Single **I**mage **R**eflection **R**emoval

Ishani Parekh
Gaurav Agrawal

Goal: Reduce reflection artifacts in images captured by mobile cameras

References: Methods published in [1] (***LB14***) and [2] (***SK15***)

Datasets:

➔ Images in SIR2 dataset [3]
➔ Images used in LB14 and SK15 ([1], [2])
➔ Mobile images captured by us
➔ Synthetic images constructed by us

IQ Metrics: Visual inspection, Structure Index, Normalized Cross Correlation

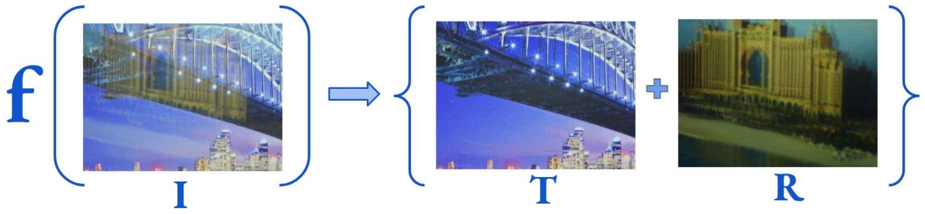
[1] Y. Li and M. S. Brown. Single image layer separation using relative smoothness. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[2] Y. C. Shih, D. Krishnan, F. Durand, and W. T. Freeman. Reflection removal using ghosting cues. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3193-3201

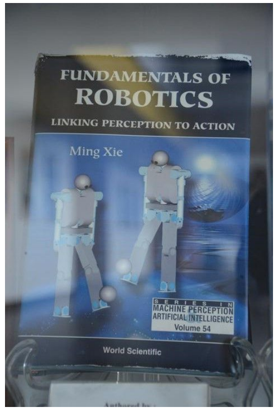
[3] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, Alex C. Kot. Benchmarking Single-Image Reflection Removal Algorithms. The IEEE International Conference on Computer Vision (ICCV), 2017, pp. 3922-3930

SIRR Problem Formulation

- Reflection removal is a layer separation problem
  - Image = T + R
  - Inherently ill posed with no single “right” answer
- Assume priors on statistics of T and R layers
  - Acts as regularization on the solution space
  - Image priors: gradient sparsity, GMM
  - T, R asymmetry: R smoothness, R ghosting
- Formulated as cost function minimization problem



LB14 (Li, Brown, CVPR 2014)



- "Relative Smoothness of R vs. T"
- Assume gradient sparsity for T component
    - Estimate gradients using first order derivative filters
- Assume R to be smoother than T
    - Estimate smoothness using Laplacian
- Maximize joint probability P(T, R)
    - i.e. minimize negative log (P_1(T) . P_2(R))
    - Solved iteratively using *half quadratic separation* method
    - Add range constraints to bound s.t. 0 <= T <= I

Gradient

Long tail of gradients

$$P_1(x) = \frac{1}{z} \max\left\{ e^{-\frac{x^2}{\sigma_1^2}}, \varepsilon \right\}$$

$$P_2(x) = \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2}{\sigma_2^2}}$$

$$\min_T \sum_i \left(\sum_j \rho(T * f_j) + ((I - T) * f_k)^2 . \lambda \right)$$

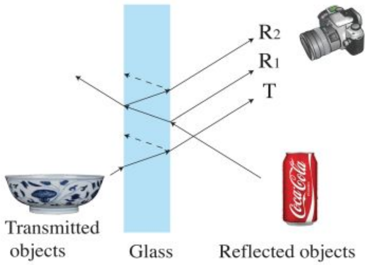
$$\rho(x) = \min\left\{ \frac{x^2}{k}, 1 \right\}$$

SK15 (Shih, Krishnan et al. CVPR 2015)

- “Ghosting Cues”
- Assume ghosting in R due to double reflections
- Assume 8x8 patch based prior based on GMM for T and R
  - 200 mixture components
  - Trained over 2M patches sampled from natural images
- Estimate ghosting kernel k
  - Use 2D autocorrelation of Laplacian
- Minimize sum-of-squared differences error
  - Augmented with joint probability P(T, R) under GMM prior
  - P estimated as sum of probabilities over all overlapping patches (EPLL)
- Solve iteratively using half-quadratic separation + L-BFGS

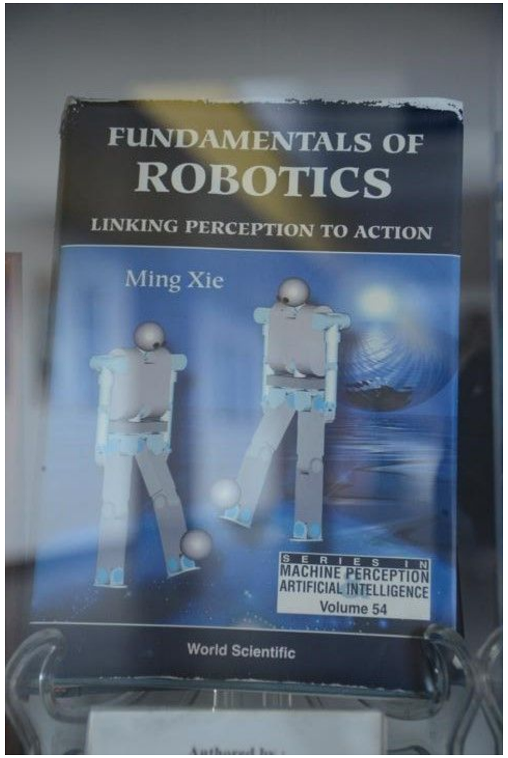
Ghosting kernel

$$I = T + R \otimes k$$

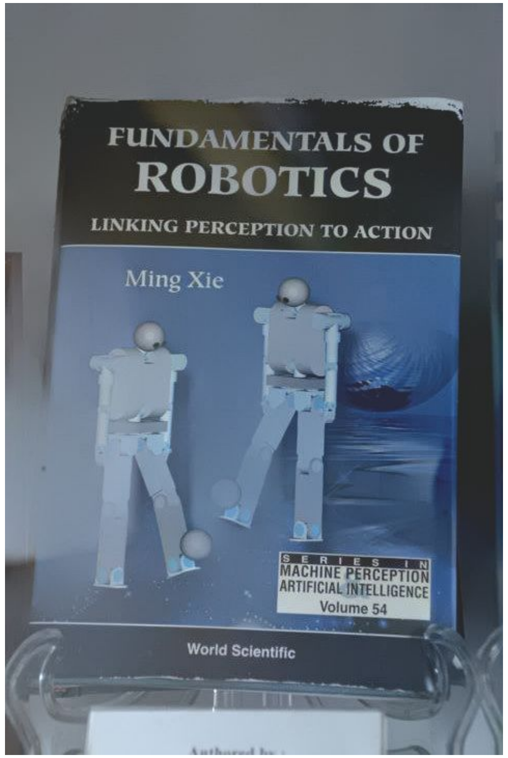


$$\min_{T,R} \left(\frac{1}{\sigma^2} \|I - T - R \otimes k\|^2 - \sum_i \log(GMM(P_i T)) - \sum_i \log(GMM(P_i R)) \right)$$

Results with LB14 (Relative Smoothness)



I =



T +

R

Ref: "book" image used by [1], 480 x 725 pix

Results with LB14 (Relative Smoothness) (Contd..)

I

=

T

+

R

Ref: "SolidObjects/8" image from SIR2. 540 x 400 pix

Results with LB14 (Relative Smoothness) (Contd..)

I

=

T

+

R

Ref: "SolidObjects/7" image from SIR2. 540 x 400 pix

Results with SK15 (Ghosting Cues)

I = **T** + **R**

Ref: "apples" image used by [1], 540 x 400 pix

Results with SK15 (Ghosting Cues) (Contd..)

I

T

R

Synthetic image constructed to demonstrate SK15

Challenges

- Optimization problems non-convex
  - Solved using iterative methods, variable time to convergence
  - Sensitive to initialization
- Optimization solved over entire image
  - Memory and computation quickly becomes prohibitive
- Must make assumptions about T and R
  - Unfortunately these assumptions aren’t robust
    - SB14: R may be sharp and not diffuse
    - LK15: T may contain repeating features, ghosting in R may be minimal

Bad result with LB14 (Relative Smoothness)

I

=

T

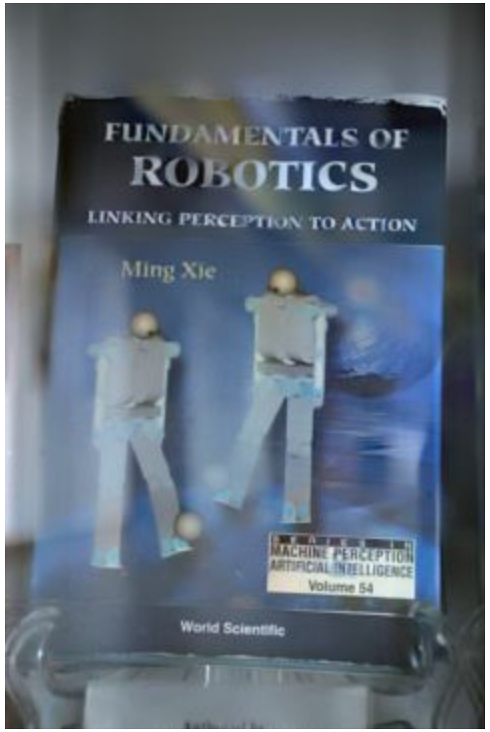
+

R

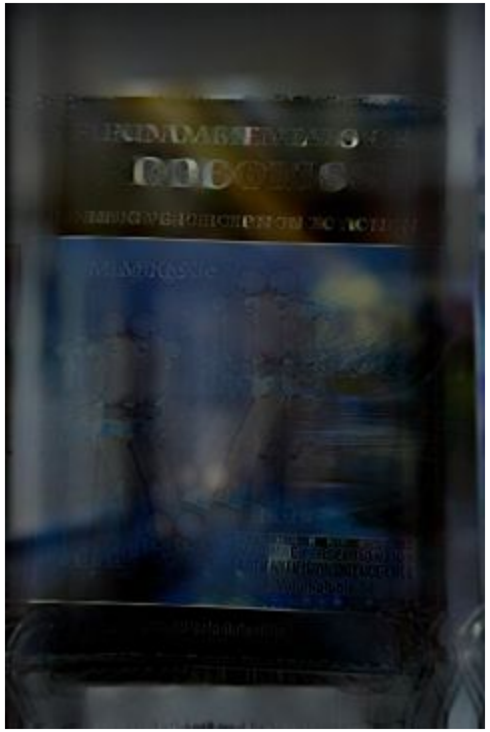
Ref: “apples” image used by [1], 540 x 400 pix

Bad result with SK15 (Ghosting Cues)

I



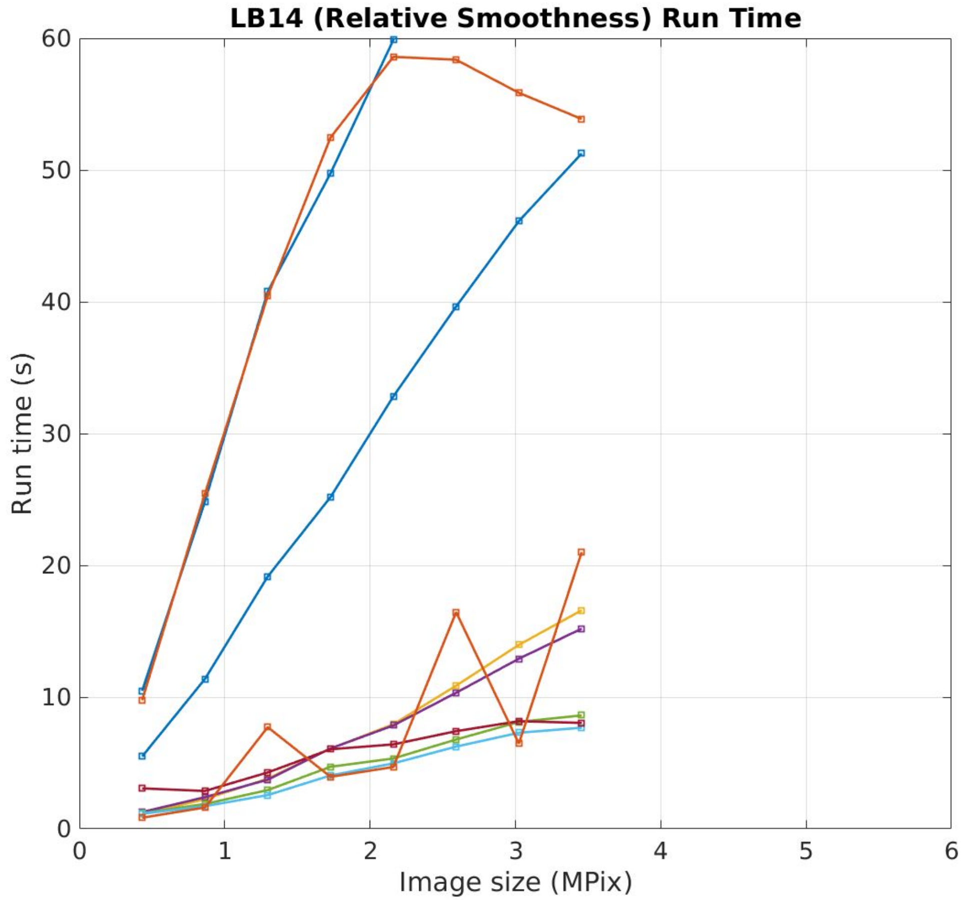
T



R

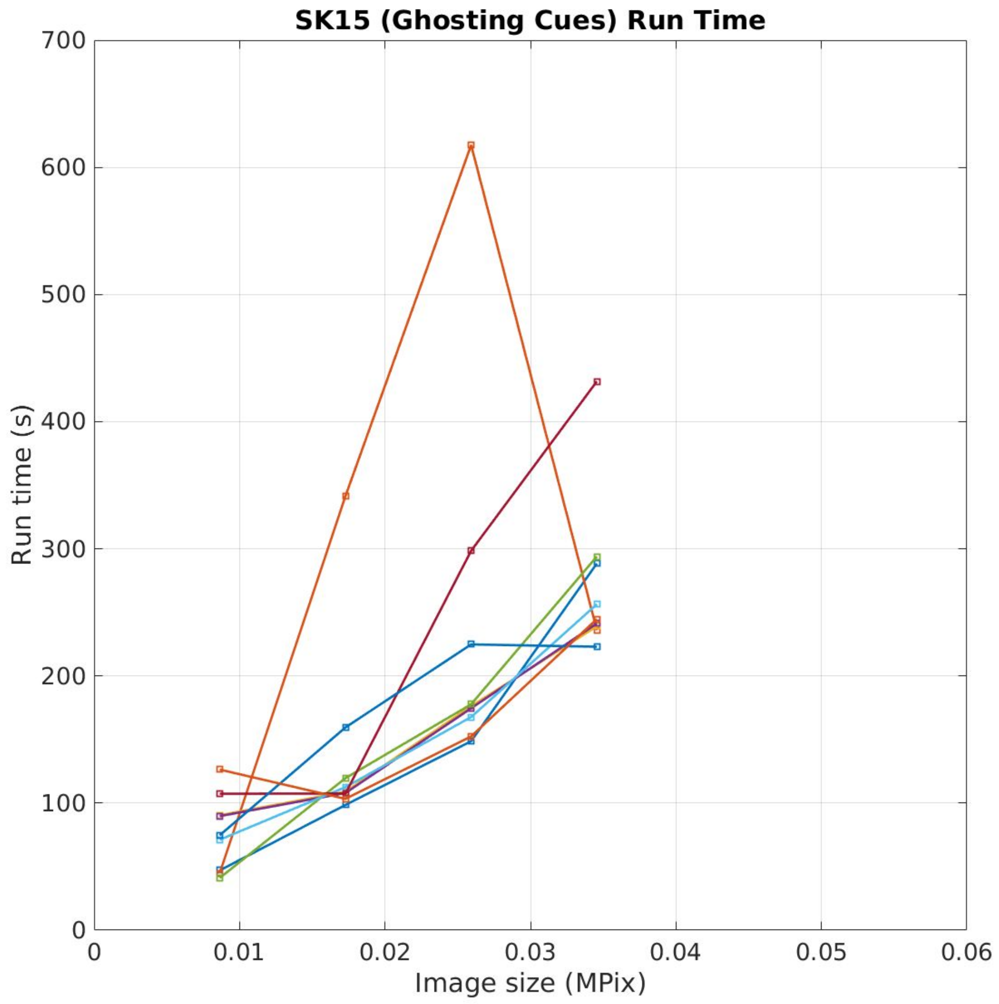
Ref: "book" image used by [1], 480 x 725 pix

Run time of LB14 (Relative Smoothness)



- 9 images from SIR2 dataset
- MATLAB runtime on Linux workstation
  - 12 core Intel Xeon E5-1650 @ 3.60GHz
- Large image-to-image variation for a fixed size
- Large images will take several minutes to run
  - Upto **9 mins** measured on 12 Mpix images

Run time of SK15 (Ghosting Cues)

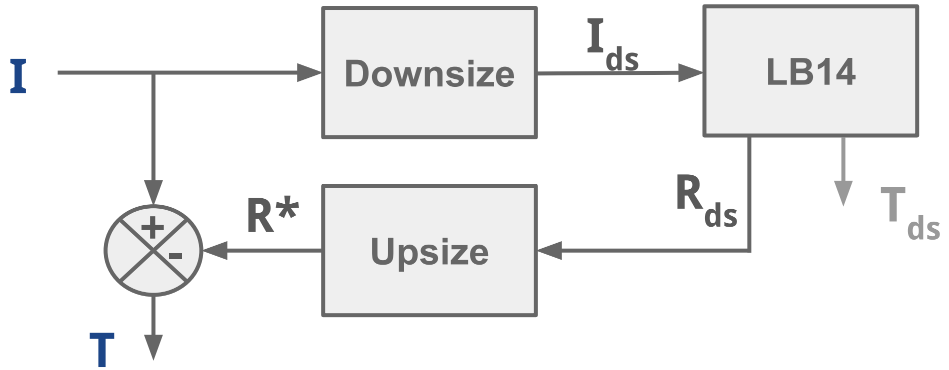


- 9 images from SIR2 dataset
- MATLAB runtime on Linux workstation
  - 12 core Intel Xeon E5-1650 @ 3.60GHz
- Large image-to-image variation for a fixed size
- Large images will take several days to run
  - **70 mins** measured on a **0.2 Mpix** (400 x 540) image

Note: Only 5 iterations of optimization were run (vs. 25 iterations in SK15)

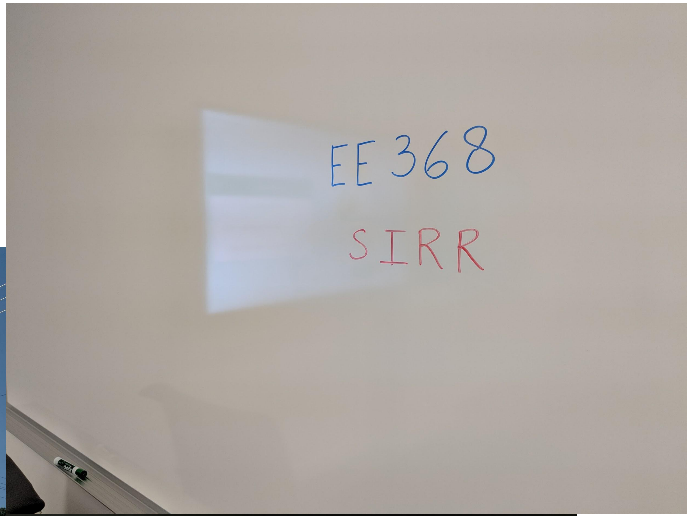
SIRR on Multi-Mpixel images

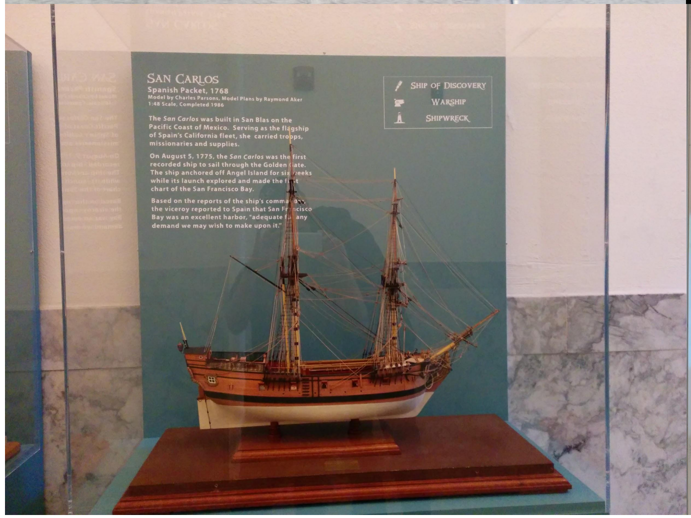
- SK15 is too compute and memory intensive
    - Even 500 x 500 pixel image will take multiple hours
- LB14 can be run on larger images
    - But 9 minutes on 12 Mpixel image is still too large
- Ideas to speed up LB14
    - Run LB14 on downsized image; upsample R and subtract from full-res image to get T
    - Reduce the number of iterations; relax convergence criterion

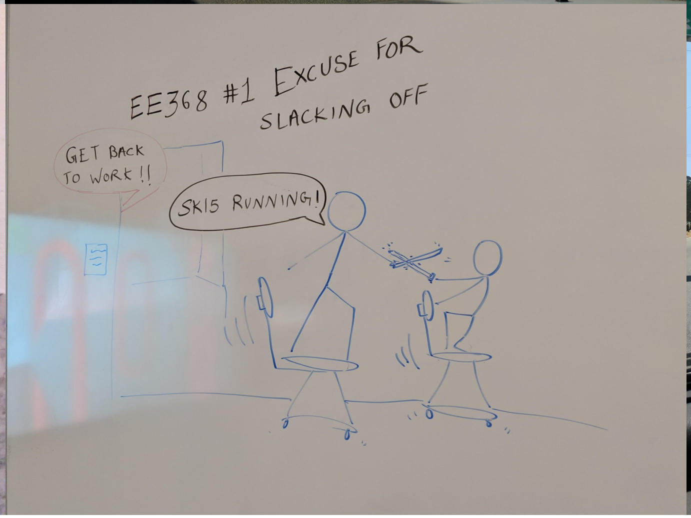


Images For Evaluation

5 images taken with our mobile phones, 12 MPixel each







Runtime Improvement

Runtimes on 6 test images of 12 MPixel each

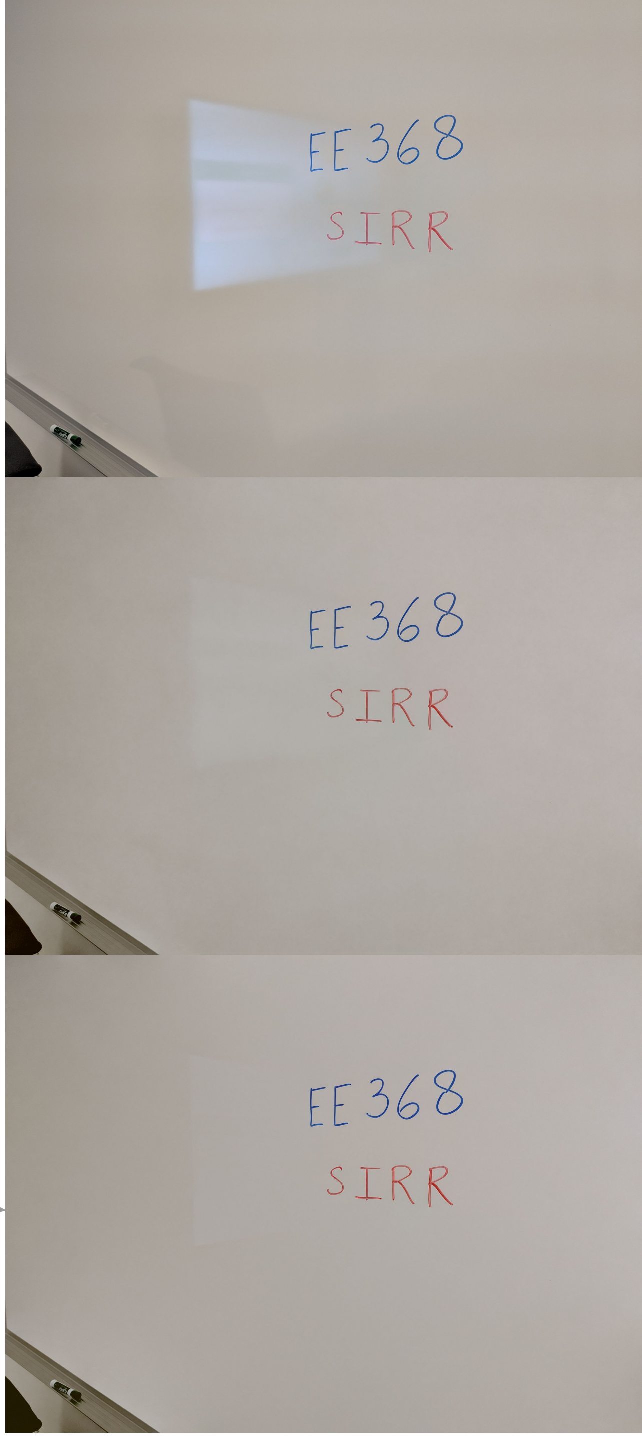
	LB14 Runtime (sec)	With parameter Tuned LB14 (sec)	With Downscaled LB14 (sec)	With Both optimizations (sec)
Boat	61	9	16	3
Cardash	150	75	33	16
Fort	45	12	11	4
Trees	214	71	47	5
XKCD	489	44	97	11
Whiteboard	493	48	97	12

MATLAB runtime on Linux workstation (12 core Intel Xeon E5-1650 @ 3.60GHz)

I

T

T*



Next steps

- Measure image quality degradation
- Try other ideas to improve run time without quality degradation