Mobile Visual Computing

Kari Pulli
Research Fellow
Nokia Research Center Palo Alto
Overview

• Mobile Augmented Reality
  • Matching geo-located image collections
  • Tracking with recognition
  • Point & Find

• Computational Photography
  • High-dynamic range imaging
  • Mobile panoramas

• Mobile GPUs for image processing
  • OpenGL ES
  • OpenCL
Use images to find out what you’re pointing at

From an image... ...to information
System Overview

GPS

Memorial Church

Stanford Memorial Church stands at the center of the campus, and is the University's architectural

Server

Gabriel Takacs, Vijay Chandrasekhar, Natasha Gelfand, Yingen Xiong, Wei-Chao Chen, Thanos Bismpigiannis, Radek Grzeszczuk, Kari Pulli, Bernd Girod

Outdoor Augmented Reality on Mobile Phone using Loxel-Based Visual Feature Organization

ACM International Conference on Multimedia Information Retrieval (MIR'08)
"Bag of Words" Matching

Query Image

Geometric Feature Consistency Check

Prefetched Data

Database Images
Computing Visual Words

$D_{xx}$

$D_{xy}$

$D_{yy}$

$D_{xx} D_{yy} - (0.9 D_{xy})^2$

Blob Response
Computing Visual Words

Orient along dominant gradient

Oriented Patch
Computing Visual Words

Gradient Field

SURF Descriptor

\[ \sum d_x \]
\[ \sum d_y \]
\[ \sum |d_x| \]
\[ \sum |d_y| \]

SIFT Descriptor

\[ \sum \]
Feature Descriptor Clustering

Average “Visual Words” That Match Across Images
Feature Descriptor Pruning
Select the Most Descriptive “Visual Words”
How Many Visual Words are Needed?

![Graph showing the relationship between Percent and Kernel Budget with True Positives and False Positives. The graph indicates a 4x Reduction at a certain point.]
Data Organization

[Diagram showing relationships between Kernel and Loxel]
**V. Chandrasekhar, G. Takacs, D. Chen, S. S. Tsai, R. Grzeszczuk, B. Girod**

**CHoG: Compressed Histogram of Gradients: A Low Bit-Rate Feature Descriptor**

IEEE Conf. on Computer Vision and Pattern Recognition (CVPR09)
Gradient Binning

Asymmetry arises from patch orientation
Gradient Binning

![Graph showing correct match fraction against incorrect match fraction for VQ-3, VQ-5, VQ-7, and VQ-9]
Histogram Compression

Gradient histogram

Quantized histogram

Tree code

0.46
1/2

0.21
1/4

0.08
1/16

0.09
1/16

0.160

0.08
0.16

0.09
0.21

0.46

0.16

0.05
0.1

0.1
0.05
-0.05
-0.1
-0.1
-0.05
Spatial Binning
Feature Matching Performance

Descriptor Size (bits)

Equal Error Rate (%)
Compressed Domain Matching

Gradient histogram → Quantized histogram → Tree index → Look-up table

Dist(·)
Nearest Neighbor Search

Query Time (sec)

SIFT

CHoG

10^3 query descriptors  10^6 database descriptors

372

47

28

Exact

kD-tree

Approx

kD-tree

Metric tree

0.3% errors
Landmark-based navigation

1. Keep walking straight, Gates Hall will be to your front left
2. Turn right here, Gilbert Hall will then be to your front left
3. Keep walking straight, past Gilbert Hall to your left
4. Keep walking straight, past Herrin Labs to your left
11. Keep walking straight, past Main Quad Math Corner to your left
12. Keep walking straight, toward West Gate
13. Keep walking straight, into West Gate
14. Keep walking straight, Hoover Tower is in the distance
Path Loxels

- The path is divided into small 30x30m cells
- Directions are generated for each cell locally
Good landmarks

- Image count is an indicator of landmark popularity
- Require good visibility from current location
- Prefer landmarks that are straight ahead
Choosing images

- The center of the cluster (most features in common with other images) is likely to be a good representation.
Augment images

- Use known locations to estimate camera direction
- Draw an arrow in the image
- Generate relative text directions

H. Hile, R. Grzeszczuk A. Liu, R. Vedantham, J. Kosecka, G. Borriello
Landmark-based Pedestrian navigation with Enhanced Spatial Reasoning
Pervasive 2009.
Real-time Tracking and Pose Estimation for AR

Julien Pilet, et al.
EPFL

Daniel Wagner, et al.
Graz University of Technology

Georg Klein et al.
University of Oxford

• Using corners as features has limitations
  • Weak recognition capability
    • Limited number of objects
    • Small environment
Goal: Bring AR to outdoor environment

• Location-based context information
  - Need: **Large-scale scene recognition**
  - Must use scale-invariant features with strong descriptors for matching

• Target mobile devices
  - Need: **Efficient real-time tracking on the mobile platform**
  - Cannot detect and match scale-invariant features for every frame!

Duy-Nguyen Ta, Natasha Gelfand, Wei-Chao Chen, Kari Pulli
SURFTrac: Efficient Tracking and Continuous Object Recognition using Local Feature Descriptors
IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'09)
SURFTrac - Detection

• Predict and detect features within local neighborhood regions of scale-space pyramid
  • Avoid searching the entire image pyramid
**SURFTrac - Matching**

- **Strategy 1:** Use edge-response measures
  - Measure how likely a feature is along an edge
  - Very fast to compute

\[
H(x, \sigma) = \begin{bmatrix}
L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\
L_{xy}(x, \sigma) & L_{yy}(x, \sigma)
\end{bmatrix}
\]

\[
r_2 = \frac{trace(H)^2}{det(H)}
\]
• Strategy 2: Use template matching in Hessian domain
SURFTrac - Matching

• Strategy 2: Use template matching (NCC or SSD)

Feature i, Frame 1

Feature j, Frame 1
**SURFTrac - Matching**

- Strategy 2: Use template matching (NCC or SSD)

Feature i, Frame 2

Feature j, Frame 2
• Strategy 2: Use template matching (NCC or SSD)

Feature i, Frame 3

Feature j, Frame 3
• Strategy 2: Use template matching (NCC or SSD)

Feature i, Frame 4

Feature j, Frame 4
Strategy 2: Use template matching (NCC or SSD)

Feature i, Frame 5

Feature j, Frame 5
• Strategy 2: Use template matching (NCC or SSD)

Feature i, Frame 6

Feature j, Frame 6
Comparison: Template matching wins

Accuracy is approximated by $\frac{N_{\text{inliers after RANSAC}}}{N_{\text{total found matches}}}$
Impact of z-translation

Impact of panning

Impact of tilting

Impact of rolling

Δz

Δangle

accuracy (%)

accuracy (%)

accuracy (%)

accuracy (%)

1 2 3 4 5 6 7 8 9 10 11 12

1 2 3 4 5 6 7 8 9 10 11 12

1 2 3 4 5 6 7 8 9 10 11 12

1 2 3 4 5 6 7 8 9 10 11 12

- Template matching
- Edge response
SURFTrac is 3x – 5x faster than repeated SURF

- Nokia N95 with Texas Instrument OMAP2 @ 330 MHz
  - ~20x slower than a laptop
- Image size: 256 x 192

<table>
<thead>
<tr>
<th>Methods</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SURF</td>
<td></td>
</tr>
<tr>
<td>Detection only</td>
<td>0.357</td>
</tr>
<tr>
<td>Detection and Matching</td>
<td>0.678</td>
</tr>
<tr>
<td>SURFTrac</td>
<td></td>
</tr>
<tr>
<td>Template matching only</td>
<td>0.115</td>
</tr>
<tr>
<td>Template matching + RANSAC</td>
<td>0.133</td>
</tr>
<tr>
<td>Edge response only</td>
<td>0.111</td>
</tr>
<tr>
<td>Edge response + RANSAC</td>
<td>0.134</td>
</tr>
</tbody>
</table>
Matching and Tracking with an Image Database

Camera images

Databases images
Matching and Tracking with an Image Database
Product Vision: Nokia Point & Find
Bridges The Physical World With The Digital World

Physical World

Digital World

Future release
Product Vision: Nokia Point & Find
User Flow With Minimum Clicks To Information For Many Use Cases

1. Open Reader

2. Read Tag

3. View Content

1b. Change Tab
Broad solutions: across industries Nokia Point & Find enables new business opportunities

Publishers/Magazines
- Point at different ads and pages in a magazine and get additional relevant info and calls to action
- Point at exhibits and objects on display to learn more and interact
- Point at cars at a car show or dealership and learn more about the vehicles

Out of Home Ads
- Point at street posters and billboards to learn more and interact with ad
- Make attractions within a theme park connect to mobile interactive experiences (Disney)

Retail/Consumer Products
- Point at products on display and on shelves and get details, reviews, promos, and price comparisons
- Point at apartments to “check available apartments” or point at house for sale to get detailed info and photo tour

Events, Trade Shows, Venues, Museums, Theme Parks
- Point at exhibits and objects on display to learn more and interact
- Make attractions within a theme park connect to mobile interactive experiences (Disney)

Automotive
- Point at an area or object in a car to get info, tutorials, and explanations

Real Estate
- Point at apartments to “check available apartments” or point at house for sale to get detailed info and photo tour

Brands, Ad Agencies, Media Buyers
- Build engaging interactive campaigns

Consumer/Long Tail
- Enable consumers to build their own interactive experiences and make part of their social networks

Nokia
Point of sale example: point at products in store and get details, promos, videos, more

Shopping in a store...

Point phone at a product (camera)

Instant Results:

- Detailed product info
- Video demo
- Virtual tour
- Special offers (coupon or rebate)
- Accessories
- Send message to friend about product
Vision
High-dynamic range imaging

Orazio Gallo, Wei-Chao Chen, Natasha Gelfand, Marius Tico, Kari Pulli
Artifact-free High Dynamic Range Imaging
IEEE International Conference on Computational Photography (ICCP’09)
Interactive mobile panorama

- Automatic capture based on camera motion tracking (2D)
- High resolution images for panorama stitching
Unlimited viewing angle
Image registration

• Identify corresponding features between input images
Simple stitching produces ghosting artifacts

• Simplest combination
  • alpha blending across the overlap

• Ghosting, if
  • objects move
  • registration is imperfect
  • there is parallax
Good seams between images get rid of ghosting
How to calculate the seams?

• **Graph cut**
  - Popular
  - General: can simultaneously calculate good seams when N images overlap
  - Slow
  - Uses lot of memory (need all the images)

• **Dynamic programming**
  - Add images one at a time
  - Calculate a good seam between the image and previous collection
  - Much faster (30X – 90X !!)
  - Uses much less memory since don’t need to have all the images in memory
Colors will not in general match

• 3A
  • Auto exposure, auto white balance, [auto focus]

• Poisson blending
  • gets details from the gradients of the next image
  • forces colors to blend continuously
Match color curves using the overlap area
Poisson doesn’t always work (and is slow)

Poisson blending

Color correction & simple alpha blending
Results on the phone

- Nokia N95 8G, 18 images, each 1024x768
Mobile graphics HW for image processing

• Modern mobile GPUs support shaders
  • OpenGL ES 2.0 from Khronos Group
  • Starting to ship this year in volumes
  • Designed for 3D graphics, but can be used also to accelerate image processing
  • But image processing on GPUs is quite different from CPU programming

• OpenCL
  • For heterogeneous multiprocessing, same programming language for CPU and GPU (and even other back ends)
  • Also from Khronos Group
  • Desktop implementations already shipping
    • Mobile implementations still experimental
POWERVR Graphics IP Cores Roadmap

- **MBX Series4**
  - 2002-2006
  - DirectX + OpenGL ES + OpenVG
  - POWERVR MBX
  - POWERVR MBXLite

- **SGX Series5**
  - 2007-2008
  - USSE
  - POWERVR SGX535
  - POWERVR SGX540
  - POWERVR SGX531
  - POWERVR SGX520
  - POWERVR SGX530
  - POWERVR SGX545

- **SGX Series5XT**
  - 2009-
  - USSE2
  - POWERVR SGX543MP2-16
  - POWERVR SGX543

- **VGX Series5**
  - 2009-
  - OpenVG
  - POWERVR VGX150

All Series 5 and Series 5XT cores support OpenCL Embedded Profile
All DirectX-compatible cores additionally support OpenCL Full Profile
POWERVR USSE (Universal Scalable Scheduler Engine)

Thread Scheduling

Task Queues
- 16 pending per pipe

Thread Queues (16 pending per pipe)

Active Threads
- 4 active per pipe

Active Threads

Vertex Source

Pixel Source

Coarse Grain Scheduler (CGS)

Execution Unit 1

Thread which hits data stall switched out, ready thread switched in with no cycle loss
Fragment processing of frame N happens simultaneously with geometry processing of frame N+1 while the CPU processes frame N+2. Unified architecture means all DSP resources are available for OpenCL use.
Speeding up the warping process

- Warp the rectangular image to spherical coordinates using vertex transformations and rasterization
- The speed was tested on
  - GPU with OpenGL ES 2.0
  - CPU with OpenCL
  - CPU with C, hand optimized to use fixed-point maths
### Results

- Results for 7 frames, average frame size 1300 x 930 pixels
- Tests run on OMAP Zoom: CPU 550 MHz, SGX GPU 110 MHz

<table>
<thead>
<tr>
<th>Method</th>
<th>Per frame</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>14.5 sec</td>
<td>101.5 sec</td>
</tr>
<tr>
<td>GPU by OpenGL ES 2.0</td>
<td>0.8 sec</td>
<td>5.6 sec</td>
</tr>
<tr>
<td>OpenCL with GPU backend</td>
<td>0.9 sec</td>
<td>6.3 sec</td>
</tr>
<tr>
<td>Hand optimized CPU</td>
<td>0.5 sec</td>
<td>3.6 sec</td>
</tr>
</tbody>
</table>
## Time distribution on GPU execution

<table>
<thead>
<tr>
<th>Action</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transferring data to input buffer</td>
<td>0.12 ms</td>
</tr>
<tr>
<td>Upload time to input picture texture</td>
<td>28.87 ms</td>
</tr>
<tr>
<td>Upload time to mask texture</td>
<td>9.34 ms</td>
</tr>
<tr>
<td>Execution</td>
<td>346.07 ms</td>
</tr>
<tr>
<td>Download</td>
<td>191.86 ms</td>
</tr>
<tr>
<td>Extracting data</td>
<td>204.47 ms</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>780.73 ms</strong></td>
</tr>
</tbody>
</table>
Picture quality

- CPU integer optimized suffers from inaccurate tan() table
- GPU benefits from free bilinear interpolation
Data transfers create a large overhead

- Unfortunately, OpenGL ES lacks asynchronous pixel data transfers, pixel buffer objects
  - Huge overhead from synch’ing the pipes and reading data back
- Pipeline textures loading, drawing, and reading results
  - While you are reading from one buffer, the hardware should be able to process the commands buffered up for other buffers

```c
// read the 1st frame
glReadPixels()  
// read the 2nd frame
glReadPixels()
...
```

### Sequential

<table>
<thead>
<tr>
<th>Frame 1</th>
<th>Frame 2</th>
<th>Frame 3</th>
</tr>
</thead>
</table>

### Parallel

<table>
<thead>
<tr>
<th>Frame 1</th>
<th>Frame 2</th>
<th>Frame 3</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Framebuffer 1</th>
<th>Framebuffer 2</th>
</tr>
</thead>
</table>

81
Acknowledgements

- Full time NRC PA researchers
  - Wei-Chao Chen
  - Natasha Gelfand
  - Radek Grzeszczuk
  - Marius Tico
  - Ramakrishna Vedantham
  - Shawn Wang
  - Yingen Xiong

- Interns
  - Andrew Adams (Stanford)
  - Vijay Chandrasekhar (Stanford)
  - Orazio Gallo (UC Santa Cruz)
  - Chia-Kai Liang (National Taiwan Univ.)
  - Duy-Nguyen Ta (Georgia Tech)
  - Gabriel Takacs (Stanford)

- Other Nokia experts (OpenCL / GL ES)
  - Jyrki Leskelä
  - Mika Salmela
  - Jarmo Nikula

- Point & Find
  - Philipp Schloter, Sophie-Charlotte Moatti, Christophe Marle

- Faculty collaborators
  - Bernd Girod (Stanford)
  - Marc Levoy (Stanford)

- Imagination Technologies
  - Georg Kolling, Peter McGuinness