Real-Time Affine-Invariant Feature Extraction: Object Recognition Under Extreme Viewpoint Change

Presented by Valeriu Codreanu

rijksuniversiteit groningen

Technische Universiteit Eindhoven University of Technology

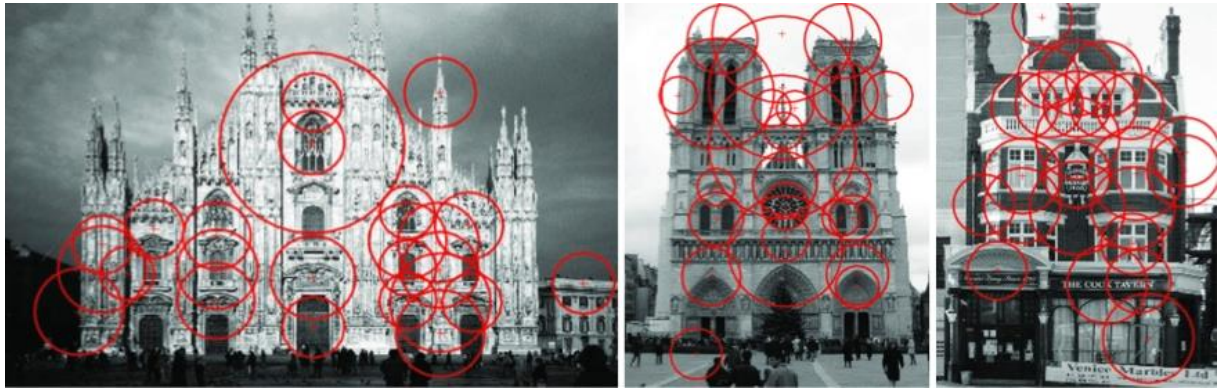
Outline

- What is object recognition?
- Viewpoint-invariant keypoint extraction
- GPU-ASIFT case study
  - Multi-GPU implementation
  - Timing/accuracy results
- Real-time GPU-ASURF
- Viewpoint-invariance overhead
- Conclusions

Object Recognition

2-Step Process:

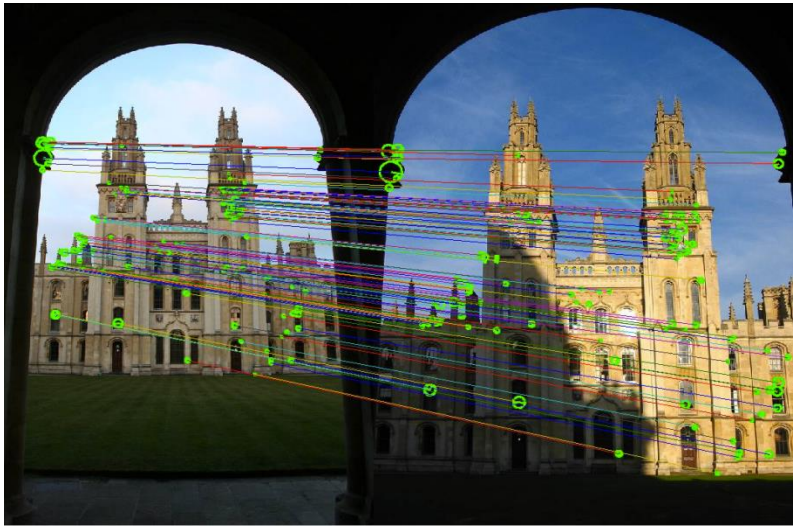
- Identify distinctive features (keypoints) using a feature extractor (SIFT/SURF/ORB/etc)



Object Recognition

2-Step Process:

- Find matching keypoints in source and query images



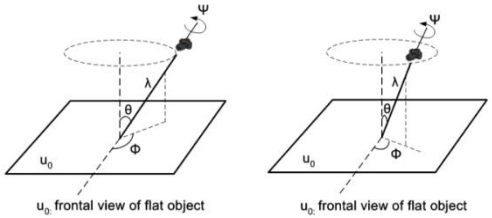
Viewpoint-invariant keypoint extraction

- ASIFT is the first fully affine-invariant CV algorithm
  - Proposed in 2009 by Morel et al.*
- Based on the SIFT algorithm
- But addressing its main deficiency
  - Low invariance to camera viewpoint change
- By simulating multiple views
- But it generates/processes a lot more information than SIFT
  - This makes it computationally expensive
  - Unfeasible for real-time feature extraction on multi-core CPUs

* J.M. Morel and G.Yu, ASIFT: A New Framework for Fully Affine Invariant Image Comparison, SIAM Journal on Imaging Sciences, vol. 2, issue 2, 2009.

Viewpoint-invariant keypoint extraction

Variation of the latitude angle (rotation)



$$u'(i,j) = \bar{u}(i\cos\phi - j\sin\phi, i\sin\phi + j\cos\phi)$$

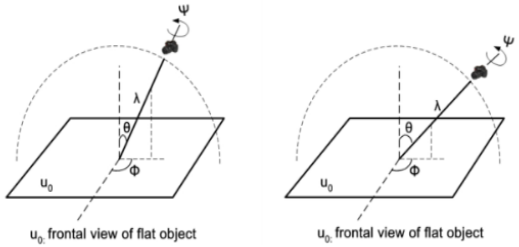
Anti-aliasing filtering kernel. Gaussian

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{x^2}{2\sigma^2}) \quad ,\sigma = c\sqrt{t^2-1}$$

Variation of the longitude angle (tilt)

$$t = |\frac{1}{\cos\theta}|.$$

$$u(x,y) \rightarrow u(x,ty)$$



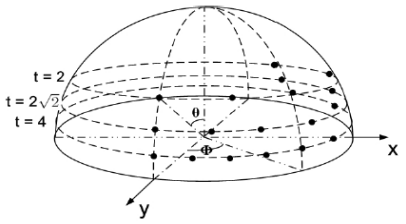
SIFT/SURF/ORB keypoint extraction

Parameter sampling

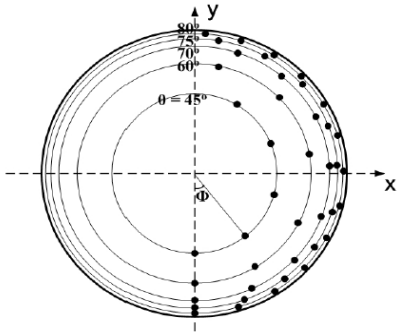
- Geometric sampling of t: $\triangle t = t_{k+1}/t_k = \sqrt{2}$.

t	1	$\sqrt{2}$	2	$2\sqrt{2}$	4	$4\sqrt{2}$
θ	$0°$	$45°$	$60°$	$69.3°$	$75.5°$	$79.8°$

- Arithmetical sampling of ϕ: $\triangle\phi = \phi_{k+1} - \phi_k = \triangle\phi = \frac{72°}{t}$.



Perspective view

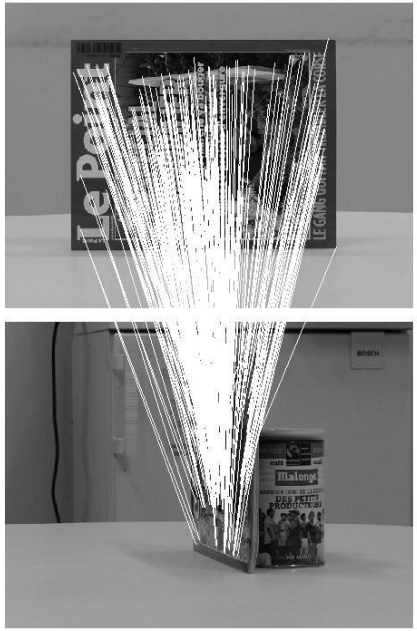


View from the zenith

Key idea :

- Although a tilt distortion is irreversible, it can be compensated by simulating a tilt of the same amount in the orthogonal direction.

ASIFT motivation

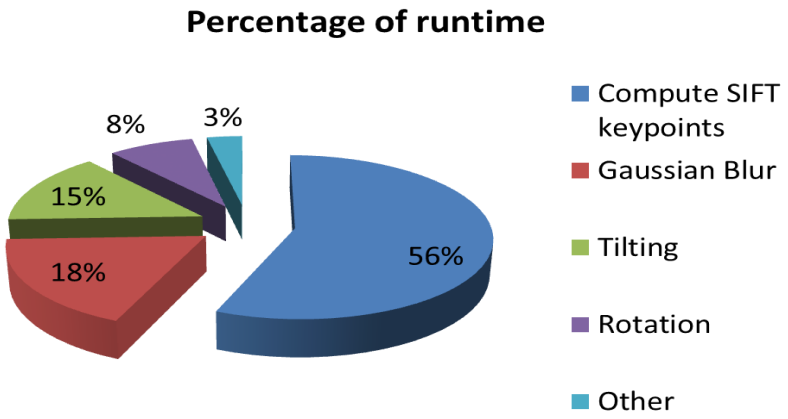


Viewpoint-invariant features (ASIFT)

Traditional features (SIFT)

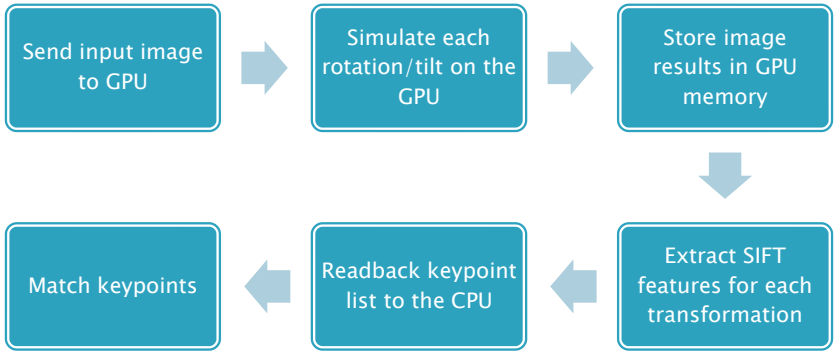
ASIFT profiling results

- All functions have a significant weight!
  - So all have to be moved to the GPU
  - Otherwise, Amdahl's law limits the achievable gain
- Data transfers between stages should be minimized



GPU-ASIFT

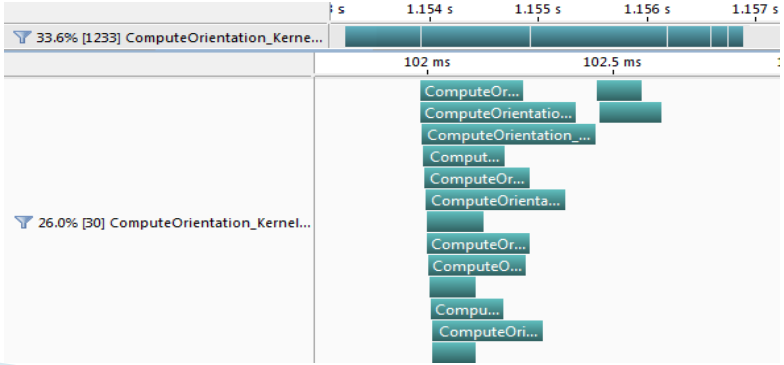
- The goal is to have as little CPU<->GPU traffic as possible:
  - Copy image data
  - Readback keypoints
- Implemented the image transformations as 3 CUDA kernels
  - Rotation
  - 1D Gaussian convolution
  - Directional sub-sampling
- A variation of SiftGPU* is used for SIFT keypoint extraction



** Changchang Wu: SiftGPU: A GPU Implementation of Scale Invariant Feature Transform (SIFT)*

SiftGPU

- CUDA implementation of SIFT
- Generates the traditional set of 128-dimensional SIFT descriptors
- Modified to fit our needs:
  - Load image data from device memory
  - Improved performance through stream concurrency
    - 20% performance increase on SiftGPU



Multi-GPU implementation

- Basic idea: The image transformations are fully independent
- Total number of transformed images is divided by the number of GPU devices
- Each GPU device extracts keypoints in parallel
- Keypoints are then aggregated on the host
- Scales almost linearly w.r.t. the nr. of GPU devices
  - 1.75-1.9x for 2 GPUs
- 6 tilts generate 43 independent image transformations
  - Can be distributed to multiple GPUs

Timing results

- Images originally sampled at 3MP
  - Upscaled and downscaled for performance measurements
- Features from 6x image area are extracted in 4x SiftGPU time
- 4x image area is computed in 2x time on GPU, 4x on CPU

$$A(t) = 1 + (|\Gamma_t| - 1)\frac{180°}{72°}$$

$$|\Gamma_t| = |\{1, \sqrt{2}, ..., \sqrt{2}^{t-1}\}|$$

- Test system
  - CPU: Intel Core i7-2600K
  - GPU: Nvidia GTX690

Resolution	single-core	quad-core	single-GPU	dual-GPU
480x640	8861	2262	390	222
600x800	13634	3464	483	272
1024x1280	35101	8414	765	425
1200x1600	54039	13136	921	492
1536x2048	90465	21840	1342	710
1936x2584	141711	34086	2075	1092

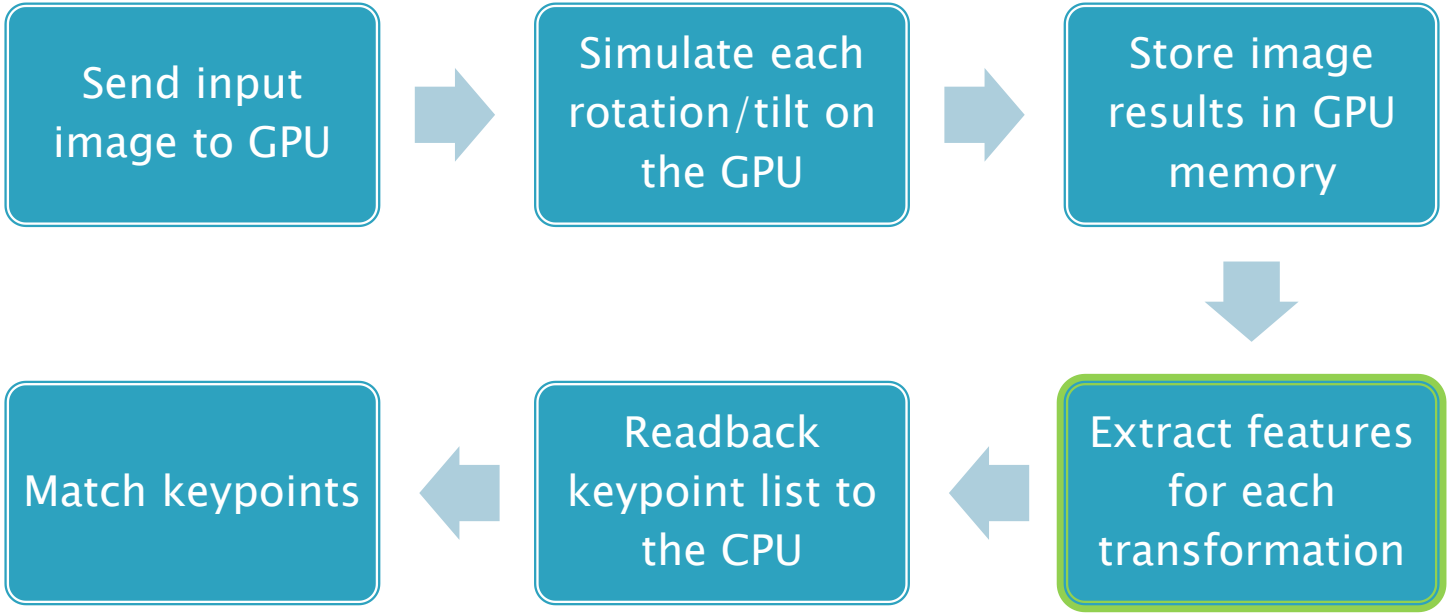
Accuracy results

Method	Number of correct matches [#]
ASIFT-CPU($t = 2, u = 0$)	160
GPU-ASIFT($t = 2, u = 0$)	159
GPU-ASIFT($t = 2, u = 1$)	445
ASIFT-CPU($t = 4\sqrt{2}, u = 0$)	753
GPU-ASIFT($t = 4\sqrt{2}, u = 0$)	497
GPU-ASIFT($t = 4\sqrt{2}, u = 1$)	1508
siftGPU	1

Method	Number of correct matches [#]
ASIFT-CPU($t = 2, u = 0$)	69
GPU-ASIFT($t = 2, u = 0$)	34
GPU-ASIFT($t = 2, u = 1$)	90
ASIFT-CPU($t = 4\sqrt{2}, u = 0$)	141
GPU-ASIFT($t = 4\sqrt{2}, u = 0$)	151
GPU-ASIFT($t = 4\sqrt{2}, u = 1$)	467
siftGPU	0

- Accuracy can be further improved by upsampling
  - Much lower computational penalty than for the CPU algorithm

OpenCV Framework overview



GPU-ASURF & GPU-AORB

- Implemented image transformations using the OpenCV GPU functionality
  - gpu::rotate
  - gpu::GaussianBlur
  - gpu::warpAffine
- Data transfers are minimized
  - Transformed images reside in GPU memory
- The SURF/ORB GPU implementations are applied on the resulting images
  - The framework can be used with any other feature extractor
- Keypoint matching is done with kNN

Performance comparison

- Multi-GPU implementation provides good performance benefits
  - 1.75x-1.9x acceleration from 2 GPUs
- GPU-ASURF applied on low-resolution images extracts features in real-time on GTX690

Resolution	GPU-ASURF	GPU-AORB	GPU-ASIFT
480x640	89	107	195
600x800	128	122	242
1024x1280	291	155	382
1200x1600	456	182	461
1536x2048	740	224	671
1936x2584	1145	341	1038

Accuracy comparison

Method	# correct matches
GPU-ASIFT (t = 2)	159
GPU-ASURF (t = 2)	75
GPU-AORB (t = 2)	24
GPU-ASIFT (t = $4\sqrt{2}$)	497
GPU-ASURF (t = $4\sqrt{2}$)	354
GPU-AORB (t = $4\sqrt{2}$)	146
SIFT	1
SURF	2
ORB	2

Method	# correct matches
GPU-ASIFT (t = 2)	34
GPU-ASURF (t = 2)	15
GPU-AORB (t = 2)	8
GPU-ASIFT (t = $4\sqrt{2}$)	151
GPU-ASURF (t = $4\sqrt{2}$)	50
GPU-AORB (t = $4\sqrt{2}$)	21
SIFT	0
SURF	2
ORB	2

Overhead of viewpoint-invariance

- Image transformation overhead is negligible in comparison to feature extraction
  - 3-11% depending on resolution
- Scales very well on multiple GPUs
  - All transformations are independent
- Programming overhead is minimal

Resolution	Overhead [ms]
480x640	10.8
600x800	13.5
1024x1280	16.9
1200x1600	20.2
1536x2048	25.8
1936x2584	38.6

Conclusions

- Simple OpenCV framework to develop viewpoint-invariant object recognition algorithms
- GPU-ASIFT provides a high level of accuracy while extracting features in near-real-time
- Real-time ASURF extraction using the OpenCV GPU framework

- The transformation framework:
  - has acceptable overhead
  - is scalable to multiple GPUs
  - can be applied to any "*non affine-invariant*" feature extractors

Thank you

Q&A