Compiling Image Processing Applications for Many-Core Accelerators
Pierre Guillou

To cite this version:
Pierre Guillou. Compiling Image Processing Applications for Many-Core Accelerators. Journées de seconde année de l’École Doctorale, Jun 2015, Paris, France. hal-01178938

HAL Id: hal-01178938
https://hal-mines-paristech.archives-ouvertes.fr/hal-01178938
Submitted on 21 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
### Mathematical Morphology Base Operators

- Arithmetic operators
  - Unary: (pixel x parameter, 1 input image)
  - Binary: (pixel x pixel, 2 input images)
- Morphological operators
- Functions:
  - Neighbor selection: min, max, avg
- Reduction operators
  - Global: min, max, sum
  - Local: max, min
- Other operators
  - Threshold, mask, log

---

### Sigma-C, a Dataflow Programming Language

```c
agent foo() { // describe agent interface

    interface { /* ... */ }

    // describe subgraph interface

    map { /* ... */ }

    // describe subgraph bar

    subgraph bar () {

        // describe subgraph interface

        map { /* ... */ }

        // connect agents to subgraph interfaces

        agent a3 = new Subgraph3 (); // ...

        connect (a3. output , a5. input1 ); // ...

        connect (a1. output0 , a2. input );

        connect (input0 , a1. input0);

    }
}
```

---

### Optimisations

- Unrolling of converging loops
- Arithmetic operators aggregation
- Generation of kernel-specific convolutions
- Data parallelization for compute-intensive operators

---

### Results: Execution Times and Energy Consumption

<table>
<thead>
<tr>
<th>Application</th>
<th>MPPA-256 (Sigma-C, 10 W)</th>
<th>MPPA-256 (FPGA, 20 W)</th>
<th>AMD 4-core (OpenCL, 90 W)</th>
<th>Tesla C 2050 (OpenCL, 40 W)</th>
<th>Tesla C 2500 (OpenCL, 240 W)</th>
<th>GMEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar999</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>antino</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>burner</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>deblending</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>id</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>id</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>retina</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>toggle</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
</tbody>
</table>

---

### Future Work

- Other programming models:
  - Phynix/OpenMP on compute clusters, communication library between clusters
  - OpenCL, via local memory package
- Improve data-parallelism to take better advantage of the current architecture
- Implement more complex algorithms: watershed, arrow, labelling, minima, ...

---

### References