From Example-Based to Local Gaussian Priors. Applications to Inpainting, HDR & Challenges Ahead

Andrés Almansa
joint work with C. Aguerrebere, T. Guillemot, A. Newson,
T. Boubekeur, J. Delon, M. Fradet, Y. Gousseau, P. Musé, P. Perez

LTCI

Workshop on New Trends in Optimization for Imaging
Sanya, January 19th 2015

Outline

- Overview
  - Example-based image inpainting
  - From example-based to model-based regularization
  - Local Gaussian Models vs GMM
- Local Gaussian Models in HDR Imaging
- Challenges ahead

Example-based image inpainting [Efros-Leung 1999, Wexler 2005]

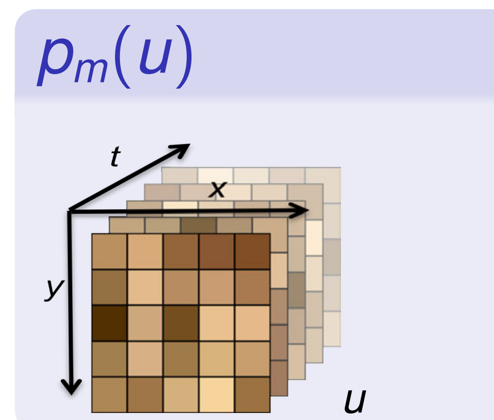
Input: *Visible part* of the image $u|_{O^c}$
Output: reconstruction of the *occluded part* $u|_O$ via

$$\min_{u|_O} \sum_{m \in O} \|p_m(u) - p_{\varphi(m)}(u)\|^2$$

where

$$\varphi(m) = \arg\min_{n \in O^c} \|p_m(u) - p_n(u)\|^2$$

is the *nearest neighbour* of $p_m(u)$:
patches in Images and videos:



Example-based image inpainting [Arias-Caselles-Facciolo 2012]

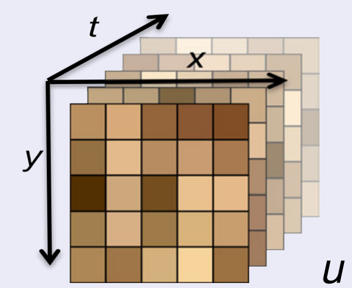
Input: *Visible part* of the image $u|_{O^c}$
Output: reconstruction of the *occluded part* $u|_O$ via

$$\min_{w,u|_O} \sum_{m\in O,\, n\in O^c} w(m,n)\|p_m(u)-p_n(u)\|^2 - T\sum_m H(w(m,\cdot))$$

under the constraint $\sum_n w(m,n) = 1, \ \forall m \in O$

where $H(f) = -\sum_n f(n)\log(f(n))$ is the entropy of the probability density distribution f.

$p_m(u)$



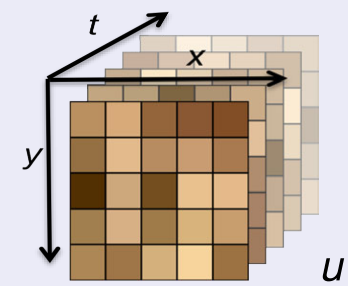
Example-based image inpainting [Arias-Caselles-Facciolo 2012]

Input: *Visible part* of the image $u|_{O^c}$
Output: reconstruction of the *occluded part* $u|_O$ via

$$\min_{w,u|_O} \sum_{m\in O,\, n\in O^c} w(m,n)\|p_m(u)-p_n(u)\|^2 - T\sum_m H(w(m,\cdot))$$

under the constraint $\sum_n w(m,n) = 1, \ \forall m \in O$

$p_m(u)$



where $H(f) = -\sum_n f(n)\log(f(n))$ is the entropy of the probability density distribution f.

Non-convex problem
Alternated minimisation of convex problems

- w-minimization *(Learn local distribution)*

$$w(m,n) = \frac{1}{Z} e^{-\frac{1}{T}\|p_m(u)-p_n(u)\|^2}$$

- u-min: *(a posteriori expectation)*

$$\hat{p}_m = \mathrm{E}\left[p \mid p_m(u)\right] = \sum_n w(m,n) p_n$$

- Aggregation: $u(m) = \sum_n \hat{p}_n[n-m]$

Example-based image inpainting [Arias-Caselles-Facciolo 2012]

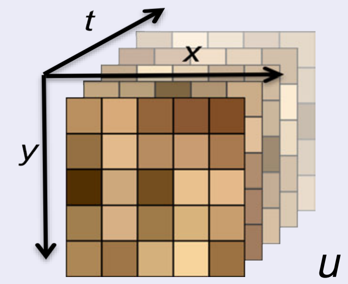
Input: *Visible part* of the image $u|_{O^c}$
Output: reconstruction of the *occluded part* $u|_O$ via

$$\min_{w,u|_O} \sum_{m\in O,\, n\in O^c} w(m,n)\|p_m(u)-p_n(u)\|^2 - T\sum_m H(w(m,\cdot))$$

under the constraint $\sum_n w(m,n) = 1, \; \forall m \in O$

where $H(f) = -\sum_n f(n)\log(f(n))$ is the entropy of the probability density distribution f.

$p_m(u)$



Non-convex problem
Alternated minimisation of convex problems

- w-minimization *(Learn local distribution)*

$$w(m,n) = \frac{1}{Z} e^{-\frac{1}{T}\|p_m(u)-p_n(u)\|^2}$$

- u-min: *(a posteriori expectation)*

$$\hat{p}_m = \mathrm{E}\left[p \mid p_m(u)\right] = \sum_n w(m,n)p_n$$

- Aggregation: $u(m) = \sum_n \hat{p}_n[n-m]$

Challenges

- Computation of w truncated and approximated by Patch Match [Barnes 2009] Other alternatives? non-structured data?
- Non-convexity: Multi-scale
- Patch similarity: ℓ^2 is ambiguous for fine textures

Video Inpaintng – *original video*

Video Inpainting [Newson-Almansa-Fradet-Gousseau-Perez 2014]

Paper/Demo/Code at http://perso.enst.fr/~almansa/video_inpainting/

Example-based vs. model-based image inpainting

Example based

- *w*-minimization *(Learn local distribution)*

$$w(m,n) = \frac{1}{Z} e^{-\frac{1}{T}\|p_m(u)-p_n(u)\|^2}$$

- *u*-min: *(a posteriori expectation)*

$$\hat{p}_m = \mathrm{E}\left[p \mid p_m(u)\right] = \sum_n w(m,n) p_n$$

- Aggregation: $u(m) = \sum_n \hat{p}_n[n-m]$

Model based

- *w*-minimization *(Learn local model)*

$$w(m,\cdot) \sim N(\mu_m, \Sigma_m) \text{ that fits}$$

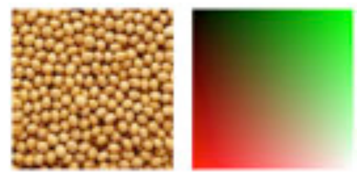
$$\{p_n(u) \ : \ \|p_m(u) - p_n(u)\|^2 < T\}$$

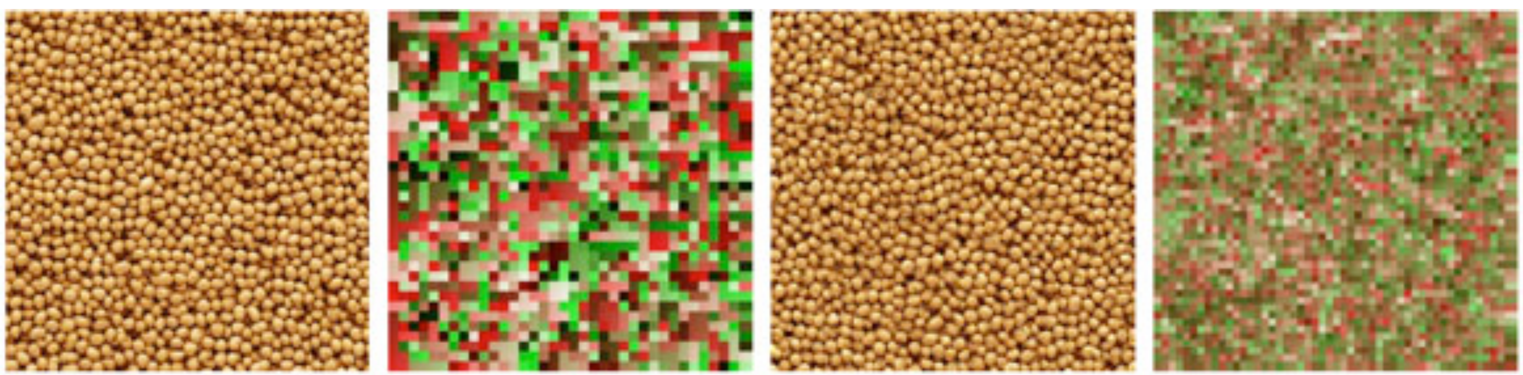
- *u*-minimization: estimate $\hat{p}_n$ by:
  - EAP (blurry), or...
  - MAP, or...
  - Random synthesis near $p_m(u)$ based on $N(\mu_m, \Sigma_m)$
- Aggregation: $u(m) = \sum_n \hat{p}_n[n-m]$

- fast algorithms on unstructured data (CovTree)
- synthesize vs. copy

Model-based image inpainting [Raad-Desolneux-Morel 2014]

original





Synthesized (example-based) Synthesized (model-based)

Non-Local Means denoising [Buades-Coll-Morel 2005]

Input: *Noisy image* $\tilde{u} = u + n$ where $n \sim N(0, \sigma^2 Id)$.
Output: *Estimated clean image* $\hat{u}$ via

$$\max_u \sum_{m,n} w(m,n)\|p_m(u) - p_n(\tilde{u})\|^2 - T \sum_m H(w(m,\cdot))$$

under the constraint $\sum_n w(m,n) = 1, \ \forall m \in O$

Example based

- w-minimization *(Learn local distribution)*

$$w(m,n) = \frac{1}{Z} e^{-\frac{\|p_m(u) - p_n(\tilde{u})\|^2 - T}{T}}$$

- u-minimization: *(a posteriori expectation)*

$$\hat{p}_m = \sum_n w(m,n) p_n$$

- Aggregation: $\hat{u}(m) = \sum_n \hat{p}_n[n-m]$

Non-Local Bayes denoising [Lebrun-Buades-Morel 2013]

Input: *Noisy image* $\tilde{u} = u + n$ where $n \sim N(0, \sigma^2 Id)$.
Output: *Estimated clean image* $\hat{u}$ via

$$\max_u \Pr\left[p_m(u) \mid p_n(\tilde{u}),\ N(\mu_m, \Sigma_m)\right]$$

s.t. $N(\mu_m, \Sigma_m)$ fits $\{p_n(u)\ :\ \|p_n(u) - p_m(u)\| < \delta\}$

Model based

- *w*-minimization *(Learn Local Gaussian Model)*

$$\boldsymbol{\mu}_m = \frac{1}{Z} \sum_n e^{-\frac{\|p_m(u) - p_n(\tilde{u})\|^2}{\sigma^2}} p_n(\tilde{u})$$

$$\boldsymbol{\Sigma}_m = \frac{1}{Z} \sum_n e^{-\frac{\|p_m(u) - p_n(\tilde{u})\|^2}{\sigma^2}} \bar{p}_n(\tilde{u}) \bar{p}_n(\tilde{u})^T - \sigma^2 Id$$

- *u*-minimization: *(MAP)*

$$\hat{p}_m = \arg\min_q \frac{1}{\sigma^2} \|q - p_m(\tilde{u})\|^2 + (q - \boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1} (q - \boldsymbol{\mu}_m)$$

- Aggregation: $\hat{u}(m) = \sum_n \hat{p}_n[n - m]$

Piecewise Linear Estimators [Yu-Mallat-Sapiro 2012]

Input: *Perturbed image* $\tilde{u} = Au + n$ where $n \sim N(0, \sigma^2 Id)$.
Output: *Restored image* $\hat{u}$ via

$$\max_{u(m),\, k(m)} \Pr\left[p_m(u) \mid p_n(\tilde{u}),\ N(\mu_{k(m)}, \Sigma_{k(m)})\right]$$

with $k = 1, \ldots, 20$
s.t. $N(\mu_{k(m)}, \Sigma_{k(m)})$ fits $\{p_n(u)\ :\ k(m) = k(n)\}$

Model based

- initialization: $\hat{u}^0$, $(\boldsymbol{\mu}_k^0, \boldsymbol{\Sigma}_k^0)$, $k(m)$
- Relearn Gaussian Models $(\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)$ to fit $\{p_m(u)\ :\ k(m) = k\}$
- Signal estimation ($\hat{p}_m$) and model selection ($k(m)$)

$$(\hat{p}_m, k(m)) = \arg\max_{q,k} \Pr\left[q \mid p_m(\tilde{u})\,,\ N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right]$$

- Aggregation: $\hat{u}^i(m) = \sum_n \hat{p}_n[n - m]$

Learning-based restoration [Zoran-Weiss 2011]

Offline learning

Input: a huge database of *natural image patches* $\mathbf{p}_i \in \mathcal{P}$.
Output: Gaussian Mixture Model $\{N(\mu_k, \Sigma_k) \,:\, k = 1, \dots, 250\}$ fitting the data
(several days worth of computation)

Restoration

Input: *Perturbed image* $\tilde{u} = Au + n$ where $n \sim N(0, \sigma^2 Id)$.
Gaussian Mixture Model $\{N(\mu_k, \Sigma_k) \,:\, k = 1, \dots, 250\}$
(representing the manifold of natural image patches)
Output: *Restored image* $\hat{u}$ via

$$\max_{u(m),\, k(m)} \Pr\left[p_m(u) \,\middle|\, p_n(\tilde{u}),\, N(\mu_{k(m)}, \Sigma_{k(m)})\right]$$

Covariance Tree [Guillemot-Almansa-Boubekeur 2014]

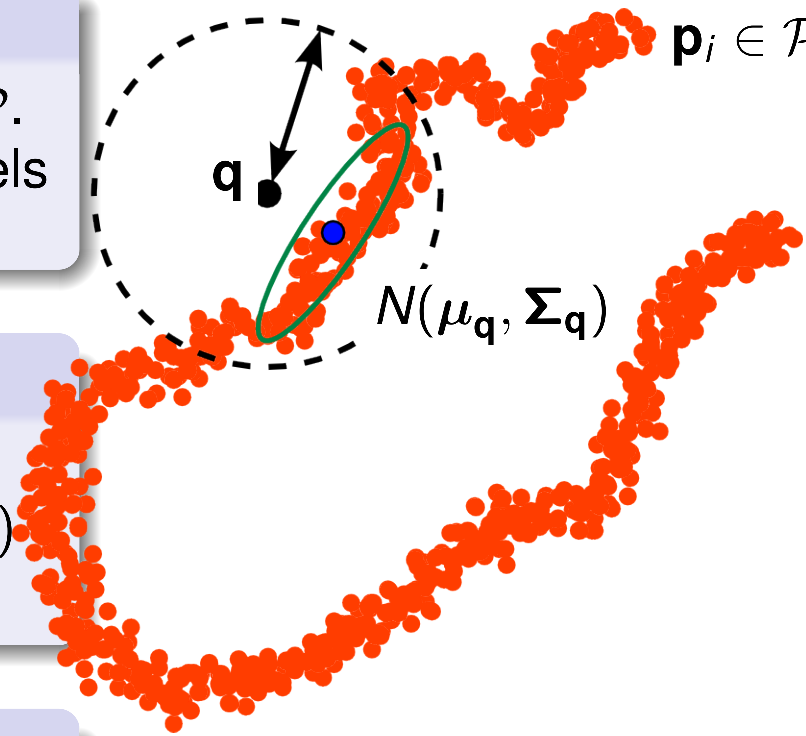
Learning

Input: a huge database of *data points* $\mathbf{p}_i \in \mathcal{P}$.
Output: pre-computed Local Gaussian Models at several *scales* and *locations*

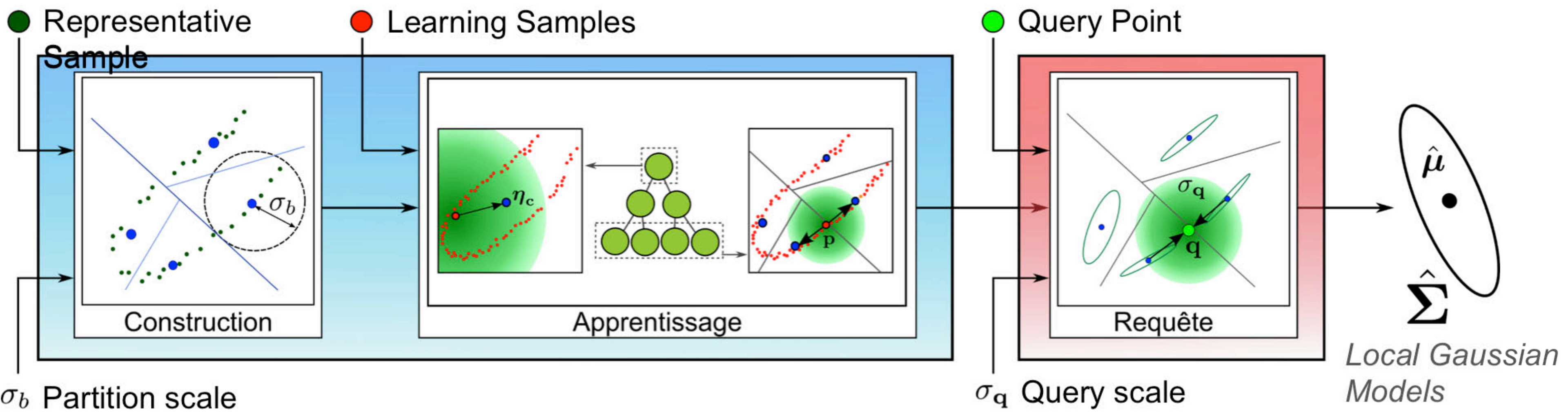
Query

Input: a query point $\mathbf{q}$ and a scale σ
Output: accurate approximation of $N(\mu_{\mathbf{q}}, \mathbf{\Sigma}_{\mathbf{q}})$ fitting $\mathcal{P}|_{B(\mathbf{q},\sigma)}$

Bayesian Restoration



Covariance Trees [Guillemot-Almansa-Boubekeur 2014]



Learning-based denoising

Bruité | NLB | CovTree NLB | CovTree + Dictionnary

PSNR : 22.4 dB | PSNR : 30.2 dB | PSNR : 30.0 dB | PSNR : 31.1 dB

Covariance Trees [Guillemot-Almansa-Boubekeur 2014]

Challenges

- Time-dependent data
- Non-gaussian noise
- Incomplete patches

Single-Shot High Dynamic Range Imaging using Local Gaussian Models

C. Aguerrebere **Andrés Almansa** J. Delon Y. Gousseau P. Musé

LTCI

Workshop on New Trends in Optimization for Imaging
Sanya, January 19th 2015

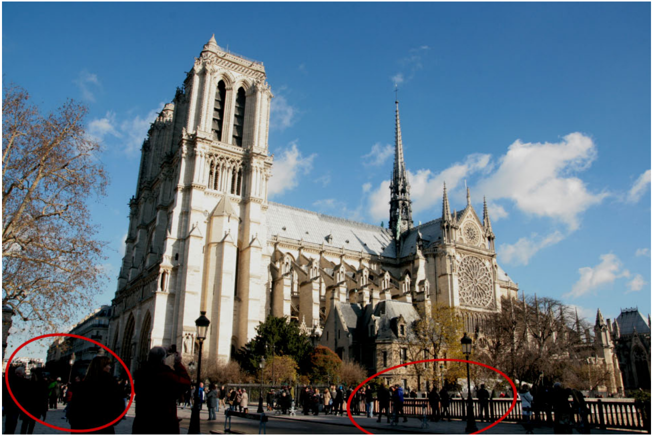
High Dynamic Range Imaging (HDR)

Capture a scene containing a large range of intensity levels...

Limited dynamic range of the camera → loss of details in bright and/or dark areas.

High Dynamic Range Imaging (HDR)

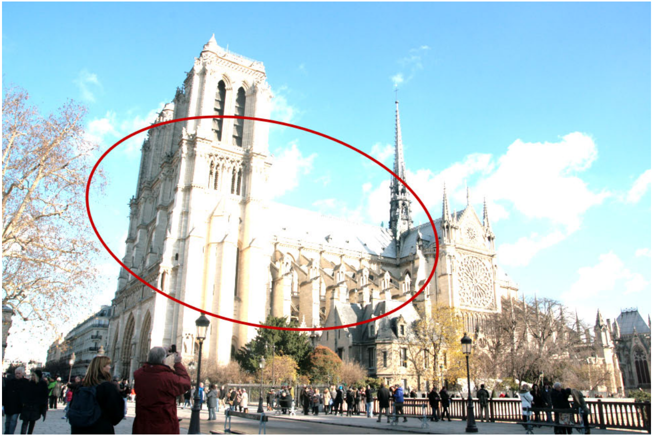
... using a standard digital camera.



Limited dynamic range of the camera → loss of details in bright and/or dark areas.

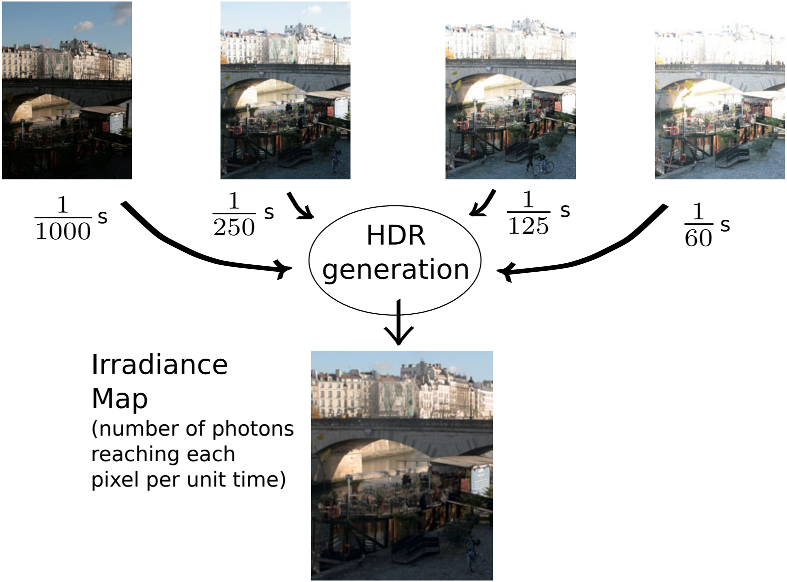
High Dynamic Range Imaging (HDR)

... using a standard digital camera.



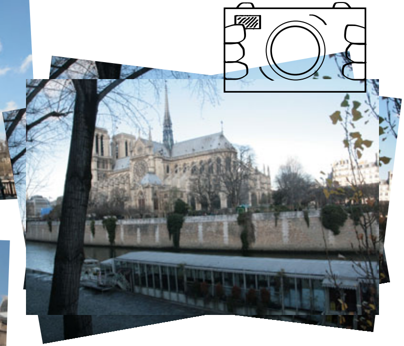
Limited dynamic range of the camera $\rightarrow$ loss of details in bright and/or dark areas.

HDR imaging - Multi-image approach



Challenges of Multi-image HDR Imaging

moving
objects



camera
motion

noise

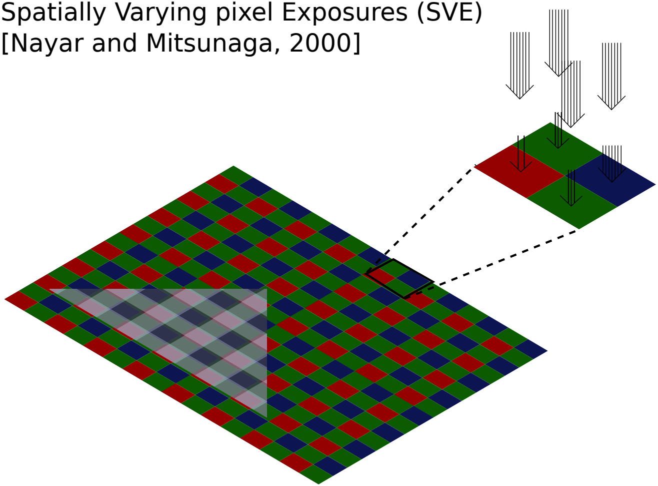
Challenges of Multi-image HDR Imaging

Input frames: camera + object motion

Result: ghosting artifacts

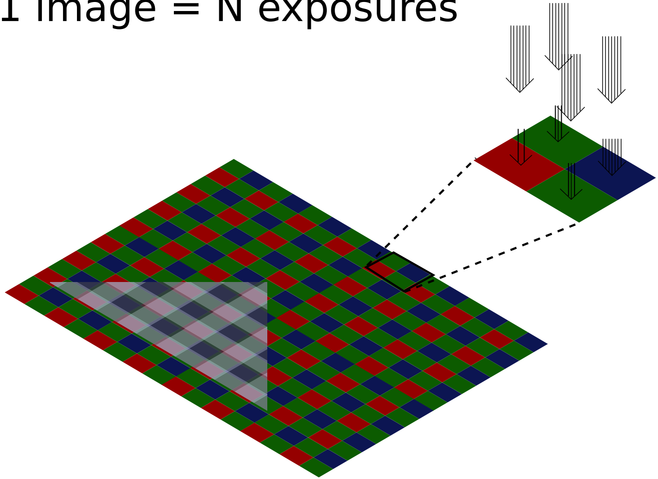
Alternative: Single-image HDR

Spatially Varying pixel Exposures (SVE)
[Nayar and Mitsunaga, 2000]



Alternative: Single-image HDR

1 image = N exposures



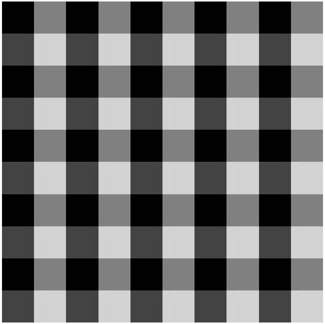
SVE Single-image HDR

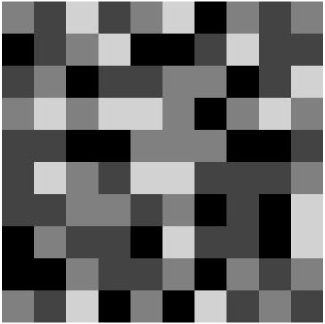
- ✓ No need for image alignment.
- ✓ No need for motion detection.
- ✓ No ghosting problems.
- ✓ No large saturated regions to fill.

- × Unknown pixels to be restored (over and under exposed pixels).
- × Noise.
- × Need to modify the standard camera.
  - Alternative without camera modification [Hirakawa and Simon, 2011].

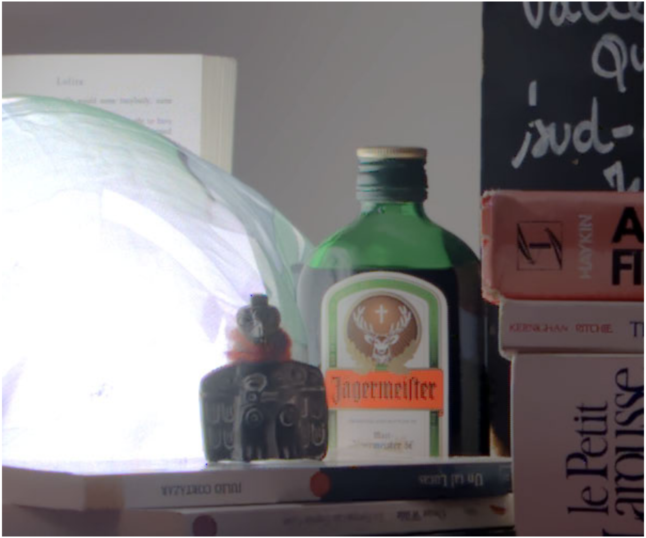
SVE: Regular or Random?

Random pattern to avoid aliasing [Schöberl et al., 2012]

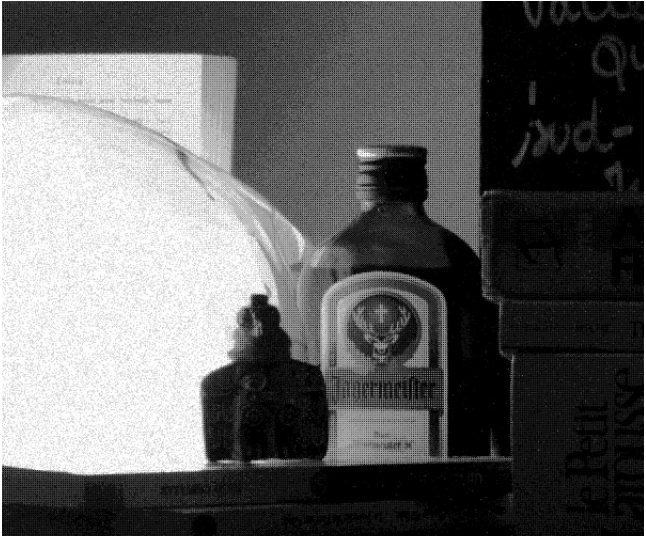




Single-image HDR - Problem to solve

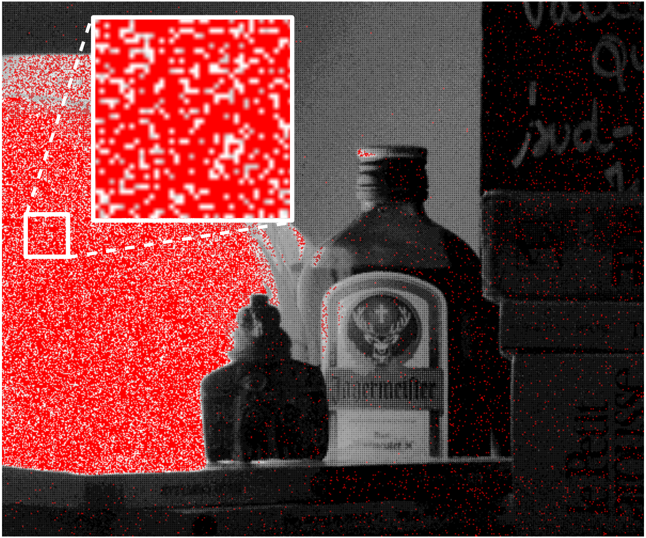


Single-image HDR - Problem to solve

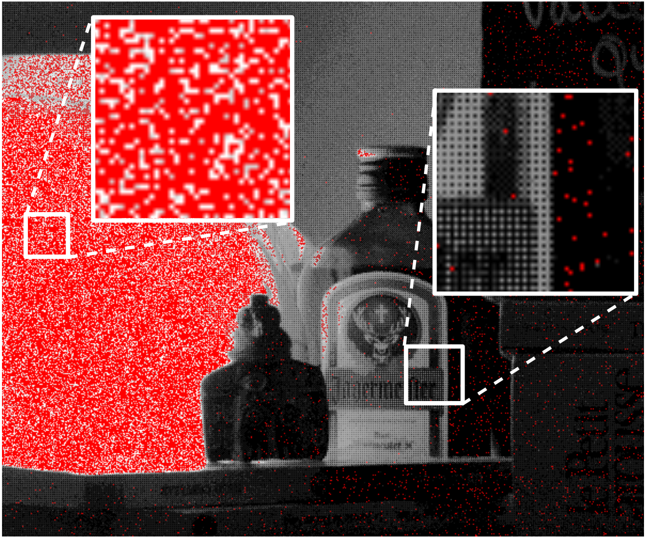


Single-image HDR - Problem to solve

Single-image HDR - Problem to solve



Single-image HDR - Problem to solve



Our approach

Extension of Piecewise Linear Estimators (PLE) [Yu et al., 2012]

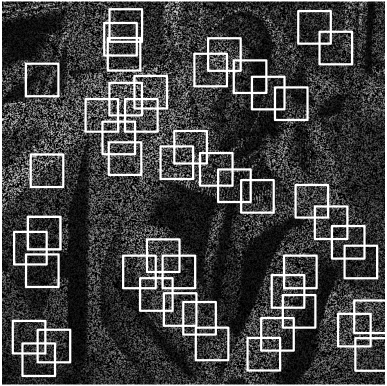
Our approach

Extension of Piecewise Linear Estimators (PLE) [Yu et al., 2012]

Our approach

Extension of Piecewise Linear Estimators (PLE) [Yu et al., 2012]

Patch-based method

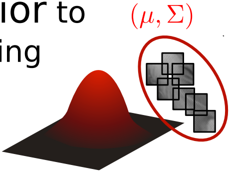


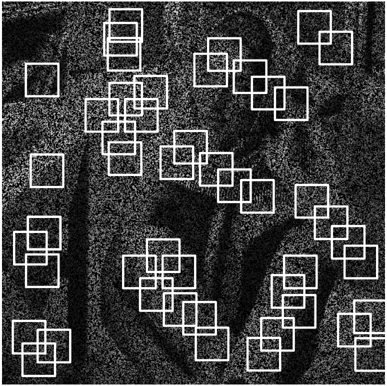
Our approach

Extension of Piecewise Linear Estimators (PLE) [Yu et al., 2012]

Patch-based method

Gaussian prior to restore missing information



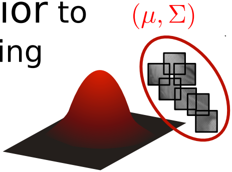


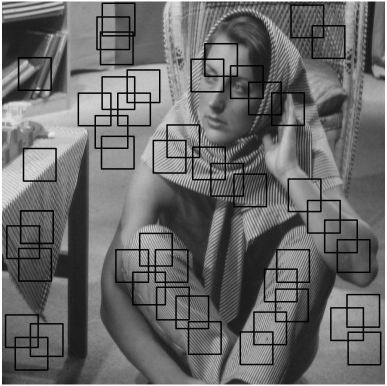
Our approach

Extension of Piecewise Linear Estimators (PLE) [Yu et al., 2012]

Patch-based method

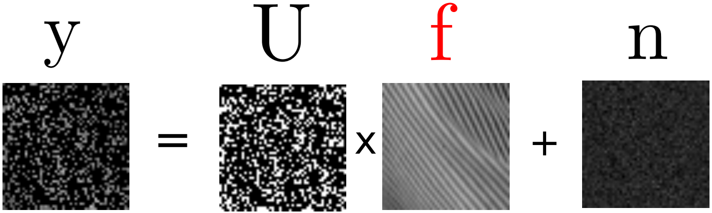
Gaussian prior to restore missing information



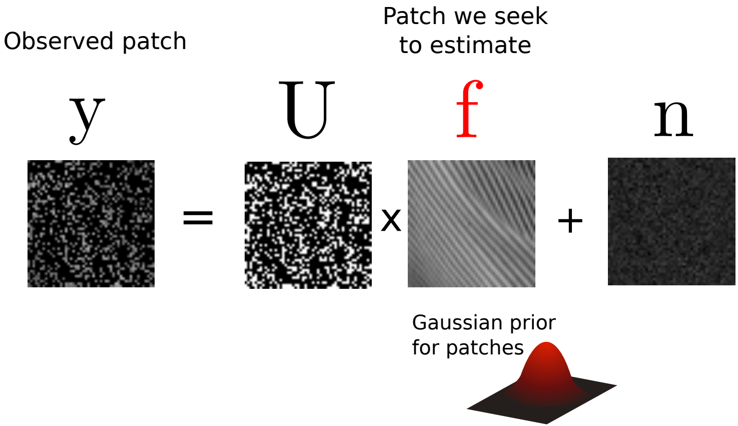


PLE Patch Model

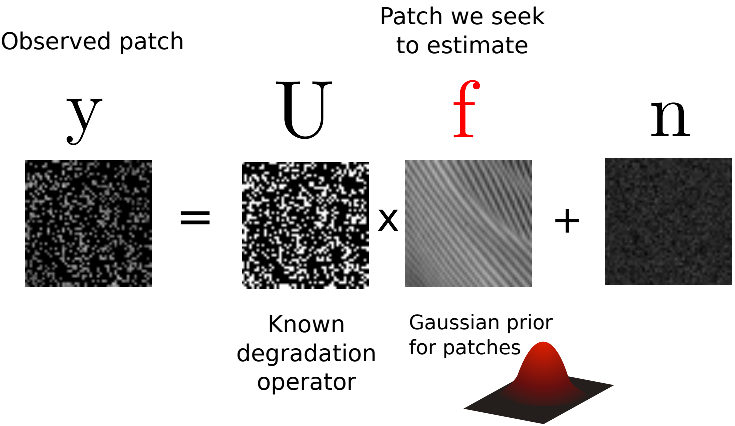
Observed patch



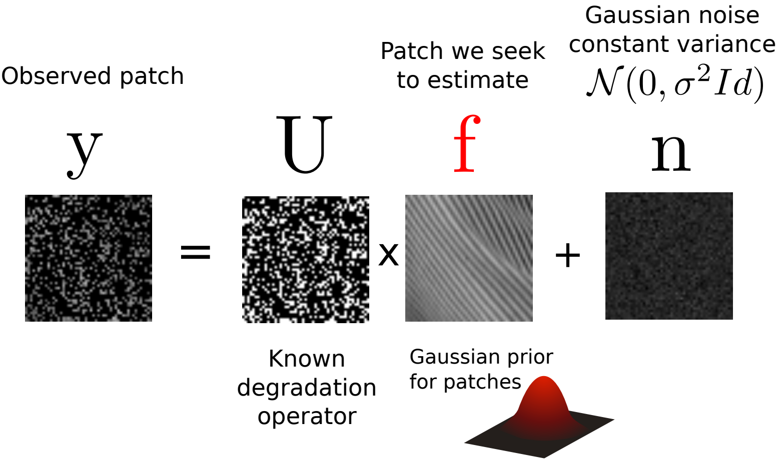
PLE Patch Model



PLE Patch Model



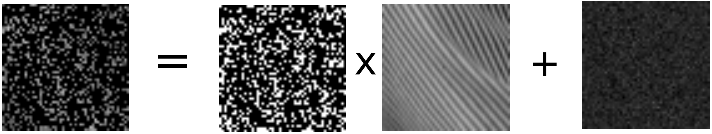
PLE Patch Model



Patch Model for Raw Data

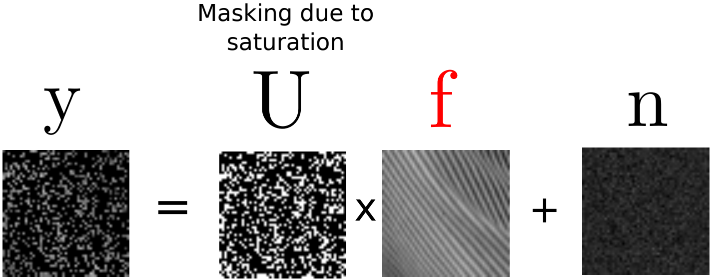
RAW data:

$$\mathbf{y} = \mathbf{U} \times \mathbf{f} + \mathbf{n}$$



Patch Model for Raw Data

RAW data:



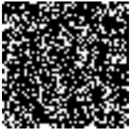
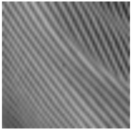
Patch Model for Raw Data

RAW data:

Masking due to saturation

~~Gaussian noise constant variance~~ $\cancel{\mathcal{N}(0, \sigma^2 Id)}$

$$\mathrm{y} = \mathrm{U} \times {\color{red}\mathrm{f}} + \mathrm{n}$$

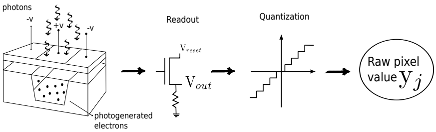
 =  x  + 

Gaussian noise
variable variance
dependent on $\color{red}\mathrm{f}$

Patch Model for Raw Data

RAW data:

Gaussian noise
variable variance
dependent on f



Main noise sources:

✓ Shot noise

✓ Readout noise

$$y_j \sim \mathcal{N}(\mathrm{f}_j, \sigma^2(\mathrm{f}_j))$$

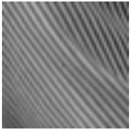
$$\boxed{\sigma^2(\mathrm{f}_j) = \frac{g^2 o a \tau \mathrm{f}_j + \sigma_R^2}{(g^2 o a \tau)^2}}$$

f_j irradiance
τ exposure time
g camera gain
o SVE optical gain
a photo-response non-uniformity factor
μ_r, σ_R^2 readout noise mean and variance

Patch Model for Raw Data

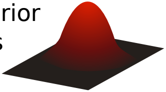
Observed patch

$$\mathrm{y} = \mathrm{U} \times \mathrm{f} + \mathrm{n}$$

 =  x  + 

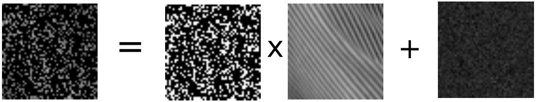
$\mathcal{N}(\mu, \mathrm{U}\Sigma\mathrm{U}^T + \Sigma_{\mathrm{n}}(\mathrm{f}))$ $\mathcal{N}(\mu, \Sigma)$ $\mathcal{N}(0, \Sigma_{\mathrm{n}}(\mathrm{f}))$

Gaussian prior
for patches



Patch Reconstruction

$$\mathrm{y} = \mathrm{U} \times \mathrm{f} + \mathrm{n}$$

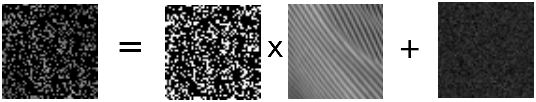


$\mathcal{N}(\mu, \mathrm{U}\Sigma\mathrm{U}^T + \Sigma_\mathrm{n}(\mathrm{f}))$ $\mathcal{N}(\mu, \Sigma)$ $\mathcal{N}(0, \Sigma_\mathrm{n}(\mathrm{f}))$

Minimize mean squared error : $\mathrm{W} = \arg\min_{W} \mathrm{E}[(W\mathrm{y} - \mathrm{f})^2]$

How to set Gaussian prior μ and Σ?

Patch Reconstruction



$\mathcal{N}(\mu, \mathrm{U}\Sigma\mathrm{U}^T + \Sigma_\mathrm{n}(\mathrm{f}))$ $\mathcal{N}(\mu, \Sigma)$ $\mathcal{N}(0, \Sigma_\mathrm{n}(\mathrm{f}))$

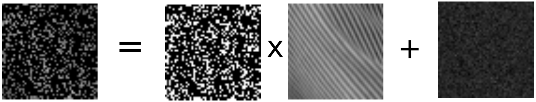
Minimize mean squared error :
$$\mathrm{W} = \arg\min_W \mathrm{E}[(W\mathrm{y} - \mathrm{f})^2]$$

Wiener filter:
$$\mathrm{W} = \Sigma\,\mathrm{U}^T(\mathrm{U}\Sigma\,\mathrm{U}^T + \Sigma_\mathrm{n}(\mathrm{f}))^{-1}$$

How to set Gaussian prior μ and Σ?

Patch Reconstruction

$$\mathrm{y} = \mathrm{U} \times \mathrm{f} + \mathrm{n}$$



$\mathcal{N}(\mu, \mathrm{U}\Sigma\mathrm{U}^T + \Sigma_\mathrm{n}(\mathrm{f}))$ $\quad$ $\mathcal{N}(\mu, \Sigma)$ $\quad$ $\mathcal{N}(0, \Sigma_\mathrm{n}(\mathrm{f}))$

Minimize mean squared error : $\mathrm{W} = \arg\min_W \mathrm{E}[(W\mathrm{y} - \mathrm{f})^2]$

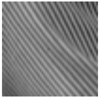
Wiener filter: $\mathrm{W} = \Sigma\,\mathrm{U}^T(\mathrm{U}\Sigma\,\mathrm{U}^T + \Sigma_\mathrm{n}(\mathrm{f}))^{-1}$

Patch reconstruction: $\boxed{\hat{\mathrm{f}} = \mathrm{W}(\mathrm{y} - \mathrm{U}\mu) + \mu}$

How to set Gaussian prior μ and Σ?

Patch Reconstruction

$$\mathrm{y} \qquad \mathrm{U} \qquad \mathrm{f} \qquad \mathrm{n}$$

 =  x  + 

$\mathcal{N}(\mu, \mathrm{U}\Sigma\mathrm{U}^T + \Sigma_{\mathrm{n}}(\mathrm{f}))$ $\quad$ $\mathcal{N}(\mu, \Sigma)$ $\quad$ $\mathcal{N}(0, \Sigma_{\mathrm{n}}(\mathrm{f}))$

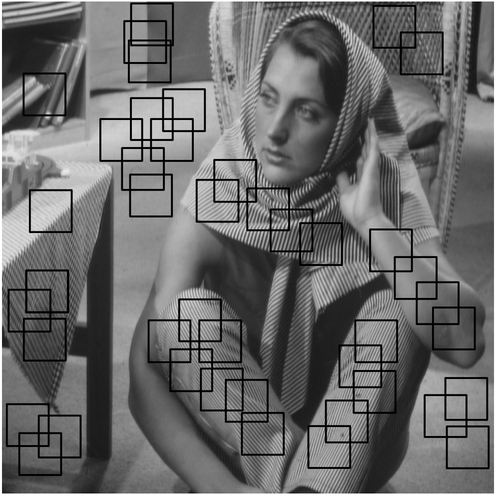
Minimize mean squared error : $\mathrm{W} = \arg\min_{W} \mathrm{E}[(W\mathrm{y} - \mathrm{f})^2]$

Wiener filter: $\mathrm{W} = \Sigma\,\mathrm{U}^T(\mathrm{U}\Sigma\,\mathrm{U}^T + \Sigma_{\mathrm{n}}(\mathrm{f}))^{-1}$

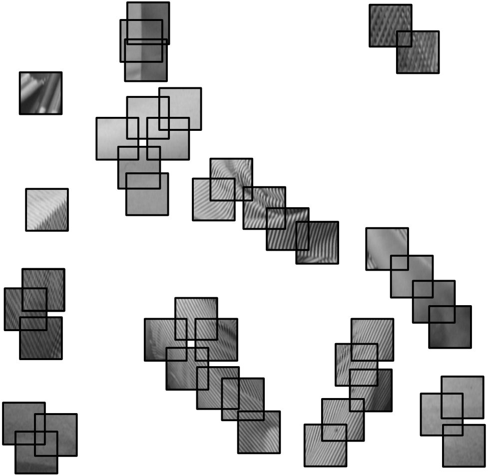
Patch reconstruction: $\boxed{\hat{\mathrm{f}} = \mathrm{W}(\mathrm{y} - \mathrm{U}\mu) + \mu}$

How to set Gaussian prior μ and Σ?

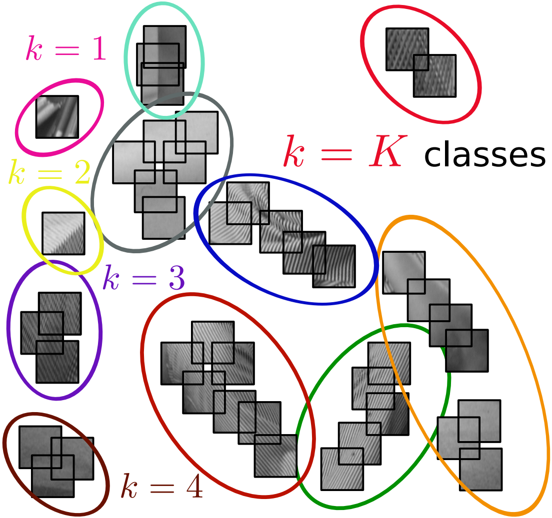
Gaussian models for image patches



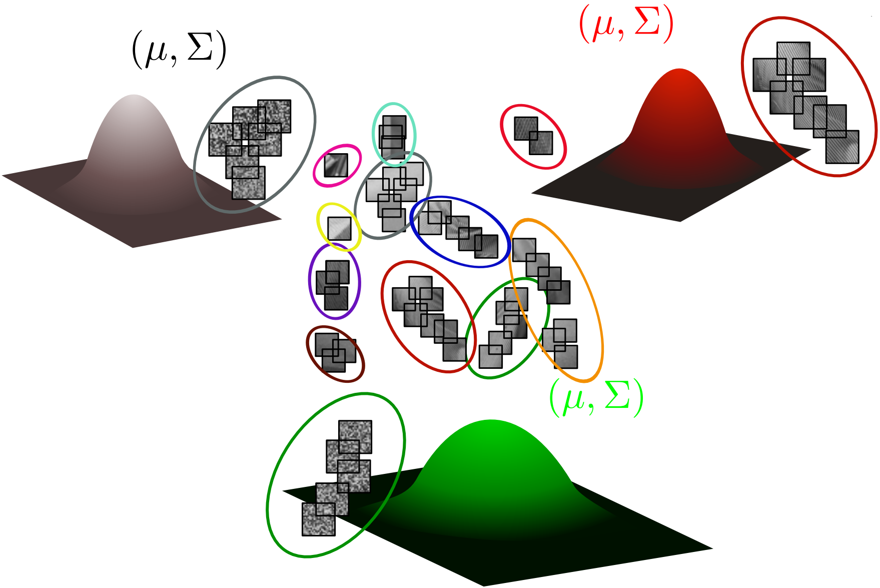
Gaussian models for image patches



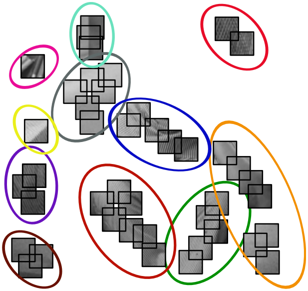
Gaussian models for image patches



Gaussian models for image patches



Class parameters estimation



For each class:

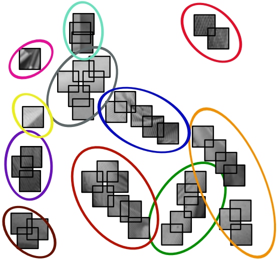
$k = 1, \ldots, K$

$$\begin{cases} \tilde{\mu}_k = \dfrac{1}{|C_k|} \displaystyle\sum_{i \in C_k} \hat{\mathrm{f}}_i \\ \tilde{\Sigma}_k = \dfrac{1}{|C_k|} \displaystyle\sum_{i \in C_k} (\hat{\mathrm{f}}_i - \tilde{\mu}_k)(\hat{\mathrm{f}}_i - \tilde{\mu}_k)^T \end{cases}$$

How to choose the best class?

$$\hat{k} = \arg\max_k \left(\text{posterior probability } p(\mathbf{f}|\mathbf{y}, \mu_k, \mathbf{\Sigma}_k)\right)$$

Summary: iterative procedure



Estimation Step:

1. Patches assigned to classes

2. Patches restored with chosen class (μ_k, Σ_k)

$$\hat{\mathrm{f}} = \mathrm{W}(\mathrm{y} - \mathrm{U}\mu_k) + \mu_k$$

Class Update Step:

$$\begin{cases} \tilde{\mu}_k = \dfrac{1}{|C_k|} \displaystyle\sum_{i \in C_k} \hat{\mathrm{f}}_i \\ \tilde{\Sigma}_k = \dfrac{1}{|C_k|} \displaystyle\sum_{i \in C_k} (\hat{\mathrm{f}}_i - \tilde{\mu}_k)(\hat{\mathrm{f}}_i - \tilde{\mu}_k)^T \end{cases}$$

2 to 3 iterations

Applied in sub-regions of size 128 x 128

Initialization

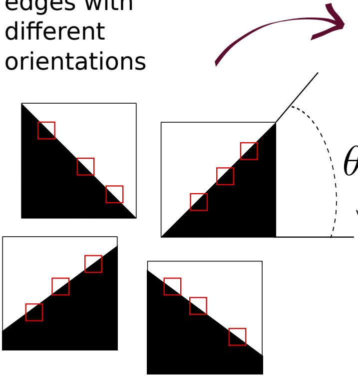
K classes to set

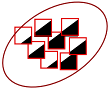
Initialization

K classes to set

(K - 1) classes

edges with
different
orientations





$$\begin{cases} \tilde{\mu}_k = 0 \\ \tilde{\Sigma}_k = \frac{1}{|C_k|} \sum_{i \in C_k} (\hat{\mathrm{f}}_i - \tilde{\mu}_k)(\hat{\mathrm{f}}_i - \tilde{\mu}_k)^T \end{cases}$$

$$\theta_k = \frac{\pi k}{K-1}$$

$$\forall k = 1, \ldots, K-1$$

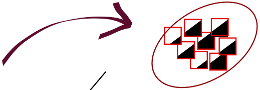
Initialization

K classes to set

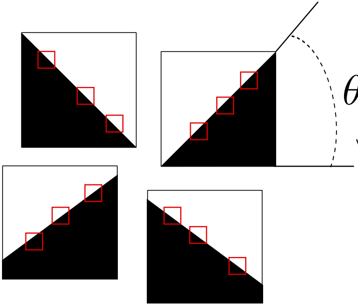
(K - 1) classes

K = 20
patch size = 8x8

edges with
different
orientations



$$\begin{cases} \tilde{\mu}_k = 0 \\ \tilde{\Sigma}_k = \frac{1}{|C_k|} \sum_{i \in C_k} (\hat{\mathrm{f}}_i - \tilde{\mu}_k)(\hat{\mathrm{f}}_i - \tilde{\mu}_k)^T \end{cases}$$



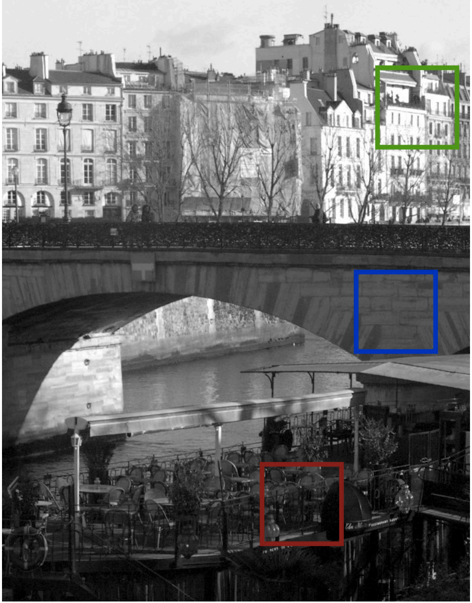
$$\theta_k = \frac{\pi k}{K-1}$$

$$\forall k = 1, \ldots, K-1$$

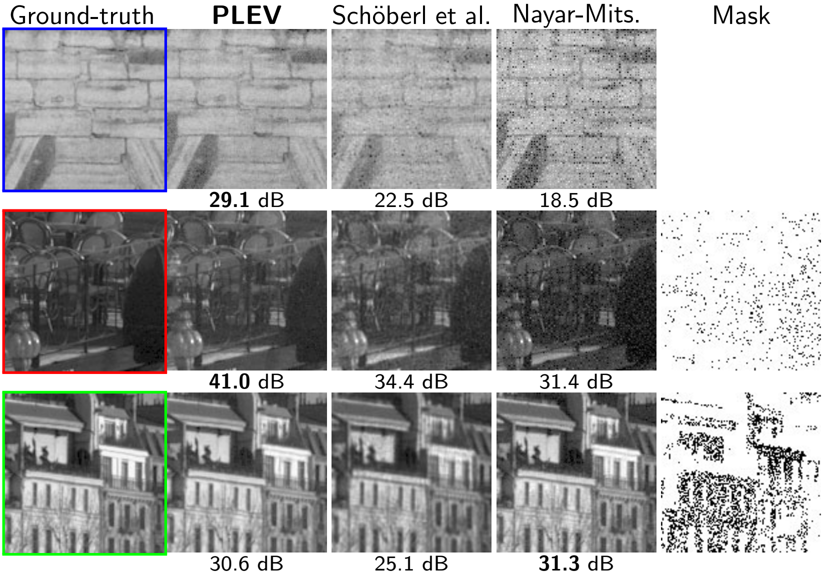
+

DCT
for isotropic
patterns

Results synthetic data



Results synthetic data



Improvement: Patch-based Bayesian restoration method (on-going work)

- Inspired from:

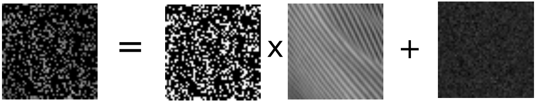
  Piecewise Linear Estimators (PLE) [Yu et al., 2012] High performance in interpolation of missing pixels.

  Non Local Bayes (NLB) [Lebrun et al.,2013] State-of-the-art denoising method.

- General restoration method.

Patch Reconstruction

$$\mathrm{y} = \mathrm{U} \times \mathrm{f} + \mathrm{n}$$



$\mathcal{N}(\mu, \mathrm{U}\Sigma\mathrm{U}^T + \Sigma_{\mathrm{n}}(\mathrm{f}))$ $\mathcal{N}(\mu, \Sigma)$ $\mathcal{N}(0, \Sigma_{\mathrm{n}}(\mathrm{f}))$

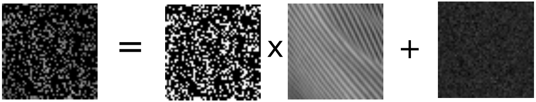
Patch reconstruction: $\boxed{\hat{\mathrm{f}} = \mathrm{W}(\mathrm{y} - \mathrm{U}\mu) + \mu}$

Wiener filter: $\mathrm{W} = \Sigma\,\mathrm{U}^T(\mathrm{U}\Sigma\,\mathrm{U}^T + \Sigma_{\mathrm{n}}(\mathrm{f}))^{-1}$

How to set Gaussian prior μ and Σ?

Patch Reconstruction

$$\mathrm{y} \qquad \mathrm{U} \qquad \mathrm{f} \qquad \mathrm{n}$$



$$\mathcal{N}(\mu, \mathrm{U}\Sigma\mathrm{U}^T + \Sigma_{\mathrm{n}}(\mathrm{f})) \qquad \mathcal{N}(\mu, \Sigma) \quad \mathcal{N}(0, \Sigma_{\mathrm{n}}(\mathrm{f}))$$

Patch reconstruction: $\boxed{\hat{\mathrm{f}} = \mathrm{W}(\mathrm{y} - \mathrm{U}\mu) + \mu}$

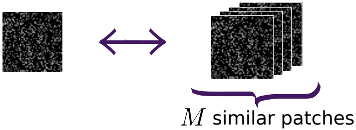
Wiener filter: $\mathrm{W} = \Sigma\,\mathrm{U}^T(\mathrm{U}\Sigma\,\mathrm{U}^T + \Sigma_{\mathrm{n}}(\mathrm{f}))^{-1}$

How to set Gaussian prior μ and Σ?

How to set Gaussian prior parameters μ and Σ?

Inspired by NLB denoising power:

- Estimate Gaussian parameters (μ, Σ) locally from similar patches.



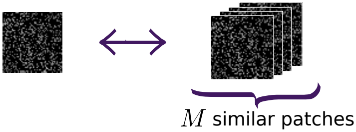
- Classical MLE formulas:

$$\tilde{\mu} = \frac{1}{M} \sum_{i=1}^{M} \tilde{\mathrm{f}}_i \qquad \tilde{\Sigma} = \frac{1}{M} \sum_{i=1}^{M} (\tilde{\mathrm{f}}_i - \tilde{\mu})(\tilde{\mathrm{f}}_i - \tilde{\mu})^T$$

How to set Gaussian prior parameters μ and Σ?

Inspired by NLB denoising power:

- Estimate Gaussian parameters (μ, Σ) locally from similar patches.



- Classical MLE formulas:

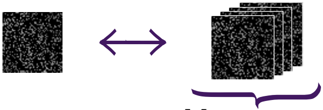
$$\tilde{\mu} = \cancel{\frac{1}{M}\sum_{i=1}^{M} \tilde{\mathrm{f}}_i} \qquad \tilde{\Sigma} = \frac{1}{M}\cancel{\sum_{i=1}^{M} (\tilde{\mathrm{f}}_i - \tilde{\mu})}(\tilde{\mathrm{f}}_i - \tilde{\mu})^T$$

MLE cannot be used due to missing pixels!

How to set Gaussian prior parameters μ and Σ?

Inspired by NLB denoising power:

- Estimate Gaussian parameters (μ, Σ) locally from similar patches.



M similar patches

- Classical MLE formulas:

$$\tilde{\mu} = \cancel{\frac{1}{M}\sum_{i=1}^{M} \tilde{\mathrm{f}}_i} \qquad\qquad \tilde{\Sigma} = \frac{1}{M}\cancel{\sum_{i=1}^{M} (\tilde{\mathrm{f}}_i - \tilde{\mu})}(\tilde{\mathrm{f}}_i - \tilde{\mu})^T$$

MLE cannot be used due to missing pixels!

$\boxed{\text{Proposed solution: Maximum a posteriori (MAP) with a prior on } (\mu, \Sigma)}$

MAP to compute Gaussian parameters μ and Σ

- Hyperprior on (μ, Σ) : Normal - Wishart distribution

- MAP:

$$(\hat{\mu}, \hat{\Sigma}) = \arg\max_{\mu,\Sigma} p(\mu, \Sigma | \mathrm{y}_1, \ldots, \mathrm{y}_\mathrm{M}) \quad ,$$

$$\arg\max_{\mu,\Sigma} \prod_{j=1}^{M} \underbrace{\mathcal{N}(\mathrm{U}\mu, \Sigma^*)|_{\mathrm{y}=\mathrm{y_j}}}_{\text{M similar patches}} \underbrace{\mathcal{N}(\mu|\mu_0, \Sigma/\kappa)\mathcal{W}(\Sigma|(\nu\Sigma_0)^{-1}, \nu)}_{\text{Hyperprior on model parameters}}$$

$\rightarrow$ Inclusion of hyperprior information compensates for missing pixels.

Iterative approach

Model parameters estimation Step:

M similar patches $\rightarrow$ MAP $\rightarrow$ $(\hat{\mu}, \hat{\Sigma})$

Hyperprior on (μ, Σ) $\rightarrow$ MAP

Restoration Step:

$$\hat{\mathrm{f}} = \mathrm{W}(\mathrm{y} - \mathrm{U}\mu) + \mu$$

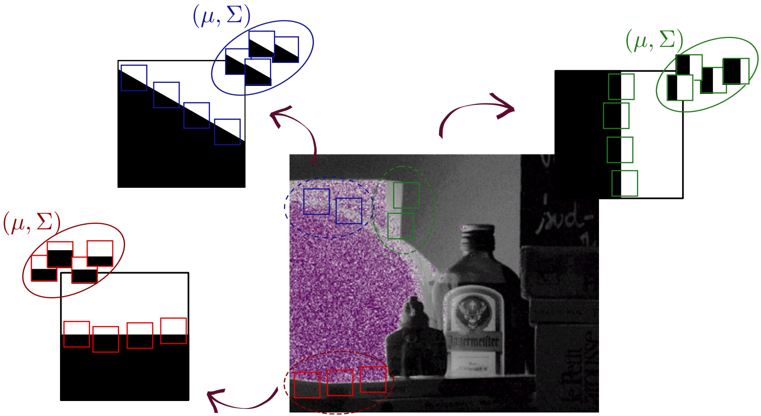
$$\mathrm{W} = \Sigma\,\mathrm{U}^T(\mathrm{U}\Sigma\,\mathrm{U}^T + \Sigma_{\mathrm{n}}(\mathrm{f}))^{-1}$$

3 to 4 iterations

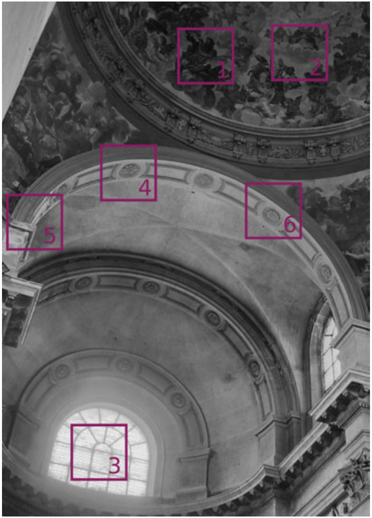
Initialization

From PLE [Yu et al., 2012]:

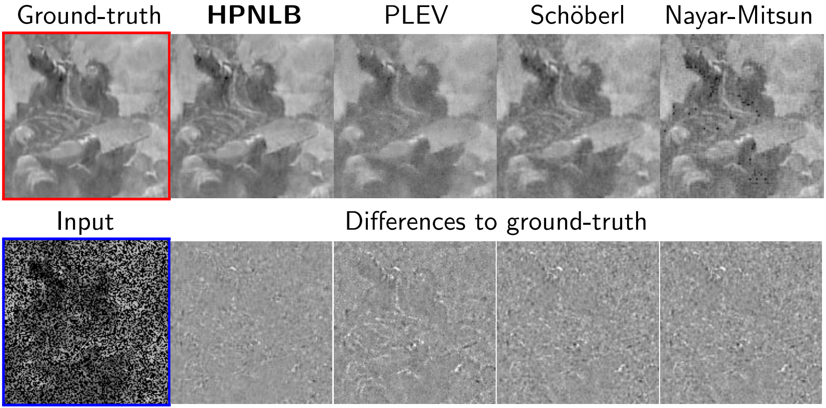
K predefined models : (K-1) edges with different orientations + DCT for isotropic patterns



Results HDR - Synthetic data

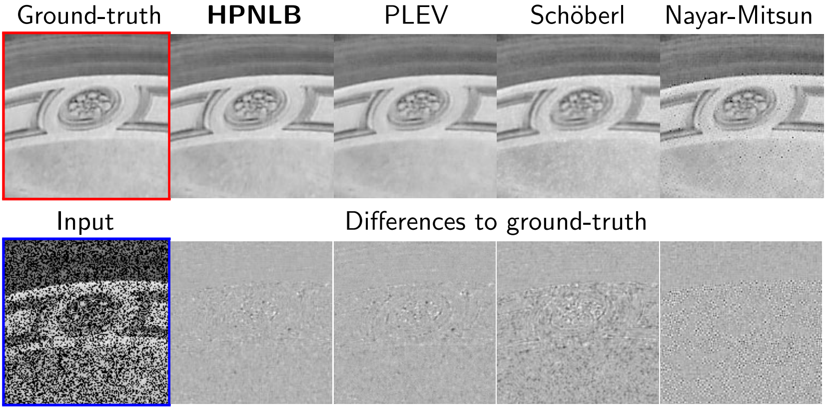


Results HDR - Synthetic data



PSNR:	**33.1dB**	29.7dB	30.4dB	29.4dB

Results HDR - Synthetic data



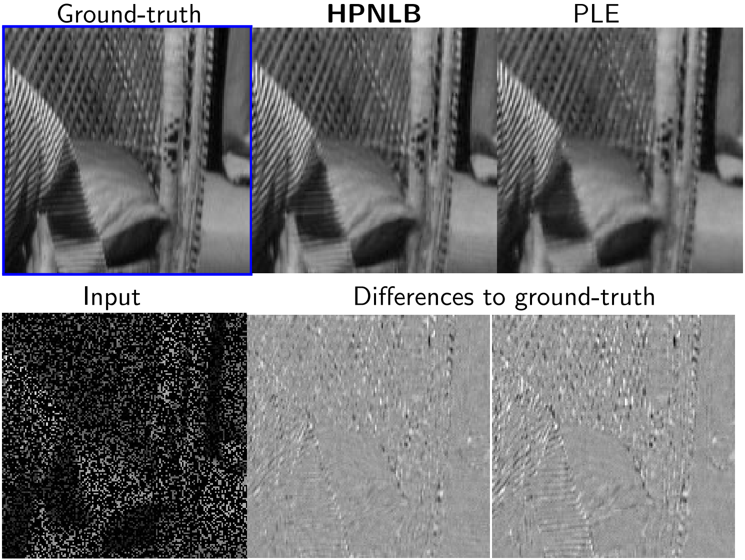
PSNR:	**35.1dB**	34.0dB	30.0dB	28.5dB

Results on other applications

70% missing pixels + additive Gaussian noise variance 5%

Results on other applications



PSNR:	**30.5dB**	28.6dB

Conclusions

- Exemplar-based patch regularization: early self-similarity model
- GMM, PLE: Extension to more inverse problems
- Local Gaussian Models: finer details, continuous classification

Challenges ahead for local Gaussian models

- Invert non-diagonal operators
- Robust neighbors in ill-posed problems
- Formal framework needed
- More flexible learning/indexing over large databases

Thanks. Questions?