Interaction feasibility - test by 3D correlation

Two molecules are represented as 3D voxel grids. They rotated and shifted until the two molecules make the most contact. The score for some relative position between A and B is the sum of voxel scores:

\[ S_\text{xyz} = \sum_{i,j,k} A_{ijk} B_{i-x,j-y,k-z} \]

The 3D floating point transform code was taken from a standard reference [4], and compiled for eight-bit saturating arithmetic for two values per result voxel: collisions and surface interactions. The original implementation was forced to use integer values, initial range limits. That gave a speedup of 100×100×100 grid in the worst-case (largest) rotated configuration. Molecules were represented with two bits per voxel, representing interior/exterior and surface/non-surface regions. This used precision requirements are defined by number of maps involved in the longest scoring arc. For a critical model of size N, the FFT is also easy to compute the range of maps involved. This technique is the foundation of many parts of the fast-convolution techniques.

3D convolution - systolic array

Correlated results are obtained by matrix multiplication in a systolic array. The convolution is computed as follows:

\[ A \ast B = C \]

Where A and B are the input arrays, and C is the output array. The convolution is performed by sliding an input window over the entire input array and performing a dot product at each position. The dot product is computed by multiplying the corresponding elements of the input arrays and summing the results.

Comparison of 3D convolution with a 2D convolution:

The main difference between the 2D and 3D convolutions is that the 3D convolution requires a 3D kernel, while the 2D convolution requires a 2D kernel. This means that the 3D convolution is more computationally intensive, but also provides a higher degree of parallelism. The 3D convolution can be performed using a systolic array, which is a type of parallel computing architecture that is well-suited for this task. The systolic array is constructed by connecting a chain of processing elements, each of which performs a small part of the computation.

Results

Related Addressing Logic

The rotation logic, by itself, was implemented on a Xilinx Spartan 3E FPGA, for timing purposes [2]. Post-place and route timing estimates gave 8-10 clock cycles, or 300 MHz speeds per voxel. The required FPGA logic cells, at 1% of available logic, Clock period and gate count could have been reduced if the bit-width was reduced. The core uses two 32-bit controlled systolic array processors to achieve the required performance.

Rotation by Address Translation

For the systems Xilinx FPGA, address translation transforms were integrated. Most of this improvement came from the highly parallel convolution array, which could perform one 256×320×320 MBC operation per clock cycle. Except core steps, every clock cycle performed useful computation since there was no need to store or address rotational information. The standard implementations of the 3D Fourier transform require a separate rotation step. Initial loading of the unrotated image was amortized over the rotated model to cover the entire model. Normal procedures would create a rotated model in that buffer.

Precision requirements are defined by number of maps involved in the longest scoring arc. For a critical model of size N, the FFT is also easy to compute the range of maps involved. This technique is the foundation of many parts of the fast-convolution techniques.