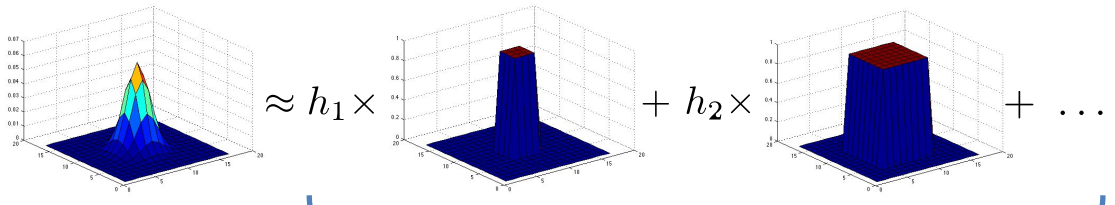
4th IEEE Int'l Workshop on Mobile Vision, 2014

Cascade of Box (CABOX) Filters for Fast Scale Space Approximation

Victor Fragoso[1], Gaurav Srivastava[2], Abhishek Nagar[2], Zhu Li[2], Kyungmo Park[2] and Matthew Turk[1]

[1]Samsung Research America

[2]University of California, Santa Barbara

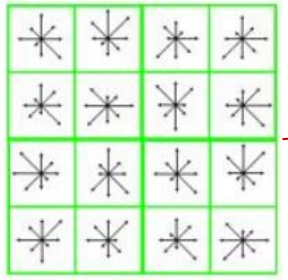


Outline

- Motivation
- Approximate Scale Space with Box Filters
- Simulation Results
- Conclusion & Future Work

Mobile Visual Search Scenarios and Problems

- Technology: key point based Image Search
  - Scale and rotation invariant key points that can be matched up in query:

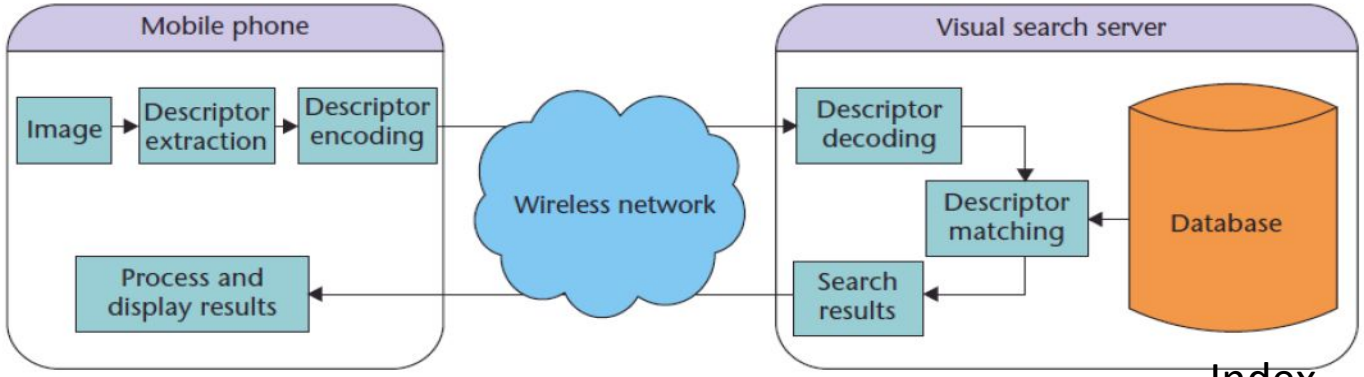


- Applications:
  - Mobile Query by Capture, Augmented Reality

- **What is wrong** by sending images/raw features over the channel ?

MPEG-7 Compact Descriptor for Visual Search (CDVS)

- MPEG CDVS Standardization Scope
  - –Front-end: image feature capture and compression
  - –Server Back-end: image feature indexing and query processing



- Objectives/Challenges:
  - –**Real-time**: front end real time performance, e.g, 640x480 @30fps
  - –**Compression**: Low bit rate over the air, achieving 100 X compression w.r.t to sending images, or 10X compression of the raw features.
  - –**Matching Accuracy**: >95% accuracy in pair-wise matching (verification) and >90% precision in identification
  - –**Indexing/Search Efficiency**: real time backend response from BIGDATA (100m) visual repository

SIFT Patent and Detection Speed

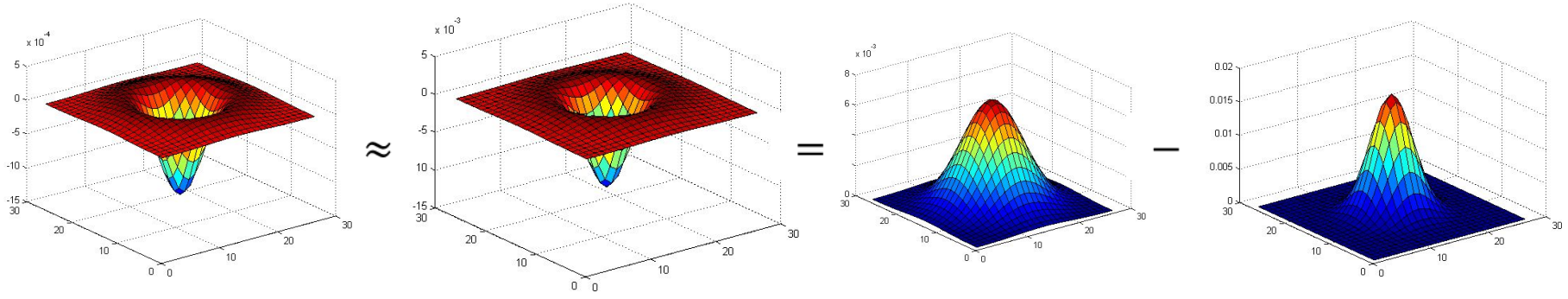
- SIFT Patent
  - SIFT patent was licensed to an unknown party, and the un-certainty of its licensing policy puts MPEG CDVS in jeopardy
  - Need to circumvent the key claims of the patent – DoG filtering
- Complexity Reduction:
  - The SIFT detection is computationally heavy
  - For mobile application deployment and especially for AR type applications, where visual queries are generated at much faster pace, the complexity need to be significantly reduced for mobile computing platform.

Outline

- Motivation
- Approximate Scale Space with Box Filters
- Simulation Results
- Conclusion & Future Work

Blob detection: Difference of Gaussians (DoG)

LoG DoG Gaussian 2 Gaussian 1



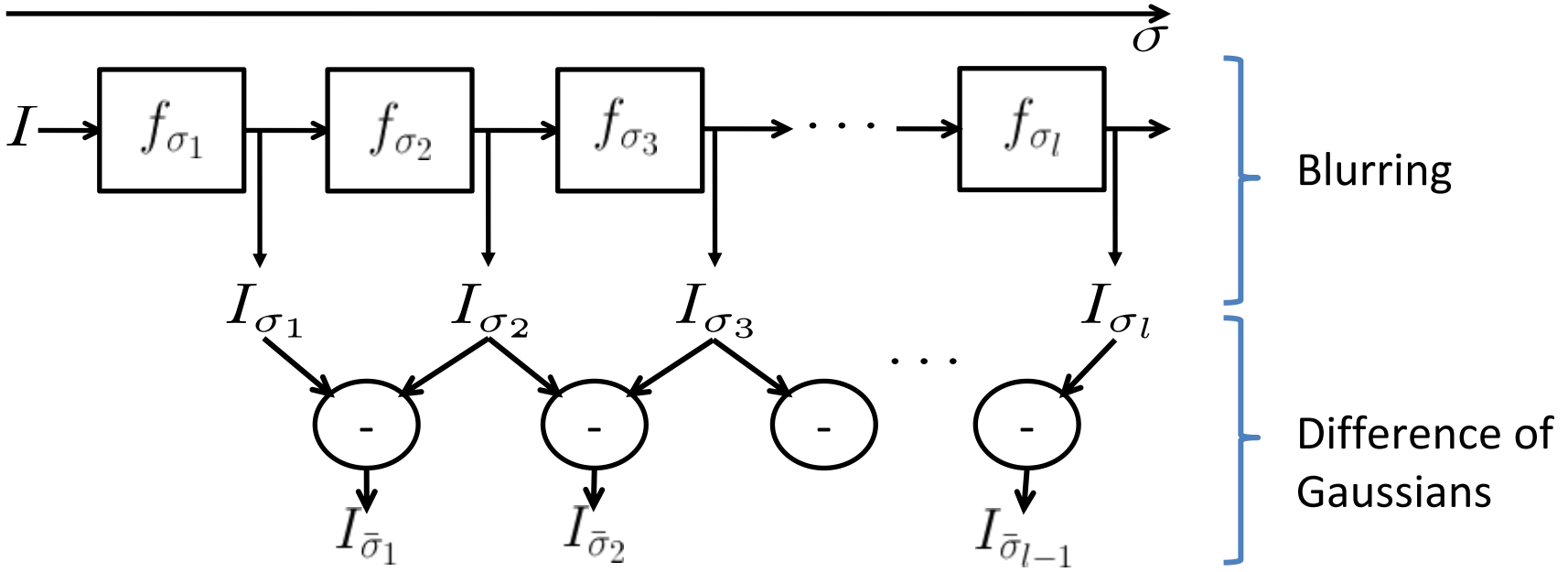
L_σ D_σ g_{σ_2} g_{σ_1}

- Convolving the filter with an image in an efficient manner:

$$I * L_\sigma \approx I * \underbrace{(g_{\sigma_2} - g_{\sigma_1})}_{D\sigma} = (I * g_{\sigma_2}) - (I * g_{\sigma_1})$$

- We can exploit the separability property of the Gaussian filter!
- Requires O(2N) multiplications and O(2N) additions per pixel!

Difference of Gaussians pipeline

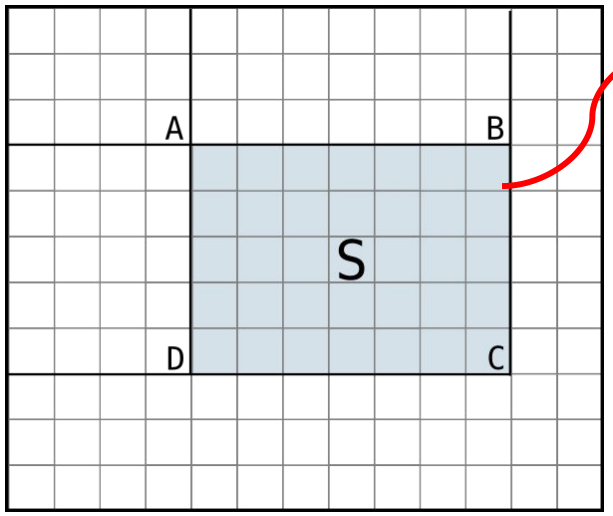


$$\bar{\sigma} = \sqrt{\frac{\sigma_a^2 + \sigma_b^2}{2}}$$

LoG $\bar{\sigma}$s calculated from the Gaussian parameters and σ_a σ_b

Can we speed up this process even further?

Gaussian kernel approximation using Box filters

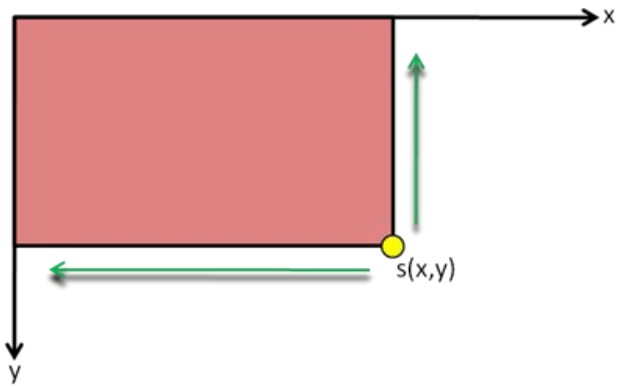


Box Filter

For a given pixel, the box filter computes the average over the rectangle (A,B,C,D).

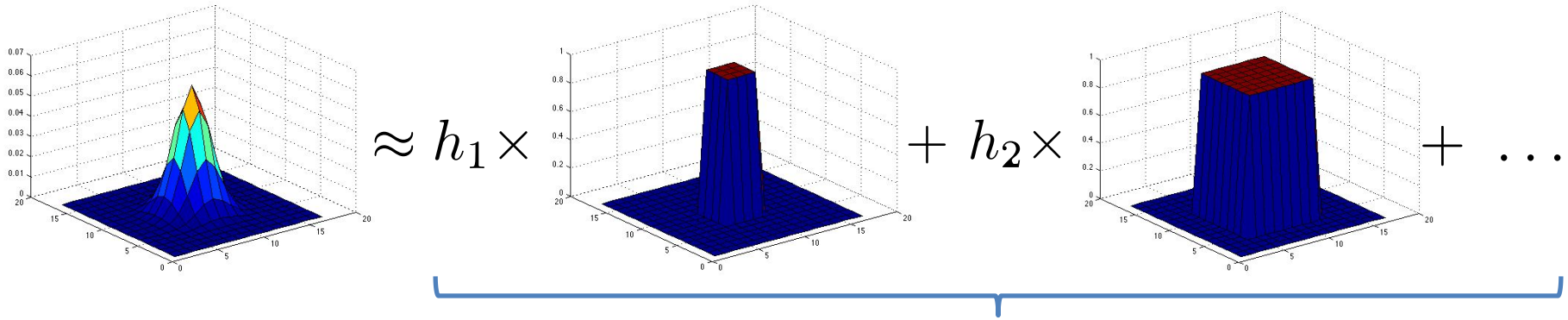
$$S = In(C) + In(A) - In(D) - In(B)$$

Requires only 4 additions and 1 multiplication!



Integral Image: A table whose entries hold the sum of the pixels from the origin to the point (x, y).

CABOX: Gaussian kernel approximation using Box filters



Gaussian Filter g_σ

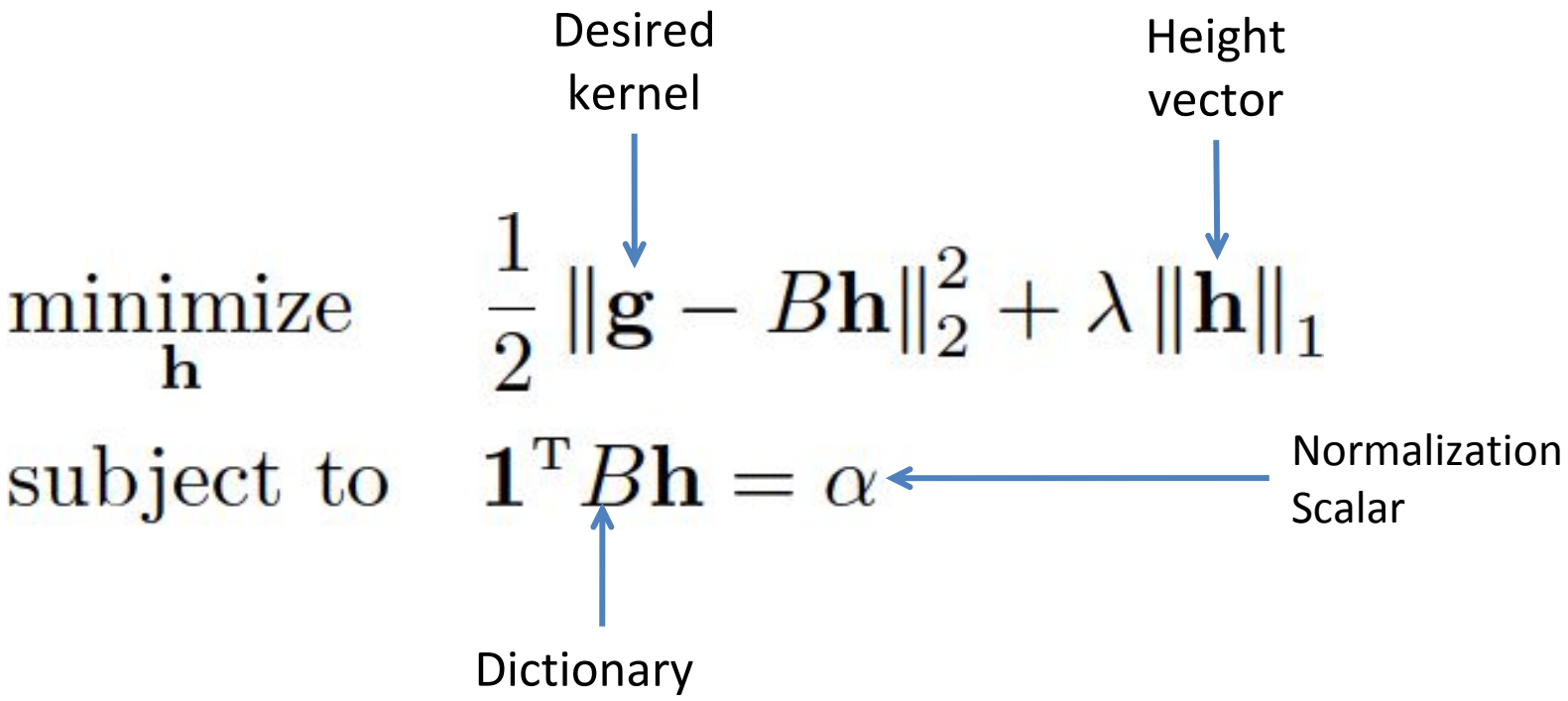
Box Filters b_σ

We need to find a coarse approximation of the Gaussian kernel using a set of k box filters.

- We can use integral images for fast convolution
- Complexity can be reduced if we use a few box filters
  - O(k) multiplications and O(4k) additions, where k is the number of box filters used.

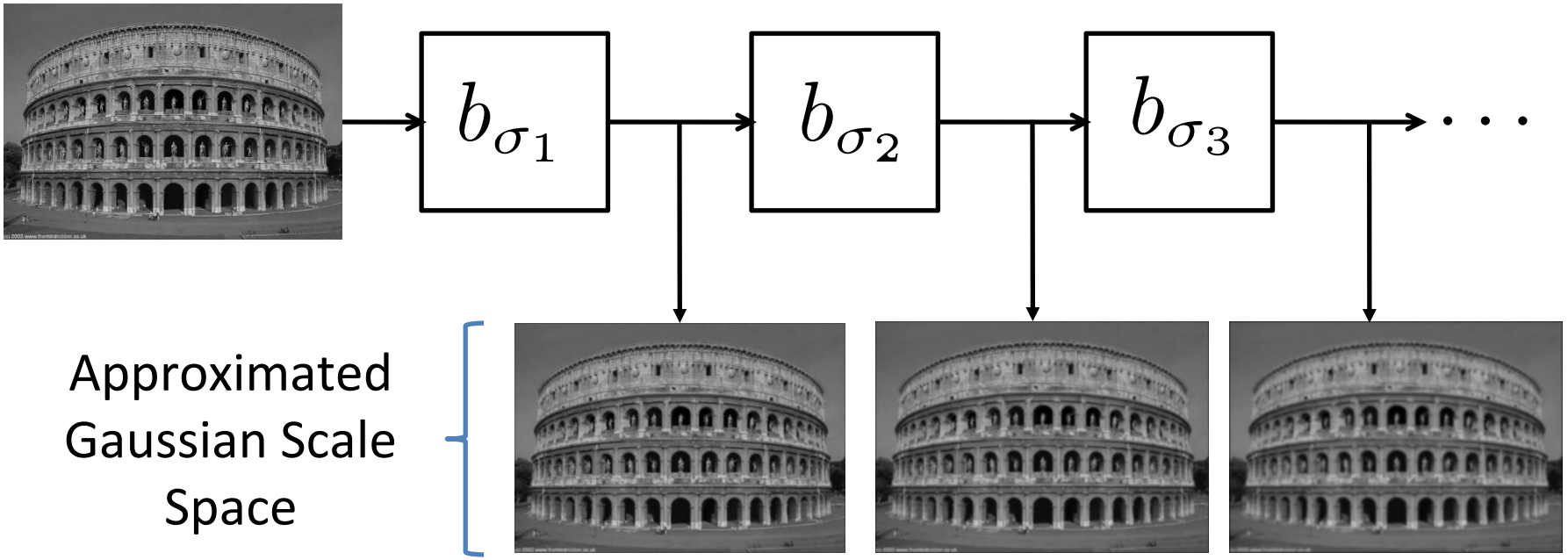
CABOX design

We can approximate a Gaussian kernel with a set of box filters by solving the following problem:



CABOX: Gaussian kernel approximation using Box filters

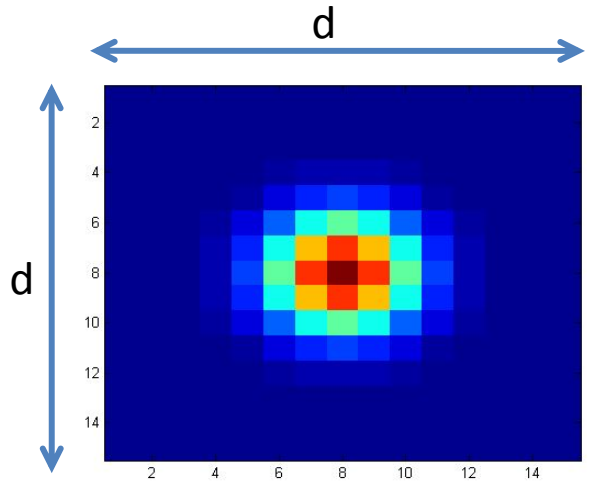
We can then plug in our approximations in the scale space pipeline.



In essence, we are doing the following approximation:

$$I * L_\sigma \approx (I * g_{\sigma_2}) - (I * g_{\sigma_1}) \approx (I * b_{\sigma_2}) - (I * b_{\sigma_1})$$

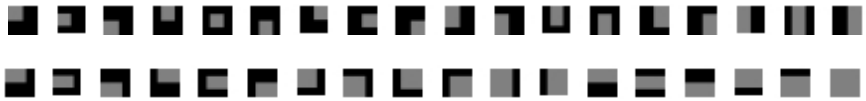
What Box filters to use for the approximation?



The size of the Gaussian kernel is calculated with the following formula:

$$d = 2\lceil 4\sigma \rceil + 1$$

Extended Dictionary: All possible rectangles with minimum size of 2.



Concentric Squares Dictionary.

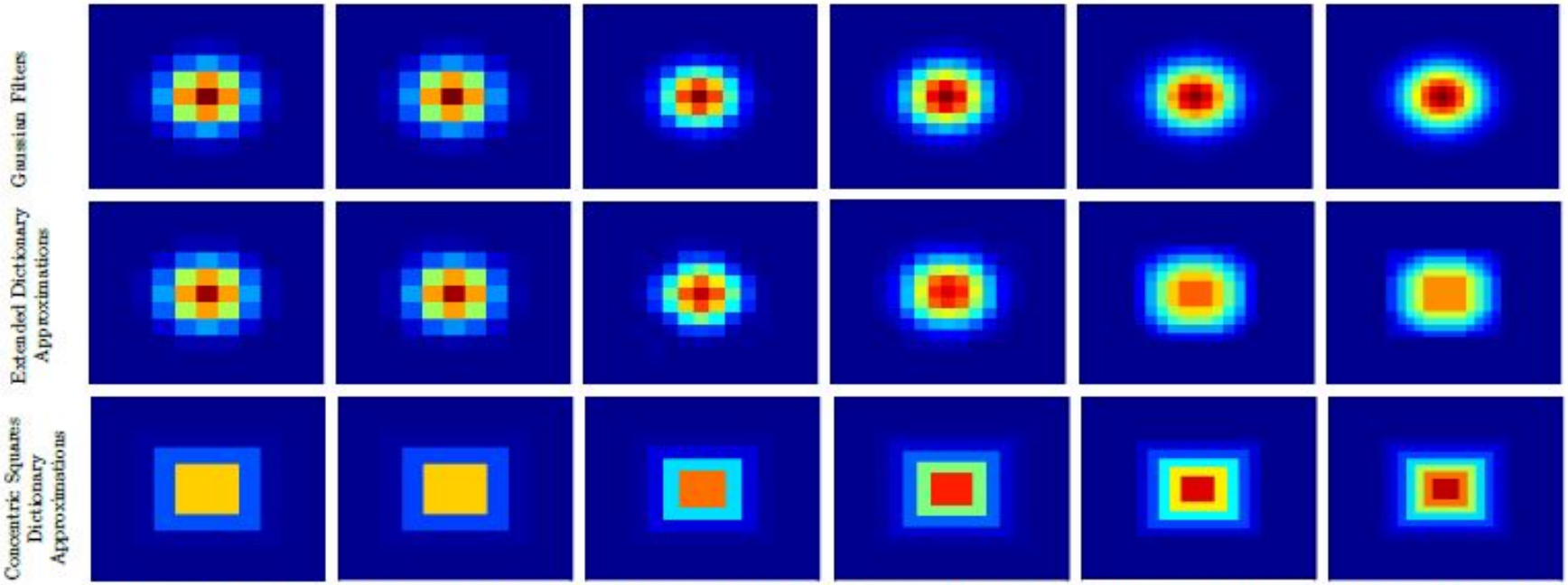
The dictionary will affect the number of boxes used when solving the optimization problem.

Approximation results

$$\text{residual} = \|\mathbf{g} - B\mathbf{h}\|_2$$

	Concentric Squares		Extended	
σ	Num. of Boxes	Residual	Num. of Boxes	Residual
1.249000	3	0.0554	43	0.0068
1.226270	3	0.0578	21	0.0067
1.545008	4	0.0358	42	0.0146
1.946588	6	0.0248	36	0.0087
2.452547	5	0.0192	16	0.0169
3.090016	8	0.0142	18	0.0163

The influence of the used dictionary determines not only the quality of the approximation but also the number of boxes required.

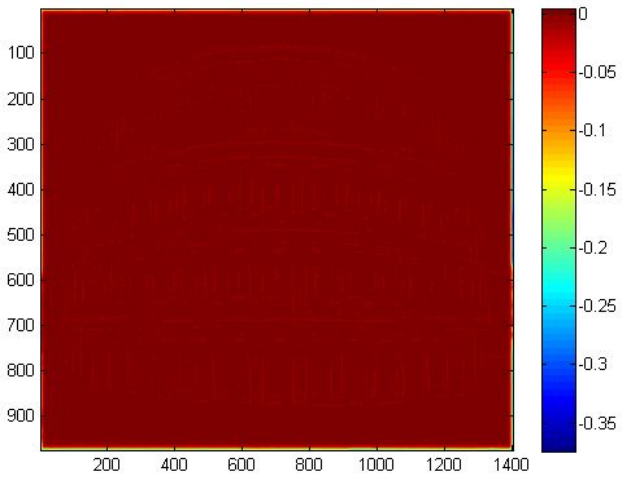


(a) $\sigma = 1.249000$ (b) $\sigma = 1.226270$ (c) $\sigma = 1.545008$ (d) $\sigma = 1.946588$ (e) $\sigma = 2.452547$ (f) $\sigma = 3.090016$

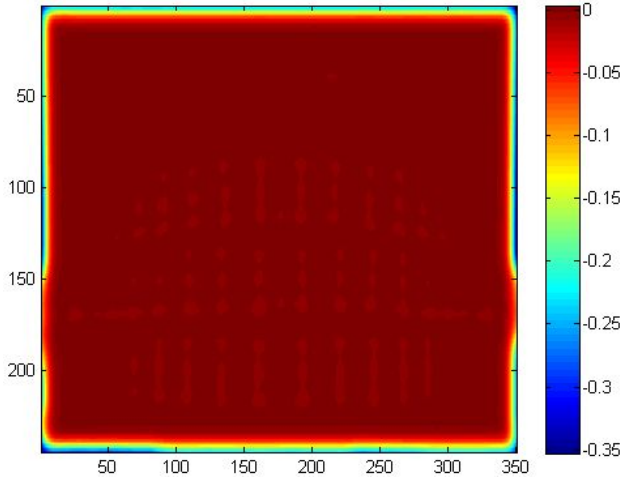
Gaussian Scale Space experiments

	Colosseum	Oxford Bldg.	Zürich
			
MSE	0.0045	0.0015	0.0063
Octaves	5	6	5
Scales	6	6	6

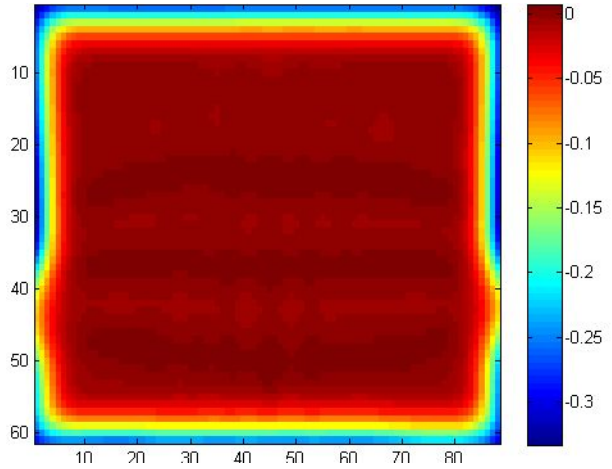
The Mean Square Error (MSE) of our approximation method across all octaves and scales is minimal, considering that the pixel intensities are in the range of 0 to 1.



Error for Scale=6
Octave=1



Error for Scale=6
Octave=3



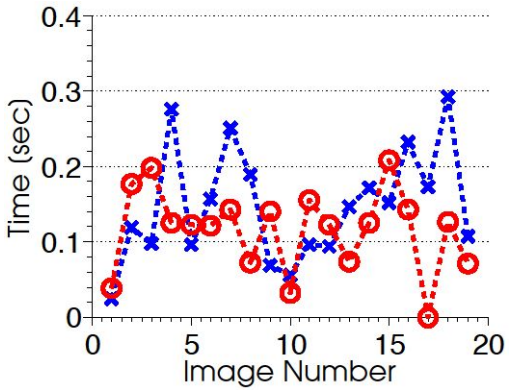
Error for Scale=4
Octave=5

Outline

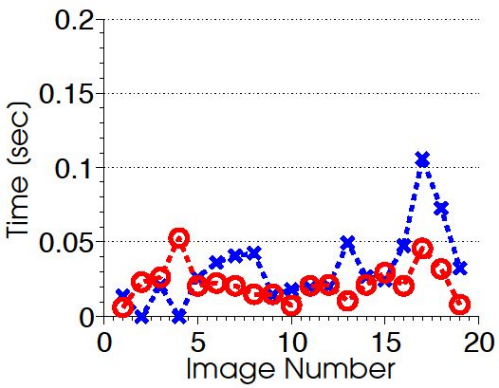
- Motivation
- Approximate Scale Space with Box Filters
- Simulation Results
- Conclusion & Future Work

Complexity Reduction

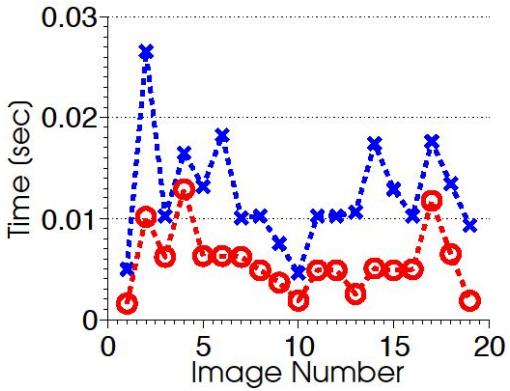
On average CABOX reduced the scale space construction time in approximately 44%.



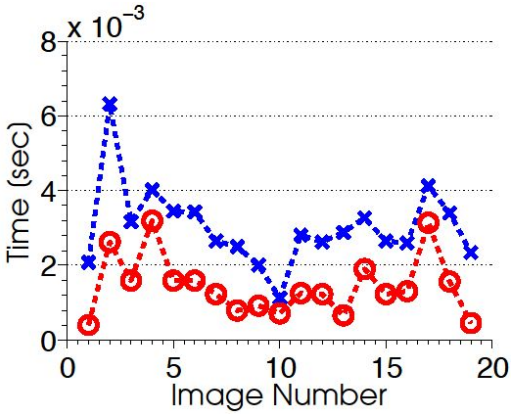
22% reduction time for 1st octave.



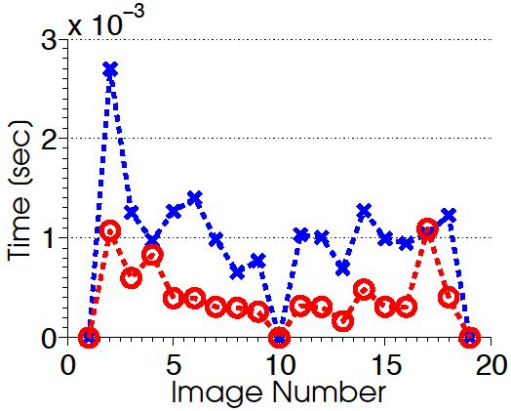
32% reduction time for 2nd octave.



54% reduction time for 3rd octave.



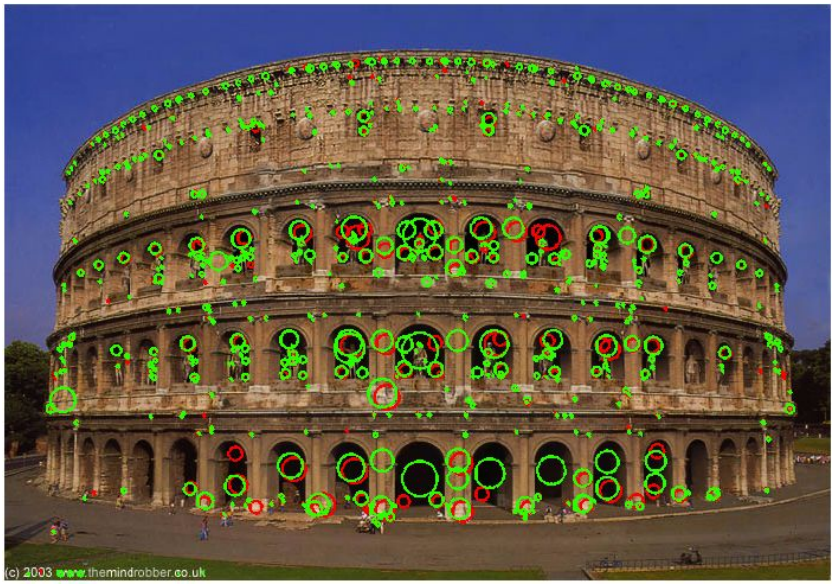
52% reduction time for 4th octave.



59% reduction time for 5th octave.

Gaussian
CABOX

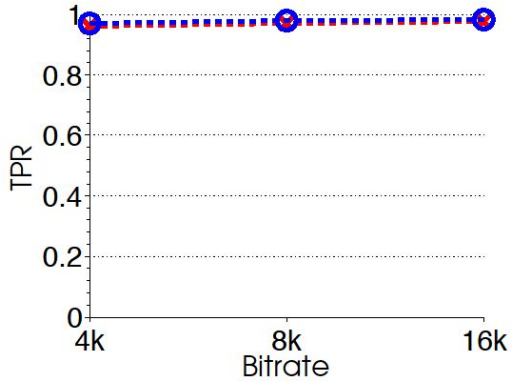
Feature detection repeatability test



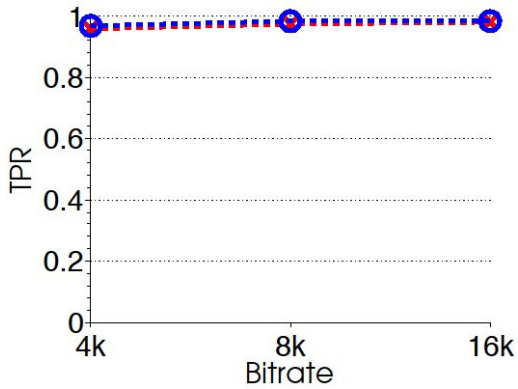
Of those features detected with CABOX (red circles), 89% are also detected by VLFeat (green circles) while the remaining 11% correspond to new features.

End-End MPEG TM Pairwise matching experiment

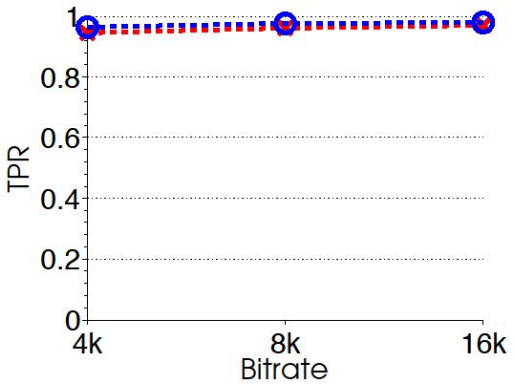
Replacing DoG with CABOX, has minimum effects on the image matching accuracy



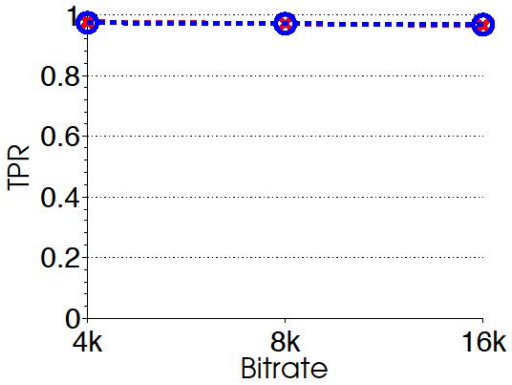
Dataset 1



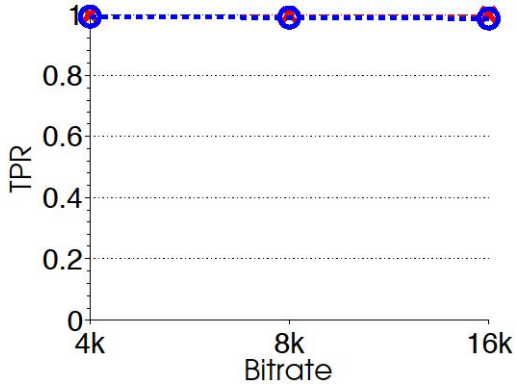
Dataset 2



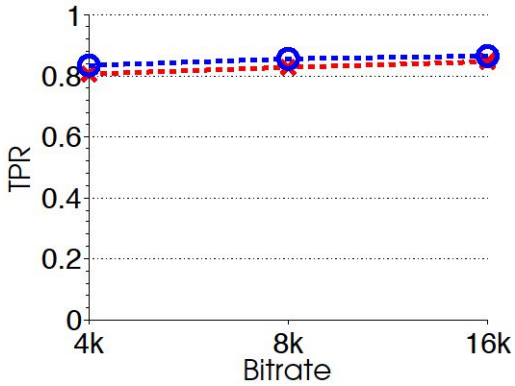
Dataset 3



Dataset 4



Dataset 5



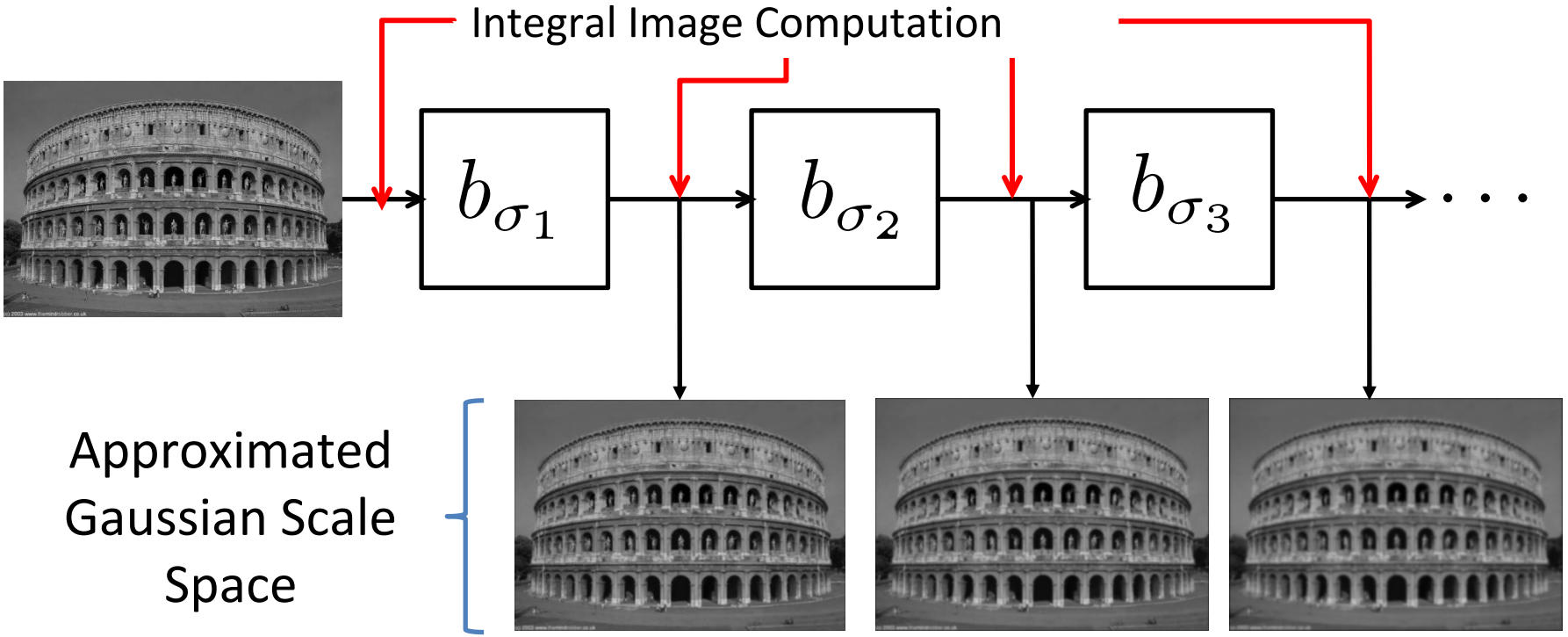
Dataset 6

Gaussian
CABOX

Summary & Future Work

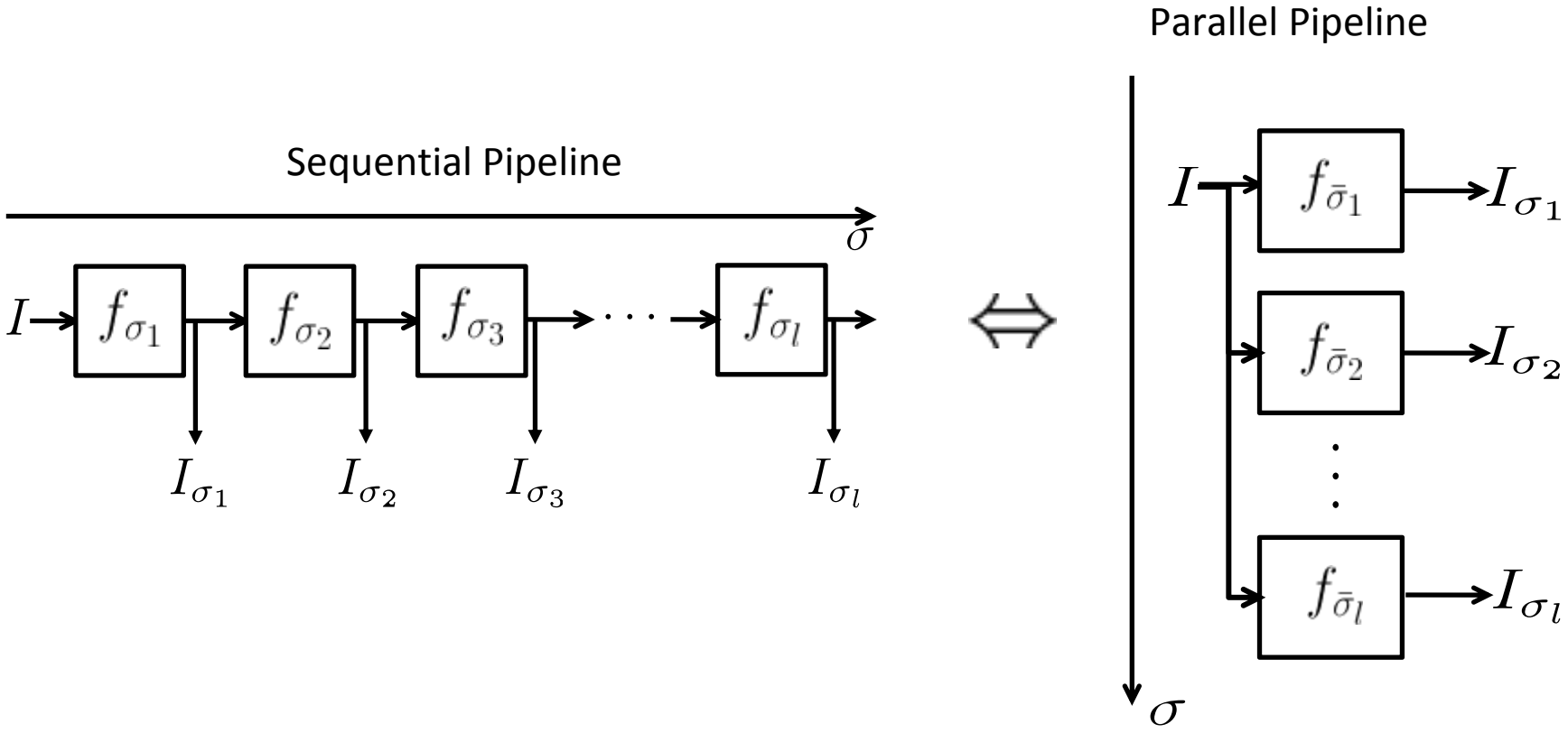
- We presented an integral domain box filter approximation of scale space which is fast
- The accuracy of scale space approximation is achieved by solving the sparsity constrained filter approximation
- Integration into the MPEG CDVS Test Model verified the robustness and efficiency of the proposed solution
- Future Work:
  - Develop Parallel CABOX filtering for even faster solution

Further Speed-up: Parallel CABOX



- Computation of integral image at every scale level!
- Approximation error propagates along the pipeline!

Parallel CABOX

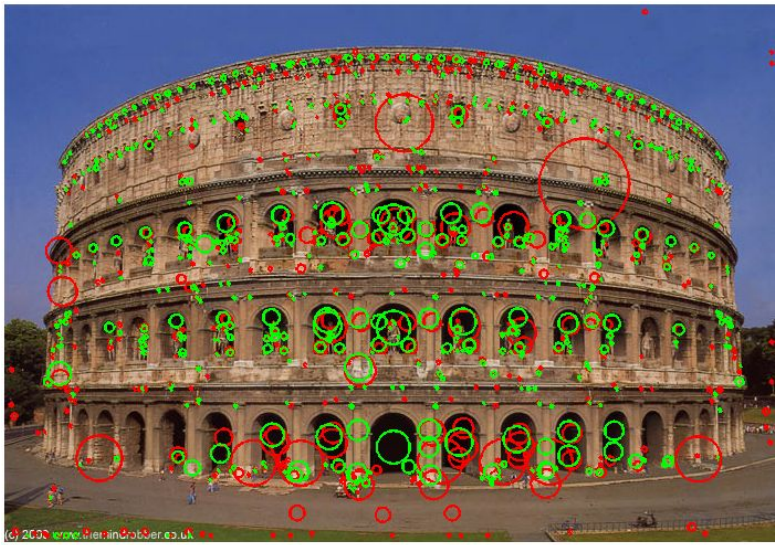


Filters used in the Parallel pipeline are computed using the following formula:

$$\bar{\sigma}_i = \sqrt{\sigma_i^2 + \bar{\sigma}_{i-1}^2}$$

$$\bar{\sigma}_1 = \sigma_1$$

Parallel CABOX Detection Repeatability



Of those features detected with Parallel CABOX (red circles), 72% are also detected by VLFeat (green circles) while the remaining 28% correspond to new features.