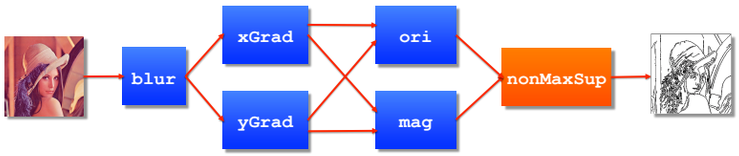
Scalable Computer Vision Applications

Rami Mukhtar, Ben Lever, and Trevor L. McDonell
{rami.mukhtar, ben.lever, trevor.mcdonell}@nicta.com.au

NICTA

High Level Description of Computer Vision Algorithms

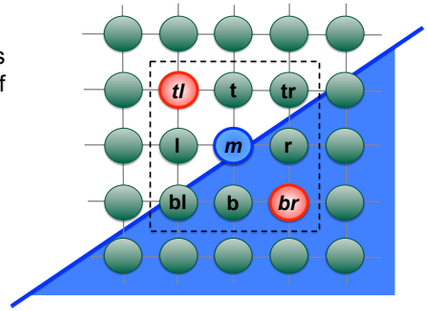
Canny Edge Detection Example




```
canny img = nonMaxSup intensity orientation
  where
    blured = blur img
    dX = xGrad blured
    dY = yGrad blured
    intensity = mag dX dY
    orientation = ori dX dY
```

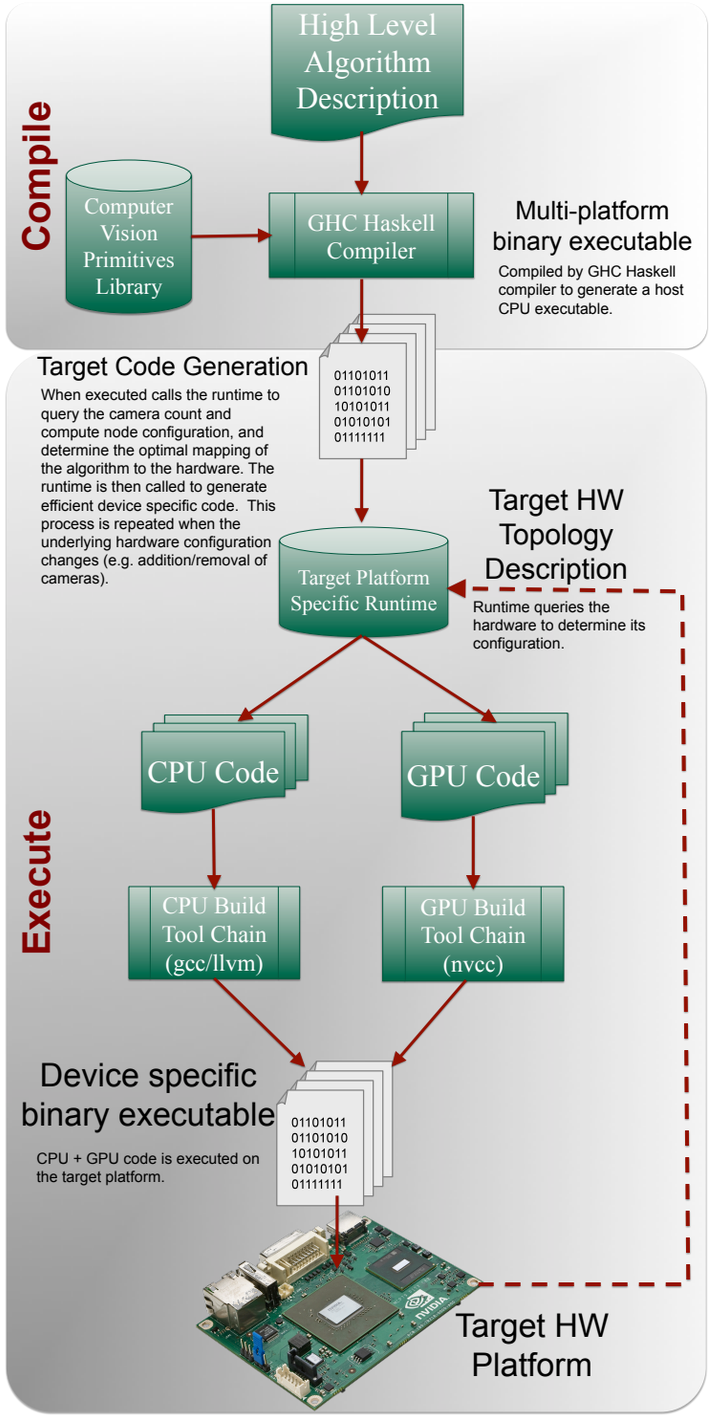
Non-Maximum Suppression

The middle pixel (***m***) is classified as an edge if its gradient intensity is greater that the intensity of the two pixels normal to its orientation (***tl*** and ***br***).



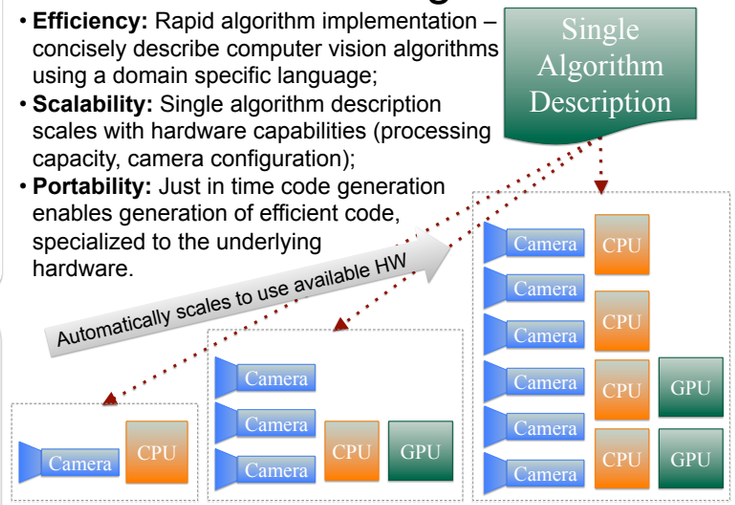

```
nonMaxSup int ori = stencil2 isMaxOri int ori

isMaxOri ((tl, t, tr),
          ( l, m,  r),
          (bl, b, br))  -- 3x3 intensity stencil
         ((o))          -- 1x1 orientation stencil
  | o == horiz   = isMax t b
  | o == vert    = isMax l r
  | o == posDiag = isMax tl br
  | o == negDiag = isMax tr bl
  | otherwise    = False
  where
    isMax a b | m < a     = False
              | m < b     = False
              | otherwise = True
```



Advantages

- **Efficiency:** Rapid algorithm implementation – concisely describe computer vision algorithms using a domain specific language;
- **Scalability:** Single algorithm description scales with hardware capabilities (processing capacity, camera configuration);
- **Portability:** Just in time code generation enables generation of efficient code, specialized to the underlying hardware.



Current Status

The project is still in an early research stage. We currently have a working 'proof of concept' framework that enables some computer vision algorithms to be described and deployed on a single camera hardware platform. Currently the algorithm designer needs to indicate which components are to be run on the GPU. The current implementation is able to make use a quad core host CPU and a single GPU. Preliminary results for the Canny edge detection example are given below.

AMD Phenom II X4 945 Quad Core 3GHz

1 CPU Core	2 CPU Core	3 CPU Core	4 CPU Core	1 CPU + GeForce GT330M GPU
1x	1.68x speedup	2.39x speedup	2.68x speedup	5.15x speedup

The above results demonstrate that a single algorithm implementation (200 lines of code) scales to use the additional hardware resources. At this early stage of the research we are not yet using shared (on-chip) GPU memory, hence we would expect the speedup ratio for the GPU case to increase significantly once on-chip memory is used.

Future Work

Our short term goals include:

- Extending the language to concisely describe a wider range of computer vision algorithms;
- Improving the efficiency of the GPU kernels for the performance primitives;
- Automating the partitioning and mapping process.

Longer term goals include:

- Language and runtime support for multiple camera inputs;
- Runtime support for distributed computation that accounts for the limitations in network capacity and bandwidth.

To track our progress go to:
http://www.nicta.com.au/research/projects/scalable_vision_machines