Speeding Up Object Detection – Fast Resizing in the Integral Image Domain

Michael Gschwandtner, Andreas Uhl and Andreas Unterweger

Department of Computer Sciences
University of Salzburg

January 5, 2014
Integral images

- Value at \((x, y)\): Pixel sum between top-left corner and \((x - 1, y - 1)\)
- Allow fast summation $\rightarrow$ fast filtering
- Perfect reconstruction possible
- Used in Viola & Jones’ object detection algorithm

![Integral Image Diagram](image.png)

Adopted from Crow (1984)
Object detection (multi-scale LBP detector from OpenCV)
Modified object detection

Image Domain

Scale 1

II Domain

Scale 2

Detector

Scale 3

Detector

Scale n

Detector

...
Integral image resizing approach

- **Dyadic case (power-of-two resizing)**
  - One division per pixel (very fast)
  - Perfect reconstruction
  - Detailed proof in paper

- **General case**
  - Reduction to bilinear interpolation plus error
  - Error is very small (results in sub-pixel shifts after reconstruction)
  - Special handling of borders (details in paper)

- **Future work:** Tilted integral image resizing
Parallelizability (dyadic case)

- Dyadic resizing with straight-forward CUDA implementation
- One pixel per thread to assess scalability

Andreas Unterweger (Univ. of Salzburg)

Speeding Up Object Detection

January 5, 2014
Quality (non-dyadic case)

- Bilinear resizing (OpenCV) vs. our resizer after reconstruction: LIVE

Andreas Unterweger (Univ. of Salzburg)
LBP detector in OpenCV with CMU/MIT frontal face data sets

OpenCV: Optimized resizers for all but integral image data types

Theoretical measurement: OpenCV with vs. without integral image calculation on scales $n > 1$ (no unoptimized resizing)

Identical detection rates with default settings (scale factor 1.1)

<table>
<thead>
<tr>
<th>CPU cores</th>
<th>System</th>
<th>Average</th>
<th>Stdev.</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A*</td>
<td>2.9%</td>
<td>0.71%</td>
<td>1.57%</td>
<td>6.78%</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>4.66%</td>
<td>0.66%</td>
<td>2.92%</td>
<td>6.89%</td>
</tr>
<tr>
<td>4</td>
<td>B**</td>
<td>6.38%</td>
<td>0.78%</td>
<td>4.38%</td>
<td>9.87%</td>
</tr>
<tr>
<td>64</td>
<td>C</td>
<td>12.6%</td>
<td>4.86%</td>
<td>4.21%</td>
<td>37.25%</td>
</tr>
</tbody>
</table>

* Intel TBB support disabled, ** 2 cores with hyper-threading
Conclusion

- Approach to resize integral images
  - Highly parallelizable
  - No quality impact for dyadic rescaling
  - Low quality impact on non-dyadic rescaling

  Used in multi-scale object detector
  - Notable speedup on 2-core system with HT (6.38% on average)
  - Significant speedup on 64-core system (12.6% on average)
  - No impact on detection rate
  - Can be used for other multi-scale detectors as well
Thank you for your attention!

Questions?