REVISITING CLASSICAL PROBLEMS OF IMAGE PROCESSING:

Looking for new ways to address longstanding problems

Mauricio Delbracio

Duke University

SIAM Conference on Imaging Science
24 May 2016

Thanks to…

I'd like to thank many people: UdelaR, ENS-Cachan, Duke University, Colleagues, Family, Friends,

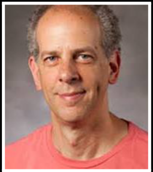
Alicia Fernández
UdelaR

Gregory Randall
UdelaR

Pablo Musé
UdelaR

Andrés Almansa
Télécom ParisTech

Jean-Michel Morel
ENS-Cachan



Guillermo Sapiro
Duke University

Ives Rey-Otero
NC State University

Cecilia Aguerrebere
Duke University

REVISITING CLASSICAL PROBLEMS OF IMAGE PROCESSING:

Looking for new ways to address longstanding problems

Mauricio Delbracio

Duke University

SIAM Conference on Imaging Science
24 May 2016

Outline

Looking for simpler ways to address longstanding problems:

1. Recovering the Camera Point Spread Function
2. Removing Camera Shake via Fourier Burst Accumulation
3. Boosting Stochastic Renderers by Auto-similarity Filtering

Act one

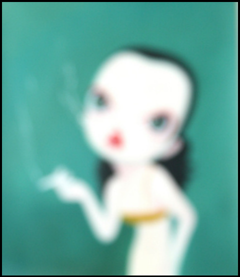
Recovering the Camera Point Spread Function

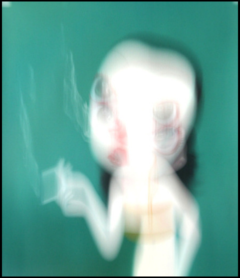
Joint work with: A. Almansa[1], P. Musé[2] and J.-M. Morel[3]

[1]Telecom Paristech, [2]UdelaR, [3]ENS-Cachan

Image blur can be a consequence of

- Camera misusing or scene configuration (Extrinsic):
  - Wrongly setting the camera focus
  - Only an specific interval of depths in focus
  - Camera shake, scene motion

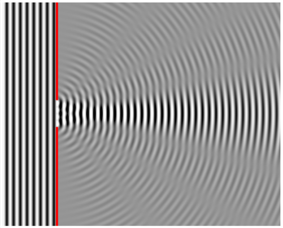




Blur Sources

Image blur can be a consequence of

- Physical camera phenomena (Intrinsic):
  - Light diffraction
  - Sensor averaging
  - Lens aberration
  - Optical anti-aliasing filter



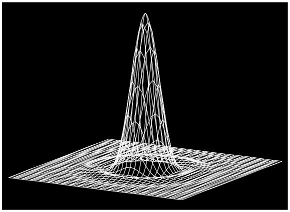
Our Goal

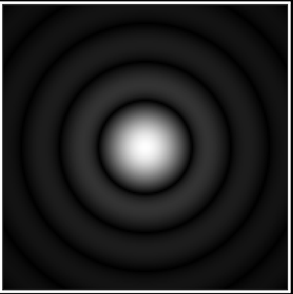
Accurately estimate a function, called Point Spread Function (PSF), that models the blur due to intrinsic camera phenomena.

Our Goal

Accurately estimate a function, called Point Spread Function (PSF), that models the blur due to intrinsic camera phenomena.

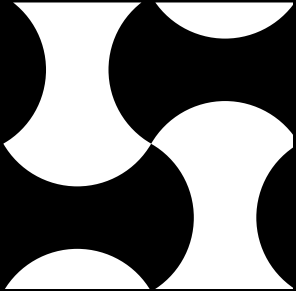
Image ideally obtained from a null-area point light source (impulse response).





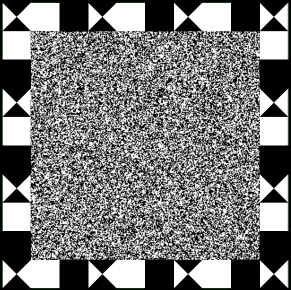
Non-blind estimation: use a calibration pattern

- How do we choose the calibration pattern?
- Local, accurate, subpixel PSF estimation - is it possible?



"knife-edge"

Joshi, et al. '08

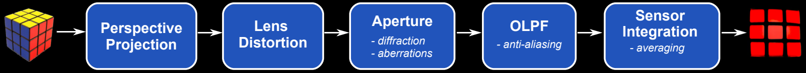


"Bernoulli pattern"

Delbracio, et al. '12

Problem statement

Image Formation Model

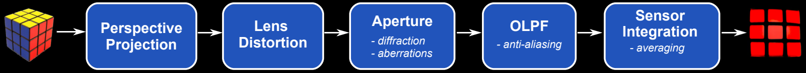


$$\mathbf{v} = \mathbf{S}_1 \left(D\left(u\right) * h \right) + \mathbf{n}$$

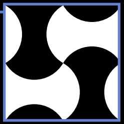
- $\mathbf{v}$ is the acquired image.
- u is the continuous image.
- h local convolution kernel due to all PSF like effects.
- $D(\cdot)$ geometrical transformation.
- $\mathbf{S}_1$ bi-dimensional ideal ideal sampling operator (sensor array).
- $\mathbf{n}$ models all noise sources of the acquisition process.

Problem statement

Image Formation Model



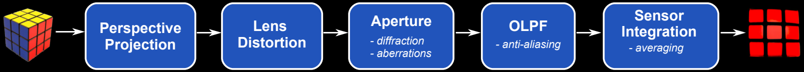
$$\mathbf{v} = \mathbf{S}_1 \left(D \left(u \right) * h \right) + \mathbf{n}$$



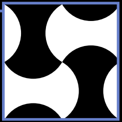
- $\mathbf{v}$ is the acquired image.
- u is the continuous image.
- h local convolution kernel due to all PSF like effects.
- $D(\cdot)$ geometrical transformation.
- $\mathbf{S}_1$ bi-dimensional ideal ideal sampling operator (sensor array).
- $\mathbf{n}$ models all noise sources of the acquisition process.

Problem statement

Image Formation Model



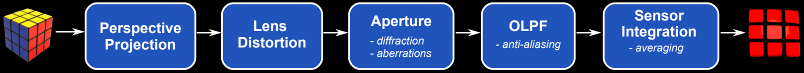
$$\mathbf{v} = \mathbf{S}_1 \left(D(u) * h \right) + \mathbf{n}$$



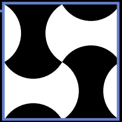
- $\mathbf{v}$ is the acquired image.
- u is the continuous image.
- h local convolution kernel due to all PSF like effects.
- $D(\cdot)$ geometrical transformation.
- $\mathbf{S}_1$ bi-dimensional ideal ideal sampling operator (sensor array).
- $\mathbf{n}$ models all noise sources of the acquisition process.

Problem statement

Image Formation Model



$$\mathbf{v} = \mathbf{S}_1 \left(D \left(u \right) * h \right) + \mathbf{n}$$



- **v** is the acquired image.
- *u* is the continuous image.
- *h* local convolution kernel due to all PSF like effects.
- $D(\cdot)$ geometrical transformation.
- $\mathbf{S}_1$ bi-dimensional ideal ideal sampling operator (sensor array).
- **n** models all noise sources of the acquisition process.

  - Model is local: h may change all over the image.

Problem statement (cont.)

Discrete Image Formation Model

- h is band-limited in $\mathrm{supp}(\hat{\mathrm{h}}) = [-\mathrm{s}\pi, \mathrm{s}\pi]^2$, e.g., $\mathrm{s} = 3 - 4$
- Take samples at rate at least $\mathrm{s}\times$ to correctly sample the PSF

Problem statement (cont.)

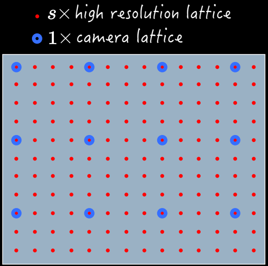
Discrete Image Formation Model

- h is band-limited in $\mathrm{supp}(\hat{h}) = [-s\pi, s\pi]^2$, e.g., $s = 3 - 4$
- Take samples at rate at least $s\times$ to correctly sample the PSF

Continuous model can be replaced by a $s\times$ oversampled discrete model,

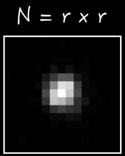
$$v = \mathcal{S}_s (u * h) + n$$

- u : samples at $s\times$ of the low-pass filtered distorted pattern image.
- $\mathcal{S}_s$: s−subsampling operator $(\mathcal{S}_s u)(x) = u(sx)$



Mathematical Formulation

Solve an inverse problem based on prior information about the small spatial support of the PSF.



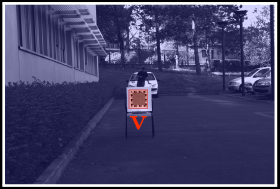
Mathematical Formulation

$N = r \times r$

Solve an inverse problem based on prior information about the small spatial support of the PSF.

$$\tilde{h} = \underset{h \in \mathbb{R}^N}{\arg\min} \left\| S_s C[u] h - v \right\|^2$$

- v: observation
- u: $s\times$ resolution rasterized distorted pattern
- $C[u]$: convolution with u in matrix form
- S_s: $s-$sub-sampling operator in matrix form



Mathematical Formulation (cont.)

Solution to the inverse problem

$$\tilde{h} = (S_s C[u])^{\dagger} v.$$

Mathematical Formulation (cont.)

Solution to the inverse problem

$$\tilde{\mathrm{h}} = (\mathrm{S_s C[u]})^\dagger \mathrm{v}.$$

The mean square error is given by

$$\mathrm{MSE}(\tilde{\mathrm{h}}) = \underbrace{\left\|(\mathrm{S_s C[u]})^\dagger\right\|_\mathrm{F}^2}_{\gamma} \underbrace{\sigma^2}_{\text{noise level}}$$

Mathematical Formulation (cont.)

Solution to the inverse problem

$$\tilde{h} = (S_s C[u])^{\dagger} v.$$

The mean square error is given by

$$\mathrm{MSE}(\tilde{h}) = \underbrace{\|(S_s C[u])^{\dagger}\|_F^2}_{\gamma} \underbrace{\sigma^2}_{\text{noise level}}$$

To minimize the error, one has to minimize

$$\gamma(S_s C[u]) := \|(S_s C[u])^{\dagger}\|_F^2 = \sum_{i=1}^{N} \sigma_i^{-2},$$

where $\{\sigma_1, \sigma_2, \ldots, \sigma_N\}$ are the singular values of $S_s C[u]$.

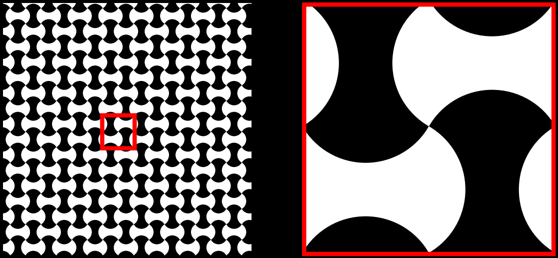
- γ controls the noise amplification,
- should be as low as possible.

Proposition (Lower bound for optimal patterns)

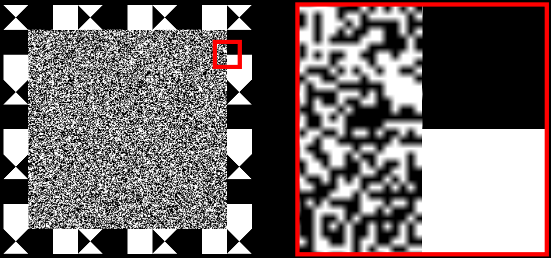
$$\min_{a \leq u_{ij} \leq b} \gamma(S_s C[u]) \geq \frac{1}{MN} \left(\frac{1}{b^2} + \frac{4(N-1)^2}{(b-a)^2} \right),$$

- $M = m \times n$ is the observation window size
- $N = r \times r$ is the kernel size
- Constraints $a \leq u_{ij} \leq b$ are linked to the physical realization and dynamic range of the sensors.

Comparing calibration patterns



Slant-edge pattern - Joshi et al. [2008]

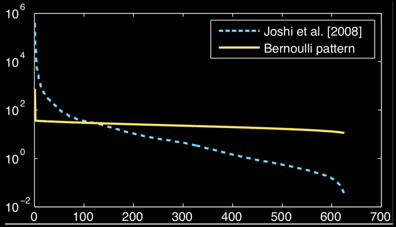


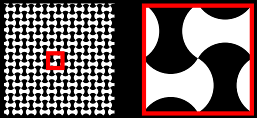
Bernoulli pattern - Delbracio et al. [2012]

Optimality of the Bernoulli pattern

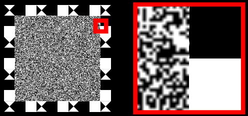
Why this i.i.d Bernoulli(0.5) random noise pattern?

Singular values of $S_s C[u]$





Slant-edge pattern - Joshi et al. [2008]

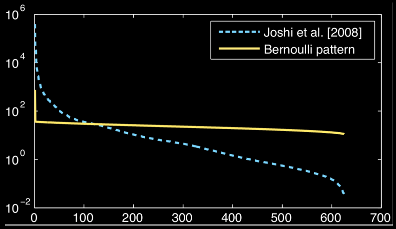


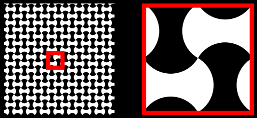
Bernoulli pattern - Delbracio et al. [2012]

Optimality of the Bernoulli pattern

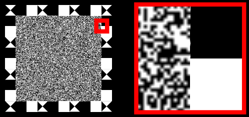
Why this i.i.d Bernoulli(0.5) random noise pattern?

Singular values of $S_s C[u]$





Slant-edge pattern - Joshi et al. [2008]



Bernoulli pattern - Delbracio et al. [2012]

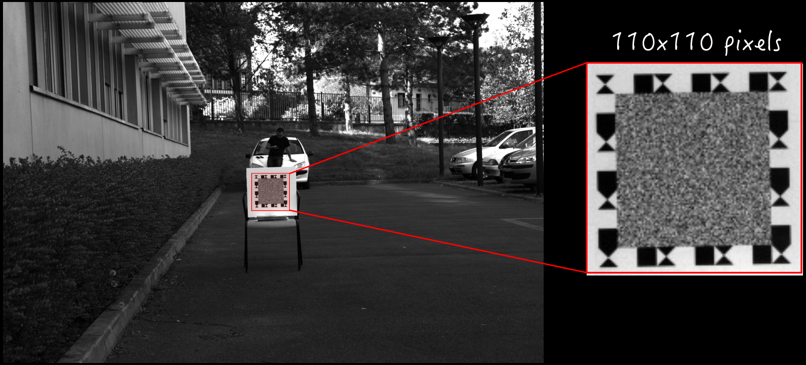
γ value:

	9×9	17×17	25×25	33×33
Theoretical bound	0.10	0.35	0.70	1.15
Bernoulli pattern	0.19	0.69	1.54	2.98
Joshi et al. [2008]	99.44	1133.05	6445.87	58419.08

Experiments

Real camera examples

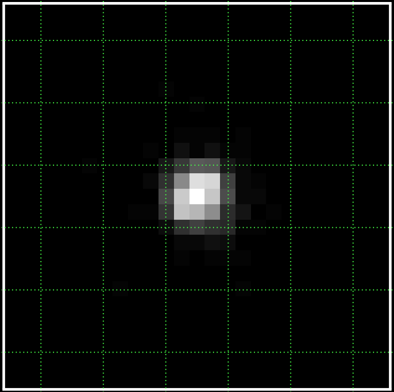
Canon EOS 400D - Tamron AF 17-50mm F/2.8 XR Di-II lens, 50mm, Green channel 1.



IPOL: Image Processing Online – ipol.im

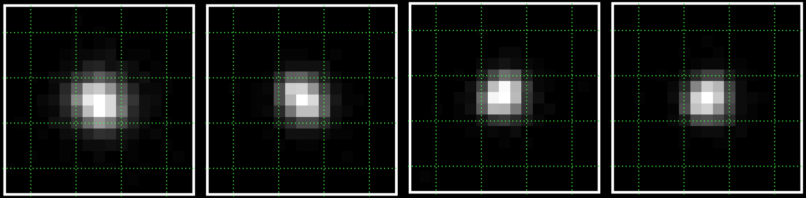
Detailed Description + Online Demo + Source Code

Estimation at $4\times$ the sensor resolution for the Green channel 1.



Canon EOS 400D - Tamron AF 17-50mm F/2.8 XR Di-II lens, f/5.6, Green channel 1.

$4\times$ PSF estimation for the four Bayer pattern channels (RAW output).



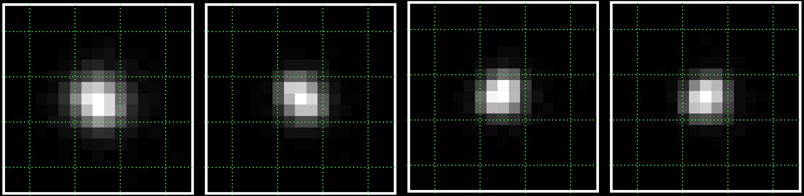
red blue green1 green2

- Red PSF larger than green and blue ones (diffraction)

Experiments

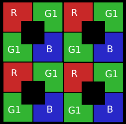
Different color channels

$4\times$ PSF estimation for the four Bayer pattern channels (RAW output).



red blue green1 green2

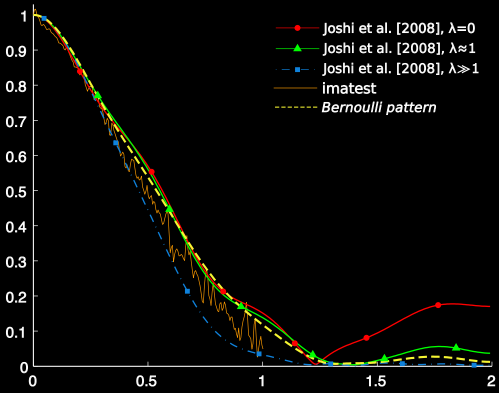
- Red PSF larger than green and blue ones (diffraction)
- Blue and green are symmetric (sensor active area L-shaped)



Comparison of several methods

- Joshi et al. [2008] very sensitive to regularization $+\lambda\|\nabla h\|^2$

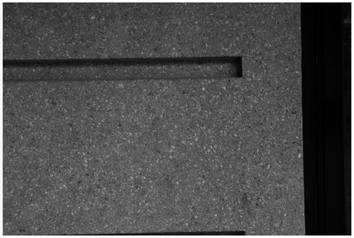
MTF horizontal profile



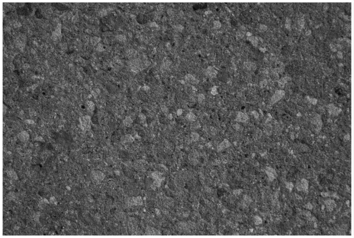
Can we avoid the calibration pattern?

Can we avoid the calibration pattern?

Yes! Take two parallel photos (same scene) different distances



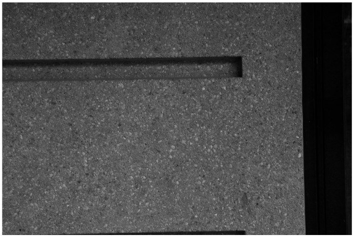
Farthest photograph v_2



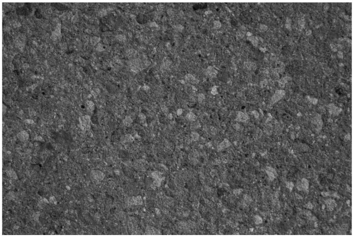
Closest photograph v_1

Can we avoid the calibration pattern?

Yes! Take two parallel photos (same scene) different distances



Farthest photograph v_2



Closest photograph v_1

- If acquired sufficiently far from each other: the PSF can be estimated from the relative blur between the two images

Relative blur between two images

- $\mathrm{v}_1, \mathrm{v}_2$ two fronto-parallel views (same scene), zooms $\lambda_1 < \lambda_2$.

Definition (inter-image kernel)

Any k satisfying

$$\mathrm{v}_2 = \mathrm{H}_\lambda \mathrm{v}_1 * \mathrm{k}, \quad \lambda := {}^{\lambda_2}/_{\lambda_1}.$$

Relative blur between two images

- v_1, v_2 two fronto-parallel views (same scene), zooms $\lambda_1 < \lambda_2$.

Definition (inter-image kernel)

Any k satisfying

$$v_2 = H_\lambda v_1 * k, \quad \lambda := \lambda_2/\lambda_1.$$

Proposition

Under (mild assumptions), there is a unique inter-image kernel k,

$$H_\lambda h * k = h,$$

Relative blur between two images

- v_1, v_2 two fronto-parallel views (same scene), zooms $\lambda_1 < \lambda_2$.

Definition (inter-image kernel)

Any k satisfying

$$v_2 = H_\lambda v_1 * k, \quad \lambda := {}^{\lambda_2}/_{\lambda_1}.$$

Proposition

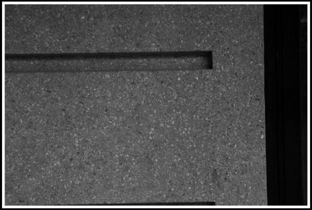
Under (mild assumptions), there is a unique inter-image kernel k,

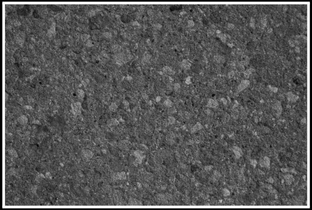
$$H_\lambda h * k = h,$$

and h can be obtained from k as:

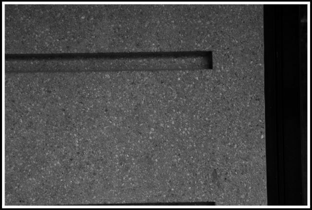
$$h = \lim_{n\to\infty} H_{\lambda^{n-1}} k * \ldots * H_\lambda k * k.$$

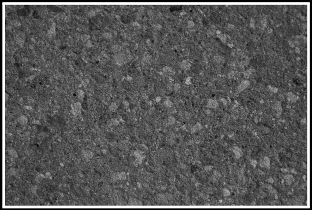
Results: A running example

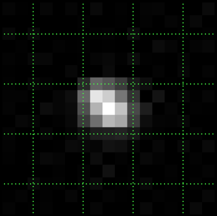

farthest image


closest image

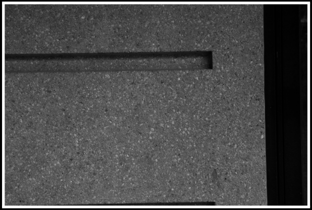
Results: A running example


farthest image

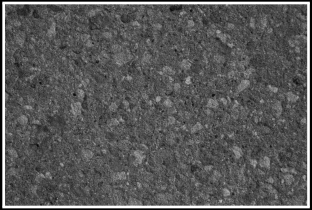

closest image


$4\times$ Inter-image kernel

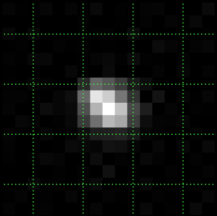
Results: A running example



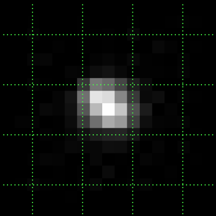
farthest image



closest image

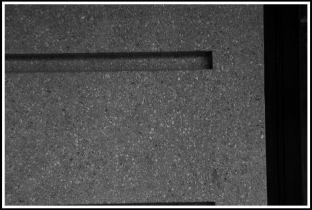


$4\times$ Inter-image kernel

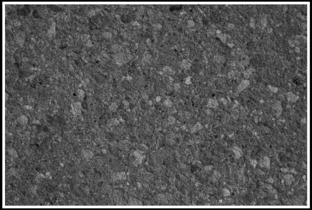


$4\times$ PSF

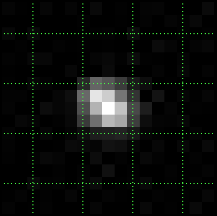
Results: A running example



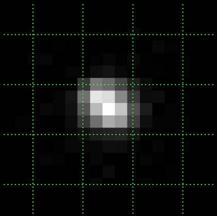
farthest image



closest image



$4\times$ Inter-image kernel

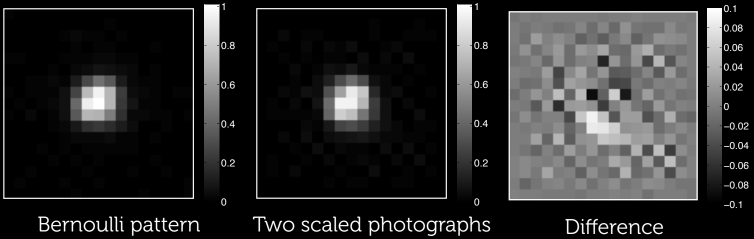


$4\times$ PSF

IPOL: Image Processing Online – ipol.im

Detailed Description + Online Demo + Source Code

Bernoulli pattern vs. Two-scaled photographs



- Estimations at $4\times$ the camera resolution

Recap

- Avoid regularization: Chose the right calibration pattern
- Subpixel, accurate PSF estimation is well-posed if calibration pattern carefully chosen
- Bernoulli random pattern near optimal
- Calibration pattern can be “avoided” by taking two images of the same scene

Act two

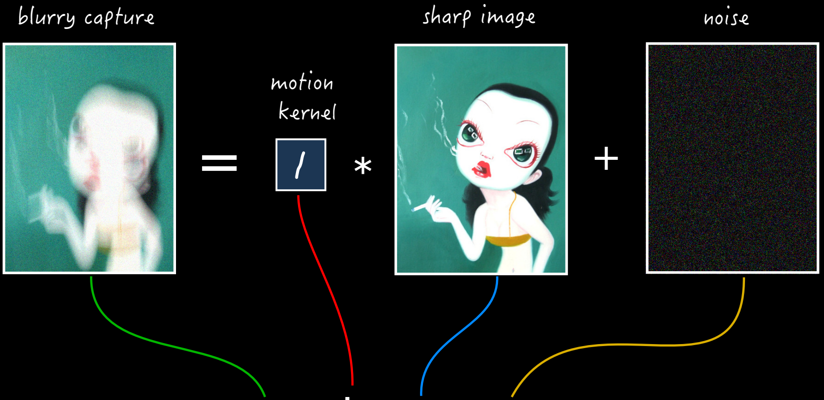
Removing Camera Shake Blur via Fourier Burst Accumulation

Joint work with: G. Sapiro
Duke University

Deblurring

Deblurring

Shift-invariant blurring model



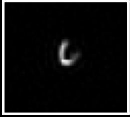
$$v = k * u + n$$

Standard Deblurring: Blind Deconvolution

1. Get a blurry image v

Standard Deblurring: Blind Deconvolution

2. Estimate the motion kernel k



Standard Deblurring: Blind Deconvolution

3. Perform nonblind deconvolution to recover $\tilde{u}$

Standard Deblurring: Blind Deconvolution

1. Get a blurry image v
2. Estimate the motion kernel k
3. Perform nonblind deconvolution to recover ũ

But...

- Who actually cares about the motion kernel?
- Even if the kernel is perfectly known the inversion is ill-posed

Standard Deblurring: Blind Deconvolution

1. Get a blurry image v
2. Estimate the motion kernel k
3. Perform nonblind deconvolution to recover ũ

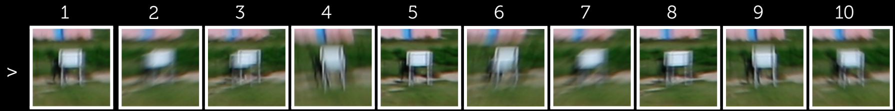
But...

- Who actually cares about the motion kernel?
- Even if the kernel is perfectly known the inversion is ill-posed

Is it possible to avoid explicit inversion?

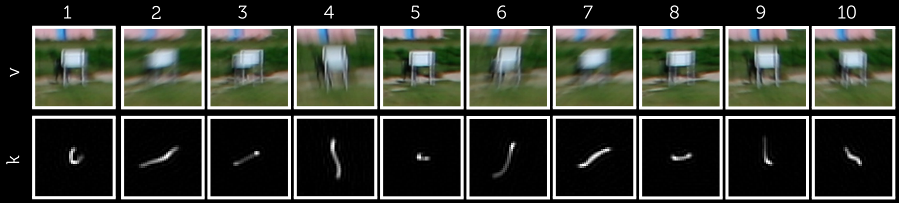
Burst photography

- In 2016… we can take a burst of 6-12 images



Burst photography

- In 2016… we can take a burst of 6-12 images



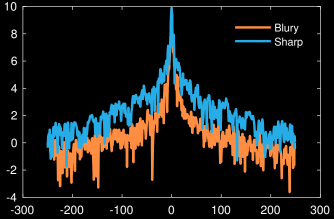
- Hand shake/tremor is random:
- Different images → Different blur (in general)

Claim (Blurring kernels do not amplify the spectrum)

sharp

blury

Fourier spectrum (vertical)



A basic principle

Claim (Blurring kernels do not amplify the spectrum)

Let $k(x) \geq 0$ and $\int k(x) = 1$. Then, $|\hat{k}(\zeta)| \leq 1, \forall \zeta$.

Proof.

$$\left|\hat{k}(\zeta)\right| = \left|\int k(x)e^{ix\cdot\zeta}dx\right| \leq \int |k(x)|\,dx = \int k(x)dx = 1.$$

□

Deblurring by Fourier Burst Accumulation

The basic Fourier Burst Accumulation algorithm:

1. Take a burst

Deblurring by Fourier Burst Accumulation

The basic Fourier Burst Accumulation algorithm:

1. Take a burst
2. Align the images (with respect to the center one)

Deblurring by Fourier Burst Accumulation

The basic Fourier Burst Accumulation algorithm:

1. Take a burst
2. Align the images (with respect to the center one)
3. Combine the images in the Fourier domain (FBA)

Fourier Burst Accumulation (FBA)

$$\bar{u}(x) = \mathcal{F}^{-1}\left(\sum_{i=1}^{M} w_i(\zeta)\cdot\hat{v}_i(\zeta)\right)(x), \quad w_i(\zeta) = \frac{|\hat{v}_i(\zeta)|^p}{\sum_{j=1}^{M}|\hat{v}_j(\zeta)|^p},$$

Deblurring by Fourier Burst Accumulation

The basic Fourier Burst Accumulation algorithm:

1. Take a burst
2. Align the images (with respect to the center one)
3. Combine the images in the Fourier domain (FBA)
4. Inverse Fourier

Fourier Burst Accumulation (FBA)

$$\bar{u}(x) = \mathcal{F}^{-1}\left(\sum_{i=1}^{M} w_i(\zeta) \cdot \hat{v}_i(\zeta)\right)(x), \quad w_i(\zeta) = \frac{|\hat{v}_i(\zeta)|^p}{\sum_{j=1}^{M} |\hat{v}_j(\zeta)|^p},$$

Deblurring by Fourier Burst Accumulation

The basic Fourier Burst Accumulation algorithm:

1. Take a burst
2. Align the images (with respect to the center one)
3. Combine the images in the Fourier domain (FBA)
4. Inverse Fourier
5. Unsharp masking (Optional)

Fourier Burst Accumulation (FBA)

$$\bar{u}(x) = \mathcal{F}^{-1}\left(\sum_{i=1}^{M} w_i(\zeta)\cdot \hat{v}_i(\zeta)\right)(x), \quad w_i(\zeta) = \frac{|\hat{v}_i(\zeta)|^p}{\sum_{j=1}^{M} |\hat{v}_j(\zeta)|^p},$$

Deblurring by Fourier Burst Accumulation (cont.)

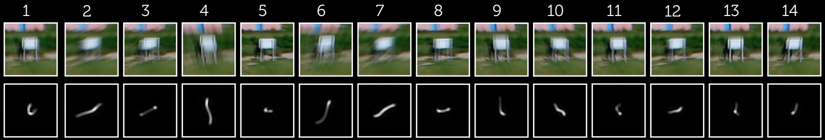
Fourier Burst Accumulation (FBA)

$$\bar{u}(x) = \mathcal{F}^{-1}\left(\sum_{i=1}^{M} w_i(\zeta) \cdot \hat{v}_i(\zeta)\right)(x), \quad w_i(\zeta) = \frac{|\hat{v}_i(\zeta)|^p}{\sum_{j=1}^{M} |\hat{v}_j(\zeta)|^p},$$

- (v_i): M input aligned images, $\hat{v}_i = \mathcal{F}(v_i)$ – Fourier Transform
- The larger $|\hat{v}_i(\zeta)|$, the more $\hat{v}_i(\zeta)$ contributes to $\bar{u}$
- p controls the aggregation procedure (soft-max):
  - If $p = 0$: arithmetic average
  - If $p = \infty$: max
  - $0 < p < \infty$: Balance Max/Mean

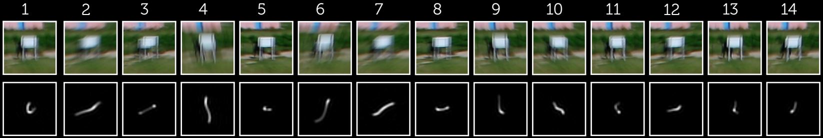
Fourier Burst Accumulation: An example

Input Burst



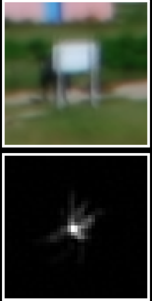
Fourier Burst Accumulation: An example

Input Burst



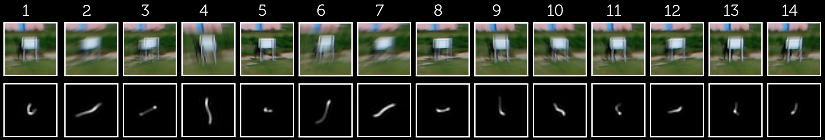
Fourier Burst Accumulation

$p = 0$

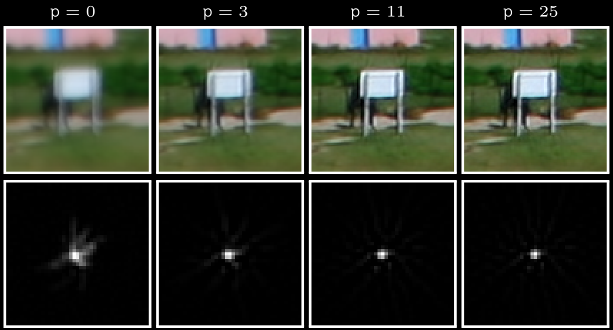


Fourier Burst Accumulation: An example

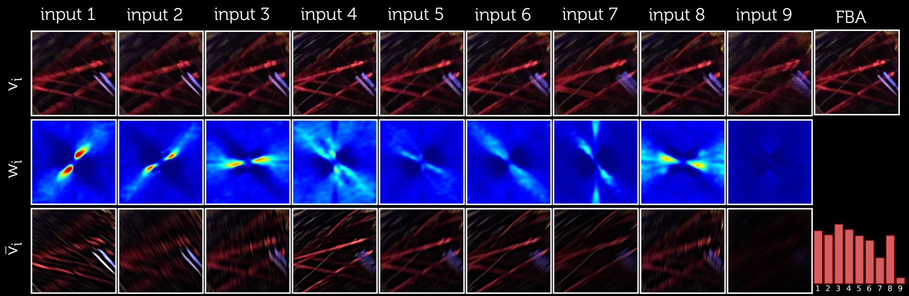
Input Burst



Fourier Burst Accumulation



Anatomy of the Fourier Aggregation

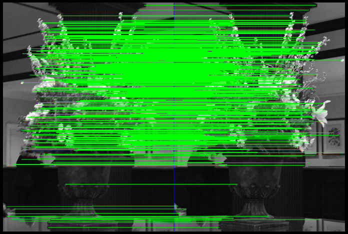


$$\bar{u}(x) = \sum_{i=1}^{M} \underbrace{\mathcal{F}^{-1}\Big(w_i(\zeta)\cdot \hat{v}_i(\zeta)\Big)(x)}_{\bar{v}_i(x)}, \quad w_i(\zeta) = \frac{|\hat{v}_i(\zeta)|^p}{\sum_{j=1}^{M} |\hat{v}_j(\zeta)|^p}, \quad p = 11.$$

Align images

- Can use gyroscope and accelerometer (Work in progress...)

- Current efficient approach
  - SIFT + Ransac (Homography)
  - Optical Flow (in low resolution)



More Results

Woods 2/12

Woods 7/12

Woods 8/12

Woods 9/12

Woods Align & Average (p = 0)

Woods FBA $p = 11$

Cabo Polonio 1/14

Cabo Polonio 2/14

Cabo Polonio 3/14

Cabo Polonio 4/14

Cabo Polonio 5/14

Cabo Polonio 6/14

Cabo Polonio 7/14

Cabo Polonio 8/14

Cabo Polonio 9/14

Cabo Polonio 10/14

Cabo Polonio 11/14

Cabo Polonio 12/14

Cabo Polonio 13/14

`Cabo Polonio` Align & Average ($p = 0$)

Cabo Polonio FBA $p = 11$

Cabo Polonio 5/14 (Best Frame)

More Results

Typical Shot | Best Shot | Align and average | Šroubek & Milanfar [2012] | Zhang et al. [2013] | FBA

Extension to videos

Recap

- Fourier weighted average to remove camera shake blur
- No (explicit) inversion, no kernel estimation, no deconvolution
- Not universal (blurs in the burst need to be different)

Act three

Boosting Stochastic Renderers by Auto-similarity Filtering

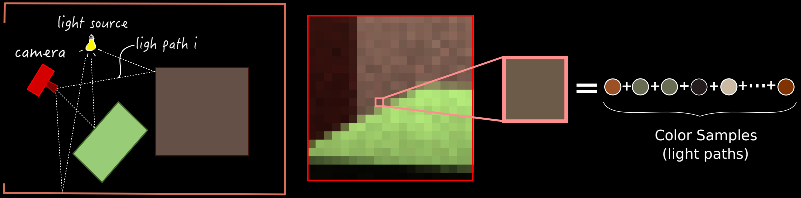
Joint work with: P. Musé[1], T. Buades[2], J. Chauvier[3], N. Phelps[3], J.-M. Morel[4]

[1]UdelaR, [2]UiB, [3]Eon-Software, [4]ENS-Cachan

Realistic Image Synthesis

Goal: Generate images from a 3D virtual scene

- Ray-tracing: popular technique for resolving the equilibrium of light in a scene (rendering equation [Kajiya 1986]).
- Pixel color = average of values along light paths
  - cast from image pixel, through camera aperture, bouncing in the scene and reaching a light source.

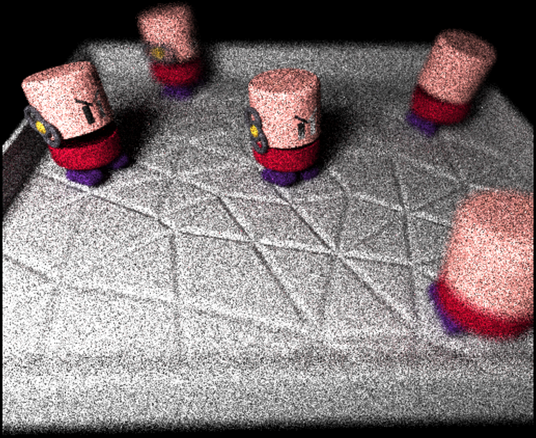


Unfortunately...

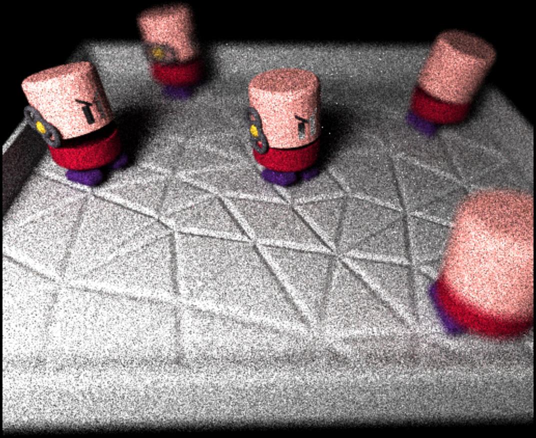
- Only a finite number of rays can be cast
- To avoid artifacts, rays are cast randomly
- Equivalent to solving the light equilibrium through a Monte Carlo integration procedure
- Variance converges linearly with number of samples

Realistic Image Synthesis

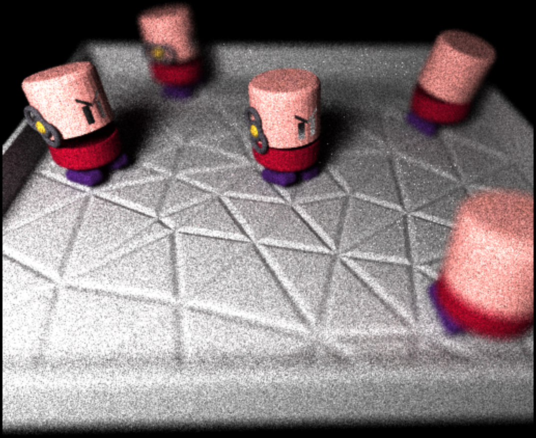
Monte Carlo Rendering Noise



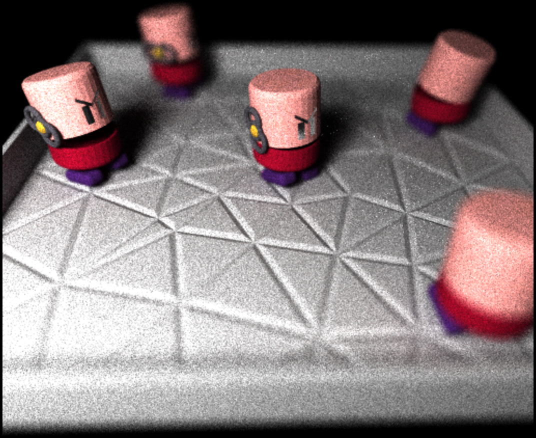
32 samples per pixel (spp) [8s]



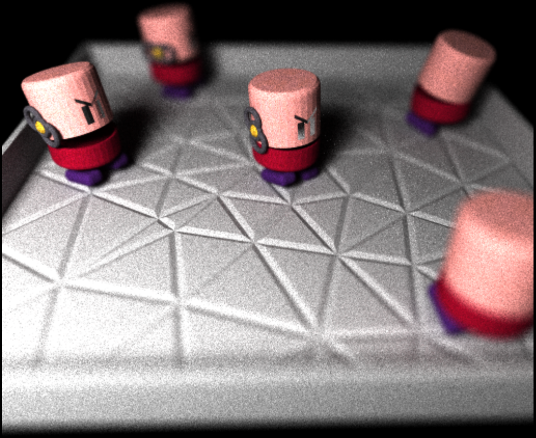
64 samples per pixel (spp) [16s]



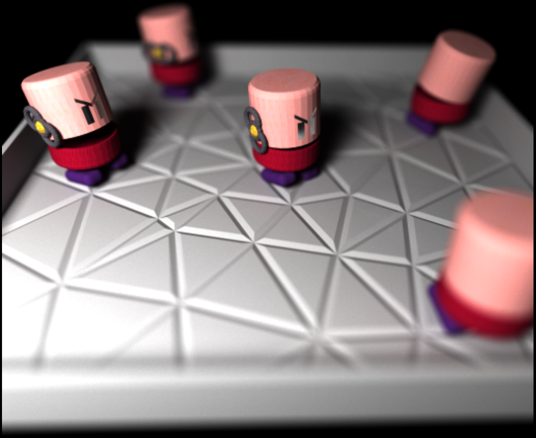
128 samples per pixel (spp) [32s]



256 samples per pixel (spp) [64s]



512 samples per pixel (spp) [128s]



65536 samples per pixel (spp) [16384s]

A General Principle: Auto-similarity

"Similar pixels must be denoised jointly,
being different samples of the same model."

A General Principle: Auto-similarity

"Similar pixels must be denoised jointly, being different samples of the same model."

Pixel similarity based on:

- Gaussian filter: spatial proximity
- Sigma/Bilateral filter: pixel color [Lee, et al., '83], [Tomasi et al., '98]
- NLmeans/BM3D: patch color [Buades et al. '05], Dabov et al, '07]
- LARK, GLIDE: kernels [Takeda, et al. '07], [Talebi, et al. '14]

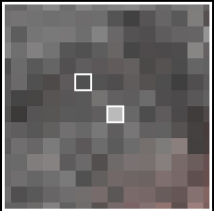
A General Principle: Auto-similarity

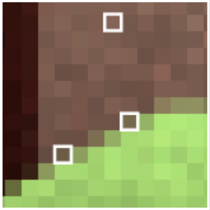
"Similar pixels must be denoised jointly,
being different samples of the same model."

Pixel similarity based on:

- Gaussian filter: spatial proximity
- Sigma/Bilateral filter: pixel color [Lee, et al., '83], [Tomasi et al., '98]
- NLmeans/BM3D: patch color [Buades et al. '05], Dabov et al, '07]
- LARK, GLIDE: kernels [Takeda, et al. '07], [Talebi, et al. '14]

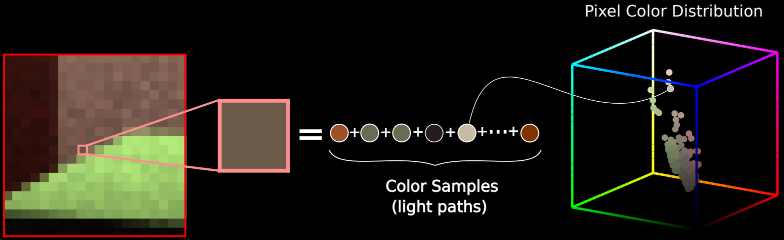
Main difficulty:
Hard to distinguish noise from intrinsic pixel variability (bias)





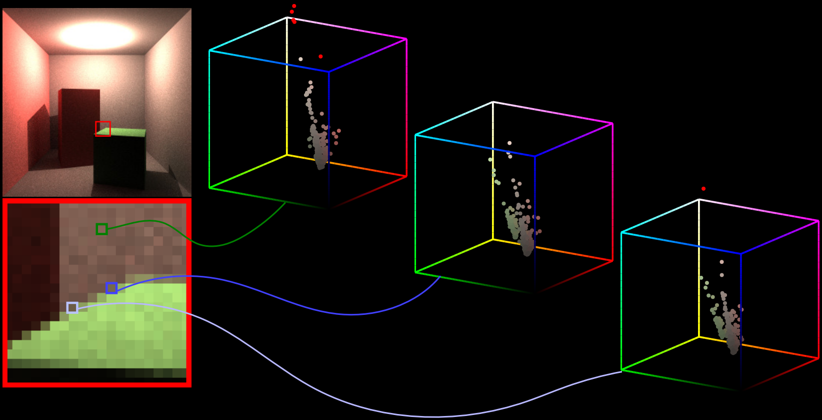
Empirical Ray Color Distribution

- During rendering a lot of information is computed
- In particular: RGB color of each ray hitting a given pixel
- Color histograms of rays cast from each pixel
- Use this histogram to define pixel similarity



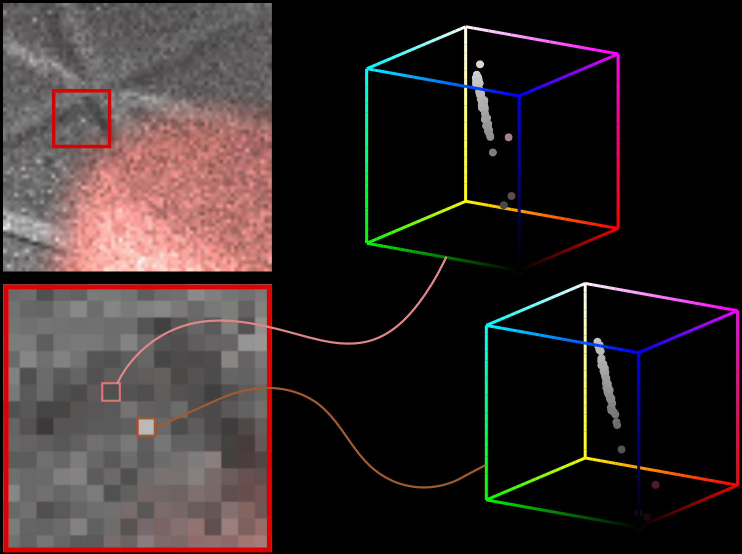
Empirical Ray Color Distribution

Measuring pixel similarity



Empirical Ray Color Distribution

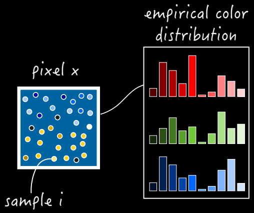
Measuring pixel similarity



Empirical Ray Color Distribution

Measuring pixel similarity

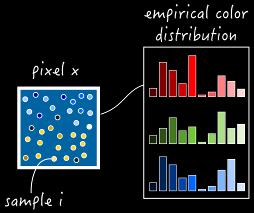
- Pixel similarity measured from the binned empirical color distributions
- Tri-stimulus color images: $3 \times 1\text{D}$ histograms (one per color channel)

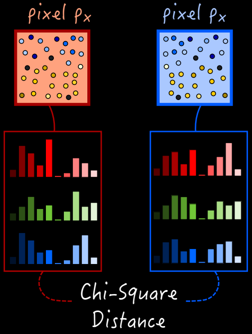


Empirical Ray Color Distribution

Measuring pixel similarity

- Pixel similarity measured from the binned empirical color distributions
- Tri-stimulus color images: $3 \times 1D$ histograms (one per color channel)





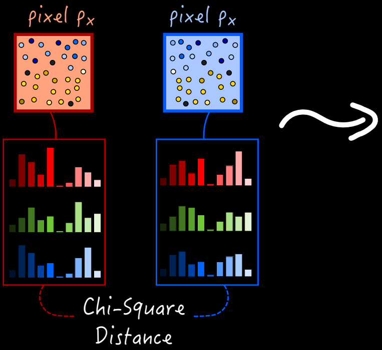
Given n_b-binned empirical color distributions $(h_i(x))$ and $(h_i(y))$ of pixels x and y, the Chi-Square distance is given by

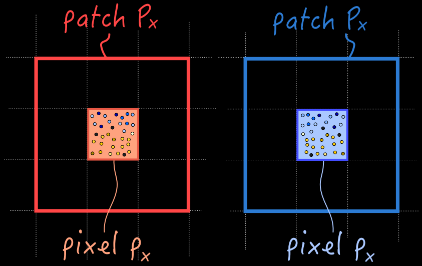
$$d_{\chi^2}(p_x, p_y) = \sum_{k=1}^{n_b} \frac{(h_k(x) - h_k(y))^2}{h_k(x) + h_k(y)}$$

Empirical Ray Color Distribution

Measuring pixel similarity

- Extended to patches for spatial coherence





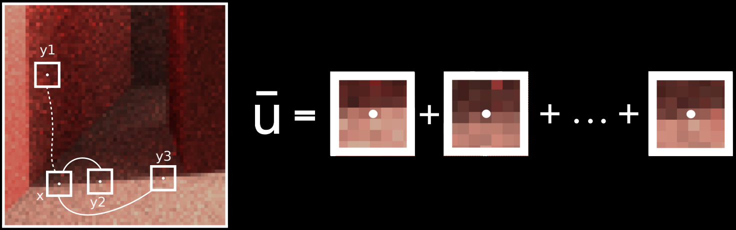
$$d_{\chi^2}(P_x, P_y) = \sum_{|t|\leq w} d_{\chi^2}(p_{x+t}, p_{y+t})$$

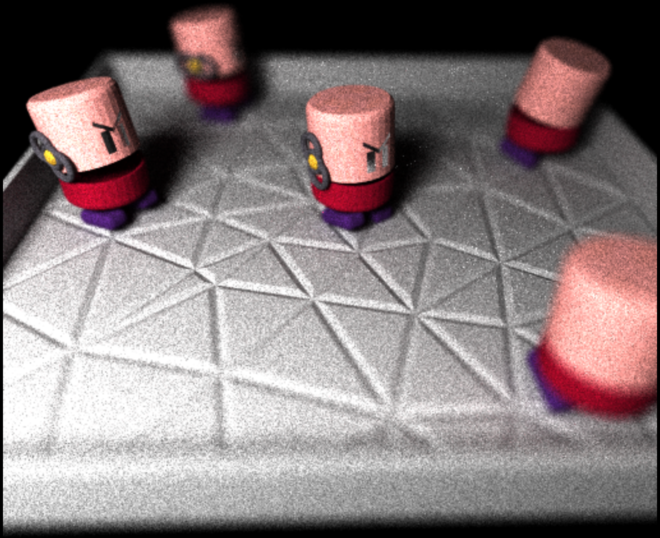
Distribution-driven average

- Given a noisy patch, all the patches that are at a distance less than κ are considered to be similar.

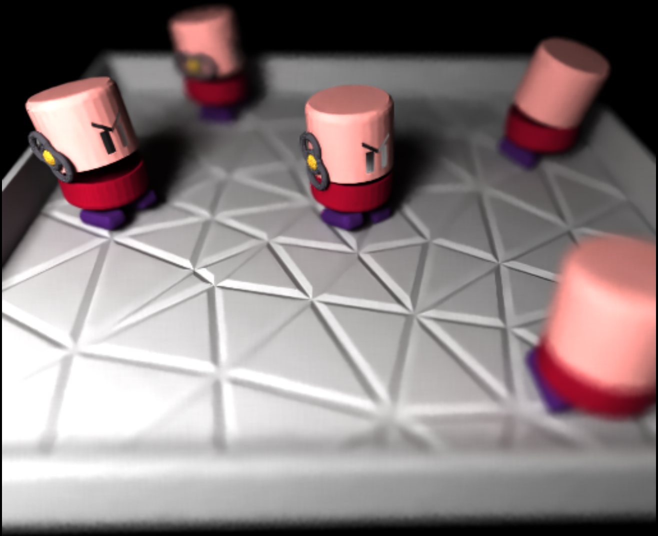
Distribution-driven average

- Given a noisy patch, all the patches that are at a distance less than κ are considered to be similar.
- Replace a patch with the average of the similar ones.

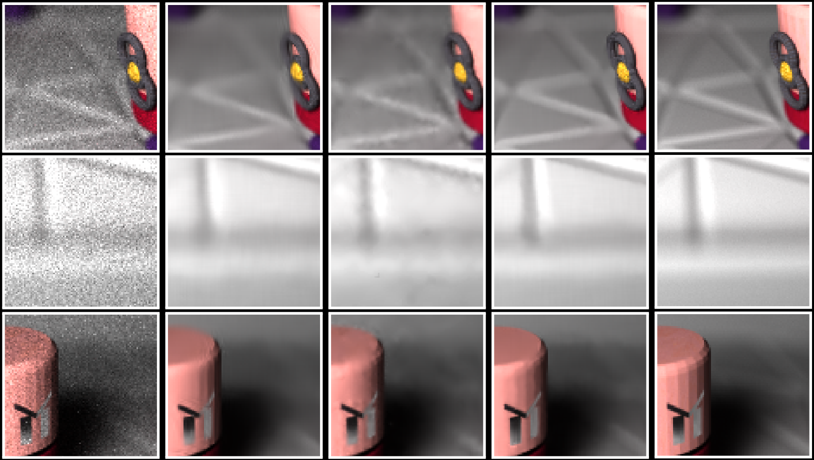




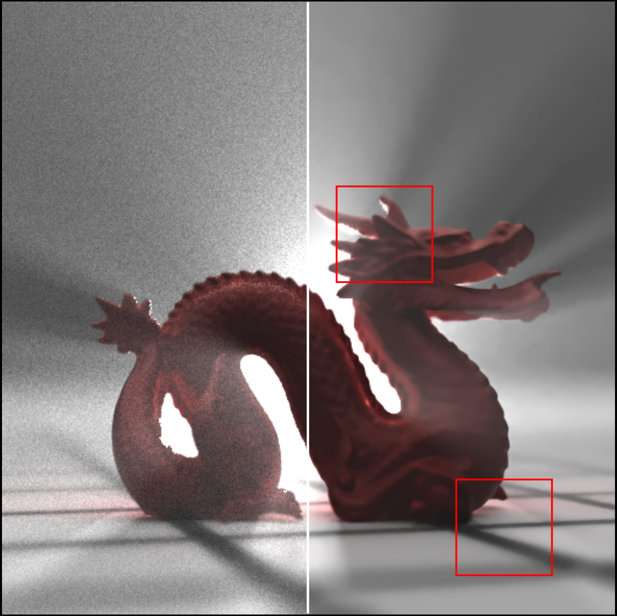
Input `toasters` scene 256spp. psnr: 33.7dB



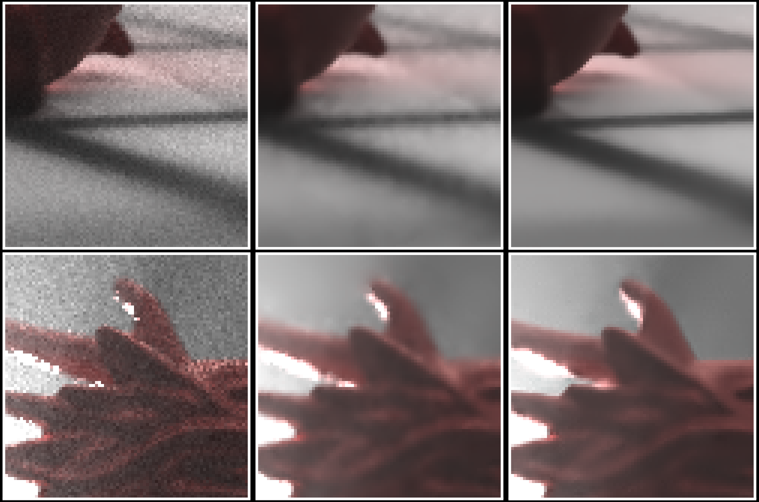
Filtered `toasters` scene. psnr: 38.1dB. Gain +14.7dB $\approx 33\times$ more samples.



mc eq. time	NL-means eq. time	ASR eq. time	**RHF**	Reference MC
24.8 db [88.8s]	36.1db [88.9s]	35.7db [88.1s]	38.1db [88.8s]	



Light interaction with participating media rendered through photon mapping



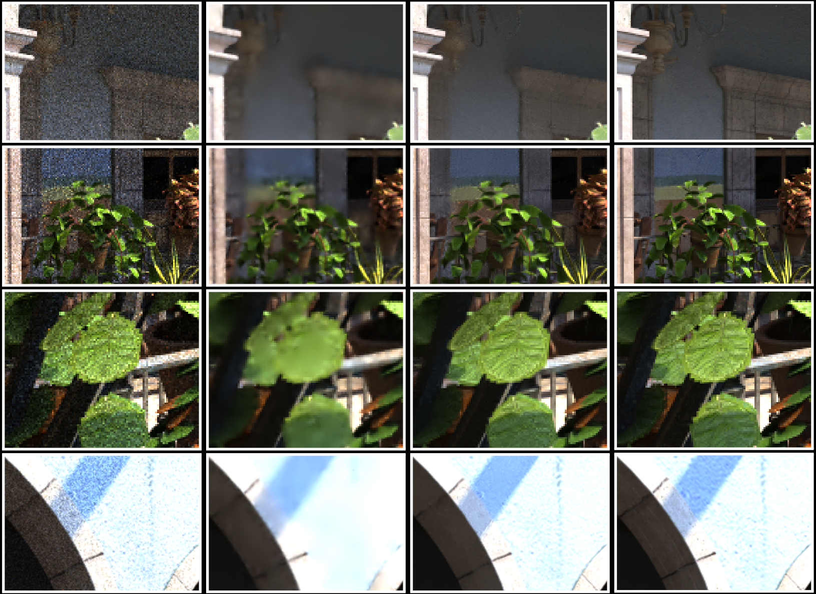
PM+FG ASR RHF

Results in `dragon-fog` scene (close-ups).

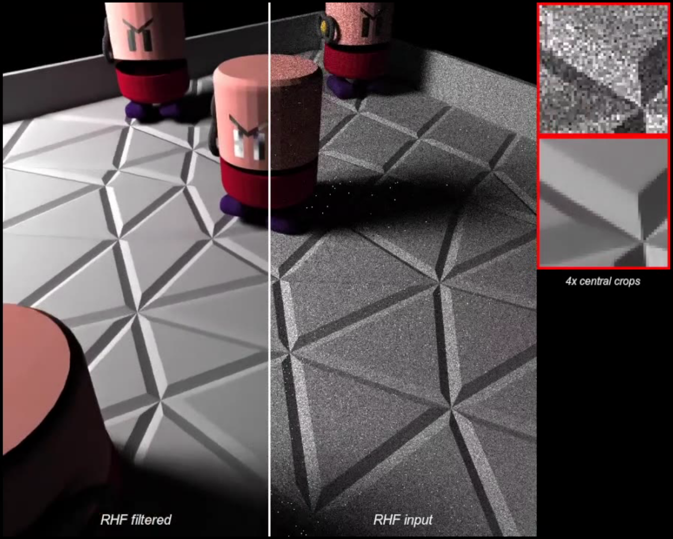
Input 256spp `san-miguel` scene. psnr: 24.1dB.

Filtered 256spp `san-miguel` scene. psnr: 29.8dB. Gain +5.7dB $\approx 4\times$ more samples.

mc noisy input	ASR eq. time	**RHF**	Reference MC



24.1dB [3521s]	26.2dB [4133s]	29.8dB [3952s]	



IPOL: Image Processing Online – ipol.im

Detailed Description + Online Demo + Source Code

I presented three different problems where we found a different and simpler solution:

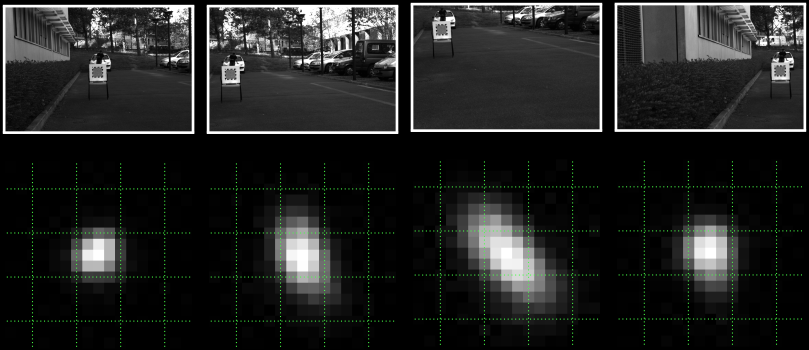
1. Instead of adding regularization we "avoid" it by choosing the best single capture (or pair of captures).
2. Instead of deblurring via deconvolution do a weighted fusion in the Fourier domain.
3. Instead of casting more rays, re-use them using the "auto-similarity" principle.

Thanks.

Experiments

Different locations

Lens aberration is more significant in image borders.



center   left   top-left   right

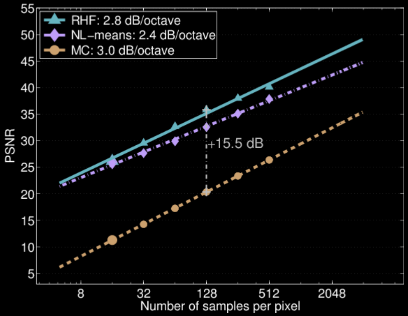
RHF: Time Complexity

- Complexity of the filtering at each scale is $O(N \cdot w \cdot b \cdot n_b)$ where N is the number of pixels, b search block size, w patch size, n_b number of bins.

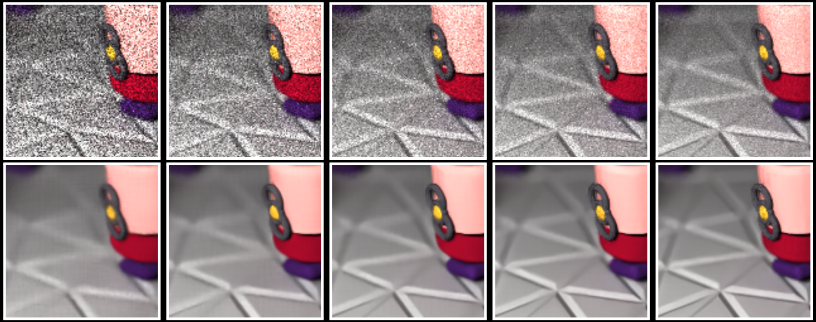
- Computational cost is independent of the number of samples.

- In the case that two scales are used the computational cost increases by about 25%,

- If n_s scales are used the computational cost is bounded from above by 133% of the filtering time at the finest resolution.



- Pure MC: duplicating spp $\rightarrow$ +3dB
- RHF slope gain is $+2.8$dB
- RHF: $+15$dB $\approx 35\times$ more samples.



32spp 64spp 128spp 256spp 512spp

Multi-scale Extension

MC noise is white: To remove low-frequency noise filter at different scales and then fuse the filtered images.

noisy input

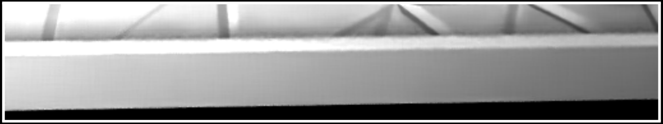
single-scale denoised

Multi-scale Extension

MC noise is white: To remove low-frequency noise filter at different scales and then fuse the filtered images.

noisy input

single-scale denoised



three-scale denoised