Many image processing algorithms can be broken down into atomic operations on arrays. There are unary operations such as additive inverse, absolute value, or square-root and binary operations such as addition or multiplication. While the arrays typically have elements of a single type, this element-type may be a signed or unsigned integer with 8, 16, 32, or more bits, a floating point number, or a complex number.

A computer vision library thus needs to implement binary operations for various combinations of element-types. Furthermore binary operations can occur as array-array-, array-scalar-, or as scalar-scalar-operation. This kind of requirements make it very hard to write a computer vision library in a statically typed language such as C++. On the other hand a naive implementation in a dynamically typed programming language such as Ruby will not satisfy real-time constraints.

However recently the DotGNU project has released libJIT which is a just-in-time compiler library for i386, x86-64, and other processors. The combination of Ruby, libJIT, and other free software allows interactive development of real-time algorithms in an unprecedented way.

Previous work: [WAD07], [WADB08], [HOR]

This kind of requirements make it very hard to write a computer vision library in a statically typed language such as C++. On the other hand a naive implementation in a dynamically typed programming language such as Ruby will not satisfy real-time constraints.

However recently the DotGNU project has released libJIT which is a just-in-time compiler library for i386, x86-64, and other processors. The combination of Ruby, libJIT, and other free software allows interactive development of real-time algorithms in an unprecedented way.

Previous work: [WAD07], [WADB08], [HOR]

References

Above: A machine vision extension for the Ruby programming language.