Efficient Pattern Recognition Algorithm Including a Fast Retina Keypoint FPGA Implementation

Lester Kalms  
Technische Universität  
Dresden, Germany  
lester.kalms@tu-dresden.de

Maximilian Hajduk  
Technische Universität  
Dresden, Germany  
maximilian.hajduk@tu-dresden.de

Diana Göhringer  
Technische Universität  
Dresden, Germany  
diana.goehringer@tu-dresden.de

Lester Kalms  
Barcelona, 09. September 2019
Content

1. Motivation & Overview
2. Pattern Recognition Algorithm
3. HW Implementation of FREAK
4. Evaluation
5. Conclusion & Outlook
Motivation

What is pattern recognition?

• Feature detector: finds points of interest in images (e.g. corners, edges or blobs)
• Feature descriptor: describes features to be comparable with others

Why pattern recognition?

• Still algorithms make use of pattern recognition (e.g. object tracking or SLAM)
• Most neural networks need a high amount of input data
Overview

First full FPGA implementation of the FREAK descriptor

• Related work does not contain orientation computation
• Contains optimized pattern generator (partially pre-computed)
• Frequency optimized integral image with parallel prefix sum

Improved pattern recognition algorithm

• Compared to different combinations of AKAZE, ORB, BRISK and FREAK
• Improved and combined AKAZE detector & FREAK descriptor
• Added a Retain Best function
SW Implementation
AKAZE feature detection & FREAK feature description
HW Implementation

Integral Image Function

SW: \[ I_{x,y} = i_{x,y} + I_{x-1,y} + I_{x,y-1} - I_{x-1,y-1} \]

HW: \[ s = s + i_{x,y} \]

\[ a = [(y > 0) \rightarrow (s + buf_x)] \land [(y \leq 0) \rightarrow s] \]

\[ I_{x,y} = buf_x = a \]
HW Implementation
Integral Image Function

**SW:**

\[ I_{x,y} = i_{x,y} + I_{x-1,y} + I_{x,y-1} - I_{x-1,y-1} \]

**HW:**

\[ s = s + i_{x,y} \]

\[ a = [(y > 0) \rightarrow (s + buf_x)] \land [(y \leq 0) \rightarrow s] \]

\[ I_{x,y} = buf_x = a \]
HW Implementation

Integral Image Function

**SW:**

\[ I_{x,y} = i_{x,y} + I_{x-1,y} + I_{x,y-1} - I_{x-1,y-1} \]

**HW:**

\[ s = s + i_{x,y} \]

\[ a = [(y > 0) \rightarrow (s + buf_x)] \land [(y \leq 0) \rightarrow s] \]

\[ I_{x,y} = buf_x = a \]
HW Implementation
Integral Image Function

- 4 Pixels each clock cycle
- High frequency

<table>
<thead>
<tr>
<th>Input Image</th>
<th>Output Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1 1</td>
<td>1 2 3 4</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>5 6 7 8</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>10 12 14 16</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>15 18 21 24</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>20 24 28 32</td>
</tr>
</tbody>
</table>

CSA = Carry Save Adder
HW Implementation

FREAK Descriptor

Implementation

- VHDL
- Fixed-Point
- Pipelined
HW Implementation
FREAK Descriptor (Pattern)

FREAK sampling pattern similar to the retinal ganglion cells [1]
HW Implementation
FREAK Descriptor

- Boundary
  - kp [1]

- Pattern unrotated
  - pattern [42], kp [1]

- Intensity
  - intensity [42]

- Orientation
  - orientation [1]

- Pattern rotated
  - pattern [43], kp [1]

- Intensity
  - intensity [43]

- Descriptor
  - descriptor [8]

- AKAZE

- DMA

configuration
HW Implementation
FREAK Descriptor (intensity)

Intensity value calculation

\[ i = \frac{I_{y_b,x_r} - I_{y_b,x_l} + I_{y_t,x_l} - I_{y_t,x_r}}{(x_r - x_l) \cdot (y_b - y_t)} \]

FREAK sampling pattern similar to the retinal ganglion cells [1]
HW Implementation
FREAK Descriptor

Boundary
kp [1]

Pattern
unrotated
kp [1]
pattern [42], kp [1]

Intensity
intensity [42]

Orientation
orientation [1]

Pattern
rotated
pattern [43], kp [1]

Intensity
intensity [43]

Descriptor
descriptor [8]

4x request [42]
4x data [42]

Send Keypoint
keypoint [1]

Send Descriptor

AKAZE
configuration

DMA
HW Implementation

FREAK Descriptor
HW Implementation
FREAK Descriptor

Diagram of HW Implementation for FREAK Descriptor, showing the flow of data through various stages such as Boundary, Pattern, Orientation, Intensity, and Descriptor, with FIFOs and DMA involved in the configuration process.
**Evaluation**

**Software (Repeatability)**

**Setup**
- Core-i7 7700
- Oxford Dataset: 800x640 - 1000x700
- Quality: luminance, blur & compression
- Position: Scale, rotation & viewpoint

<table>
<thead>
<tr>
<th></th>
<th>Position</th>
<th>Quality</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>detect + describe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>akaze + akaze</td>
<td>21.6</td>
<td>63.3</td>
<td>85.4</td>
</tr>
<tr>
<td>akaze + freak</td>
<td>23.1</td>
<td>57.2</td>
<td>84.7</td>
</tr>
<tr>
<td>akaze + brisk</td>
<td>19.4</td>
<td>59.4</td>
<td><strong>87.8</strong></td>
</tr>
<tr>
<td>akaze + orb</td>
<td>11.5</td>
<td>37.8</td>
<td>86.2</td>
</tr>
<tr>
<td>brisk + brisk</td>
<td>17.3</td>
<td>66.1</td>
<td>85.1</td>
</tr>
<tr>
<td>brisk + freak</td>
<td><strong>21.7</strong></td>
<td>58.7</td>
<td>81.4</td>
</tr>
<tr>
<td>brisk + orb</td>
<td>11.2</td>
<td>49.8</td>
<td>82.0</td>
</tr>
<tr>
<td>orb + orb</td>
<td>15.0</td>
<td>61.2</td>
<td>82.2</td>
</tr>
<tr>
<td>orb + freak</td>
<td>19.3</td>
<td>65.8</td>
<td>84.9</td>
</tr>
<tr>
<td>orb + brisk</td>
<td>16.9</td>
<td><strong>66.8</strong></td>
<td><strong>85.8</strong></td>
</tr>
<tr>
<td>proposed</td>
<td><strong>23.1</strong></td>
<td><strong>70.2</strong></td>
<td><strong>87.9</strong></td>
</tr>
</tbody>
</table>
Evaluation
Hardware (Timing)

Setup

- ZedBoard (Zynq 7020) & Vivado 2018.2
- Vhdl, Fixed-point data types, 2048 key-points per frame

Frequency / FPS (detector)

- 185 MHz (synthesis)
- Simulation: for 150 MHz -> 850 fps (5.85 Gbit/s bandwidth)
- Real system: for 167 MHz -> 73.4 fps (including DMAs)

Frequency (Integral)

- 204 MHz (synthesis)
- 3ns for parallel prefix sum
## Evaluation

### Hardware (Resources)

<table>
<thead>
<tr>
<th>Module</th>
<th>FF</th>
<th>LUT</th>
<th>LUTRAM</th>
<th>BRAM</th>
<th>DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel sum</td>
<td>596</td>
<td>448</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Controller</td>
<td>209</td>
<td>51</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Integral (%)</td>
<td>(0.75) 800</td>
<td>(0.97) 515</td>
<td>(0.01) 2</td>
<td>(1.79) 2.5</td>
<td>(0.00) 0</td>
</tr>
<tr>
<td>Boundary</td>
<td>334</td>
<td>131</td>
<td>33</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Pattern unrotated</td>
<td>311</td>
<td>12</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Pattern rotated</td>
<td>1673</td>
<td>1570</td>
<td>104</td>
<td>4.5</td>
<td>3</td>
</tr>
<tr>
<td>Intensity</td>
<td>1183</td>
<td>1719</td>
<td>24</td>
<td>4.5</td>
<td>5</td>
</tr>
<tr>
<td>Orientation</td>
<td>1292</td>
<td>1292</td>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Descriptor</td>
<td>2414</td>
<td>1837</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Send keypoint</td>
<td>130</td>
<td>59</td>
<td>0</td>
<td>1.5</td>
<td>0</td>
</tr>
<tr>
<td>Send descriptor</td>
<td>187</td>
<td>67</td>
<td>0</td>
<td>1.5</td>
<td>0</td>
</tr>
<tr>
<td>Descriptor (%)</td>
<td>(7.63) 8115</td>
<td>(17.17) 9133</td>
<td>(1.06) 184</td>
<td>(13.93) 19.5</td>
<td>(9.09) 20</td>
</tr>
</tbody>
</table>
# Evaluation

## HW/SW (Comparison)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Inliers Ratio</th>
<th>Avg. Inliers Ratio (arithmetic mean)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed SW</td>
<td>91.4</td>
<td>78.7</td>
</tr>
<tr>
<td>Proposed HW</td>
<td>90.9</td>
<td>77.0</td>
</tr>
<tr>
<td>ORB + FREAK</td>
<td>88.4</td>
<td>72.5</td>
</tr>
<tr>
<td>ORB + BRISK</td>
<td>88.0</td>
<td>74.0</td>
</tr>
</tbody>
</table>
Conclusion

Summary

• Proposed implementation of AKAZE detector + FREAK descriptor
• Improved algorithm, parameters and added Retain Best function
• HW implementation of FREAK detector + Integral image
• Achieved good computation time and repeatability in comparison to others
• Achieved a low resource consumption

Outlook

• Combine AKAZE detector with FREAK descriptor in HW
• Integrate it into computer vision algorithm
Acknowledgment

This work has been funded by the German Federal Ministry of Education and Research BMBF as part of the PARIS project under grant agreement number 16ES0657.

References

Thank you for your attention!

Any questions?