GPU Programming in Computer Vision

Preliminary Meeting

Björn Häfner,
Benedikt Löwenhauser,
Thomas Möllenhoff
What you will learn in the practical course

- Introduction to NVIDIA CUDA Framework
- Introduction to parallel computing on GPUs
- How to parallelize basic computer vision algorithms in CUDA/C++
- Practical project experience
- Team work, presentation skills
Important Dates

• Preliminary Meeting: 3. February 2017 (today)
• Registration in the matching system on 3. - 8. February 2017
  • List your preferred practical courses
  • Send an email to cuda-ss17@vision.in.tum.de describing your prior knowledge in C/C++, Computer Vision/Image processing along with a short motivational statement
• Matching Results: 15. February 2017
• Only assigned students are allowed to attend !!!
• See docmatching.in.tum.de/index.php/schedule
Course Organisation

- 4-5 weeks block course in the semester break (beginning of September - mid of October)
- 1 week lecture and exercise session
- 3 weeks project phase
- Our computer lab will be open for students
- Computers are equipped with recent GPUs (GTX 750), one for each student.
- Students will work in groups: 24 students, ideally 8 groups, each has 3 students.
- Every group will be assigned to one advisor.
Course Structure

• First Week
  • Theoretical lecture in the morning
  • Hands-on programming exercises in the afternoon

• Following 3-4 weeks
  • Project phase, one project to each group
    • Your own ideas,
    • Project Proposals, any related topic to Computer Vision, Image Processing, Machine Learning

• Final presentation of the projects
Evaluation Criteria

- Successful completion of the exercises (0,3 bonus)
- Gained expertise in CUDA/parallel programming
- Quality of your final project
  - Successful completion of the project
    - Projects will be evaluated by the project advisors
  - Your talk
Regular Attendance Is Required

- Attendance at classes/exercises is mandatory
- In case of absence: Medical attest
- The practical course is intended as a 4 week „full-time“ project
Motivation on GPU programming

CPU: 4 - 32 cores

GPU: 3072 cores
CPU vs GPU
Motivation on GPU programming

- Allows you to do some cool stuff!
- Student projects from the previous years…
High Resolution Maps from Aerial Footage
Dense Visual Odometry
Poisson Image Editing

Poisson Solution to Guided Interpolation

- Guided Interpolation

\[ f^* : \text{Destination function} \]
\[ f : \text{Unknown function} \]

Guidance vector field

Gradient field of a source function

S: Image domain
\( \partial \Omega \): Closed subset of \( S \)

Poisson Solution to Guided Interpolation

With source guiding gradient

Properties of the Poisson’s Equation

\[
\min_f \int_\Omega |\nabla f - \nabla v|^2 \quad \text{with} \quad f|_{\partial \Omega} = f^*|_{\partial \Omega}
\]

\[
\Delta f = \text{div} \, \nabla v \quad \text{over} \ \Omega \quad \text{with} \ f|_{\partial \Omega} = f^*|_{\partial \Omega}
\]

\[
\text{div} \, \nabla v = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}
\]

- Second-order variations extracted by Laplacian operator are the most significant “perceptually”
- Scalar function on a bounded domain is uniquely defined by its values on the boundary and its Laplacian in the interior
- Poisson equation therefore has a unique solution
Depth Adaptive Super Pixels
Results: Speed

1:100

GTX 980  Xeon 2640 v3 cores

...or 1:42 when using AVX on CPU

Results: Validation

World’s first 2.7 A proteasome density from EM data
(previous record: 2.8 A)
Kinect Fusion
Enjoy the practical course!