FPL 2012

Hardware Implementation of Motion Blur Removal

Cabral, Amila. P. , Chandrapala, T. N.
Ambagahawatta ,T. S. , Ahangama, S.
Samarawickrama, J. G.

University of Moratuwa

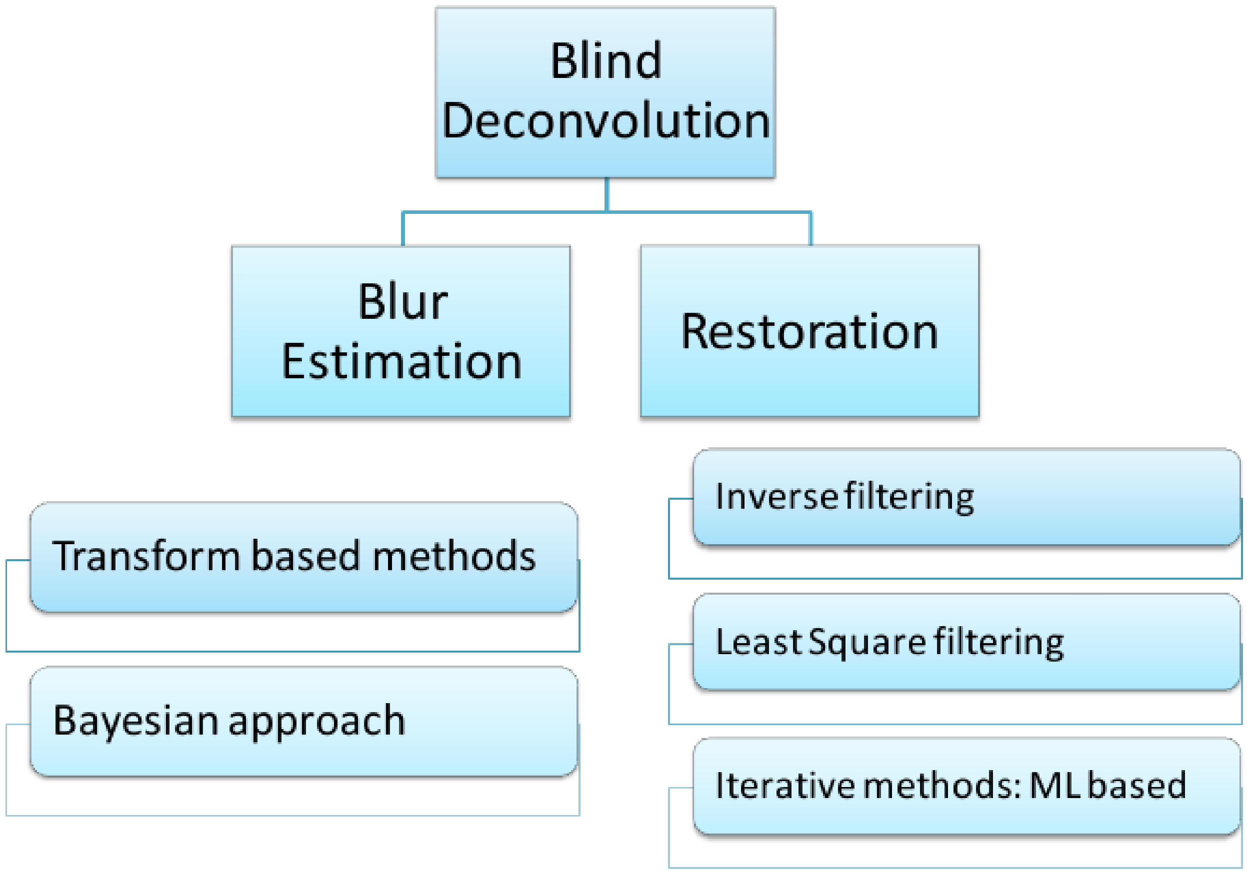
Problem and Motivation

- Photographic images and videos are highly susceptible to **Motion Blur** due **camera shakes**
- To remove **Uniform** motion blur with **only** with image(s) itself is form of **Blind de-convolution**
  - Algorithms are complex,
  - Usually implemented in Software.
  - Difficult to achieve real-time performance

Problem and Motivation cont'd..

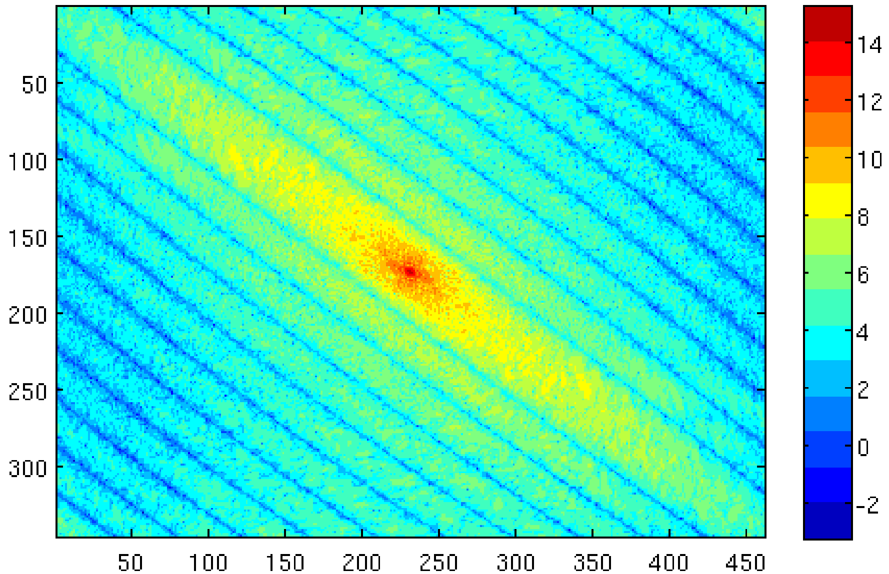
- One to one hardware mapping from software to hardware must be done carefully.

Algorithm Development

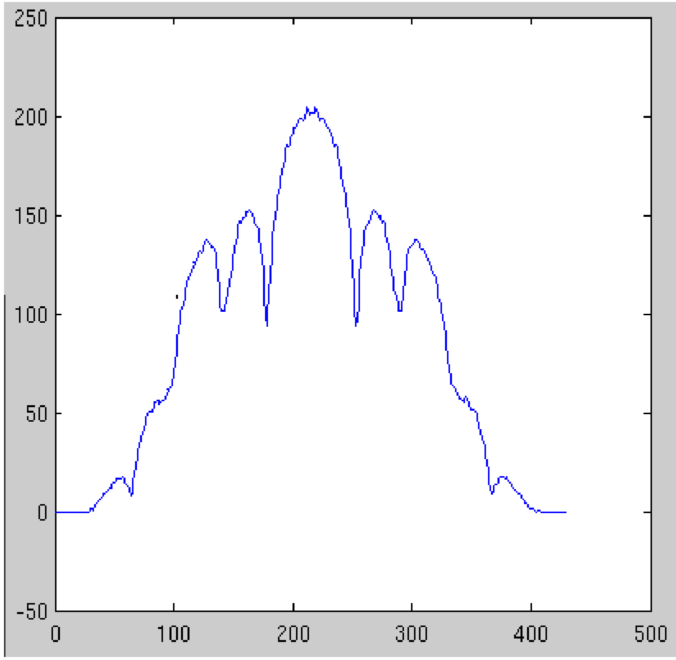


Blur Kernel Identification

Fourier domain

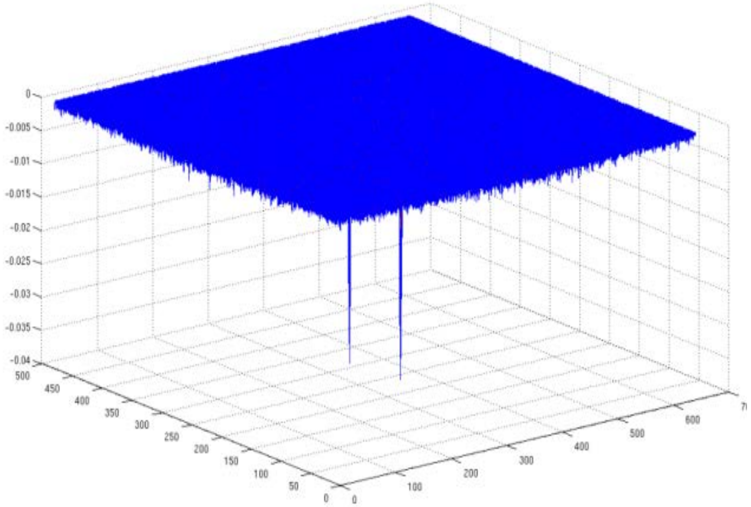


Radon transform



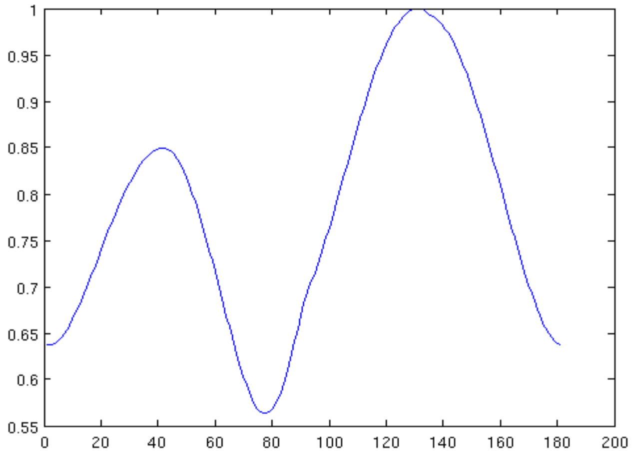
Blur Kernel Identification cont'd…

Cepstrum domain extraction



Two negative peaks -blur direction and the blur length

Directional Derivative method



Lowest value occurs at the direction of the blur

Blur Kernel Identification cont'd …

- Strengths and Weaknesses
  - Fourier Domain
    - Difficult to obtain quantitative values
  - Radon Transform
    - Non iterative
    - Computational complexity relatively high
  - Cepstrum method
    - Non iterative
    - Requires comparatively less memory
    - Acceptable accuracy
  - Directional Derivative method
    - Requires isotropic images
    - High memory usage
    - Calculation complexity is high to obtain good accuracy

Restoration Methods

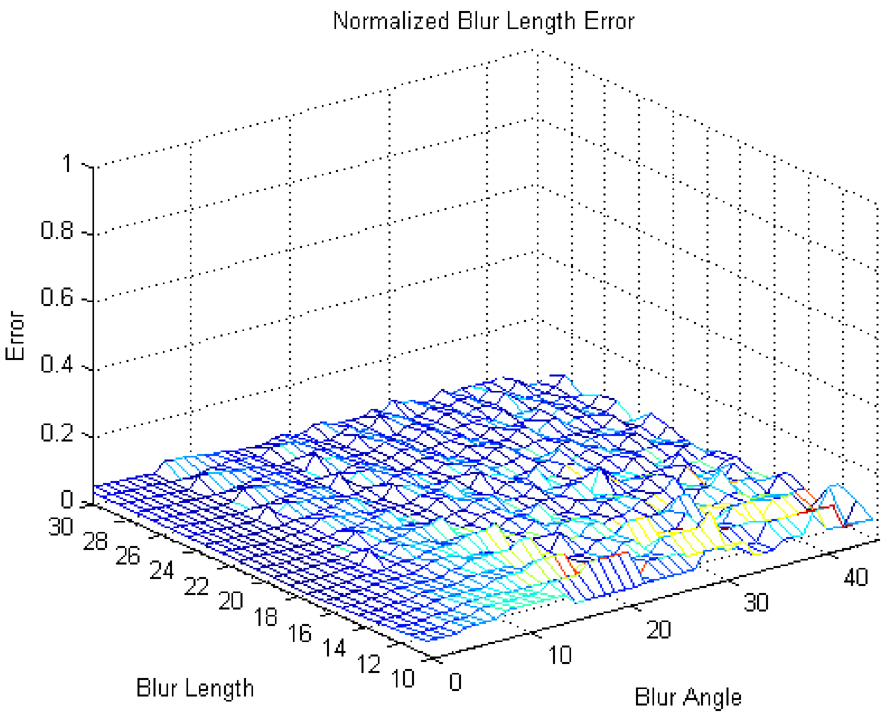
Strengths and Weaknesses

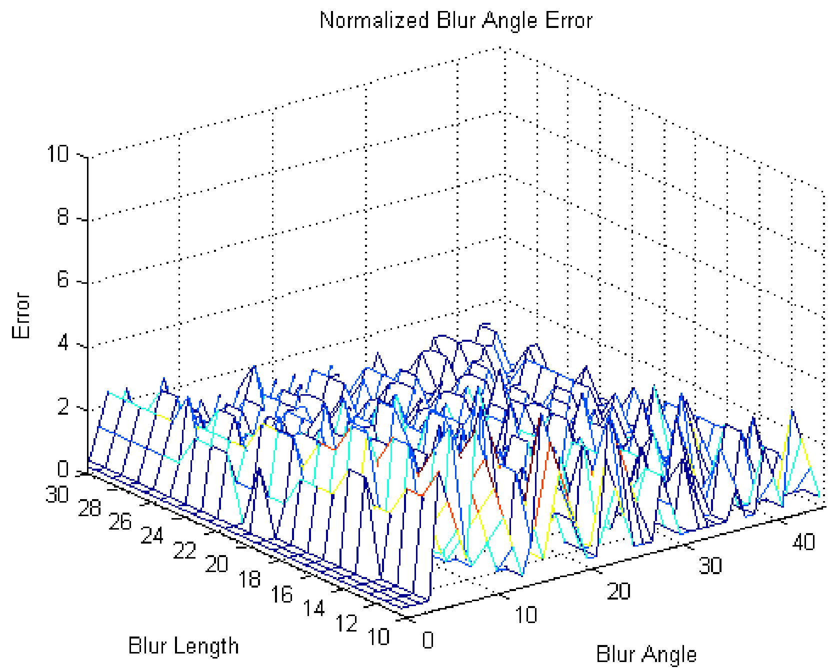
- **Least Mean Square filter (Wiener filter model)**
  - Non iterative
  - Introduces ringing effects
- **Lucy Richardson algorithm**
  - Iterative
  - Good accuracy
- **Regularized inverse method (Stationary Wiener filter model)**
  - Non iterative
  - Computational cost is relatively low

Software Implementation - Detection

Cepstrum Method: Analysis of Errors

- Length detection error
- Angle detection error





Software Implementation - Restoration

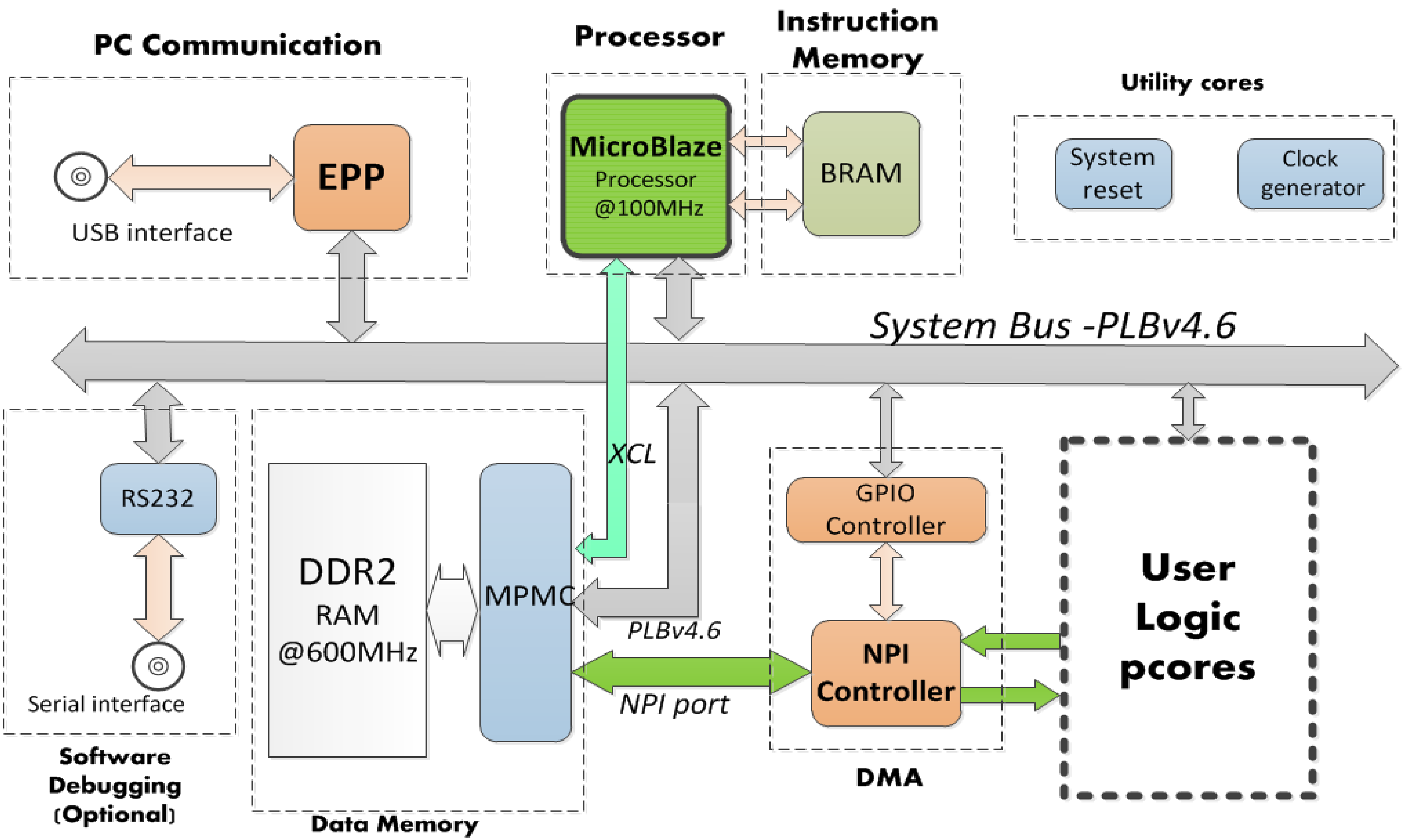
Regularized inverse filter based method

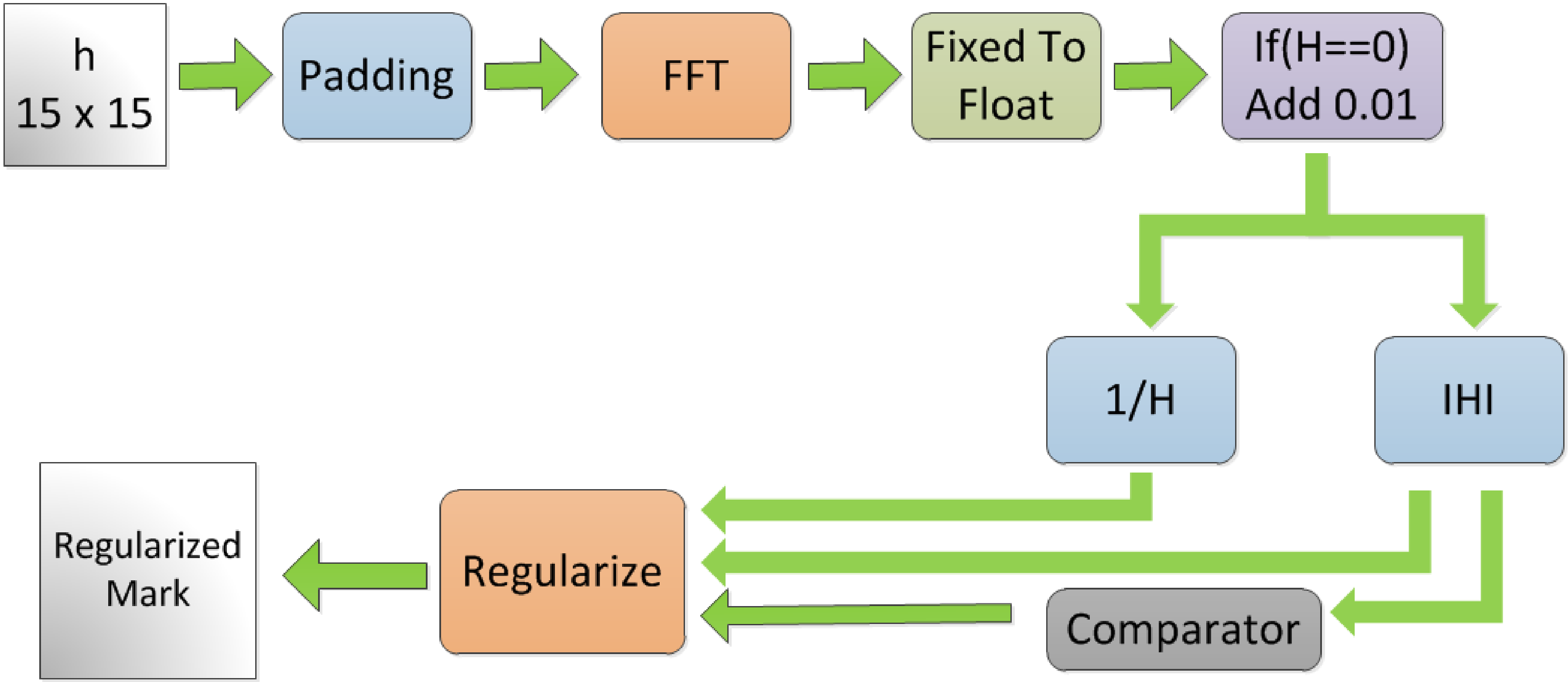
Blurred image

Filtered image

Time: For 1280x720 frame:
1.125 s (Core2 Duo with 4GB RAM at 1066MHz)

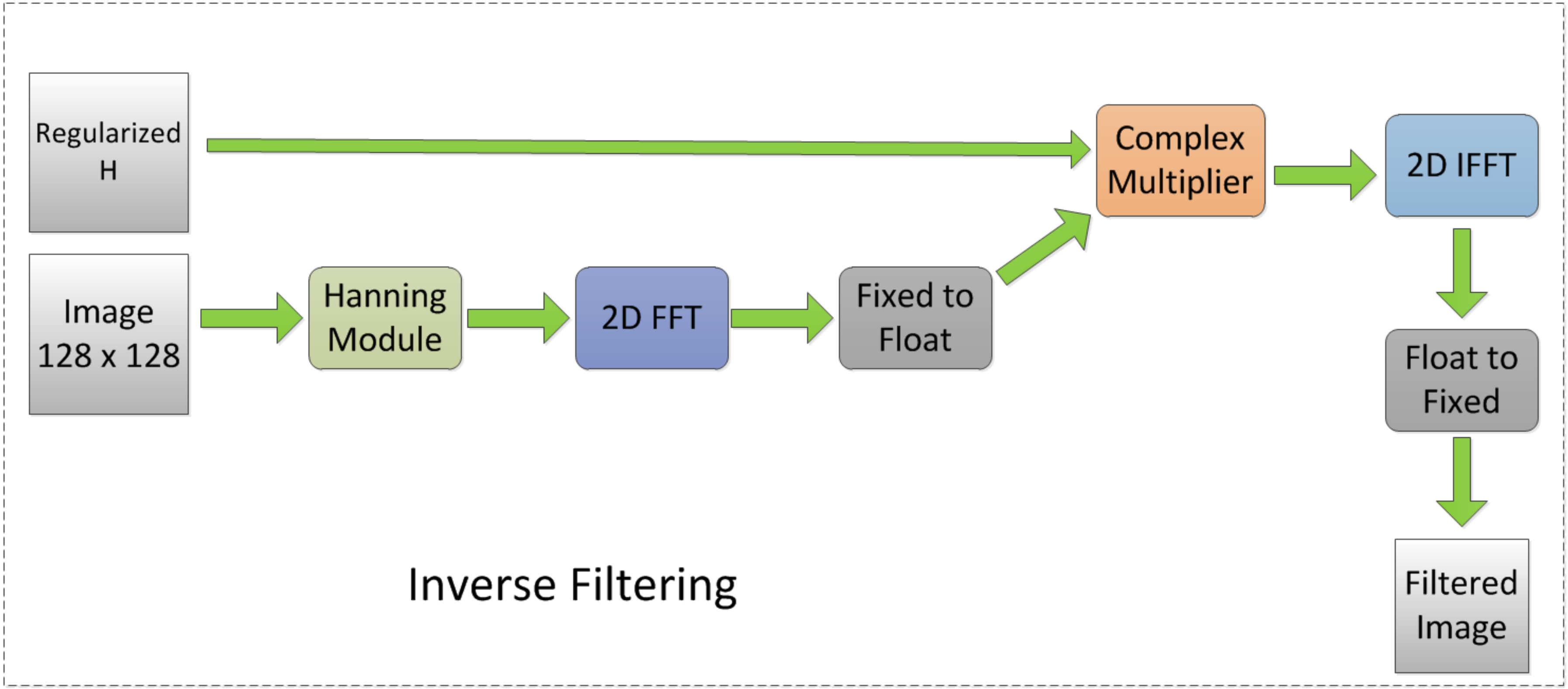
Hardware Implementation





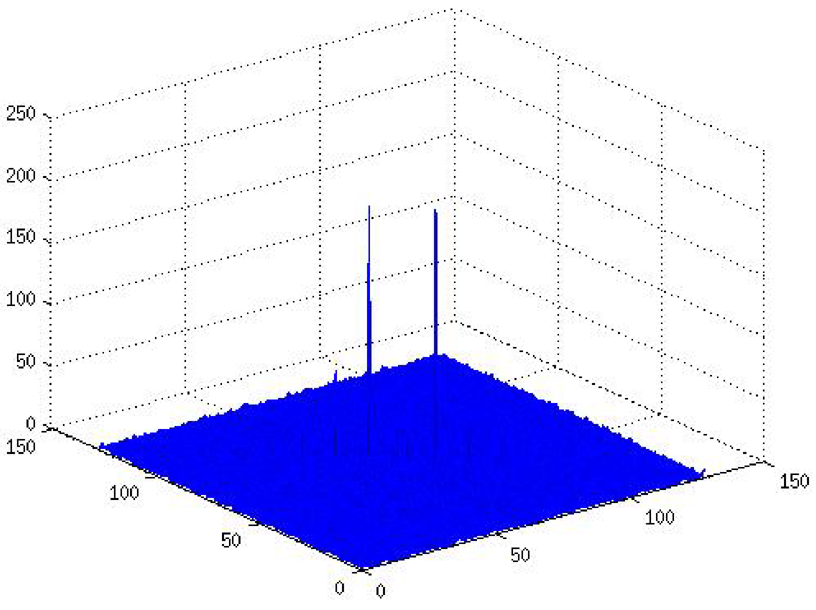
Blur Estimation

Levin et al.
Yitzhaky et al.

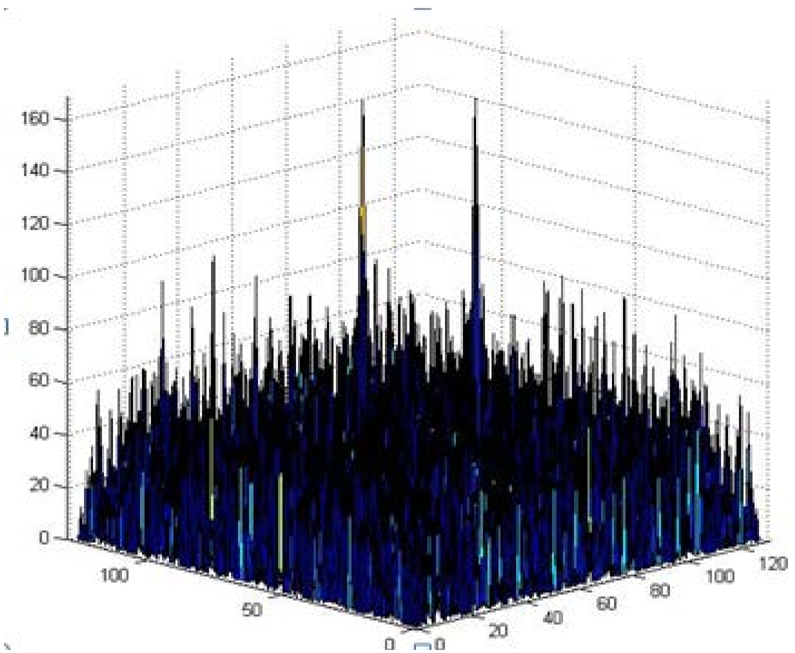


Hardware/ Software Comparison

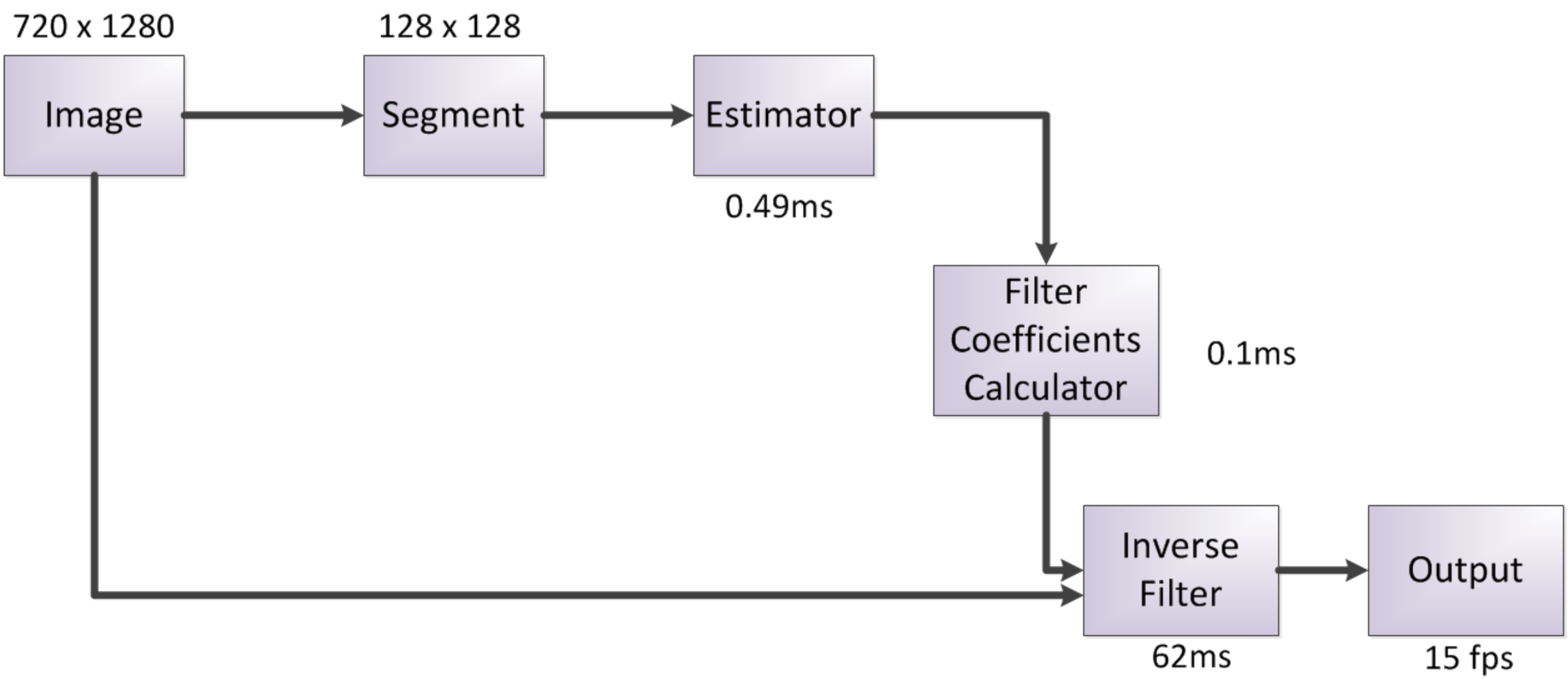
Software Implementation



Hardware Implementation



Timing Summary



Mean Absolute Error (MAE) compared to the Sofware based approach: 7.9

Time requirement for processing a 1280x720 frame: 62ms
Achievable frame rate: 15fps for 1280×720 (HD resolution)

Resource Utilization Summary

Module	DSP Slices	Slice Registers	LUTs
Estimation Module	108	16215	15923
Filter parameter calculation	61	16854	15258
Inverse Filtering	132	21795	26065
System implementation	10	6178	7793
Total	**311**	**61042**	**65039**

- *DSP - DSP48A1 slices contains an 18 x 18 multiplier, an adder, and an accumulator*
- *LUT-   contains 6-input LUT*







Applications

- Recovering Blurred images from security cameras
- Low altitude aerial photography
- Other scientific applications

Conclusion and future work

- The system presented above is suitable for an ASIC implementation to be integrated to a hand held camera.
- Extend the system for non-uniform blur

Questions ?

Bibliography

1. **A. Khireddine, K. Benmahammed , W. Puech.** *Digital image restoration by Wiener filter in 2D case*. 2006.

2. **C. T. Johnston, K. T. Gribbon, D. G. Bailey**. *Implementing Image Processing Algorithms on FPGAs*. 2010

3. **Downton, A. and Crookes, D.** *Parallel Architectures for Image Processing.* 1998.

4. **Rob Fergus, Barun Singh,Aaron Hertzmann, William T. Freeman.** *Removing Camera Shake from a Single Photograph. 2006.*

5. **Whyte, O. Sivic, J. Zisserman, A. Ponce, J. s.l.** *Non-uniform Deblurring for Shaken images.* : Computer Vision and Pattern Recognition (CVPR), 2010.

6. **Hui Ji, Chaoqiang Liu.** *Motion blur identification from image gradients*. 2008.

7. **Jo˜o P. A. Oliveira, M´rio A. T. Figueiredo, and Jos´ M. Bioucas-Dias.** *Blind Estimation of Motion Blur Parameters For Image Deconvolution*. 2007.

8. **A. Levin, Y. Weiss, F. Durand, and W. T. Freeman**, "*Understanding and evaluating blind deconvolution algorithms*," in CVPR, 2009.

9. **Y. Yitzhaky and N. S. Kopeika**, "*Identification of blur parameters from motion blurred images*," Graphical Models of Image Processing, 1996

Mean Absolute Error (MAE)

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n} |e_i|.$$

f_i is the prediction and y_i the true value.

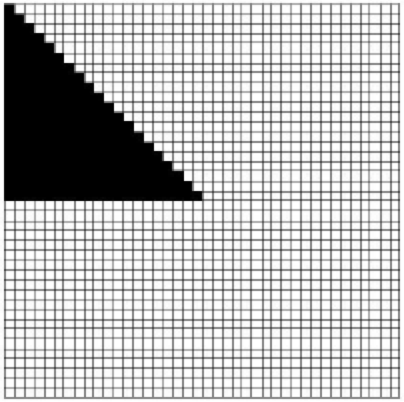
Spartan-6 FPGA Feature Summary

Device	Logic Cells[(1)]	Configurable Logic Blocks (CLBs)			DSP48A1 Slices[(3)]	Block RAM Blocks		CMTs[(5)]	Memory Controller Blocks (Max)[(6)]	Endpoint Blocks for PCI Express	Maximum GTP Transceivers	Total I/O Banks	Max User I/O
		Slices[(2)]	Flip-Flops	Max Distributed RAM (Kb)		18 Kb[(4)]	Max (Kb)						
XC6SLX4	3,840	600	4,800	75	8	12	216	2	0	0	0	4	132
XC6SLX9	9,152	1,430	11,440	90	16	32	576	2	2	0	0	4	200
XC6SLX16	14,579	2,278	18,224	136	32	32	576	2	2	0	0	4	232
XC6SLX25	24,051	3,758	30,064	229	38	52	936	2	2	0	0	4	266
XC6SLX45	43,661	6,822	54,576	401	58	116	2,088	4	2	0	0	4	358
XC6SLX75	74,637	11,662	93,296	692	132	172	3,096	6	4	0	0	6	408
XC6SLX100	101,261	15,822	126,576	976	180	268	4,824	6	4	0	0	6	480
XC6SLX150	147,443	23,038	184,304	1,355	180	268	4,824	6	4	0	0	6	576
XC6SLX25T	24,051	3,758	30,064	229	38	52	936	2	2	1	2	4	250
XC6SLX45T	43,661	6,822	54,576	401	58	116	2,088	4	2	1	4	4	296
XC6SLX75T	74,637	11,662	93,296	692	132	172	3,096	6	4	1	8	6	348
XC6SLX100T	101,261	15,822	126,576	976	180	268	4,824	6	4	1	8	6	498
XC6SLX150T	147,443	23,038	184,304	1,355	180	268	4,824	6	4	1	8	6	540

- 6295454 clock cycles, and with a 100MHz

$$z^{-1} = \frac{a}{\sqrt{a^2 + b^2}} - i\frac{b}{\sqrt{a^2 + b^2}};$$

- 16384 to 2080 data values



g. 4.16 Symmetry in Hanning window generation