HIPA^{cc}

A Domain-Specific Language and Compiler for Image Processing

HIPA^{cc} Heterogeneous Image Processing Acceleration

http://hipacc-lang.org
https://github.com/hipacc/hipacc

UNIVERSITÄT DES SAARLANDES
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Richard Membarth, Saarland University
Oliver Reiche, University of Erlangen-Nuremberg

Domain-Specific Language

DSL construct realized as compiler-known C++ classes:
- images & image pyramids
- filter mask / domain
- accessor
  - boundary handling
  - interpolation
- iteration space
- kernel

```
Image<uchar> in(width, height);
Image<float> out(width, height);
in = in_image;
out = out_image;

float filter_mask[3][3] = { ... };
Mask<float> mask(filter_mask);

BoundaryCondition bound(in, mask, CLAMP);

Accessor<uchar> acc(bound);

IterationSpace<float> iter(out);

GaussianBlur filter(iter, acc, mask, size);
filter.execute();

out_image = out.getData();
```

```
class GaussianBlur : public Kernel<float> {
  Accessor<uchar> input;
  Mask<float> mask;

public:
  GaussianBlur(IterationSpace<float> iter,
               Accessor<uchar> acc,
               Mask<float> mask)
    : Kernel(iter), input(acc), mask(mask)
  {  addAccessor(acc); }

  void kernel() {
    output() = convolve(mask, SUM, [&]() {
                 return mask() * input(mask);
               });
  }
};
```

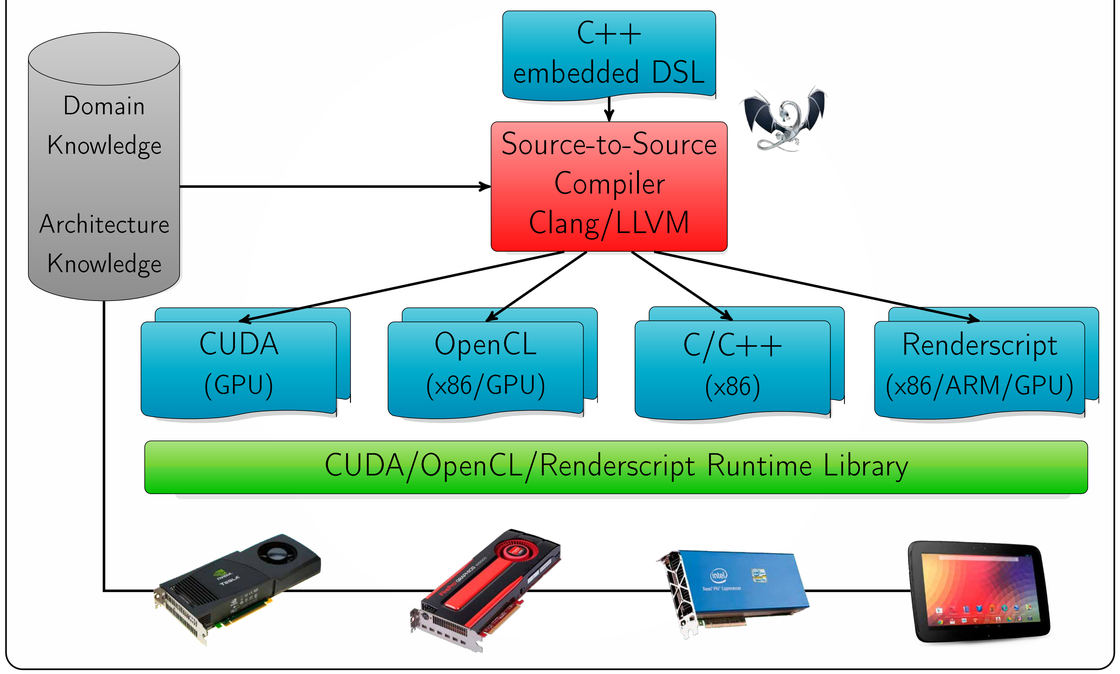
Image Manipulation

Computations on images are described in the *kernel()* and *reduce()* methods of user-defined C++ classes:
- implicit (relative) pixel access
- convolutions as C++11 lambda-function

Architecture Model

Models hardware constraints:
- registers
- shared memory
- thread pool

Determines:
- kernel configuration (tiling)
- target-specific optimizations



Optimizations

Context-specific:
- MPMD code generation
- loop unrolling
- constant propagation

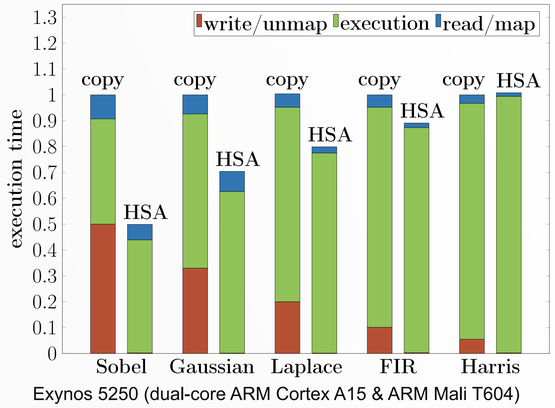
Target-specific:
- memory padding
- thread-coarsening
- memory hierarchy mapping
- vectorization

Runtime Support

Manage available resources:
- device initialization
- context management
- kernel compilation
- synchronization

Utilize hardware features:
- unified memory support
- streaming

Unified memory on Arndale Board



Exynos 5250 (dual-core ARM Cortex A15 & ARM Mali T604)

Gaussian blur filter on a Tesla K20

	Boundary Handling Mode					Speedup
	Undef.	Clamp	Repeat	Mirror	Const.	
naïve	crash	3.04	3.14	3.15	3.19	0.45×
RapidMind	5.40	6.00	crash	n/a	5.97	0.23×
Halide	n/a	4.38	n/a	n/a	n/a	0.32×
OpenCV	n/a	2.12	2.15	2.22	2.01	0.65×
NPP	2.40	n/a	n/a	n/a	n/a	0.58×
HIPA^{cc}	1.32	1.38	1.40	1.38	1.39	1.00×

image of 4096x4096 pixels; filter window size of 5x5; times in ms

Results

High productivity:
- concise and compact algorithm description

Portability:
- cross-platform support from the same high-level description

Competitive performance:
- faster than other image processing DSLs and libraries