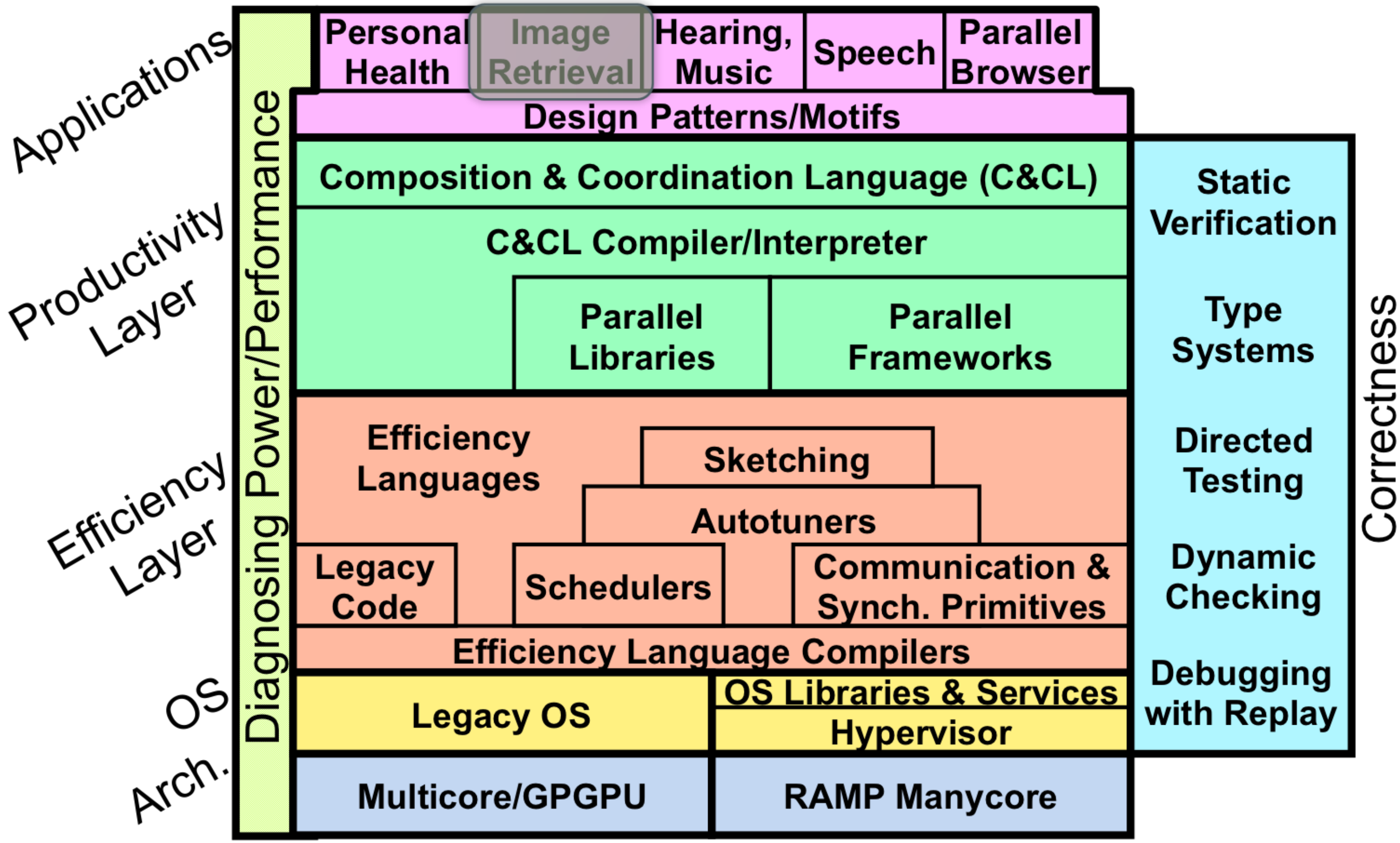
Large Displacement Optical Flow & Applications

Narayanan Sundaram, Kurt Keutzer (Parlab)

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

Parlab Stack



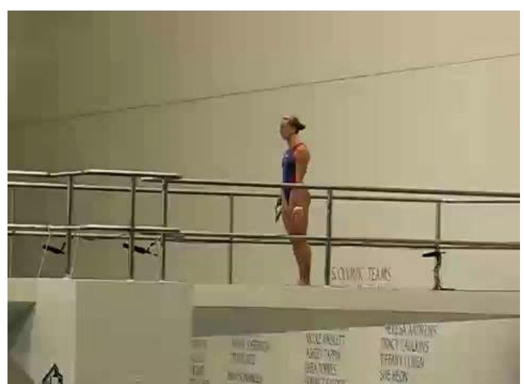
Video capture is cheap

- 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

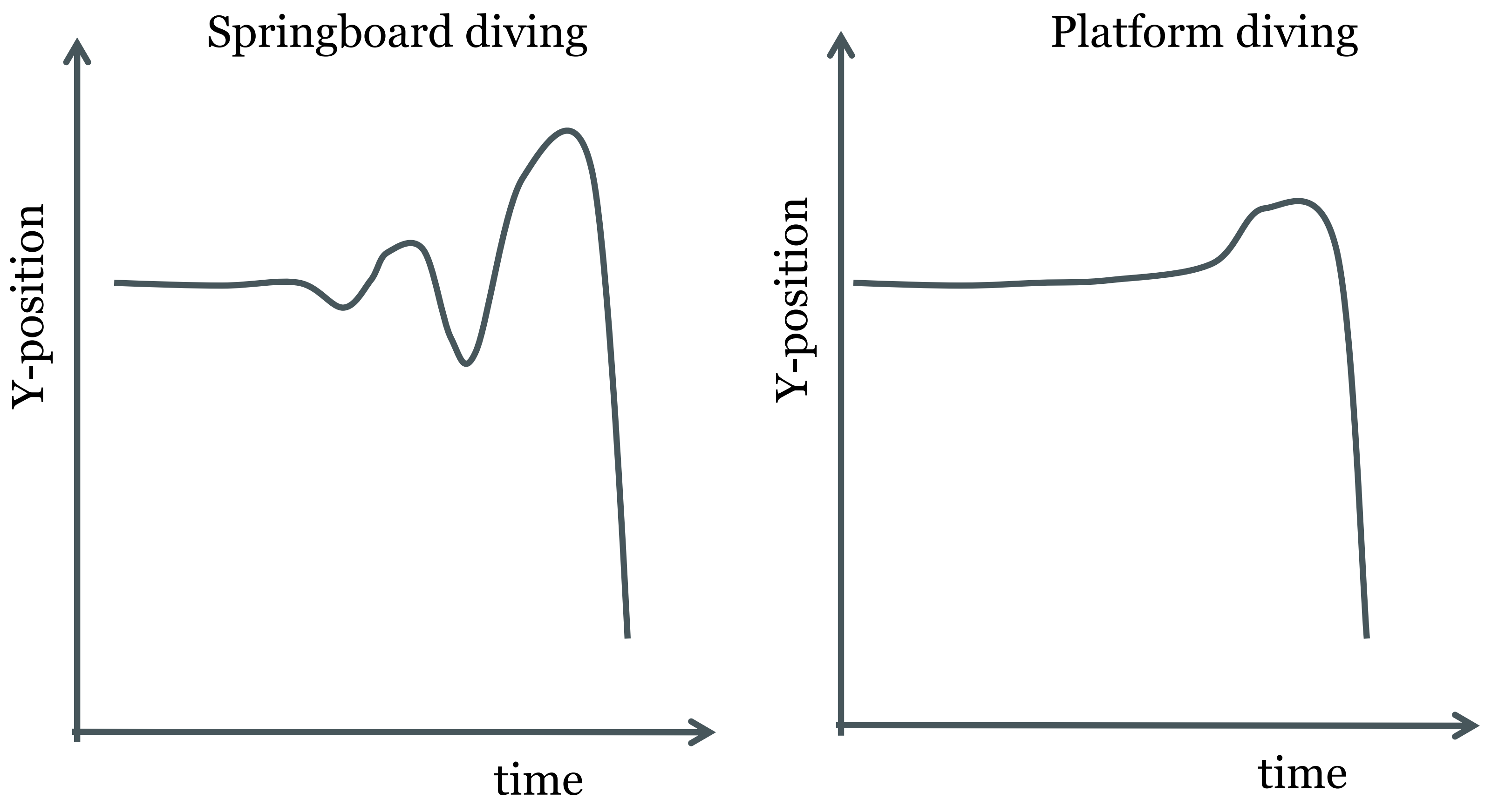
Can you see the difference?



Springboard diving Vs Platform diving

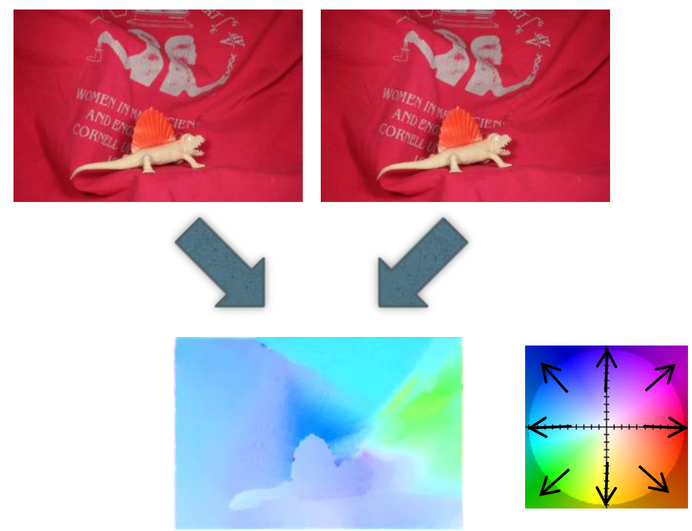

Guess the sport from trajectory



Motion is important

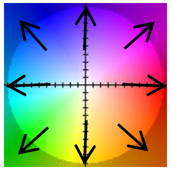
- Pre-requisite for next generation video applications
- Should track points densely and accurately over many frames.
- Optical flow provides the means to achieve good tracking

- 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

Large Displacement Optical Flow



Traditional Optical Flow

With LDOF

- 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

[1] T. Brox, J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation", IEEE Transactions on Pattern Analysis and Machine Intelligence, to appear.

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)

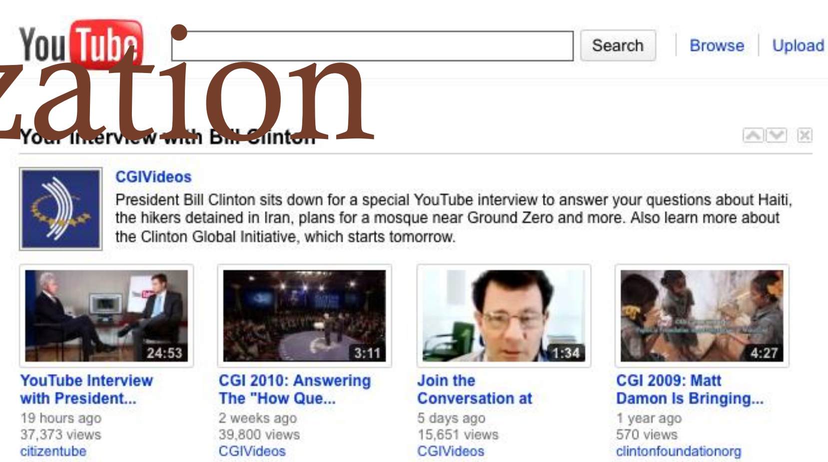
= Only about **214,000** years

- With a cluster of 36,000 nodes

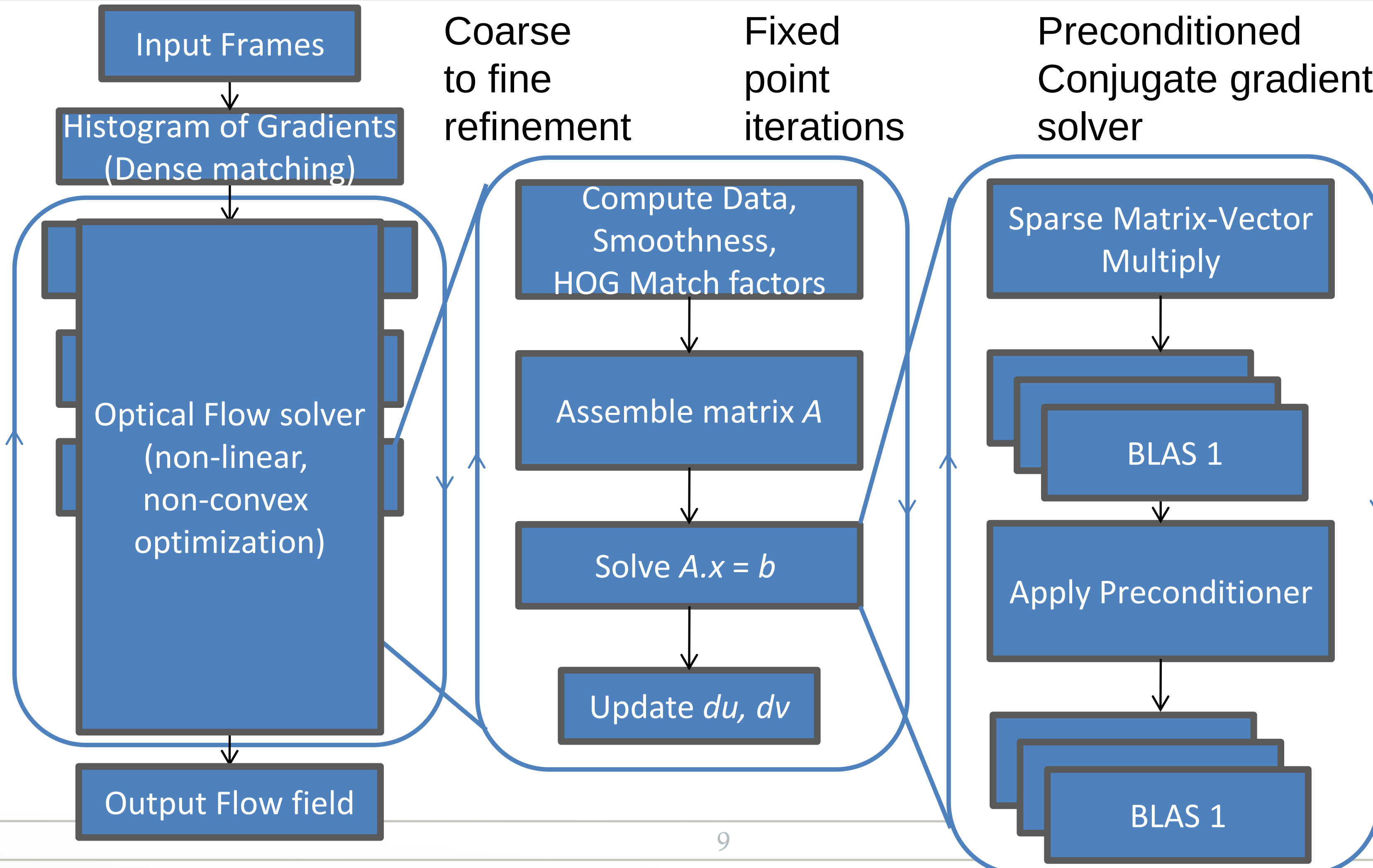
About **6** years!

Solution : Parallelization

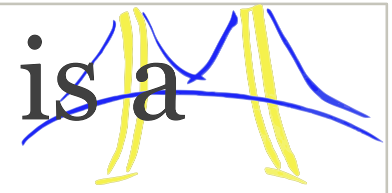
Let us run optical flow on all of youtube!



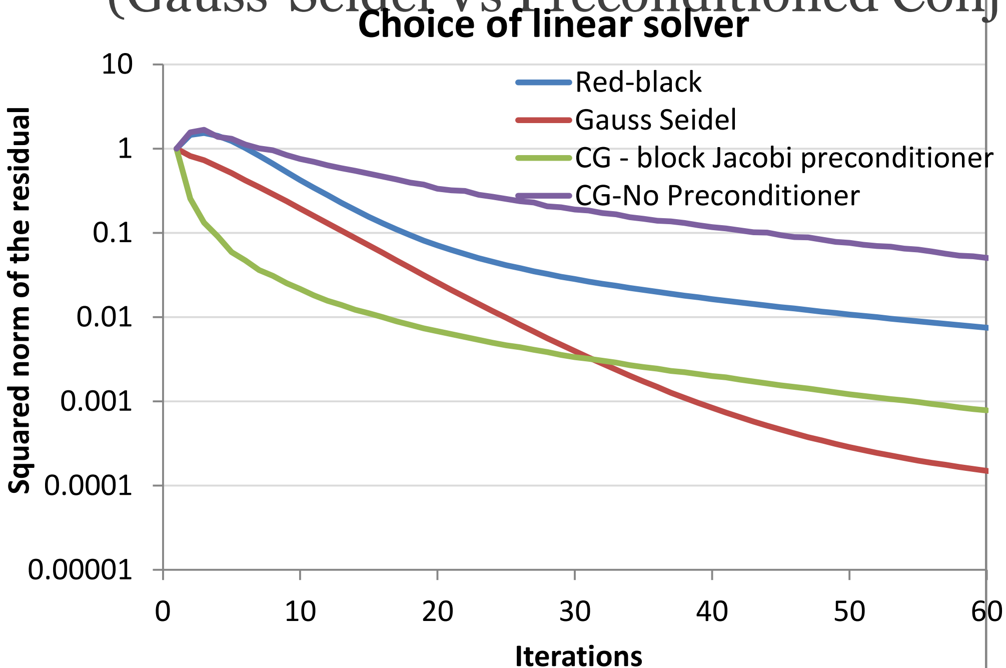
LDOF Application architecture

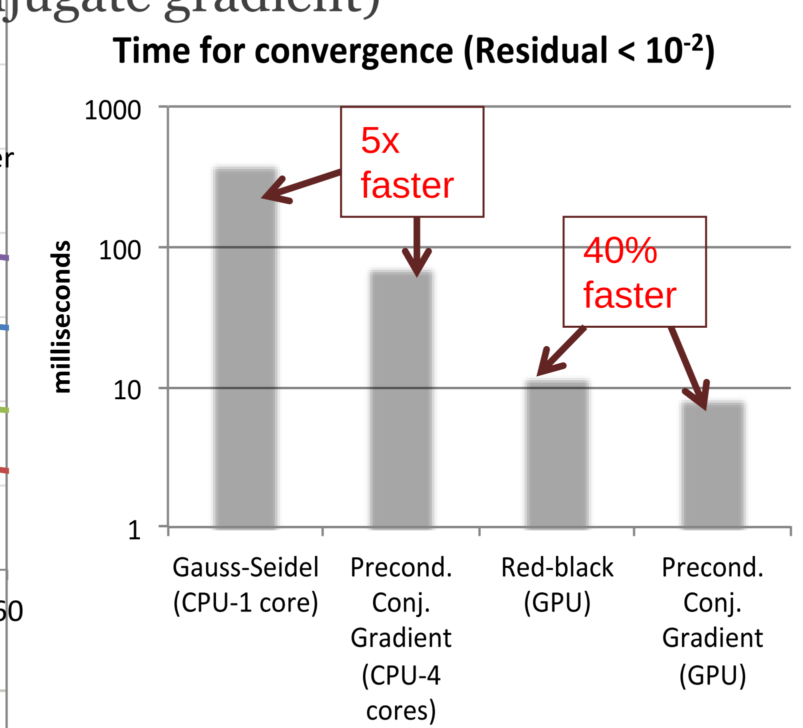


Algorithmic exploration is a must for parallelization



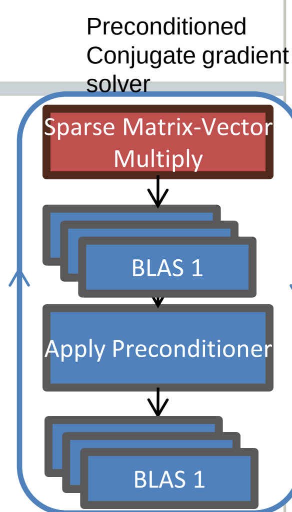
- Numeric & convergence analysis essential for efficient parallelization
- Efficient linear solvers for serial and parallel platforms are different (Gauss-Seidel Vs Preconditioned Conjugate gradient)

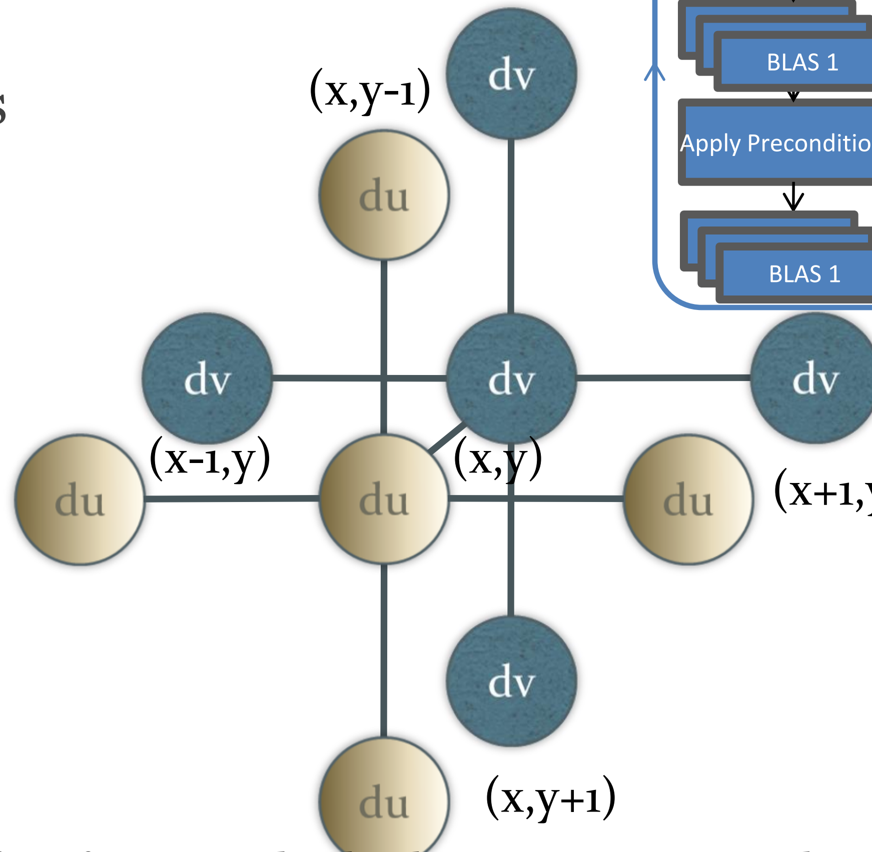




Implementing efficient Sparse Matrix Vector Multiply

- ❑ Most compute intensive component
- ❑ Linear equations in 2*# 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)

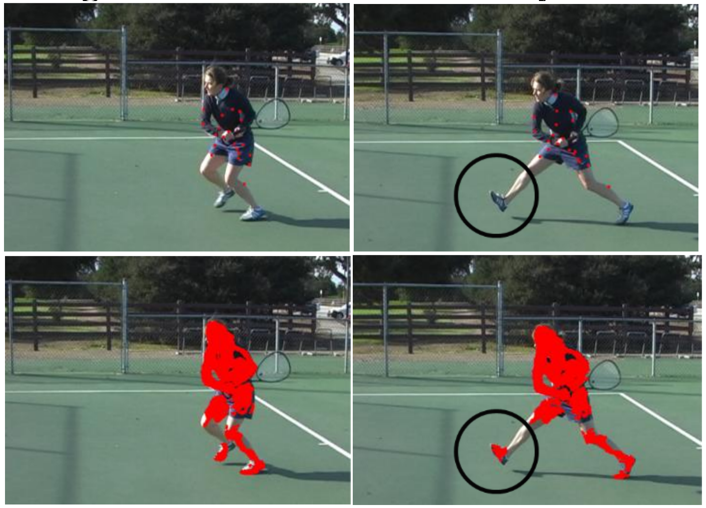




du, dv refer to x and y displacements respectively

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!



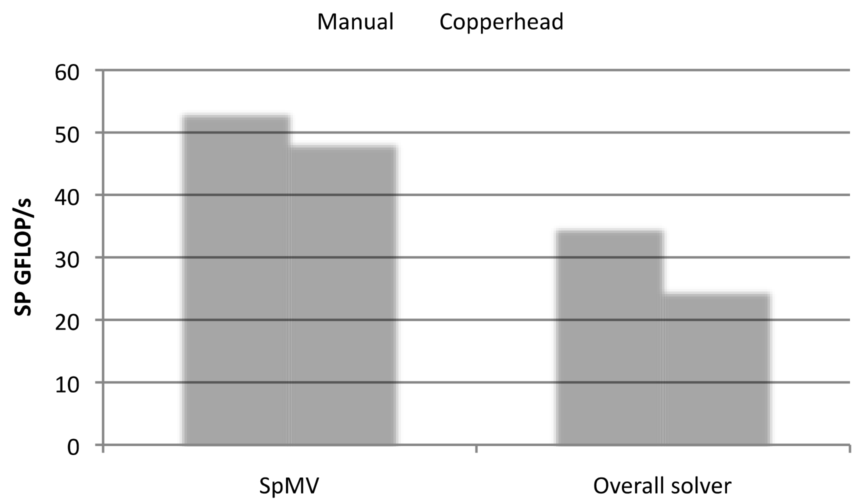
Other trackers cannot track the fast movement of the leg

LDOF is able to track fast movements

[3] Narayanan Sundaram, Thomas Brox, Kurt Keutzer, "Dense Point Trajectories by GPU-accelerated Large Displacement Optical Flow", European Conference on Computer Vision (ECCV), September 2010

Integration in Parlab stack

- The sparse linear solver used in LDOF has been ported to the Copperhead[3] 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

$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$

$\mathbf{z}_0 := \mathbf{M}^{-1}\mathbf{r}_0$

$\mathbf{p}_0 := \mathbf{z}_0$

$k := 0$

repeat

$$\alpha_k := \frac{\mathbf{r}_k^{\mathrm{T}}\mathbf{z}_k}{\mathbf{p}_k^{\mathrm{T}}\mathbf{A}\mathbf{p}_k}$$

$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{p}_k$

$\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k$

if $\mathbf{r}_{k+1}$ is sufficiently small **then** exit loop **end if**

$\mathbf{z}_{k+1} := \mathbf{M}^{-1}\mathbf{r}_{k+1}$

$$\beta_k := \frac{\mathbf{r}_{k+1}^{\mathrm{T}}\mathbf{z}_{k+1}}{\mathbf{r}_k^{\mathrm{T}}\mathbf{z}_k}$$

$\mathbf{p}_{k+1} := \mathbf{z}_{k+1} + \beta_k \mathbf{p}_k$

$k := k + 1$

end repeat

[3] B. Catanzaro, M. Garland, and K. Keutzer. Copperhead: Compiling an embedded data parallel language. PPoPP 2011

Getting the word out

- Work published at ECCV, 2010
- Available online since October 2010 at 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

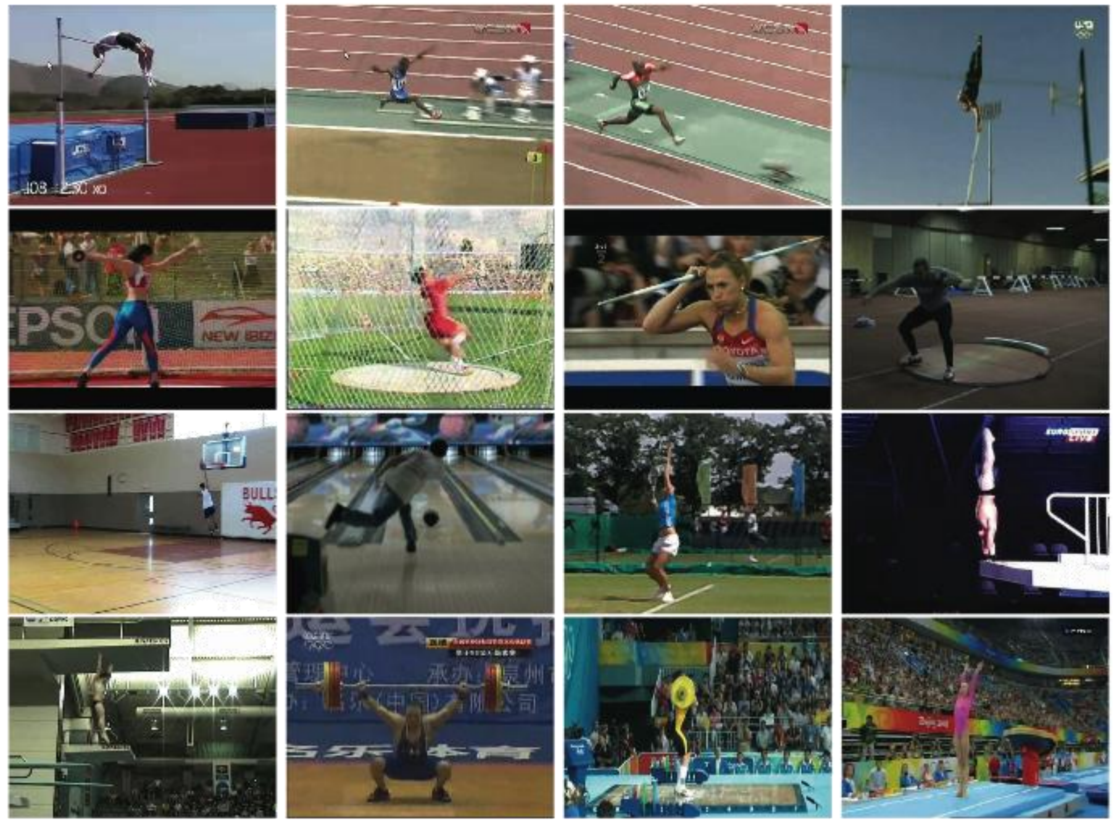
Video aesthetics

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

[1] T. Brox, J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. In IEEE Transactions on Pattern Analysis and Machine Intelligence, to appear

[2] N. Sundaram, T. Brox, K. Keutzer. Dense Point Trajectories by GPU-accelerated Large Displacement Optical Flow. In European Conference on Computer Vision (ECCV), September 2010

[3] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. Technical Report UCB/EECS-2010-104, EECS Department, University of California, Berkeley, Jul 2010.

[4] Sand, P., Teller, S.: Particle video: Long-range motion estimation using point trajectories. International Journal of Computer Vision 80 (2008) 72–91

[5] Shi, J., Tomasi, C.: Good features to track. In: CVPR. (1994) 593–600

[6] Zach, C., Gallup, D., Frahm, J.M.: Fast gain-adaptive KLT tracking on the GPU. CVPR Workshop on Visual Computer Vision on GPU’s (CVGPU) (2008)

[7] B. Catanzaro, M. Garland, and K. Keutzer. Copperhead: Compiling an embedded data parallel language. Technical Report UCB/EECS-2010-124, EECS Department, University of California, Berkeley, Sep 2010. Copperhead available at http://code.google.com/p/copperhead

[8] B. Catanzaro, N. Sundaram, and K. Keutzer. Fast support vector machine training and classification on graphics processors. In *ICML ’08: Proceedings of the 25th international conference on Machine learning, pages 104–111, 2008.*