Large Displacement Optical Flow & Applications

Narayanan Sundaram, Kurt Keutzer (Parlab)

In collaboration with
Thomas Brox (University of Freiburg)
Michael Tao (University of California Berkeley)
Parlab Stack

- Applications
- Productivity Layer
- Efficiency Layer
- Diagnosing Power/Performance
- OS
- Arch.

- Personal Health
- Image Retrieval
- Hearing, Music
- Speech
- Parallel Browser

- Design Patterns/Motifs
- Composition & Coordination Language (C&CL)
- C&CL Compiler/Interpreter
- Parallel Libraries
- Parallel Frameworks
- Efficiency Languages
- Sketching
- Autotuners
- Legacy Code
- Schedulers
- Communication & Synch. Primitives

- Efficiency Language Compilers
- Legacy OS
- OS Libraries & Services
- Hypervisor
- Multicore/GPGPU
- RAMP Manycore

- Static Verification
- Type Systems
- Directed Testing
- Dynamic Checking
- Debugging with Replay

Correctness
Video capture is cheap

- Amount of video material is increasing rapidly
- Video recording is becoming more popular
- Huge gap between video acquisition and video analysis capabilities
  - Manycore parallelism can help
Can you see the difference?

Springboard diving Vs Platform diving
Guess the sport from trajectory

- Springboard diving
- Platform diving
Motion is important

- Pre-requisite for next generation video applications
- Should track points densely and accurately over many frames.
- Optical flow provides the means to achieve good tracking

- Optical Flow involves computing the motion vectors (“flow field”) between the consecutive frames of a video

Hue indicates the direction of flow and saturation indicates the magnitude
Fast motion is very common in natural videos
- e.g. limbs in human motion, balls in sports videos
- Simple optical flow models do not handle this well.

We use the Large Displacement Optical Flow (LDOF) algorithm\[^1\]
- Crucial for accurately measuring large motion of small objects.

Incorporates both descriptor matching and optical flow in a single mathematical setting

Why should we care about optical flow runtime?

- Average Optical Flow Algorithm timing on 640x480 frame
  ~60 seconds/frame on 1 core of Nehalem

- Assume videos are 3 minutes long & algorithm scales to 4 cores linearly

- 83,400,000 videos (April 2008)
  = Only about 214,000 years

- With a cluster of 36,000 nodes
  About 6 years!

Solution: Parallelization

Let us run optical flow on all of youtube!
LDOF Application architecture

Input Frames

Histogram of Gradients (Dense matching)

Optical Flow solver (non-linear, non-convex optimization)

Output Flow field

Coarse to fine refinement

Compute Data, Smoothness, HOG Match factors

Assemble matrix A

Solve $A.x = b$

Update $du, dv$

Fixed point iterations

Preconditioned Conjugate gradient solver

Sparse Matrix-Vector Multiply

BLAS 1

Apply Preconditioner

BLAS 1

Histogram of Gradients (Dense matching)
Algorithmic exploration is a must for parallelization

- Numeric & convergence analysis essential for efficient parallelization

- Efficient linear solvers for serial and parallel platforms are different (Gauss-Seidel Vs Preconditioned Conjugate gradient)

![Graph showing squared norm of the residual vs iterations and time for convergence with different linear solvers.

- Choice of linear solver:
  - Red-black
  - Gauss Seidel
  - CG - block Jacobi preconditioner
  - CG-No Preconditioner

- Time for convergence (Residual < 10^{-2}):
  - 5x faster
  - 40% faster

Choice of linear solver

Time for convergence (Residual < 10^{-2})
Implementing efficient Sparse Matrix Vector Multiply

- Most compute intensive component

- Linear equations in $2\times \# \text{ pixels}$ variables
  - For each pixel, one variable each for $x$ and $y$ displacement

- 6 – point stencil structure, coupled equations
  - Explicitly stored matrix

- We achieve 53 SP GFlops for the SpMV computations on Fermi (GTX 280)

$du$, $dv$ refer to $x$ and $y$ displacements respectively
Results

- Runtime went down from 1 minute to 1.8 seconds
  - The original serial implementation is C++ code compiled and autovectorized using ICC running on a single thread of CPU
  - The parallel implementation uses CUDA and runs on Nvidia Fermi GPUs.

- Point tracker based on LDOF outperforms other trackers[3]
  - 46 – 66% more accurate than other state-of-the-art techniques

- Better than other algorithms and runs efficiently!

Other trackers cannot track the fast movement of the leg

LDOF is able to track fast movements

The sparse linear solver used in LDOF has been ported to the Copperhead\cite{Catanzaro2011} framework.

We see huge productivity improvements with 70% of hand-coded performance c.f. Bryan’s talk on Copperhead coming up next.

Preconditioned Conjugate Gradient

\[
\begin{align*}
    r_0 & := b - Ax_0 \\
    z_0 & := M^{-1}r_0 \\
    p_0 & := z_0 \\
    k & := 0 \\
    \text{repeat} & \\
    \alpha_k & := \frac{r_k^T z_k}{p_k^T A p_k} \\
    x_{k+1} & := x_k + \alpha_k p_k \\
    r_{k+1} & := r_k - \alpha_k A p_k \\
    \text{if } r_{k+1} \text{ is sufficiently small then exit loop end if} \\
    z_{k+1} & := M^{-1}r_{k+1} \\
    \beta_k & := \frac{r_{k+1}^T z_{k+1}}{r_k^T z_k} \\
    p_{k+1} & := z_{k+1} + \beta_k p_k \\
    k & := k + 1 \\
\end{align*}
\]

Getting the word out

- Work published at ECCV, 2010
- Available online since October 2010 at [http://www.eecs.berkeley.edu/~narayans/Software.html](http://www.eecs.berkeley.edu/~narayans/Software.html)
- Downloaded 109 times so far and is being used in
  - University of Freiburg
  - nVidia
  - Harvard University
  - Georgia Tech
  - and other places
What are the applications that need optical flow?
Video Object Segmentation

Segment video objects from a video sequence

Useful for video editing and video understanding

Running on a GPU cluster at NERSC
Video aesthetics

Identify aesthetically pleasing videos from several views of the same scene

(Data collection and user study underway)
Activity recognition

Categorize complex activities like sports
Summary

- We have designed & implemented superior optical flow and tracking algorithms that are
  - More accurate
  - Faster
  through numerical analysis, algorithmic exploration to drastically improve its applicability

- Video applications are helping, and are in turn helped by productivity frameworks

- We are applying our technology to a wide variety of video processing tasks including video object segmentation, aesthetics and action recognition
Thank You

Questions?
References


