Cross-correlation is a commonly used tool in the field of signal processing, with applications in pattern recognition, particle physics, electron tomography, and many other areas. For many of these applications, it is often the limiting factor on system performance due to its computational complexity. In this thesis, we analyze the cross-correlation algorithm and its optimization and parallelization possibilities. We then implement several optimizations of the definition-based algorithm, mainly focused on parallelization using the Graphical processing unit (GPU). Even though the definition-based algorithm provides many possibilities for parallelization, the implementation needs to solve several problems, such as the algorithm's low arithmetic intensity. Furthermore, the problems differ between computation types, which include cross-correlating a pair of inputs, one input with many other inputs, or many inputs with many other inputs. Lastly, we compare the optimizations of the definition-based algorithm with the asymptotically faster and commonly used algorithm based on the Fast Fourier Transform. Depending on the total size of the data, we achieve parity between the two algorithms for matrix sizes ranging from 60x60 to 150x150, allowing performance improvements for systems using matrices smaller than these sizes.