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

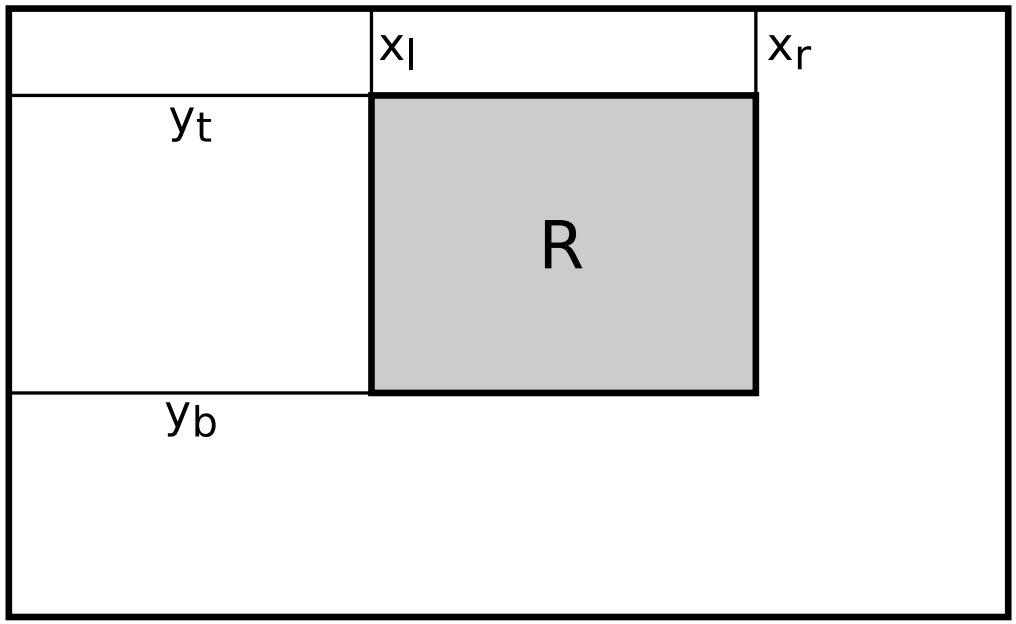
Michael Gschwandtner, Andreas Uhl and **Andreas Unterweger**

Department of Computer Sciences
University of Salzburg

January 5, 2014

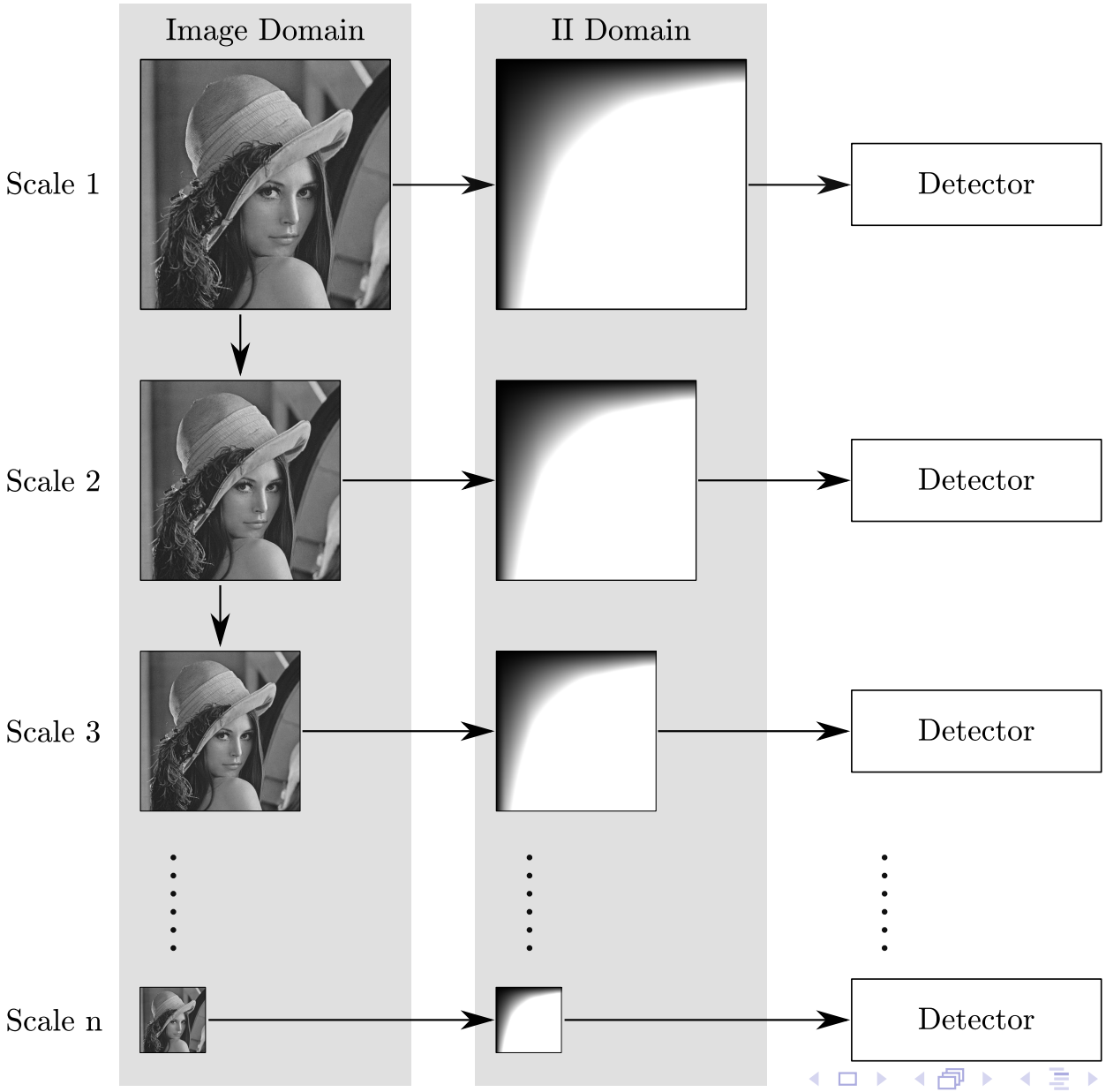
Integral images

- 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

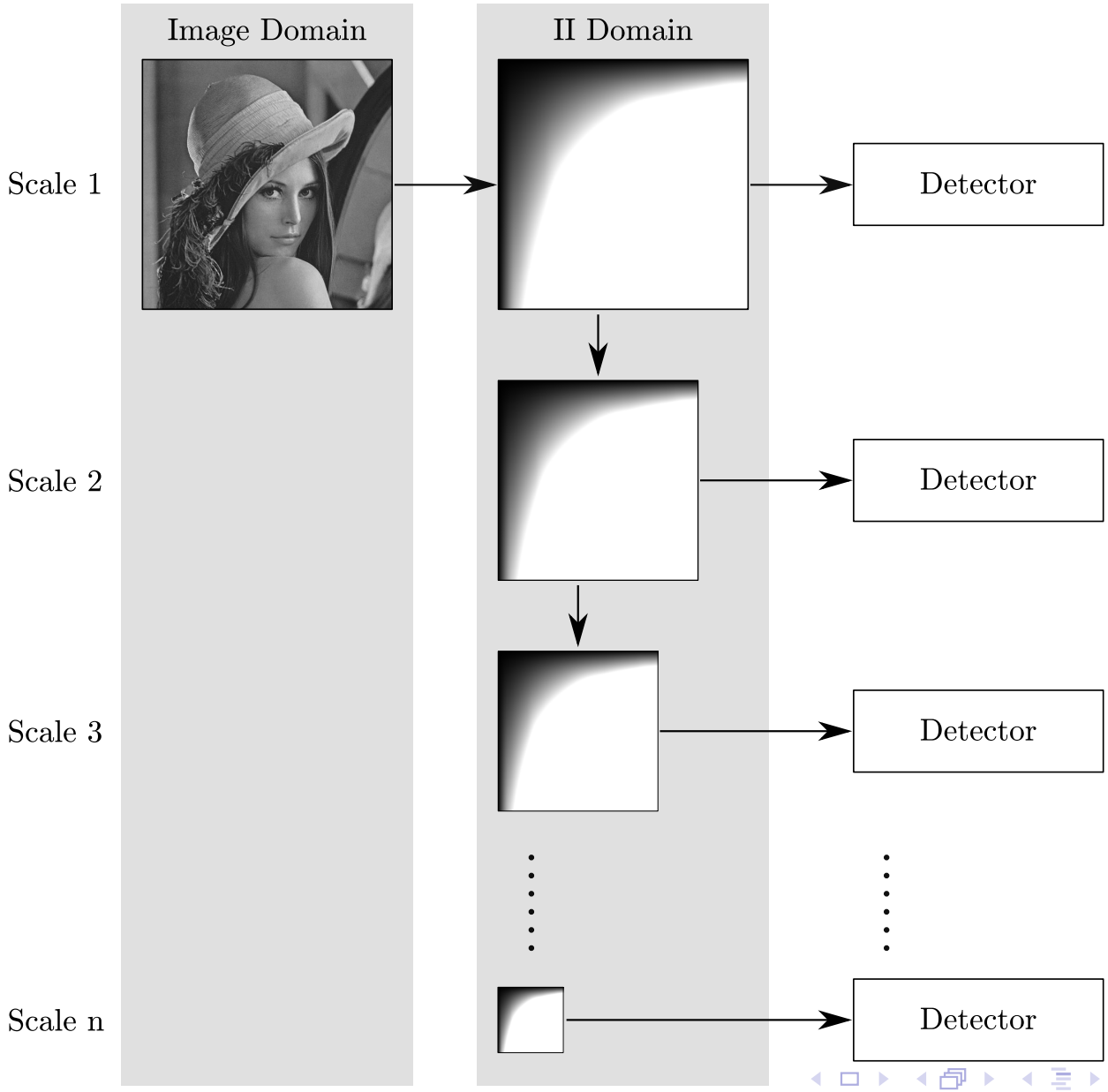


Adopted from Crow (1984)

Object detection (multi-scale LBP detector from OpenCV)



Modified object detection

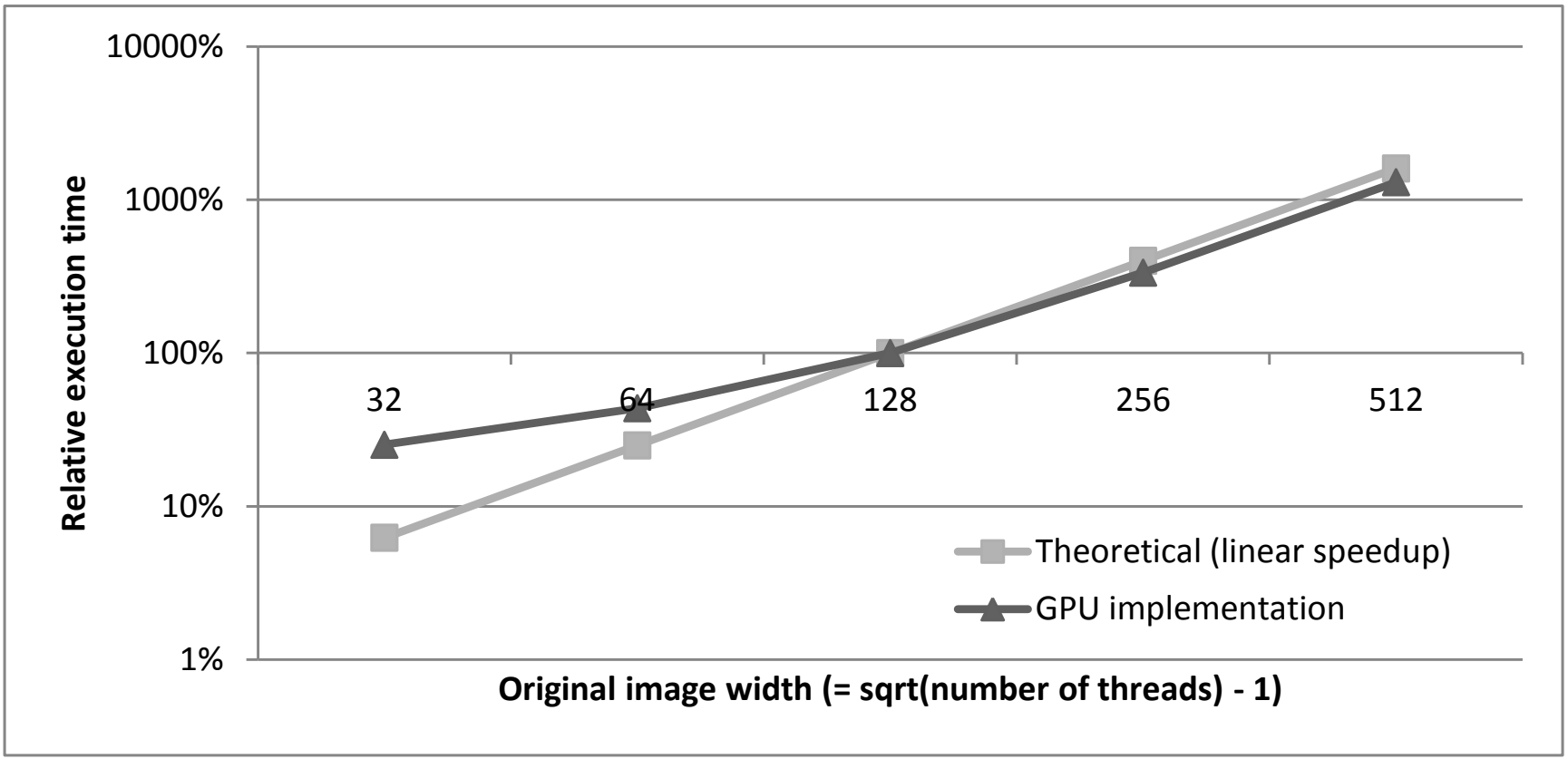


Integral image resizing approach

- Dyadic case (power-of-two resizing)
  - One division per pixel (very fast)
  - Perfect reconstruction
  - Detailled 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

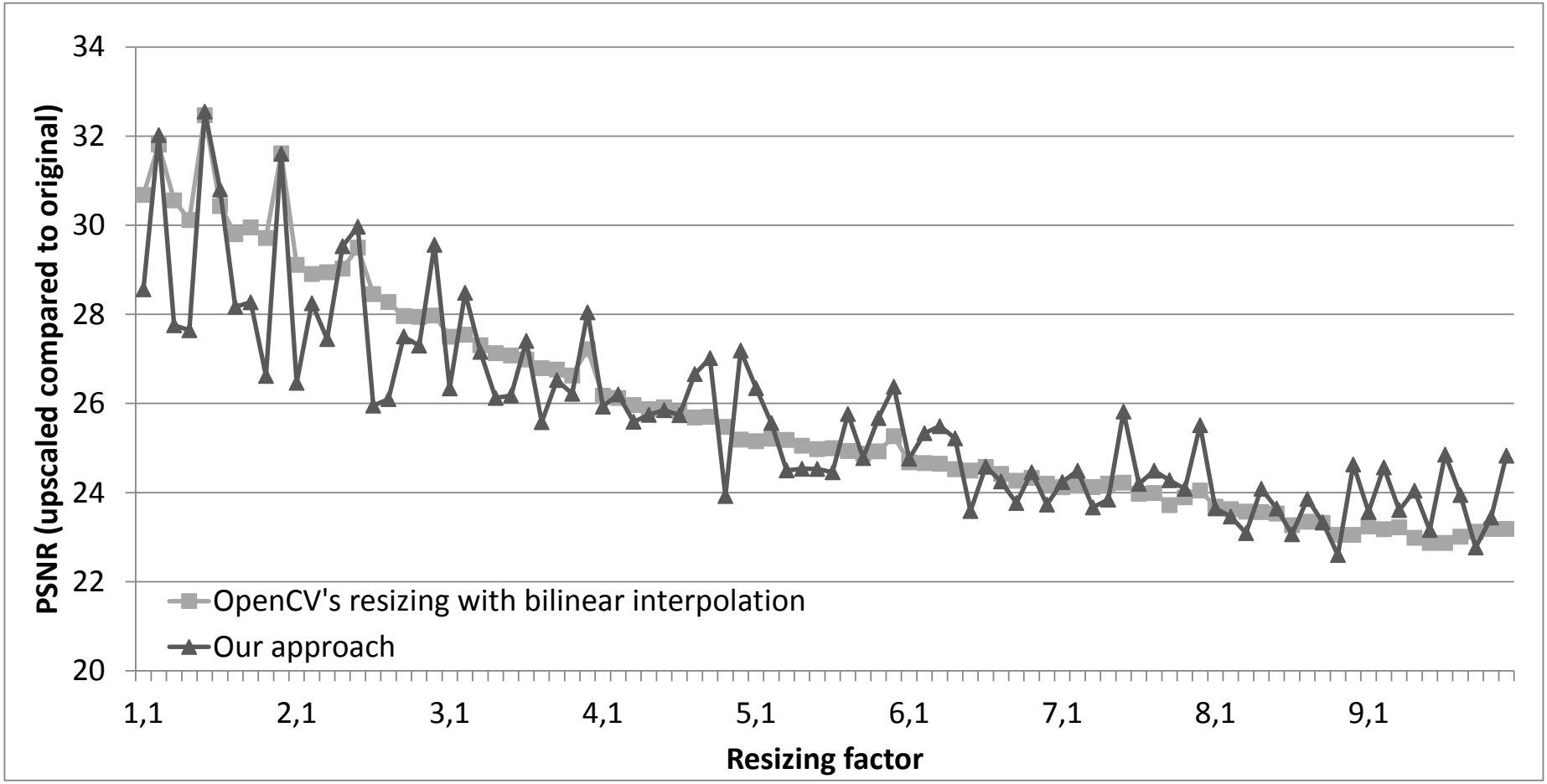
Parallelizability (dyadic case)

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



Quality (non-dyadic case)

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



Object detection speedup

- 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)

CPU cores	System	Average	Stdev.	Minimum	Maximum
1	A*	2.9%	0.71%	1.57%	6.78%
2	A	4.66%	0.66%	2.92%	6.89%
4	B**	6.38%	0.78%	4.38%	9.87%
64	C	12.6%	4.86%	4.21%	37.25%

* 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?