IMA206 Course

Patch based approaches for image processing

Florence TUPIN (LTCI, Télécom Paris, Institut Polytechnique de Paris)

May, 2022

Outline

Outline

What do we denote by “image patches”?

Definition [Oxford dictionary]

patch (noun)

A small area or amount of something

Image patches

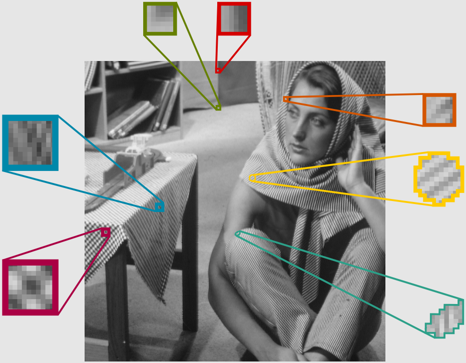
Sub-regions of the image

- shape: typically rectangular
- size: much smaller than image size

$\rightarrow$ most common use:
square regions between
5×5 and 21×21 pixels

$\rightarrow$ tradeoff:
size $\nearrow$ $\Rightarrow$ more distinctive/informative
size $\searrow$ $\Rightarrow$ more likely to find similar patches

non-rectangular / deforming shapes:
computationally complexity $\nearrow$



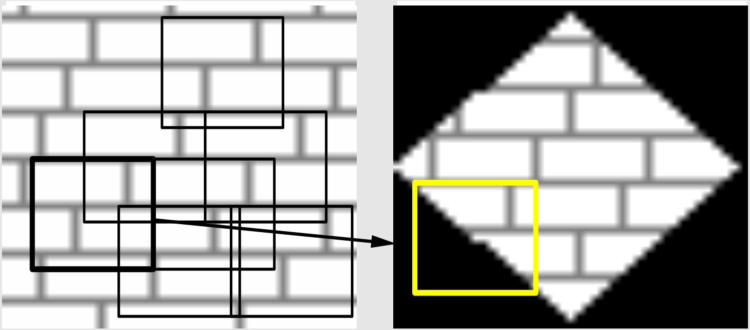
$\rightarrow$ patches capture *local context*: geometry and texture

Origins of patch-based image processing

3 success stories of patch-based models at the origin of these methods

Starting points of patch-based methods

- model for human vision (primary visual cortex)
  Theoretical and experimental works on the primary visual cortex have shed new light on the importance of patch-level image coding
- method to synthetize textures
  Examplar-based synthesis method by Efros and Leung



Efros et al., ICCV, 1999

- method to denoise images

Overview of denoising methods

Main approaches

- Linear filtering

(a) Linear filtering

Overview of denoising methods

Main approaches

- Linear filtering
- Anisotropic diffusion Perona et Malik, 1990

(a) Linear filtering

(b) Anisotropic diffusion

Overview of denoising methods

Main approaches

- Linear filtering
- Anisotropic diffusion Perona et Malik, 1990
- Prior modeling of images and energy minimization (MRF, TV,...) Rudin et al., 1992

(a) Linear filtering

(b) Anisotropic diffusion

(c) TV

Overview of denoising methods

Main approaches

- Linear filtering
- Anisotropic diffusion Perona et Malik, 1990
- Prior modeling of images and energy minimization (MRF, TV,...) Rudin et al., 1992
- Wavelet approaches Donoho et al., 1994

(a) Linear filtering

(b) Anisotropic diffusion

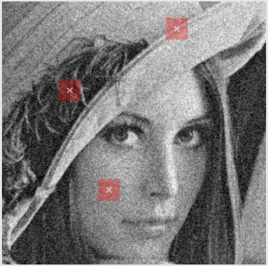
(c) TV

(d) Wavelets

Overview of denoising methods

Common ideas

- Averaging pixels sharing the same information
- Where finding them ?

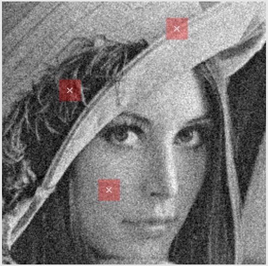


(a) Linear filtering

Overview of denoising methods

Common ideas

- Averaging pixels sharing the same information
- Where finding them ?

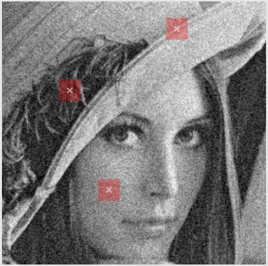


(a) Linear filtering

(b) Anisotropic diffusion

Common ideas

- Averaging pixels sharing the same information
- Where finding them ?



(a) Linear filtering

(b) Anisotropic diffusion

(c) Oracle

- Oracle : anywhere in the image as soon as the pixels share the same un-noisy value!

Overview of denoising methods

Common ideas

- Averaging pixels sharing the same information
- Where finding them ?

(a) Linear filtering (b) Anisotropic diffusion (c) Oracle

- Oracle : anywhere in the image as soon as the pixels share the same un-noisy value!

$\rightarrow$ non-local means

Selection-based filtering

$u(x)$ "true" value of pixel x
$v(x)$ noisy value (observed) of pixel x
Goal: finding the "best" $\hat{u}(x)$

Variance reduction

- If $X_1, ..., X_N$ are N i.i.d samples of mean $\boldsymbol{\mu}$ and standard deviation σ, their average has a standard deviation of $\frac{\sigma}{\sqrt{N}}$
- local linear filtering

$$\hat{u}(x) = \sum_y w(x,y)v(y)$$

averaging samples spatially close to the pixel x, $w(x,y) = k\exp(-\frac{\text{dist}^2(x,y)}{2h^2})$
- improving local linear filtering: taking gray (color) level into account

$$w(x,y) = k\exp(-\frac{\text{dissi}(x,y)}{2h'^2})$$

averaging samples radiometrically close to the pixel (if dissi(x,y) is high, $w(x,y)$ is small) [Yaroslavski 84]

Selection-based filtering

$u(x)$ "true" value of pixel x
$v(x)$ noisy value (observed) of pixel x
Goal: finding the "best" $\hat{u}(x)$

Variance reduction

- If $X_1, ..., X_N$ are N i.i.d samples of mean $\boldsymbol{\mu}$ and standard deviation σ, their average has a standard deviation of $\frac{\sigma}{N}$
- local linear filtering

$$\hat{u}(x) = \sum_y w(x,y)v(y)$$

  averaging samples spatially close to the pixel x $w(x,y) = k\exp(-\frac{\text{dist}(x,y)}{2h^2})$
- improving local linear filtering: taking gray (color) level into account

$$w(x,y) = k\exp(-\frac{\text{dissi}(x,y)}{2h'^2})$$

  averaging samples radiometrically close to the pixel (if dissi(x,y) is high, $w(x,y)$ is small) [Yaroslavski 84]

$\Rightarrow$ If the noise level is high dissi(x,y) is difficult to compute
$\Rightarrow$ Use patches to compute it !

Outline

Non-local means - Principle

- Local filter: in each pixel x, average the noisy values $v(y)$ of the pixels y in x neignborhood.

Non-local means - Principle

- Local filter: in each pixel x, average the noisy values $v(y)$ of the pixels y in x neignborhood.
- Non-local: All pixels y values are used to do the denoising, with a weight reflecting the color or radiometric similarity of y with x:

$$\hat{u}(x) = \sum_{y} \qquad v(y)$$

Non-local means - Principle

- Local filter: in each pixel x, average the noisy values $v(y)$ of the pixels y in x neignborhood.
- Non-local: All pixels y values are used to do the denoising, with a weight reflecting the color or radiometric similarity of y with x:

$$\hat{u}(x) = \sum_{y} w(x,y)v(y)$$

- Local filter: in each pixel x, average the noisy values $v(y)$ of the pixels y in x neignborhood.
- Non-local: All pixels y values are used to do the denoising, with a weight reflecting the color or radiometric similarity of y with x:

$$\hat{u}(x) = \sum_{y} w(x,y)v(y)$$

- weight depends on the dissimilarity between x and y:

$$w(x,y) = \quad \text{dissi}_{(x,y)}$$

- Local filter: in each pixel x, average the noisy values $v(y)$ of the pixels y in x neignborhood.
- Non-local: All pixels y values are used to do the denoising, with a weight reflecting the color or radiometric similarity of y with x:

$$\hat{u}(x) = \sum_{y} w(x,y)v(y)$$

- weight depends on the dissimilarity between x and y:

$$w(x,y) = e^{-\text{dissi}(x,y)}$$

Non-local means - Principle

- Local filter: in each pixel x, average the noisy values $v(y)$ of the pixels y in x neignborhood.
- Non-local: All pixels y values are used to do the denoising, with a weight reflecting the color or radiometric similarity of y with x:

$$\hat{u}(x) = \sum_y w(x,y)v(y)$$

- weight depends on the dissimilarity between x and y:

$$w(x,y) = e^{-\frac{\mathsf{dissi}_{(x,y)}}{2h^2}}$$

Non-local means - Principle

- Local filter: in each pixel x, average the noisy values $v(y)$ of the pixels y in x neignborhood.
- Non-local: All pixels y values are used to do the denoising, with a weight reflecting the color or radiometric similarity of y with x:

$$\hat{u}(x) = \sum_y w(x,y)v(y)$$

- weight depends on the dissimilarity between x and y:

$$w(x,y) = \frac{e^{-\frac{\text{dissi}_{(x,y)}}{2h^2}}}{\sum_z e^{-\frac{\text{dissi}_{(x,z)}}{2h^2}}}$$

Non-local means - Principle

- Local filter: in each pixel x, average the noisy values $v(y)$ of the pixels y in x neignborhood.
- Non-local: All pixels y values are used to do the denoising, with a weight reflecting the color or radiometric similarity of y with x:

$$\hat{u}(x) = \sum_y w(x,y)v(y)$$

- weight depends on the dissimilarity between x and y:

$$w(x,y) = \frac{e^{-\frac{\text{dissi}_{(x,y)}}{2h^2}}}{\sum_z e^{-\frac{\text{dissi}_{(x,z)}}{2h^2}}}$$

- weight depends on the dissimilarity betwwen patches around x and y

$$\text{dissi}(x,y) = \frac{1}{s^2}\|V(x) - V(y)\|^2 \triangleq \frac{1}{s^2}\sum_\delta (V(x+\delta) - V(y+\delta))^2$$

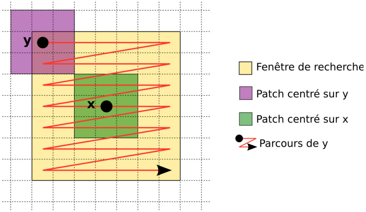
where V is the vector of all the values in the patch and s^2 is the size of the patch.

Non-local means - Algorithm in practice

- 3 loops:
  1. Go through all the pixels x

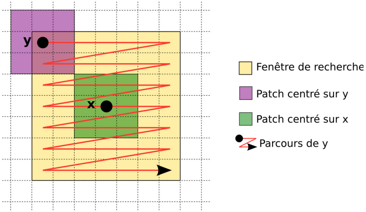
Non-local means - Algorithm in practice

- 3 loops:
  1. Go through all the pixels x
  2. Compare the patches centered on x and y to compute the weighted mean (in practice the y pixels are taken in a search window centered on x)

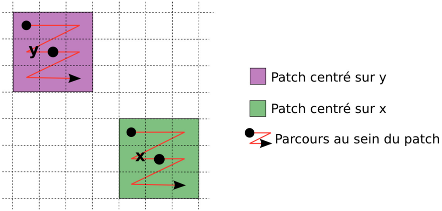


Non-local means - Algorithm in practice

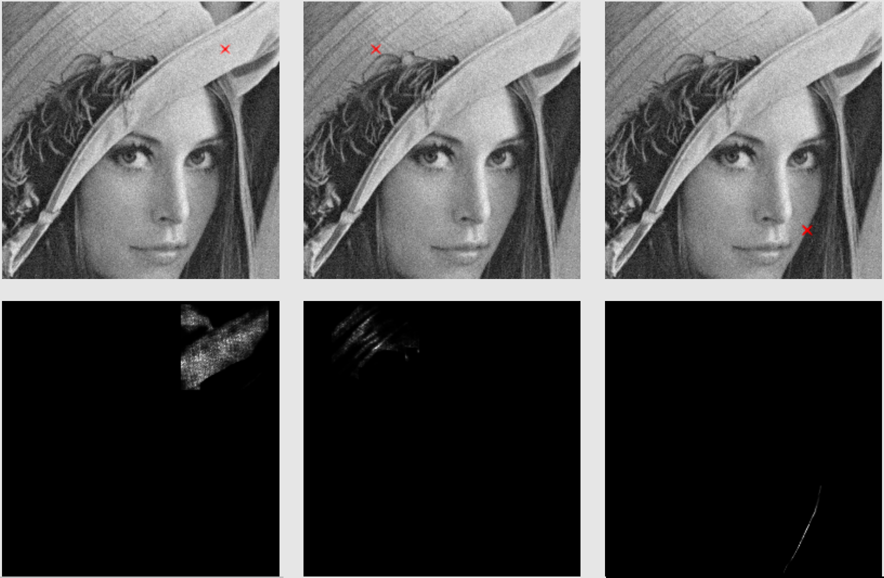
- 3 loops:
  1. Go through all the pixels x
  2. Compare the patches centered on x and y to compute the weighted mean (in practice the y pixels are taken in a search window centered on x)



  3. The dissimilarity between patches (euclidean distance between the vectors of pixel values) represents the dissimilarity between all the pixels of the patches taken 2 by 2 (quadratic sum of their differences).



Map of weights



Non-local means - Map of weights

Map of weights

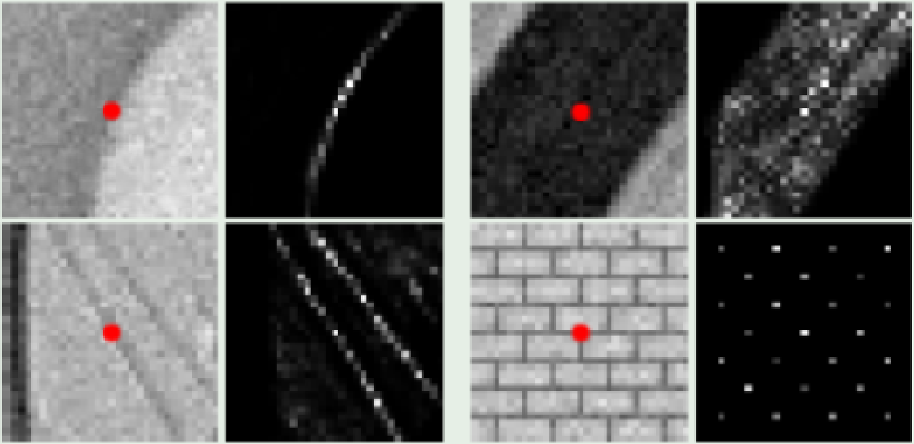
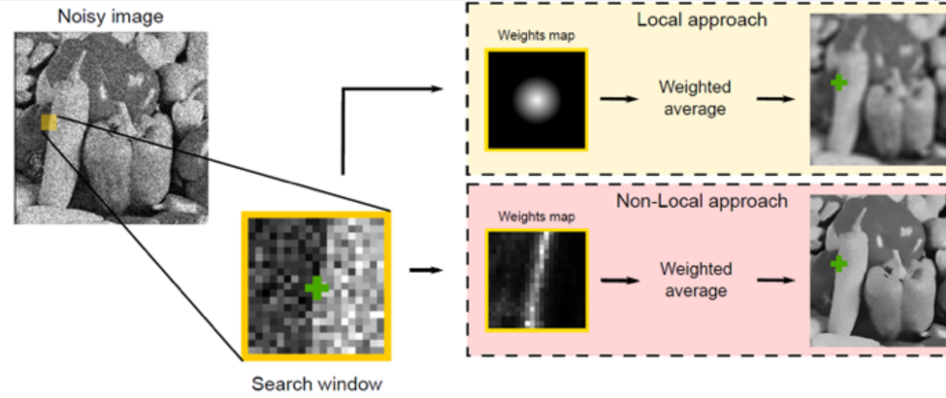


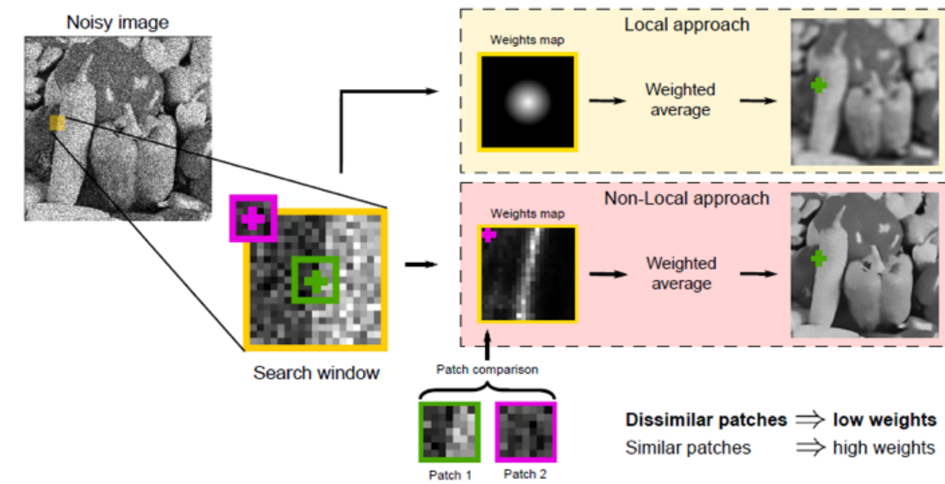
image extracted from [Buades et al., 2005]

Non-local means - Illustration

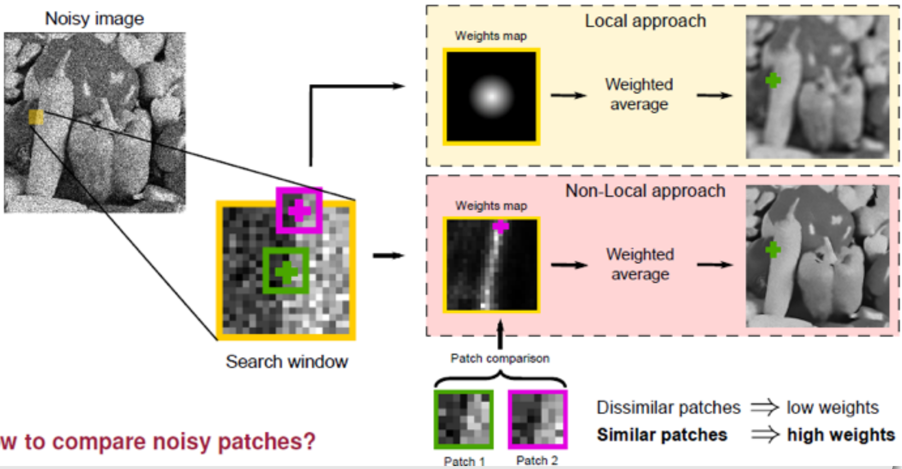
Local / non-local



Local / non-local



Local / non-local



How to compare noisy patches?

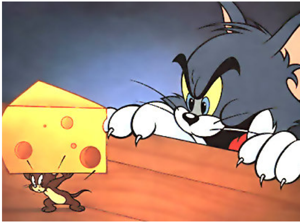
Non-local means - Illustration

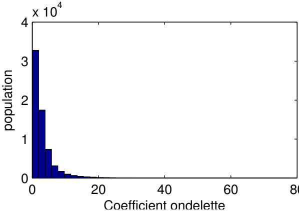
NL-means denoising


Denoising

Ill-posed problem: hypotheses have to be done

- On the kind of signal to denoise:
  - Constant / smooth
  - bounded variation / piecewise constant
  - sparcity in a wavelet basis.

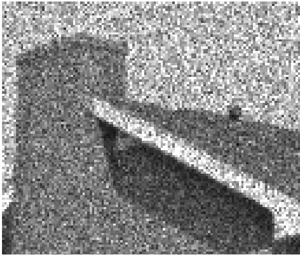




Denoising

Ill-posed problem: hypotheses have to be done

- On the kind of signal to denoise:
  - Constant / smooth
  - bounded variation / piecewise constant
  - sparcity in a wavelet basis.
- On the kind of noise:
  - additive / multiplicative / impulsive...
  - white / colored



Denoising

Ill-posed problem: hypotheses have to be done

- On the kind of signal to denoise:
  - Constant / smooth
  - bounded variation / piecewise constant
  - sparcity in a wavelet basis.
- On the kind of noise:
  - additive / multiplicative / impulsive...
  - white / colored

There is no denoising without hypotheses

Denoising

Ill-posed problem: hypotheses have to be done

- On the kind of signal to denoise:
  - Constant / smooth
  - bounded variation / piecewise constant
  - sparcity in a wavelet basis.
- On the kind of noise:
  - additive / multiplicative / impulsive...
  - white / colored

There is no denoising without hypotheses

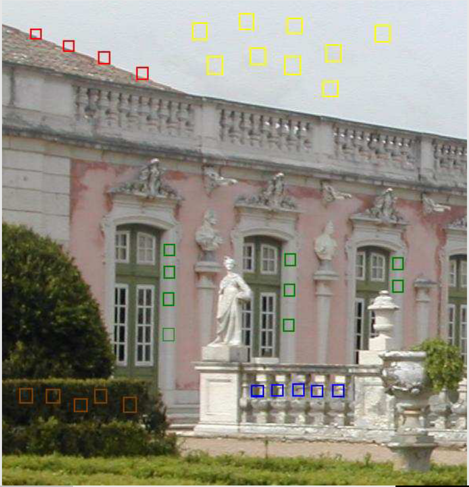
Hypotheses of NLmeans:

1. Similar patches have similar central values.
2. There are similar patches in the image (self-similarity = redundancy).
3. The noise is additive Gaussian and white.

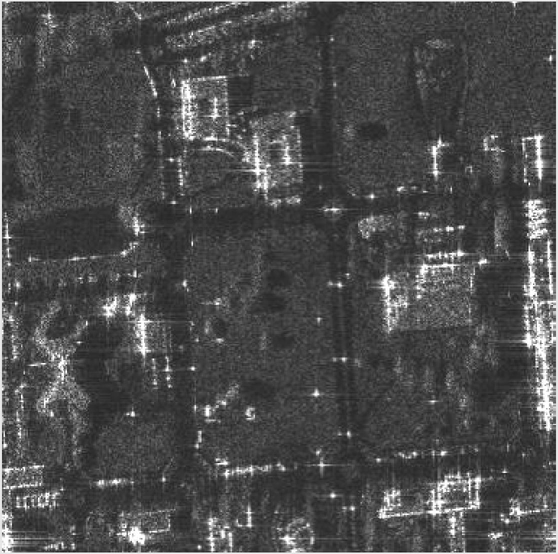
Non-local means - Hypotheses

Main hypotheses

- (H1) Redundancy: there are many similar patches in an image
- (H2) If the noisy patches are similar, their central values are similar

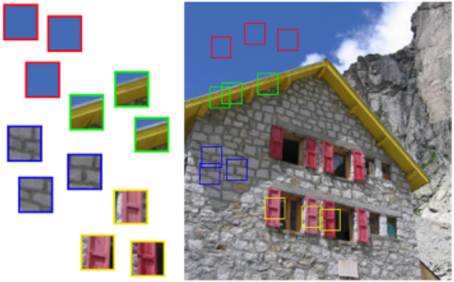


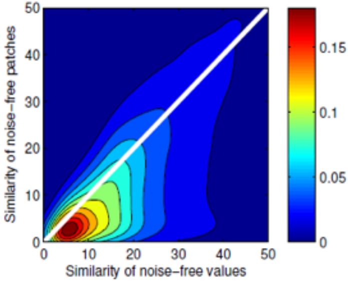
(H1) Redundancy ?



Non-local means - Hypotheses

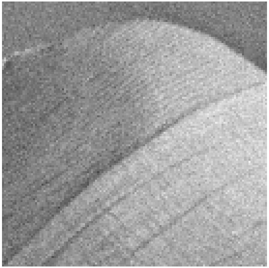
(H2) Central values vs patch similarity ?



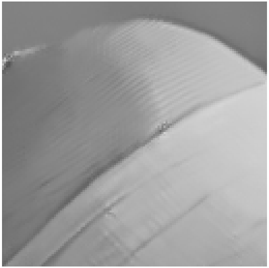


Non-local means - limits

1. Low contrasted textures and details



Noisy image ($\sigma = 10$)



Restored image

Non-local means - limits

1. Low contrasted textures and details
2. Contrasted rare patches

Noisy image ($\sigma = 10$)

Restored image

Non-local means - limits

1. Low contrasted textures and details
2. Contrasted rare patches
3. Non gaussian noise

Salt and pepper noise

Restored image

Non-local means - limits

1. Low contrasted textures and details
2. Contrasted rare patches
3. Non gaussian noise
4. Time computation

Non-local means - limits

1. Low contrasted textures and details
2. Contrasted rare patches
3. Non gaussian noise
4. Time computation
5. Parameter choice

Bias-Variance decomposition

- Case of a white gaussian noise $\mathcal{N}(0, \sigma^2)$.
- If u is the original image and v the noisy image (NLu and NLv their non local versions), we have:

$$\boldsymbol{E}|NLv(x) - u(x)|^2 = \underbrace{\boldsymbol{E}|NLv(x) - NLu(x)|^2}_{\text{"variance"}} + \underbrace{\boldsymbol{E}|NLu(x) - u(x)|^2}_{\text{"bias"}} + 2\,\underbrace{\boldsymbol{E}\left((NLv - NLu(x))(NLu(x) - u(x))\right)}_{\approx 0}.$$

Variance term

$$\boldsymbol{E}|NLv(x) - NLu(x)|^2 = \boldsymbol{E}|\sum_y w(x,y)n(y)|^2 = \sigma^2 \sum_y (w(x,y))^2$$

Minimal when $w(x,y) = \frac{1}{\text{card}(W)}$ uniform mean on the whole image ($h \to +\infty$)

Bias term

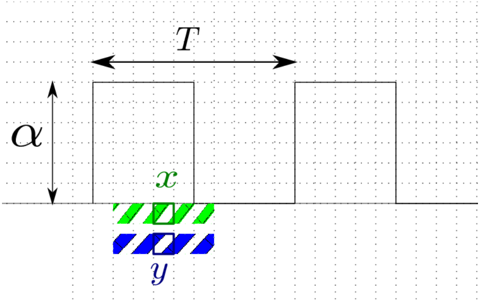
$$\boldsymbol{E}|NLu(x) - u(x)|^2 = |\sum_y w(x,y)(u(y) - u(x))|^2$$

Minimal when $w(x,y) = 1$ for $u(x) = u(y)$ and 0 elsewhere.

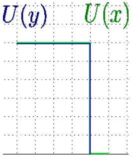
Bias / variance compromise

Variance reduction is ensured by a high value of h (tolerant selection) whereas bias limitation needs a small h (strict selection).

A toy-example in 1-dimension



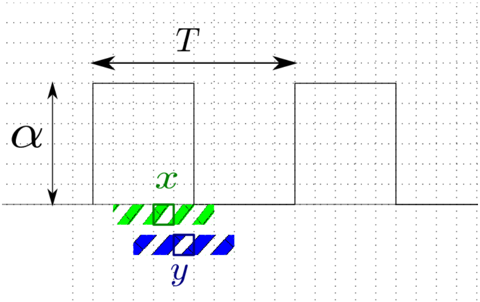
offset 0px



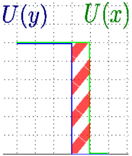
Values of the patches

Distance between patches: $\|U(x) - U(x)\|^2 = 0$

A toy-example in 1-dimension



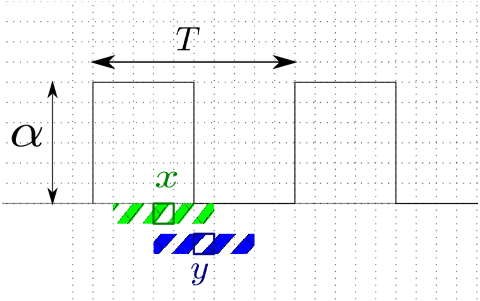
offset 1px



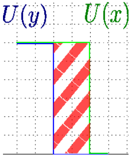
Values of the patches

Distance between patches: $\|U(x) - U(x+1)\|^2 = \frac{\alpha^2}{s}$

A toy-example in 1-dimension



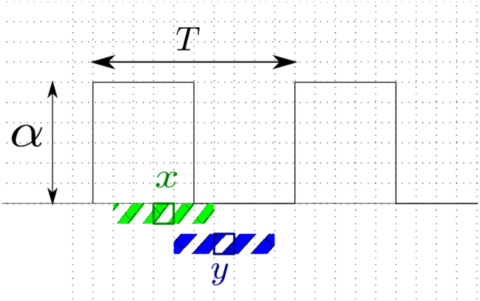
offset 2px



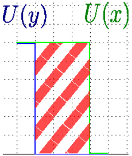
Values of the patches

Distance between patches: $\|U(x) - U(x+2)\|^2 = \frac{2\alpha^2}{s}$

A toy-example in 1-dimension



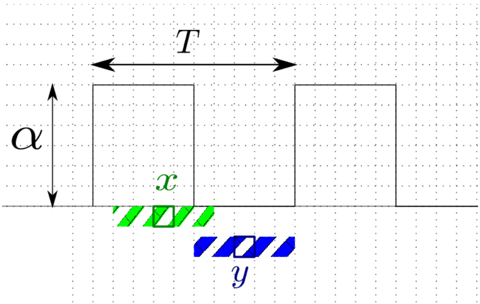
offset 3px



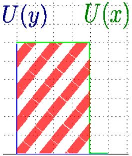
Values of the patches

Distance between patches: $\|U(x) - U(x+3)\|^2 = \frac{3\alpha^2}{s}$

A toy-example in 1-dimension



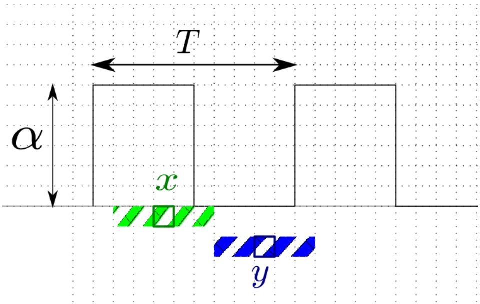
offset 4px



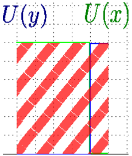
Values of the patches

Distance between patches: $\|U(x) - U(x+4)\|^2 = \frac{4\alpha^2}{s}$

A toy-example in 1-dimension



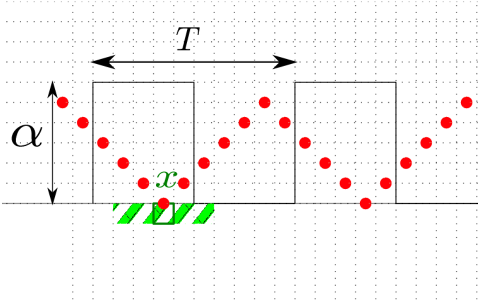
offset 5px



Values of the patches

Distance between patches: $\|U(x) - U(x+5)\|^2 = \frac{5\alpha^2}{s}$

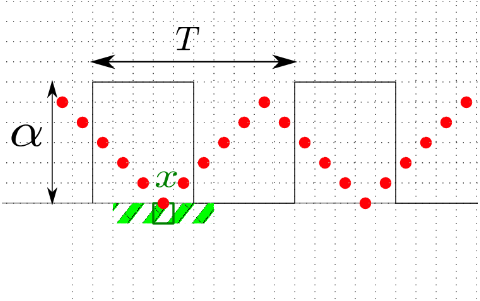
Example: periodic texture (step)



Distances to $U(x)$

Distance between patches: $\|U(x) - U(x+j)\|^2 = \frac{|j|\alpha^2}{s}$

Example: periodic texture (step)

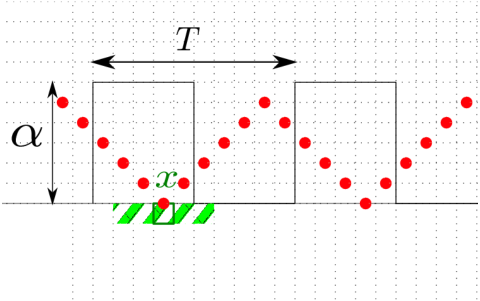


Distances to $U(x)$

Therefore:

$$NLu(x) = \frac{\sum_{-\frac{T}{2}<j\leq\frac{T}{2}} e^{-\frac{\|U(x)-U(x+j)\|^2}{2h^2}} u(x+j)}{\sum_{-\frac{T}{2}<j\leq\frac{T}{2}} e^{-\frac{\|U(x)-U(x+j)\|^2}{2h^2}}}$$

Example: periodic texture (step)

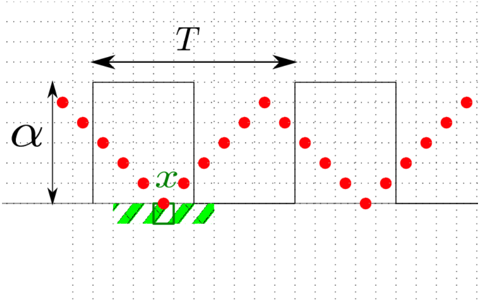


Distances to $U(x)$

Therefore:

$$NLu(x) = \frac{\alpha\left(\sum_{j=0}^{j_1} e^{-rj} - 1 + \sum_{j=0}^{j_2} e^{-rj}\right) + 0}{2\sum_{j=0}^{\frac{T}{2}-1} e^{-rj} - 1 + e^{-r\frac{T}{2}}}$$

Example: periodic texture (step)

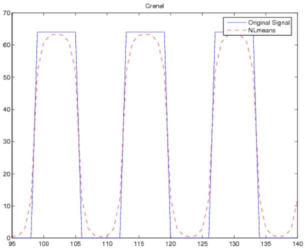


Distances to $U(x)$

Therefore:

$$NLu(x) = \frac{\alpha}{(1 - e^{-r\frac{T}{2}})(1 + e^{-r})} \left(1 - e^{-r} - 2e^{-\frac{1}{2}(\frac{T}{2}+1)r} \cosh rx\right)$$

Example: periodic texture (step)

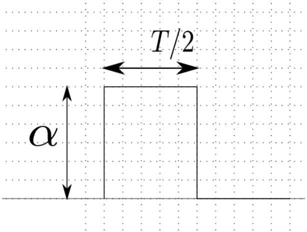


$$NLu(x) = \frac{\alpha}{(1-e^{-r\frac{T}{2}})(1+e^{-r})}\left(1 - e^{-r} - 2e^{-\frac{1}{2}(\frac{T}{2}+1)r}\cosh rx\right)$$

with $r = \frac{2}{s}\frac{\alpha^2}{h^2}$.
Comments:

- Even perfectly periodic signals are modified !
- Non-linear filter: r depends on α
- "checking" : if $h \to +\infty$, $NLu(x) \sim \frac{\alpha}{2}$ (uniformy gray image)

Example: isolated step



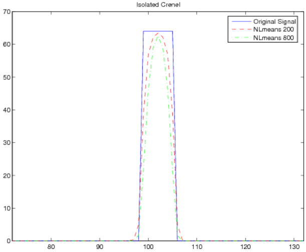
In the same way:

$$NLu(x) = \alpha \frac{1 - e^{-r} - 2e^{-\frac{1}{2}(\frac{T}{2}+1)r} \cosh rx}{(1 - e^{-r}) \left(2 \sum_{j=0}^{\frac{T}{2}} e^{-rj} - 1 + (N - T - 1) e^{-r\frac{T}{2}} \right)}.$$

with $r = \frac{2}{s} \frac{\alpha^2}{h^2}$.
Remarques:

- The result depends on the size N of the image / the size W of the search window.
- Weights of the background pixel are $e^{-r\frac{T}{2}} = e^{-\frac{T\alpha^2}{sh^2}}$. When s is large, they have an increased influence.

Example: isolated step



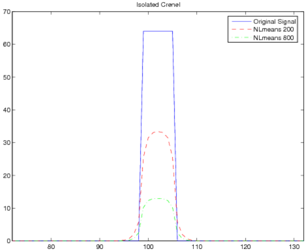
In the same way:

$$NLu(x) = \alpha \frac{1 - e^{-r} - 2e^{-\frac{1}{2}(\frac{T}{2}+1)r}\cosh rx}{(1-e^{-r})\left(2\sum_{j=0}^{\frac{T}{2}} e^{-rj} - 1 + (N-T-1)e^{-r\frac{T}{2}}\right)}.$$

with $r = \frac{2}{s}\frac{\alpha^2}{h^2}$.
Remarques:

- The result depends on the size N of the image / the size W of the search window.
- Weights of the background pixel are $e^{-r\frac{T}{2}} = e^{-\frac{T\alpha^2}{sh^2}}$. When s is large, they have an increased influence.

Example: isolated step



In the same way:

$$NLu(x) = \alpha \frac{1 - e^{-r} - 2e^{-\frac{1}{2}(\frac{T}{2}+1)r}\cosh rx}{(1-e^{-r})\left(2\sum_{j=0}^{\frac{T}{2}} e^{-rj} - 1 + (N-T-1)e^{-r\frac{T}{2}}\right)}.$$

with $r = \frac{2}{s}\frac{\alpha^2}{h^2}$.
Remarques:

- The result depends on the size N of the image / the size W of the search window.
- Weights of the background pixel are $e^{-r\frac{T}{2}} = e^{-\frac{T\alpha^2}{sh^2}}$. When s is large, they have an increased influence.

Example: loss of details

Noisy image

Example: loss of details

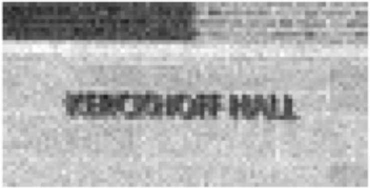
Search window $W = 11 \times 11$

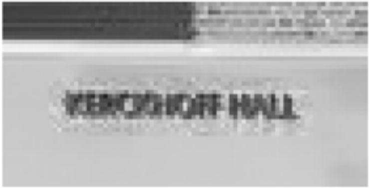
Search window $W = 61 \times 61$

- Details are lost when the search window W is too big.
- This effect increases with s.

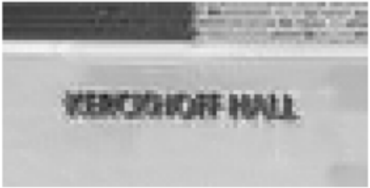
Example: influence of parameters

Influence of patch size





Patch 9×9



Patch 5×5

Example: influence of parameters

Influence of patch size

Patch 3×3

Patch 5×5

Parameters and influence

Comment on bias

Problems

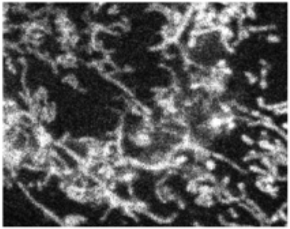
- Even the estimation of periodic signals is biased.
- The size of the search window W has a strong impact: it should not be too large !...
- Weakly contrasted details are erased.
- An area is more strongly attenuated if it is "rare" in the image (infuence of the background pixels).

Diagnostic

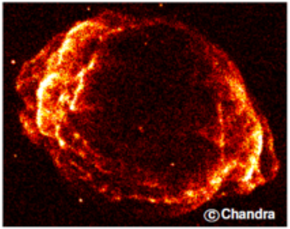
- A patch size too large makes more similar fine details and background.
- A patch size too small keeps noise fluctuations.
- Un-matching pixels may have a low weight but it is non zero because of the gaussian kernel. Their number increases with the search window W.

$\rightarrow$ the strength of non-local means is the patch not the non-locality !

Outline



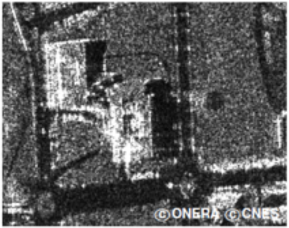
(a) Mitochondrion in microscopy



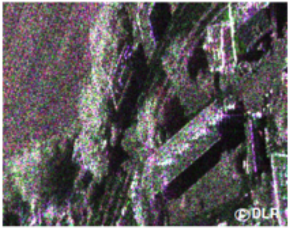
(b) Supernova in X-ray imagery

(c) Fetus using ultrasound imagery

(d) Plane wreckage in SONAR imagery

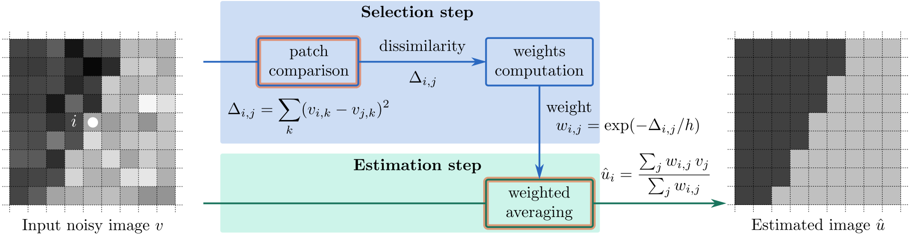


(e) Urban area using SAR imagery



(f) Polarimetric SAR imagery

Patch-based denoising – Selection-based filtering



General idea

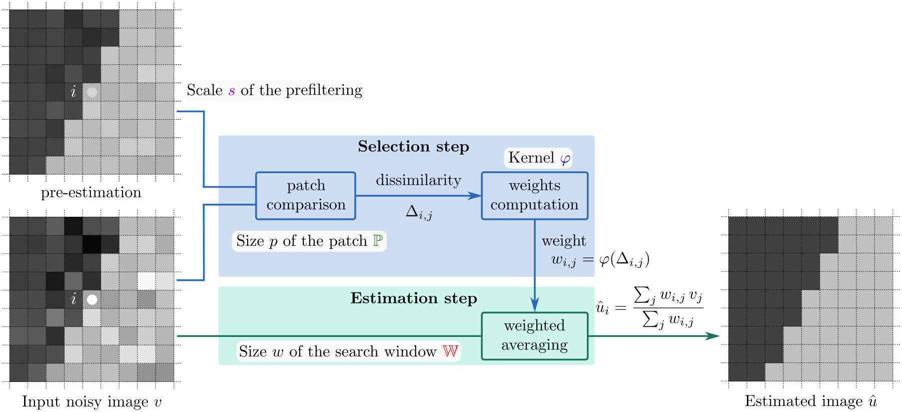
Goal: estimate the image u from the noisy image v

- Choose a pixel i to denoise
  - Inspect the pixels j around the pixel of interest i
  - Select the suitable candidates j
  - Average their values and update the value of i
- Repeat for all pixels i

2 key-steps:

- Computation of patch similarity
- Estimation step

Patch-based denoising – Selection-based filtering



Key parameters:

- Patch size
- Search window
- Kernel to convert similarity to weight (up to now Gaussian kernel)
- Pre-filtering step (preliminary filtering to improve the patch comparison)

Improvements of the nl-means method:

- Extension to different noise models
- Iterative approaches
- Automatic setting of parameters
- (Patch shapes)
- Block of patches

Noise models and estimation step

We suppose that a noise model is available: $p(v|u)$ is known (white noise here, v noisy value, u "true" value)

Estimation step

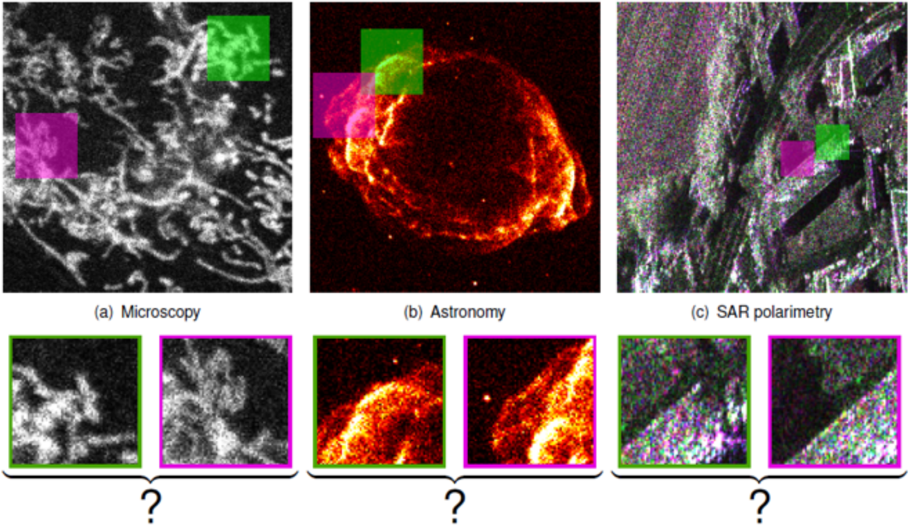
- Weighted sample mean
- Weighted maximum likelihood estimator (WMLE)
- Linear Minimum Mean Square Error estimator (LMMSE) (after wavelet transform and a first estimation step)

Estimation step: example of Gaussian or Gamma distributed data with WMLE

$$\hat{u}_i = \arg\max_{u_i} \left\{ \sum_j w_{i,j} \log p(v_j|u_i) \right\} = \frac{\sum_j w_{i,j} v_j}{\sum_j w_{i,j}}$$

Maximum likelihood estimate

Noise adaptation

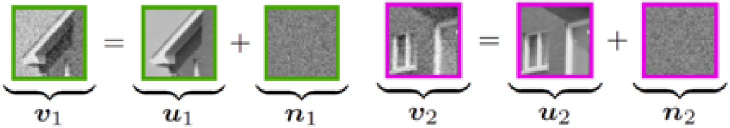


How to take into account the noise model?

Weight and patch similarity: how to compare noisy patches?

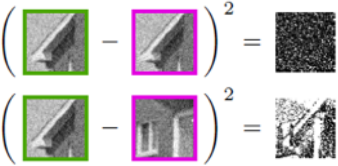
Buades et al.

- Euclidean distance between patches
- Additive White Gaussian noise implicit assumption



when $u_1 = u_2$: is low $\Rightarrow$ decide “similar”

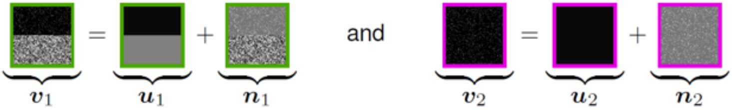
when $u_1 \neq u_2$: is high $\Rightarrow$ decide “dissimilar”



Weight and patch similarity: how to compare noisy patches?

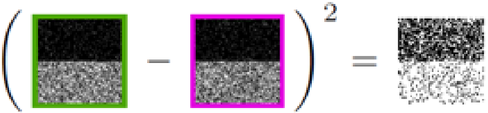
Other noise models

- Example: signal dependent noise
- Bad behaviour of the euclidean distance

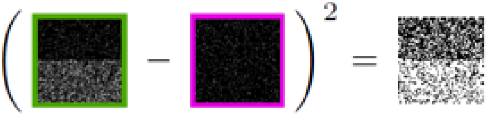


$\underbrace{\quad}_{v_1} = \underbrace{\quad}_{u_1} + \underbrace{\quad}_{n_1}$ and $\underbrace{\quad}_{v_2} = \underbrace{\quad}_{u_2} + \underbrace{\quad}_{n_2}$

when $u_1 = u_2$:



when $u_1 \neq u_2$:



Weight and patch similarity: how to compare noisy patches?

Nl-means and AWGN

- Left : noisy image
- Middle: restored image with oracle-based patch weights (patch comparison is done using the un-noisy image)
- Right: restored image with noisy-based patch weights (patch comparison is done using the noisy image)

Weight and patch similarity: how to compare noisy patches?

Nl-means and signal dependent noise

- Left : noisy image (multiplicative noise)
- Middle: restored image with oracle-based patch weights (patch comparison is done using the un-noisy image)
- Right: restored image with noisy-based patch weights (patch comparison is done using the noisy image)

Weight and patch similarity: how to compare noisy patches?

Taking into account the noise distribution

- When comparing two patches, all pixel values are compared two by two
- So the problem boils down to the comparison of v_1 and v_2 (noisy values)
- Idea: replacing the distance by an hypothesis test :

$$\mathcal{H}_0 : u_1 = u_2 = u_{12}$$
$$\mathcal{H}_1 : u_1 \neq u_2$$

- Performances measured by
  - False alarm rate: deciding "dissimilar" under $\mathcal{H}_0$
  - Detection rate: deciding "dissimilar" under $\mathcal{H}_1$
- Likelihood ratio test :

$$L(v_1, v_2) = \frac{p(v_1, v_2 | \mathcal{H}_0, u_{12})}{p(v_1, v_2 | \mathcal{H}_1, u_1, u_2)}$$

Weight and patch similarity: how to compare noisy patches?

Taking into account the noise distribution

- To compute the Likelihood Ratio Test, the true values u_1 and u_2 should be known
- Since they are unknown, they are replaced by their maximum likelihood estimates $\hat{u}_1$ and $\hat{u}_2$ using the observed values v_1 and v_2
- Generalized Likelihood Ratio Test:

$$L(v_1, v_2) = \frac{p(v_1, v_2|\mathcal{H}_0, \hat{u}_{12})}{p(v_1, v_2|\mathcal{H}_1, \hat{u}_1, \hat{u}_2)}$$

From pixel similarities to patch similarities and weights

- Comnining pixel GLRT to define weights:

$$L(P_1, P_2) = \Pi_k L(v_{1\,k}, v_{2\,k})$$

- Link between weight and dissimilarities :

$$\text{dissi}(P_1, P_2) = -\log(w(P_1, P_2))$$

- Dissimilarity associated to GLRT :

$$\begin{aligned} \text{dissi}(P_1, P_2) &= -\log(L(P_1, P_2)) \\ &= \sum_k -\log(L(v_{1\,k}, v_{2\,k})) \end{aligned}$$

Weight and patch similarity: how to compare noisy patches?

Weight and patch similarity: how to compare noisy patches?

Example of AWGN

- the noise model is given by

$$p(v|u) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(v-u)^2}{2\sigma^2})$$

- the Maximum Likelihood estimate of u if only v is available is the value $\hat{u}$ maximizing $p(v|u)$:

$$\hat{u} = \text{argmax} - \log p(v|u) = v$$

$$\hat{u}_{12} = \text{argmax} - \log p(v_1|u)p(v_2|u) = \frac{1}{2}(v_1 + v_2)$$

Therefore $\hat{u}_1 = v_1$, $\hat{u}_2 = v_2$ and $\hat{u}_{12} = \frac{1}{2}(v_1 + v_2)$

- Generalized Likelihood Ratio Test:

$$L(v_1, v_2) = \frac{p(v_1, v_2|\mathcal{H}_0, \hat{u}_{12})}{p(v_1, v_2|\mathcal{H}_1, \hat{u}_1, \hat{u}_2)} = \frac{p(v_1|\hat{u}_{12})p(v_2|\hat{u}_{12})}{p(v_1|v_1)p(v_2|v_2)} = \exp(-\frac{(v_1 - v_2)^2}{4\sigma^2})$$

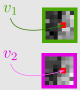
- Dissimilarity between pixels:

$$\text{dissi}(v_1, v_2) = \frac{(v_1 - v_2)^2}{4\sigma^2}$$

Euclidean distance between pixel values !...

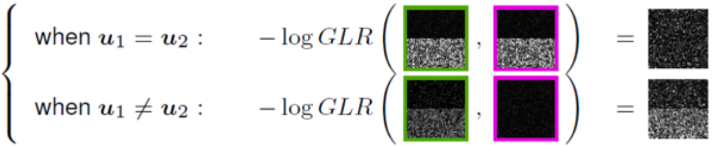
Patch similarity

- Example for multiplicative noise (Rayleigh-Nakagami distribution)
  - Likelihood test of the observed values to be explained by the same reflectivity (detection approach)
  - Generalized likelihood ratio test



$$-\log \mathrm{GLR}(v_1, v_2) = 2L \log\left(\sqrt{\frac{v_1}{v_2}} + \sqrt{\frac{v_2}{v_1}}\right) - 2L\log 2$$

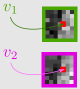
GLR



Patch similarity

- Example for multiplicative noise (Rayleigh-Nakagami distribution)
  - Likelihood test of the observed values to be explained by the same reflectivity (detection approach)
  - Generalized likelihood ratio test

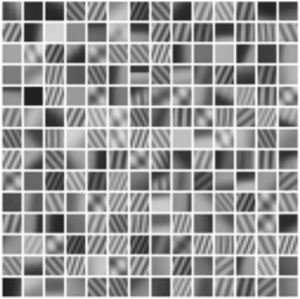
$$-\log \mathrm{GLR}(v_1, v_2) = 2L \log \left(\sqrt{\frac{v_1}{v_2}} + \sqrt{\frac{v_2}{v_1}} \right) - 2L \log 2$$

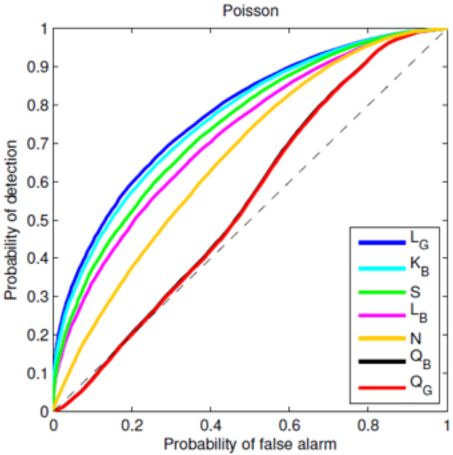


- Other strategy: information approach
  - Kullback-Leibler divergence similarity on denoised data for iterative scheme

$$\mathcal{D}_{\mathsf{KL}}(u_1, u_2) = L \frac{(u_1 - u_2)^2}{u_2 u_1}$$

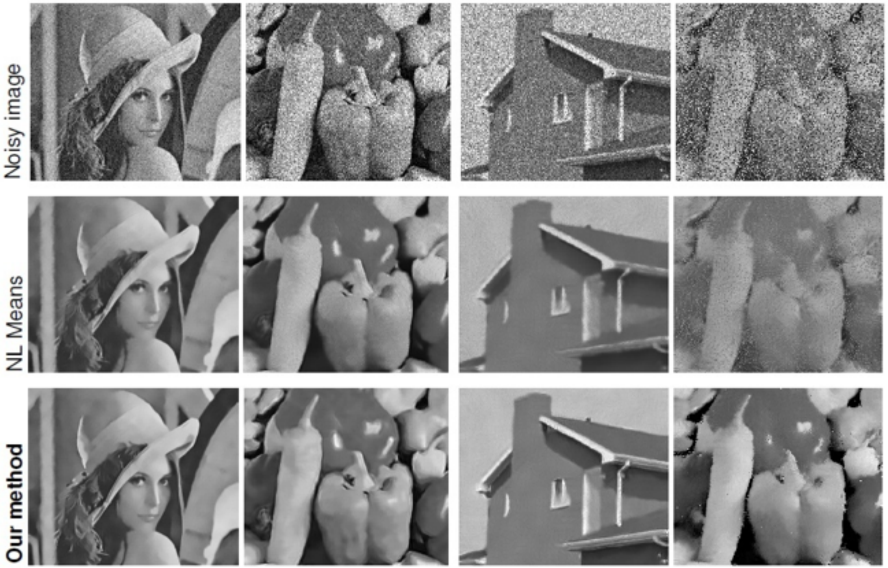
  - Comparison of the distributions inside the patches (loss of structural information but increase of robustness)
- estimation approach
  - Sigma-preselection to select the patch samples





- Generalized likelihood ratio
- Variance stabilization
- Euclidean distance
- Maximum joint likelihood [Alter et al., 2006]
- Mutual information kernel [Seeger, 2002]
- Bayesian likelihood ratio [Minka, 1998, Minka, 2000]
- Bayesian joint likelihood [Yianilos, 1995, Matsushita and Lin, 2007]

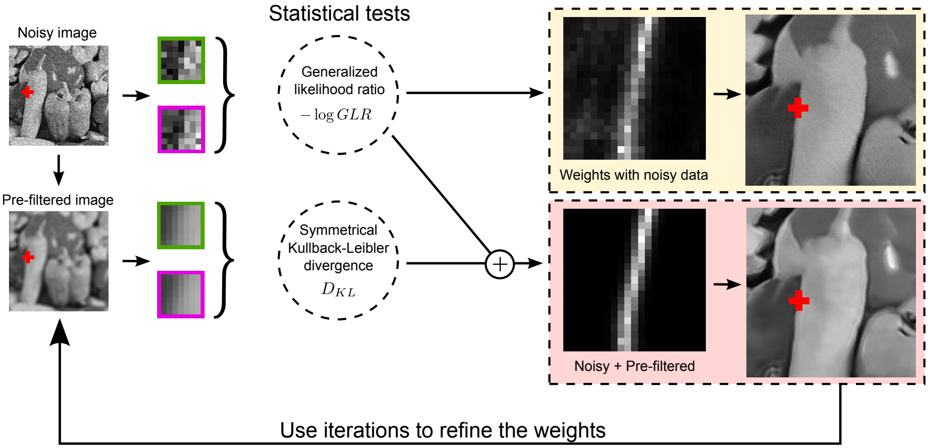
Extended non-local means for various noise models



(a) Gaussien +0.87 dB (b) Poisson +1.13 dB (c) *Speckle* +4.00 dB (d) Impuls. +3.82 dB

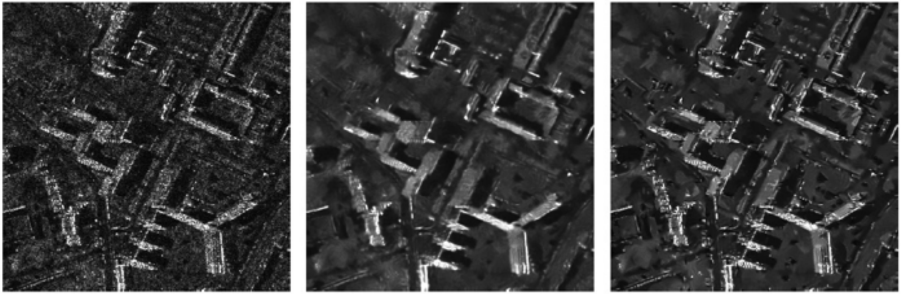
Iterative framework

similarity improvement using the current denoised estimate



Iterative framework

similarity improvement using the current denoised estimate



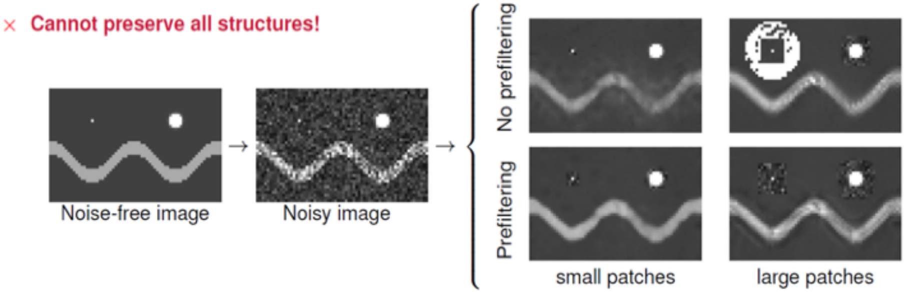
(b) A (c) $\hat{R}^1$ (d) $\hat{R}^i$

Automatic setting of parameters

Many parameters

- Search window: rare patch effect, influence of small weights
- Patch size: rare patch effect, noise halo
- Kernel (shape, discriminative power): more or less selective, bias / variance trade-off
- Pre-filtering strength: improvement for high noise level, but blurring effect

antagonist criteria: no best parameter tuning !



Automatic setting of parameters

Parameter choice should be adapted to the signal content

A good choice for a specific area can be a bad one for another one : combination of results to select locally the best one

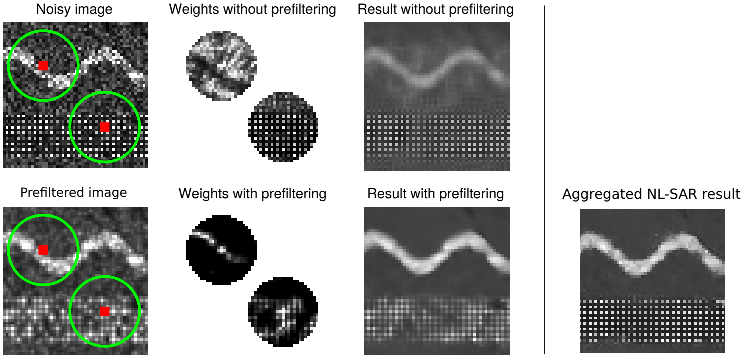
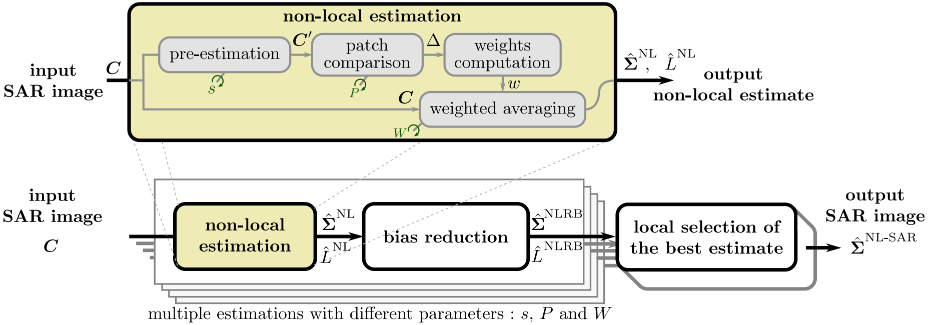


Figure: (left) Top: non local means result by comparing 7×7 patches extracted from the noisy image. Bottom: Same except patches are extracted in a prefiltered image. Two pixels of interest (in red) are focused and their associated weights in the circle searching window (in green) are displayed. (right) NL-SAR result that is an aggregation of several non local means results obtained for different prefiltering strengths, patch sizes and search window sizes.

Automatic setting of parameters

Parameter choice should be adapted to the signal content

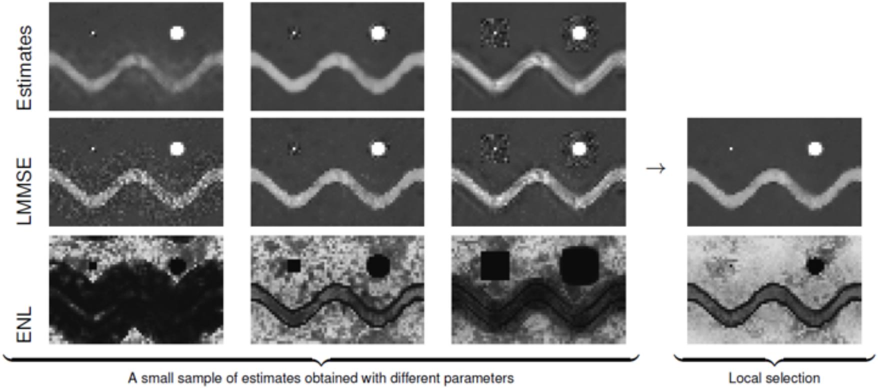
A good choice for a specific area can be a bad one for another one : combination of results to select locally the best one

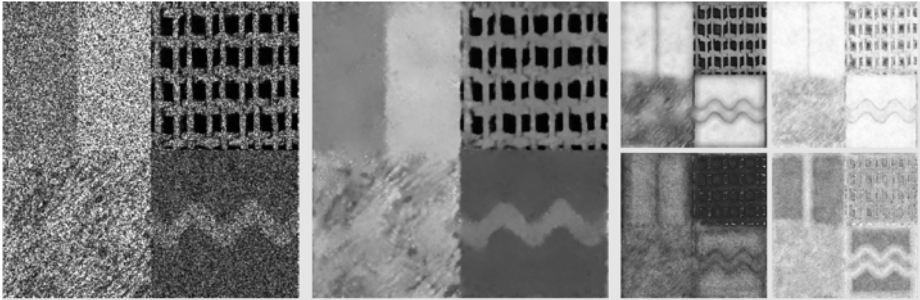


Automatic setting of parameters

Parameter choice should be adapted to the signal content

A good choice for a specific area can be a bad one for another one : combination of results to select locally the best one





(a) (b) (c)

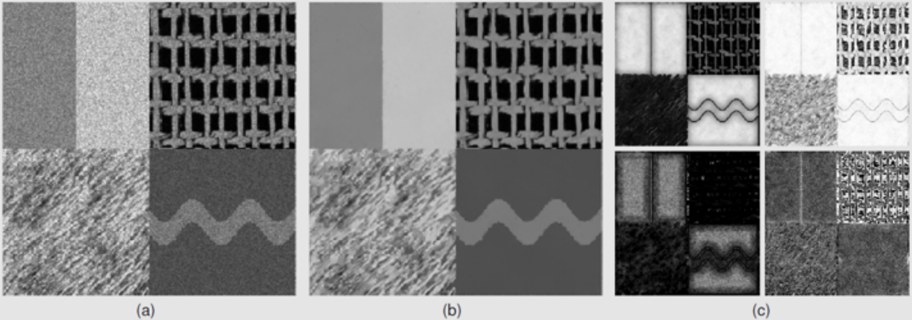
(a) Noisy image.

(b) Result of the adaptive approach.

(c) From left to right, top to bottom:

- smoothing strength (range: $[0, 20 \times 20]$),
- search window sizes (range: $[0, 20 \times 20]$),
- the patch size (range: $[3 \times 3, 11 \times 11]$),
- prefiltering strength (range: $[1, 3]$).

Automatic setting of parameters



(a) Noisy image.

(b) Result of the adaptive approach.

(c) From left to right, top to bottom:

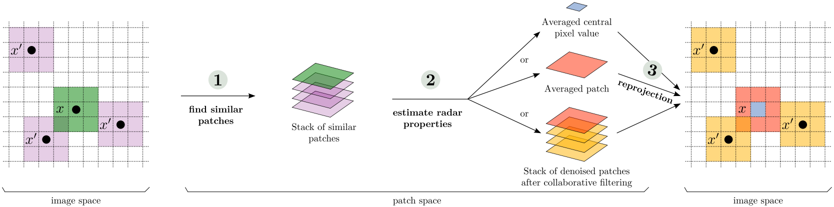
- smoothing strength (range: $[0, 20 \times 20]$),
- search window sizes (range: $[0, 20 \times 20]$),
- the patch size (range: $[3 \times 3, 11 \times 11]$),
- prefiltering strength (range: $[1, 3]$).

Denoising of 3D blocks of patches

Block of patches

- Global denoising of the block of patches
- Combination of denoised patches

More efficient use of information!



BM3D (Dabov et al.)

Principle of BM3D

- 2-steps filtering
- Step 1: global 3D filtering of the block (grouping, collaborative filtering, aggregation)
- Step 2: block of noisy and current estimate patches and second global filtering of the 3D noisy block driven by current estimates, followed by aggregation

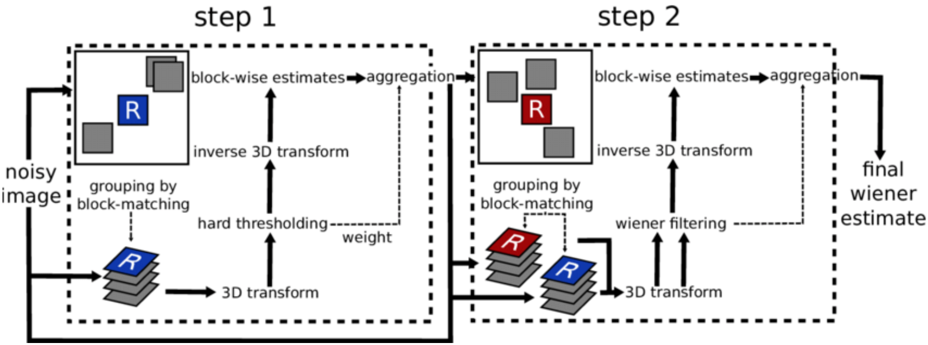


Figure: figure of Marc Lebrun (IPOL)

Principle of BM3D

- 2-steps filtering
- Step 1: global 3D filtering of the block (grouping, collaborative filtering, aggregation)
- Step 2: block of noisy and current estimate patches and second global filtering of the 3D noisy block driven by current estimates, followed by aggregation

NLBayes (Lebrun et al.)

Principle of NLBayes (Non Local Bayes)

- Main idea: using a model for the patch distribution
- Gaussian multivariate pdf with a mean (mean patch) and a covariance matrix
- Step1 : these parameters are computed empirically using the block of similar noisy patches $\overline{P}_v$ and C_{P_v}; an analytic formula gives the expression of the denoised patch (MAP estimate) called basic estimate

$$\hat{P}_u = \overline{P}_v + (C_{P_v} - \sigma^2 I)C_{P_v}^{-1}(P_v - \overline{P}_v)$$

- Step 2: improvement of the Gaussian pdf using the block of basic estimate patches and new estimation

Application to color images

- Color space: YUV system separating luminance and chromatic parts (transformation from RGB to YUV, processing, inverse transform)
- Processing of each channel separately (the distance between patches for grouping can combine the 3 channels

NLBayes (Lebrun et al.)

Noisy image ($\sigma = 30$)

NL-Bayes ($\sigma = 30$)

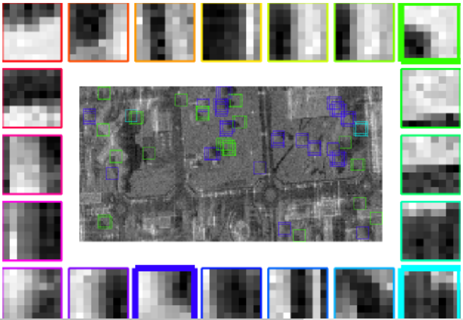
Outline

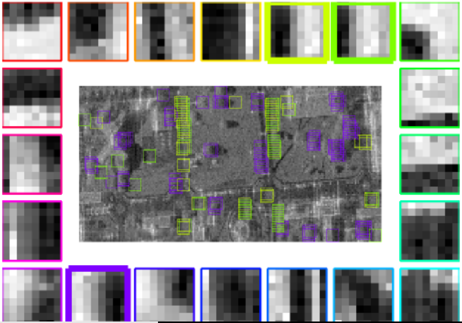
Dictionaries of patches

redundancy / dictionary

- Limits of patch-based approaches
  - Rare patch effect: redundancy not verified
  - Low contrast situations: not enough similar samples
- Solutions
  - Use a database with many examples
  - Create representative atoms of an image
  - Create a dictionary of models

- K-SVD: search the representative patches
- FoE (Field of Experts): model and learn the clique potentials (clique = neighborhood = patch)
- EPLL: create dictionaries of models of Gaussian distributed patches (GMM: Gaussian Mixture Models)





General idea

The method is based on the optimization of a functional $\sum_{ij} ||D\alpha_{ij} - P_{v_{ij}}||^2$ combining the following elements:

- Sparse coding α_{ij} of the patches of the image using a patch dictionary D
- Improvement (updating) of the dictionary to improve the sparse coding of the image
- Reconstruction of the image v using the final dictionary with aggregation

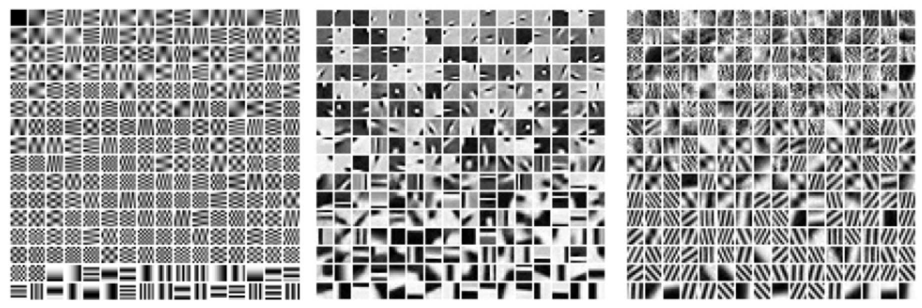


Figure: Examples of dictionaries: on the left DCT dictionary, middle K-SVD dictionary on a set of natural image, on the right K-SVD update for Barbara image

K-SVD

General idea

The method is based on the optimization of a functional $\sum_{ij} ||D\alpha_{ij} - P_{v_{ij}}||^2$ combinig the following elements:

- Sparse coding α_{ij} of the patches of the image using a patch dictionary D
- Improvement (updating) of the dictionary to improve the sparse coding of the image
- Reconstruction of the image v using the final dictionary with aggregation

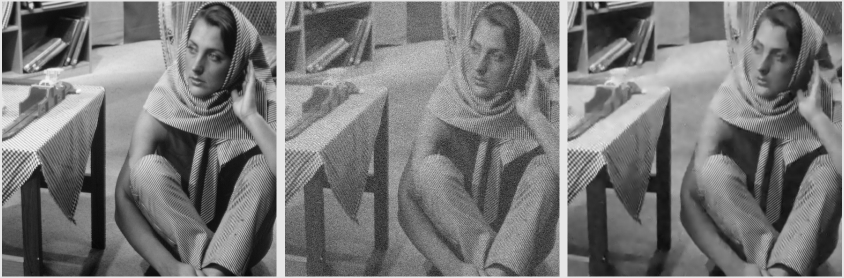


Figure: Examples of K-SVD denoising: from left to right original image, noisy image, K-SVD denoising (256 atoms in D)

General idea

Instead of using a dictionary of fixed atoms, atoms are replaced by Gaussian Mixture Models.

- A patch is a sample of a Gaussian multi-variate distribution $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.
- Create the dictionary of GMM using a database of natural image (ex 200 components learnt on 10^6 patches)
- Solve the following optimization problem $||u - v||^2 - \log(\Pi_i p(P_{u_i}|k_i))$

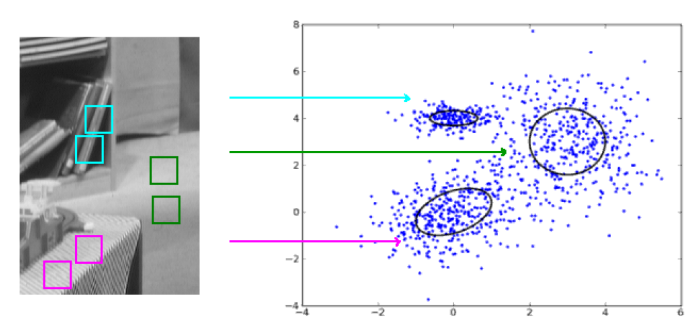


Figure: Each patch comes from one of the GMM.

EPLL and GMM

General idea

Instead of using a dictionary of fixed atoms, atoms are replaced by Gaussian Mixture Models.

- A patch is a sample of a Gaussian multi-variate distribution $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.
- Create the dictionary of GMM using a database of natural image (ex 200 components learnt on 10^6 patches)
- Solve the following optimization problem $||u - v||^2 - \log(\Pi_i p(P_{u_i}|k_i))$

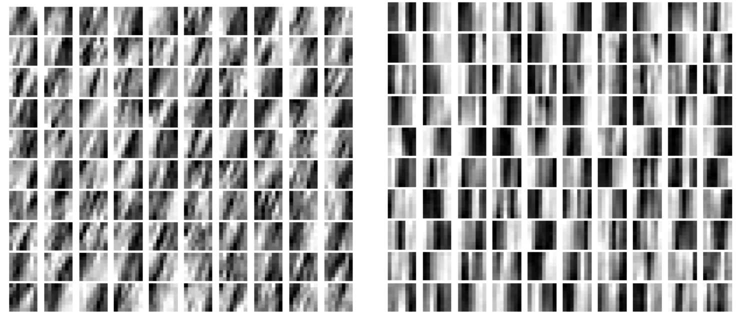


Figure: Examples of patches drawn from 2 Gaussian models, one encoding a stripe pattern (on the left) and one encoding a vertical edge (on the right)

EPLL and GMM

General idea

Instead of using a dictionary of fixed atoms, atoms are replaced by Gaussian Mixture Models.

- A patch is a sample of a Gaussian multi-variate distribution $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.
- Create the dictionary of GMM using a database of natural image (ex 200 components learnt on 10^6 patches)
- Solve the following optimization problem $||u - v||^2 - \log(\Pi_i p(P_{v_i}|k_i))$

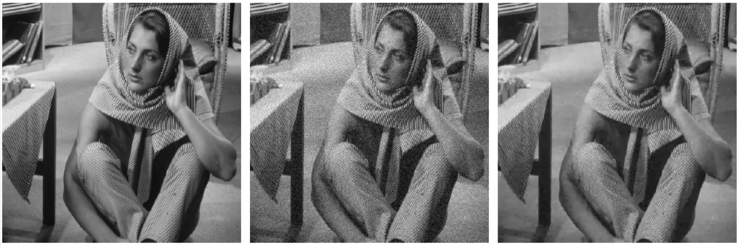


Figure: Denoising of an image using GMM: on the left original image,middle noisy image, on the right denoising with EPLL and 200 GMM.

EPLL and GMM

General idea

Instead of using a dictionary of fixed atoms, atoms are replaced by Gaussian Mixture Models.

- A patch is a sample of a Gaussian multi-variate distribution $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.
- Create the dictionary of GMM using a database of natural image (ex 200 components learnt on 10^6 patches)
- Solve the following optimization problem $||u - v||^2 - \log(\Pi_i p(P_{v_i}|k_i))$

Figure: Denoising of an image using GMM: on the left original image, on the right the color code represents the chosen GMM. Similar textures are represented by the same model.

Outline

CNN and denoising

Convolutive neural networks

- Combines non linear steps (such as truncating values below a threshold) and local filtering
  - Hierarchy of non-linear features
  - many layers: increases the considered neighborhood (receptive field)
- Different strategies
  - Residual learning: ex DnCNN (restores the noise residual image -easier to train)
  - Auto-supervised learning: ex Noise2noise (uses only noisy samples to do the training)
- Pros and Cons
  - Very effective to preserve geometric structure and textures
  - May invent plausible structures (hard to tell artifacts)

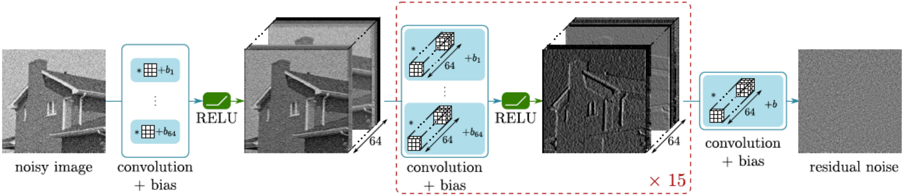


Figure: Architecture of DnCNN, Zhang et al.

Example of CNN / non-local combination (1)

General idea

Training a network using non-local information: increasing the number of channels using image redundancy, Davy et al.

- Principle
  1. find the K most similar patches
  2. collect the central values of these patches
  3. concatenate them to form K additional layers
- Key idea: the denoising can be improved when making available values from similar patches that are quite far apart

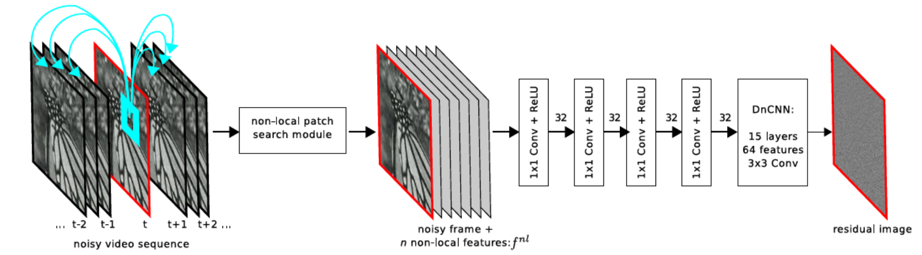


Figure: Architecture of Davy et al. network exploiting patch redundancy to create additional channels ["Non-local video denoising by CNN"].

Example of CNN / non-local combination (2)

General idea

Iterate CNN and non-local methods to reduce the artifacts created by the CNN.

- Principle
  1. The noisy and current estimate are combined iteratively: $\bar{z}_k = \lambda_k z + (1 - \lambda_k)\hat{y}_{k-1}$
  2. The current estimate is obtained by a CNN taking the decreasing noise variance into account followed by a non-local filter with updated threshold
- Key idea: correct the drawback of one method by the other

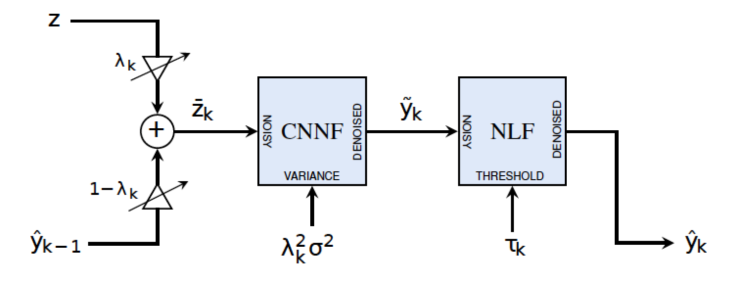


Figure: Algorithm of Cruz et al. : iterative (CNN+NLM) approach ["Nonlocality-reinforced convolutional neural networks for image denoising"]. NLF: simple averaging of the k-nearest neighbors with threshold τ_k, CNNF trained CNN with decreasing λ_k.

Example of CNN / non-local combination (3)

General idea

Introducing a non-local block inside the network to exploit the redundancy in the image or in the feature maps.

- Principle
  1. The non local block is trained to generate continuous nearest neighbors versions of the input
  2. It is then used as a building brick to define new networks architectures
  3. The new architecture is then trained in a usual way
- Key idea: introduce redundancy at different feature levels

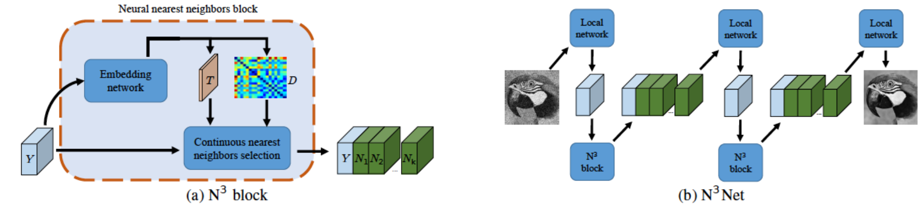
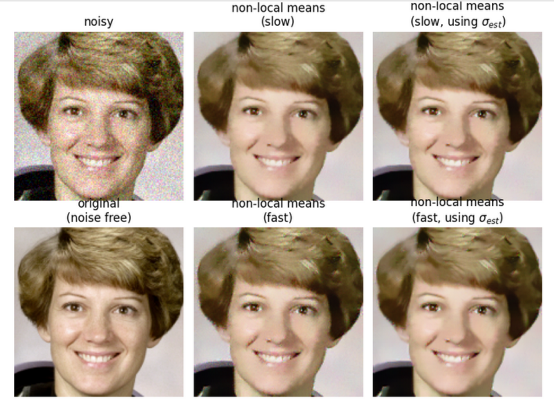


Figure: Architecture of Plotz et al. : N^3 brick and new architecture including N^3 component ["Neural Nearest Neighbors Networks"].

Python

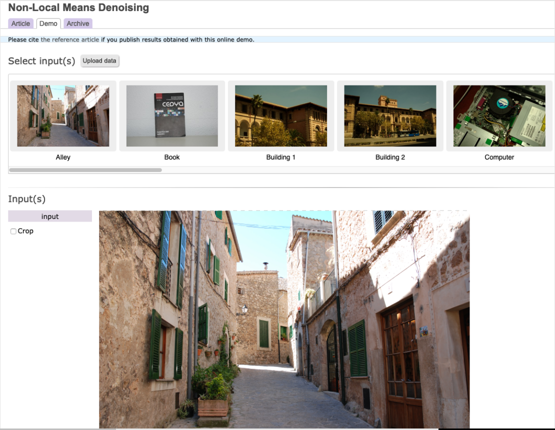
- Library skimage (scikit-image, image processing in python)
- from skimage.restoration import denoise_nl_means
- `https://scikit-image.org/docs/dev/auto_examples/filters/plot_nonlocal_means.html`



Practice of non-local approaches

IPOL

- Image Processing On Line (reproducible research, online demo + detailed paper on implementation tricks)
- `https://www.ipol.im`
- Topics : Enhancement and restoration (Denoising)



Take home messages

Patch-based methods

- Exploit the redundancy in images
- Key ingredients : patch distance, parameters, aggregation step $\Rightarrow$ must be adapted to the noise statistics
- State of the art methods before deep learning
- Recent trends try to combine the efficiency of both approaches