

# Automatische Paketisierung

Sebastian Hack

Universität des Saarlandes, Saarbrücken

„Packetization“ (also: Data Parallelization or SIMDfication) is the process of transforming scalar code, given by a CFG  $G$ , into a CFG  $G'$  that works on  $N$  scalar input values at once (where  $N$  is the target-architecture's SIMD width). One execution of  $G'$  is equivalent to  $N$  parallel executing instances of  $G$ . This technique is important for data-parallel algorithms, in particular applications in computer graphics such as ray tracing.

The benefit of packetization lies in the usage of SIMD instructions. To this end, a value in the scalar instance is mapped to a SIMD “packet” in the packetized version. However, each of the scalar instances may take different control-flow paths. To avoid splitting SIMD packets apart, control flow is almost completely replaced by predicated execution (only loop-back branches have to be kept).

We present a packetization algorithm and discuss its limitations. Furthermore, we present first results of using our algorithm in the shading system of a ray tracer: the programmer writes scalar code in C/C++ that is then automatically packetized. Although still lacking some optimizations, the packetized CFGs already outperform their scalar counterparts by an average factor of 3.6 using a vector width of 4.