, conditionals, loops, and function call and return.

A compilation of a C implementation of the Fibonacci series has the interesting property that the position of one neuron controls the number of Fibonacci terms computed. Thus, an important parameter of the program is mapped onto a spatial analogue automatically by the compiler. So far, the most complex program that has been compiled into a perspex, neural network is a C implementation of Dijkstra's solution to the Travelling Salesman Problem. The C source is about two pages long and the compiled network has about 600 neurons. If the compiler were extended to cover the whole of C then it would be possible to compile any C source. It would be possible, for example, to compile the whole of Linux into a neural network.

There is a moral here for other researchers. Develop a constructive proof of the equivalence of your favourite kind of neural network with the Turing machine then implement it as a compiler. At a stroke, this will help AI deliver massive neural networks for use in all manner of software applications. This might be useful in itself, but the perspex machine does much more than this.

For a start, the perspex machine corrects a bug in the Turing machine. The Turing machine can enter a non-deterministic state where the current symbol on its tape instructs it to enter more than one state. In this condition the Turing machine stalls until an external agency, or oracle, decides which one state to enter. By contrast, the perspex machine is always deterministic, though it can emulate this Turing non-determinism, say, by raising flags to indicate that the Turing non-determinism has been encountered. This property of the perspex machine arises from the connectivity of geometrical space and its underlying, total arithmetic.{{{1}}} This arithmetic can be used on its own to remove division by zero errors from all numerical programs, thereby creating safer and more robust software.

There is a moral here, too. If you choose not to use a total arithmetic you leave all of your software open to Turing's bug and risk your code crashing.

More profoundly, the perspex machine maps all Turing computations into geometrical stuff so that geometrical operations can be applied to them. For example, programs can be Fourier transformed and filtered so that the broadest filter band is a single neuron that approximates an entire program, and successively finer bands contain more and more neurons that, ultimately, reproduce the original program exactly.{{{4}}} This makes it theoretically possible for a compiler to construct global-to-fine processing threads for any Turing program. In other words, in theory, global reasoning can be delivered by a compiler that compiles any existing program.

And there are deeper properties too. The Walnut Cake Theorem{{{3}}} shows that, in general, when a discrete system approximates a finer discrete or continuous system it does so non-monotonically. Thus, non-monotonic reasoning is a general property of discrete machines operating in spacetime. Of course, monotonic reasoning can be had in certain special cases, but these are unrepresentative of the spectrum of computing machines that can exist in spacetime.

There is a great deal more that could be said about the perspex machine, but this must suffice. Unifying the Turing machine with geometry has produced a new class of machines, perspex machines, that describe the shape and motion of objects in the world in a natural way, one that combines symbolic and non-symbolic computation in a single machine, and one which offers geometrical methods of computation that are, theoretically, more powerful than the Turing machine. Even Turing computable simulations of the perspex machine have surprising properties that make it a very powerful virtual machine with many potential applications in AI.

James Anderson, Matthew Spanner, Christopher Kershaw
Computer Science
The University of Reading
E-mail: author@bookofparagon.com
http://www.bookofparagon.com

References

1. J. A. D. W. Anderson Exact Numerical Computation of the Rational General Linear Transformations in Vision Geometry X1, Proceedings of SPIE vol. 4794, pp 22-28, 2002.

2. J. A. D. W. Anderson Perspex Machine in Vision Geometry X1, Proceedings of SPIE vol. 4794, pp 10-21, 2002.

3. J.A.D.W. Anderson, Perspex Machine II: Visualisation in *Vision Geometry XIII*, Proceedings of the SPIE Vol. 5675, i.e., this volume.

4. J. A. D. W. Anderson Perspex Machine III: Continuity Over the Turing Operations in Vision Geometry XIII, Proceedings of SPIE vol. 5675, pp 112-123, 2005.