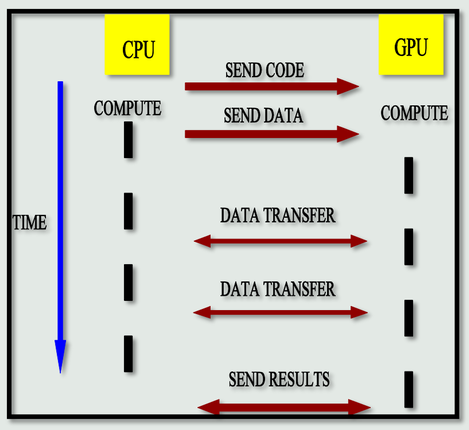
Hybrid Multi-Core Algorithms for Regular Image Filtering Applications

Shrenik Lad, Krishna Kumar Singh, Kishore Kothapalli, P.J. Narayanan

International Institute of Information Technology, Hyderabad, India

HYBRID MULTICORE COMPUTING

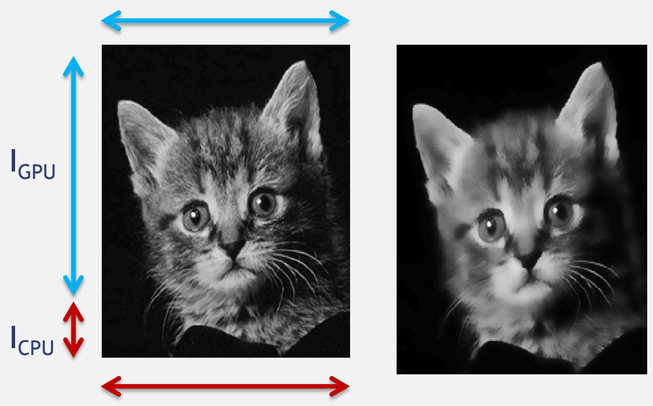
- Traditional Accelerator based computing does not involve the CPU in parallel computation.
- Not a good proposition.
- CPUs have high compute power cores.
- Computational power of CPUs is also on the rise.



- *Hybrid Multi-core Computing*
- Use all resources available(in a single platform).
- Provides a higher level of parallelism and efficiency.

DATA DECOMPOSITION

- CPU and GPU operate on separate image parts I_{GPU} and I_{CPU}, and compute filtered output using lookup tables.

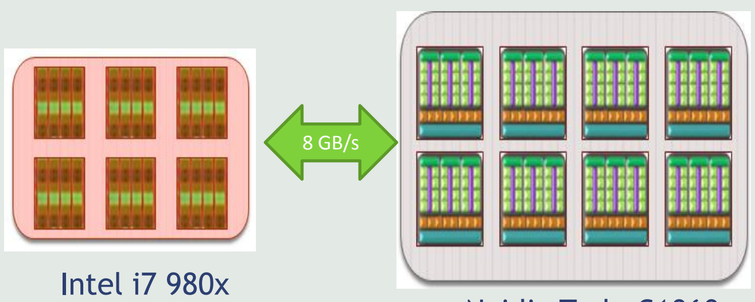


CONVOLUTION

- Common operation used in Image Processing.
- Involves only multiply-add operations. No exponential computations and hence no lookup tables.
- Fourier approach and Spatial approach for Convolution.
- On GPU, Spatial approach outperforms the Fourier approach whereas on CPU vice-versa is observed.
- Hence, using spatial approach on I_{GPU} and Fourier approach on I_{CPU} gives best results.
- Other components similar to the Bilateral Filtering implementation .

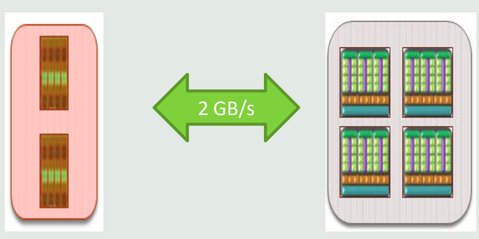
EXPERIMENTAL SETUP

HIGH-END SYSTEM



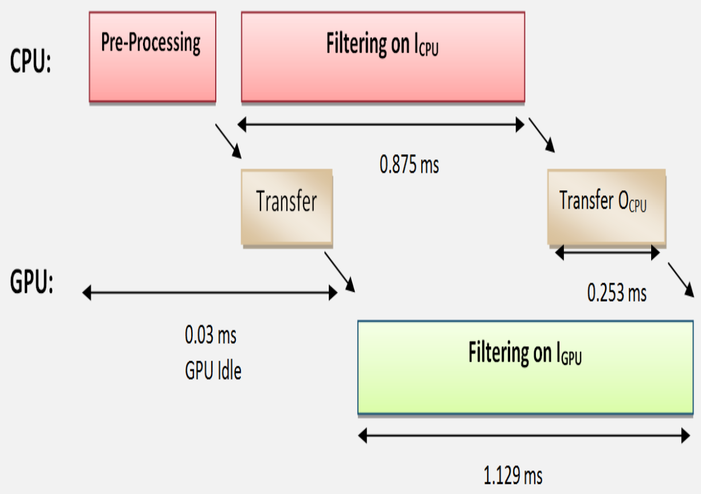
Intel i7 980x — Nvidia Tesla C1060

LOW-END SYSTEM



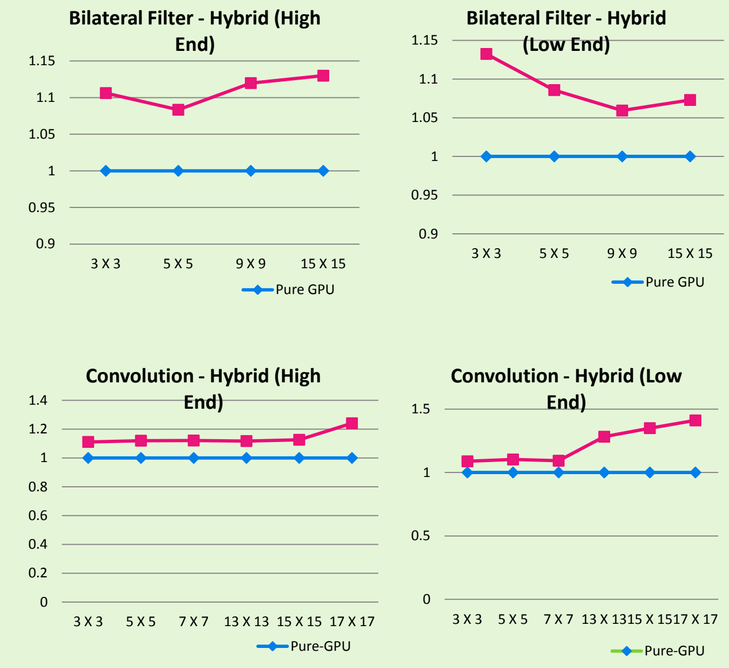
Intel Dual Core — Nvidia GT520

ALGORITHM FLOW



RESULTS

Plots for Relative speed vs. Filter Size



BILATERAL FILTER

- Bilateral filter is edge preserving and noise reducing filter used in Image Processing.
- Equation (1D) : $F(x) = \sum_{y \in N(x)} I(y) e^{\frac{-\|y-x\|^2}{2\sigma_d^2}} e^{\frac{-\|I(y)-I(x)\|^2}{2\sigma_r^2}}$
- Spatial as well as Intensity components involved.
- We pre-compute all possible transcendentals on the CPU.

X Dist / Y Dist	0	1	2	...	14	15
0	$E_{0,0}$	$E_{0,1}$	$E_{0,2}$		$E_{0,14}$	$E_{0,15}$
1	$E_{1,0}$	$E_{1,1}$	$E_{1,2}$		$E_{1,14}$	$E_{1,15}$
2	$E_{2,0}$	$E_{2,1}$	$E_{2,2}$		$E_{2,14}$	$E_{2,15}$
...						
14	$E_{14,0}$	$E_{14,1}$	$E_{14,2}$		$E_{14,14}$	$E_{14,15}$
15	$E_{15,0}$	$E_{15,1}$	$E_{15,2}$		$E_{15,14}$	$E_{15,15}$

Spatial Lookup

Intensity Diff	Exponential Term
0	I_0
1	I_1
2	I_2
3	I_3
...	...
254	I_{254}
255	I_{255}

Range Lookup

WORK DISTRIBUTION

CPU

Pre-Processing:

- Compute *Intensity* and *Spatial* Lookup tables.
- Divide Image into I_{CPU} and $I_{GPU.}$

Filtering on I_{CPU}:

- For each thread in parallel:
  compute output for a pixel using Spatial and Intensity lookup tables

GPU

Kernel code : thread (tx, ty)

- Load Image pixels from global memory to shared memory according to linear mapping.
- Syncthreads()
- Compute weighted sum for pixel (tx,ty) using Intensity and Spatial Lookup tables.

CONCLUSIONS

- Hybrid CPU+GPU Algorithms for the following 2 filtering methods:
  - Bilateral Filtering
  - Convolution
- Our implementation of Bilateral Filter is twice faster than the best GPU implementation.
- 10% average speedup compared to our Pure-GPU implementation of Bilateral Filter.
- 18% average speedup compared to our Pure-GPU implementation of Convolution.

REFERENCES

- V. W. Lee, C. Kim, J. Chhugani, M. Deisher, D. Kim, A. D.Nguyen,N. Satish, M. Smelyanskiy, S. Chennupaty, P. Hammarlund,R. Singhal, and P. Dubey, “Debunking the 100X GPU vs.CPU myth: an evaluation of throughput computing on CPU and GPU,” in Proc. ISCA, 2010.
- C. Tomasi and R. Manduchi, “Bilateral Filtering for Gray and Color Images”, Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India.
- FIALKA O, Cadk M, “FFT and Convolution Performance in Image Filtering on GPU,” in Proceedings of the Conference on Information Visualization, IEEE Computer Society.
- Nvidia Corporation, “CUDA: Compute Unified Device Architecture programming guide,” Technical report, Nvidia, Tech. Rep., 2007.