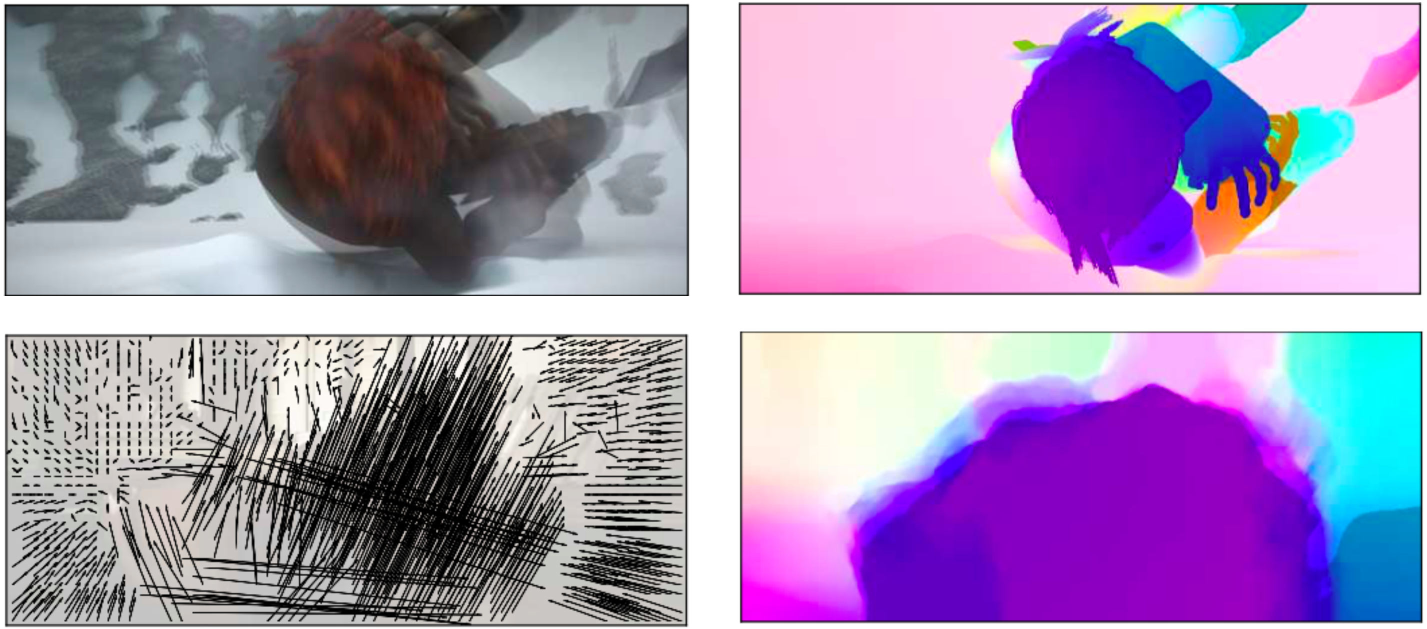
Large Displacement Optical Flow Estimation Based on Warping

SAIWEN W., YIFAN S., YIJUN P., XINYUAN Y.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Problem

Find the relative displacement of each pixel in a given pair of images.



Algorithm

$$E(\omega) = \int_{\Omega} E_D + \alpha E_S + \beta E_M \, dx$$

Data Term

Smooth term

Matching term

$$\boxed{E_D = \delta\Psi\left(\sum_{i=1}^{c} \omega^T \overline{J}_0^i \omega\right) + \gamma\Psi\left(\sum_{i=1}^{c} \omega^T \overline{J}_{xy}^i \omega\right)}$$

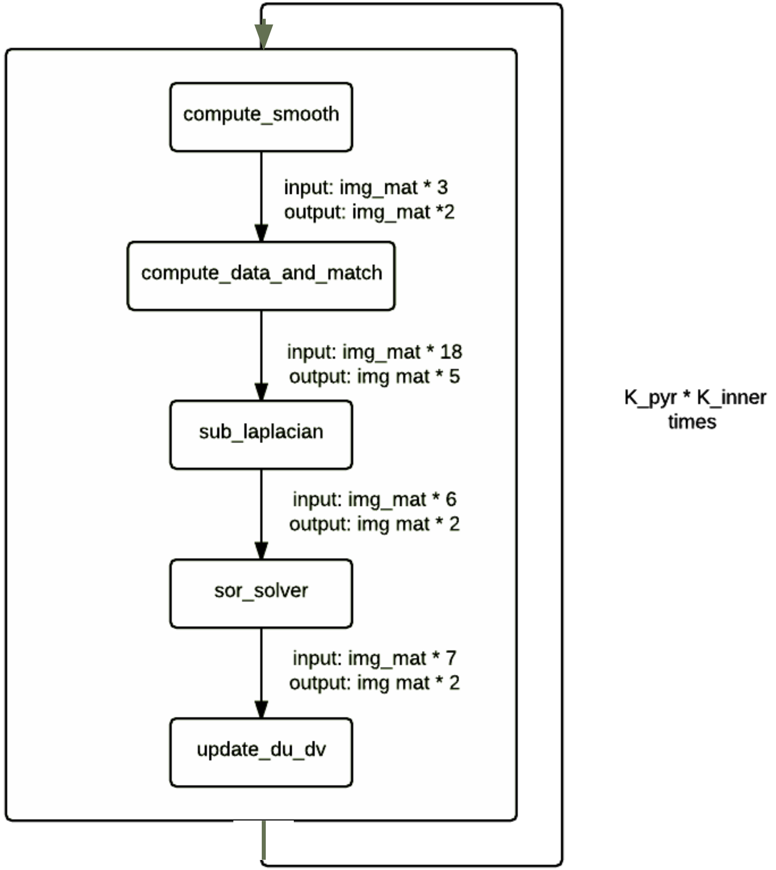
$$\boxed{E_S = \Psi\left(\|\nabla u\|^2 + \|\nabla v\|^2\right)}$$

$$\boxed{E_M = c\phi\Psi\left(\|\omega - \omega'\|^2\right)}$$

Algorithm Flow

sor_solver:
K iterations of successive over-relaxation method
Computational complexity:
    O(K_pyr * m * n * K_inner * K_solver)
Memory complexity:
    O(K_pyr * m * n * K_inner * K_solver)

compute_data and_match:
Prepare pixel-wise linear system: Ax’ = b
Computational complexity:
    O(K_pyr * m * n * K_inner)
Memory complexity:
    O(K_pyr * m * n * K_inner)



Cost Measure

Using Perf to measure cycles and memory transfer

Using Geekbench to measure peak performance of RAM
which is used in roofline model

Measure flops as (addition, multiply, division, square root)

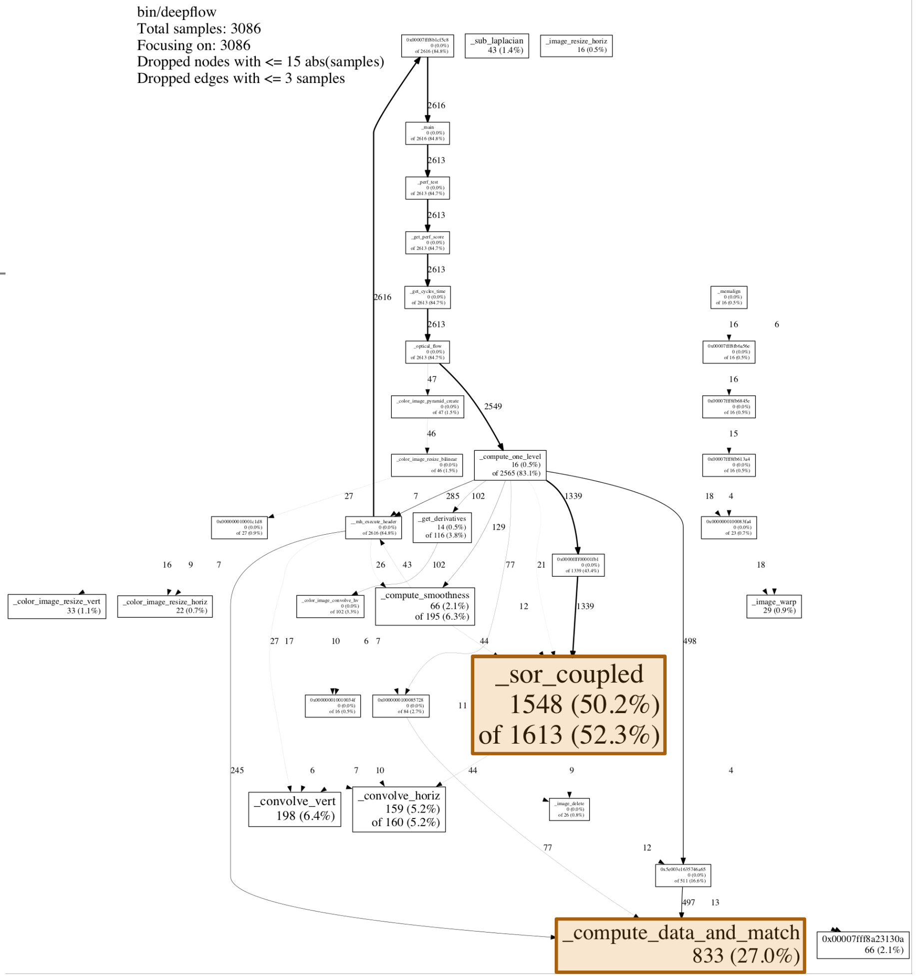
Divisions and sqrts costs 28 cycles, regard them as one flop may underestimate flop count.
Which leads to lower performance.

Bottlenecks

Using google profiler to plot calling graph

Two functions:

- Solver
- Computer data and match



Optimization Methods

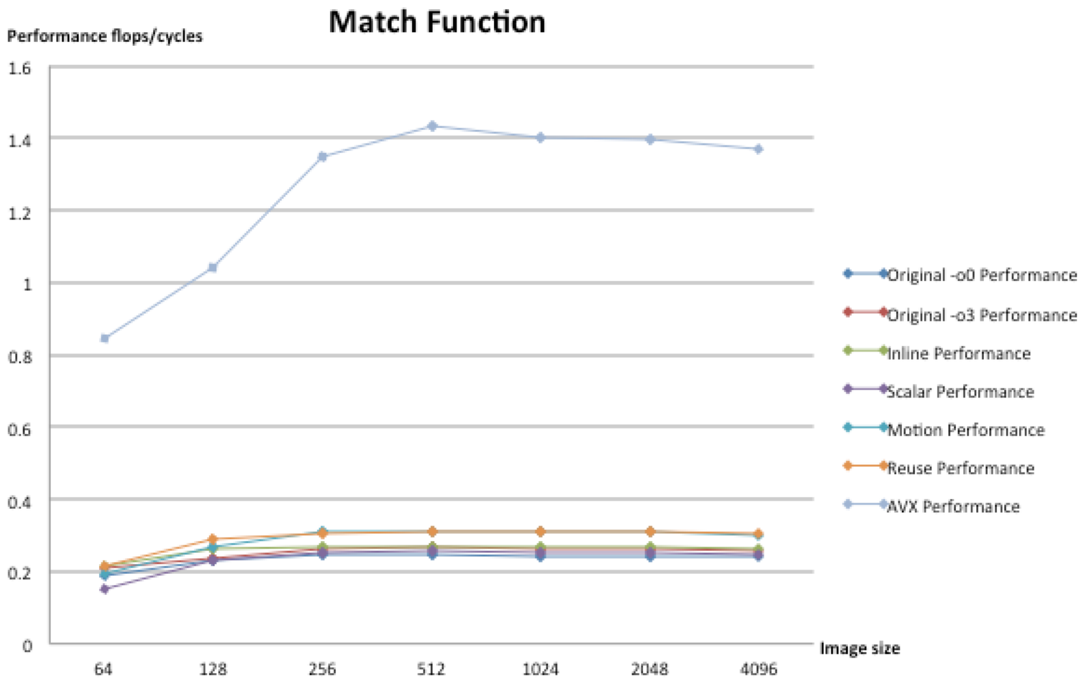
compute_data_and_match:

1. compiler flags
2. function inlining
3. scalar replacement
4. code motion
5. memory reuse
6. AVX

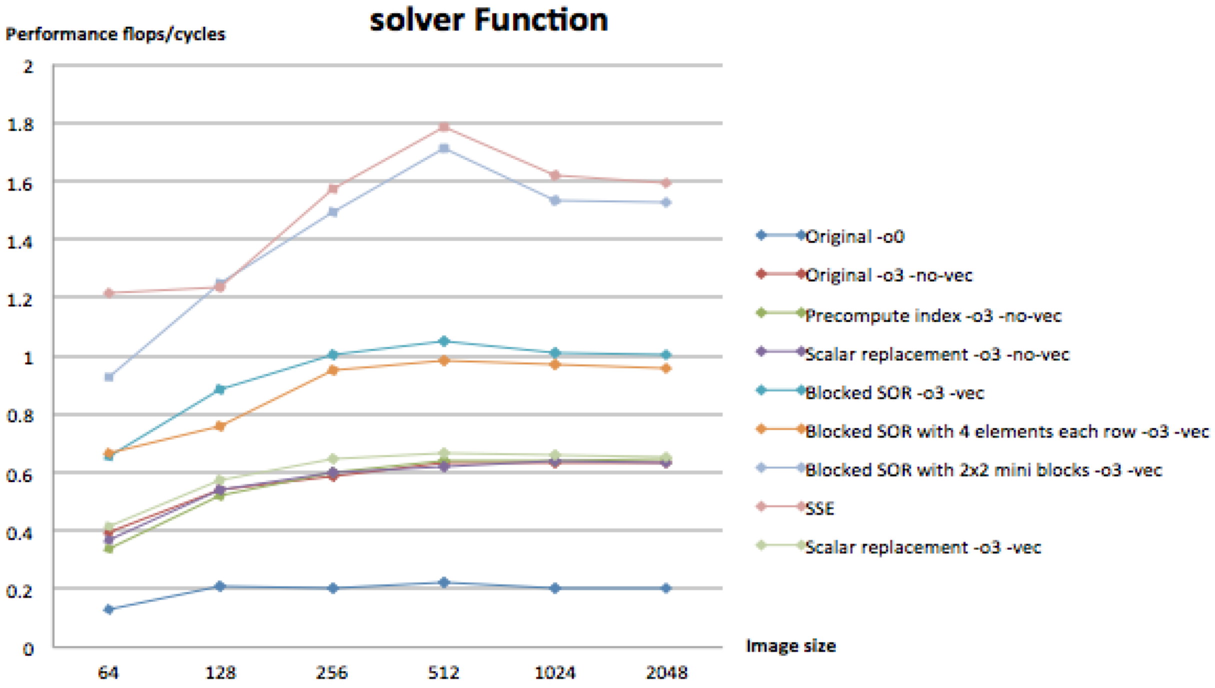
sor_solver:

1. compiler flags
2. code motion
3. scalar replacement
4. blocking (1x4, 2x2)
5. partial vectorization (SSE)

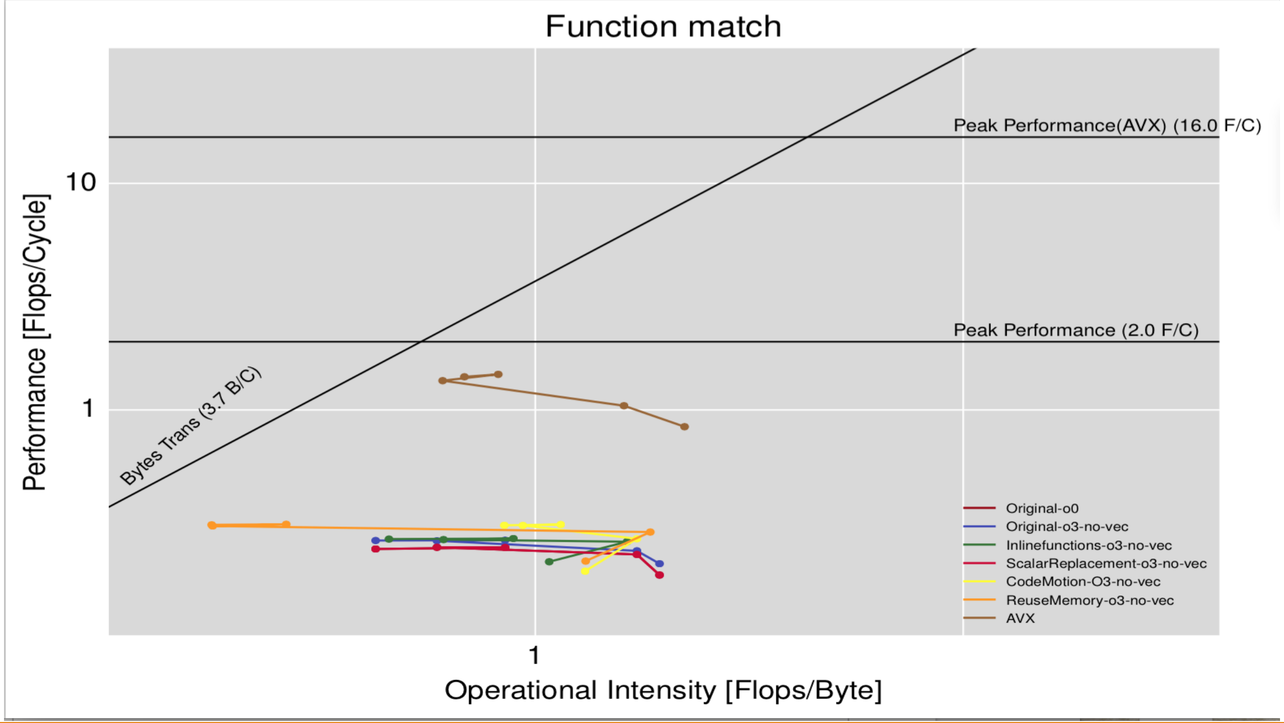
Performance Plots (match function)



Performance Plots (solver function)

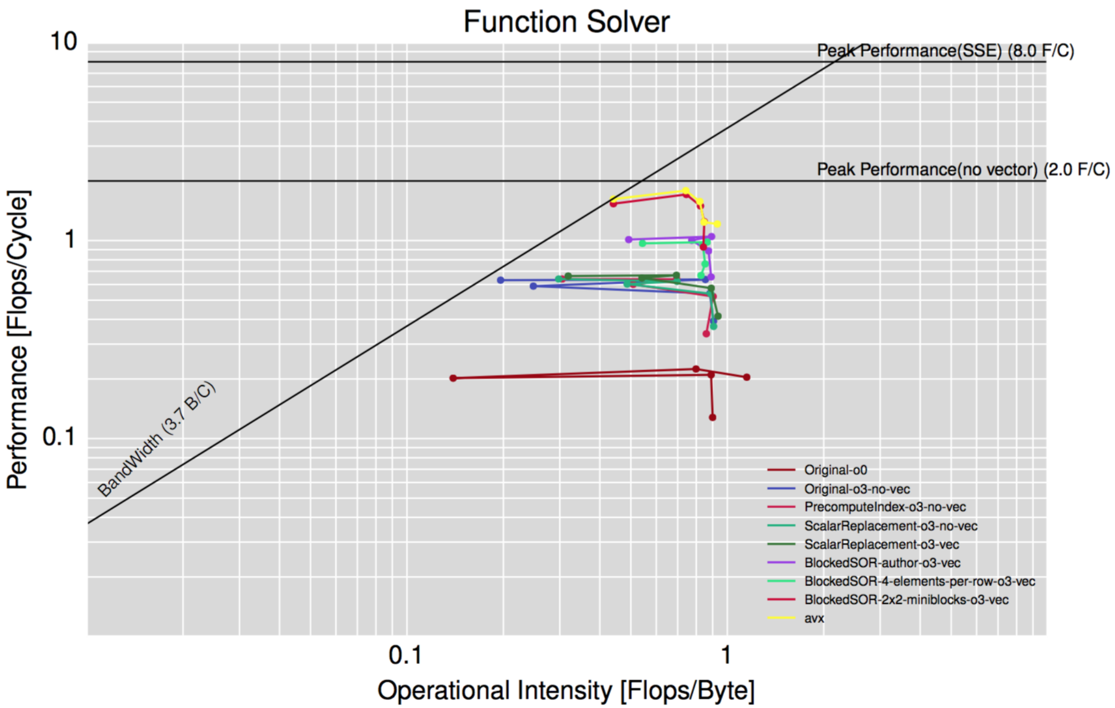


Roofline (compute_data_and_match)



Environment:
Sandy-Bridge

Roofline (solver)




how?
where?
why
asking questions
challenge
QUESTIONS
clues
ask
who?
discover
what?
Knowing
clues
how
Knowing
investigation

Theory Appendix

- Non-convex and non-linear -> incremental coarse to fine warping strategy with down sampling.

- Three loops
  - Outer loop: Move along the image pyramid.
  - Middle loop: outer layer fixed point iteration -> update flow increments and non-linear weight iteratively.
  - Inner loop: inner layer fixed point iteration -> Use successive over-relaxation method to approximate the solution.

$$u_{i,j}^{n,r+1} := (1-w)u_{i,j}^{n,r} + w\frac{\left(\left(I_1 - I_2 + I_{2_x}u_{i,j}^{n} - I_{2_y}(v_{i,j}^{n,r} - v_{i,j}^{n})\right) I_{2_x} + \alpha^2 \mathrm{A}(u_{i\pm1,j\pm1}^{n,r+1})\right)}{I_{2_x}^2 + \alpha^2},$$

$$v_{i,j}^{n,r+1} := (1-w)v_{i,j}^{n,r} + w\frac{\left(\left(I_1 - I_2 - I_{2_x}(u_{i,j}^{n,r+1} - u_{i,j}^{n}) + I_{2_y}v_{i,j}^{n}\right) I_{2_y} + \alpha^2 \mathrm{A}(v_{i\pm1,j\pm1}^{n,r+1})\right)}{I_{2_y}^2 + \alpha^2}$$

Performance Appendix

With M, N as the height and the width of the input image.

Solver

- **Cost** $41MNK_iK_mK_o$ flops
- **Data** $84MNK_iK_mK_o$ bytes
- **Operational intensity** 0.48 $\Theta(1)$

Compute data and match

- **Cost** $263MNK_mK_o$ flops
- **Data** $136MNK_mK_o$ bytes
- **Operational intensity** 1.96 $\Theta(1)$