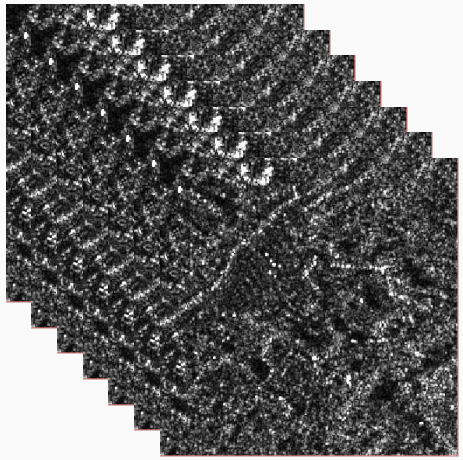
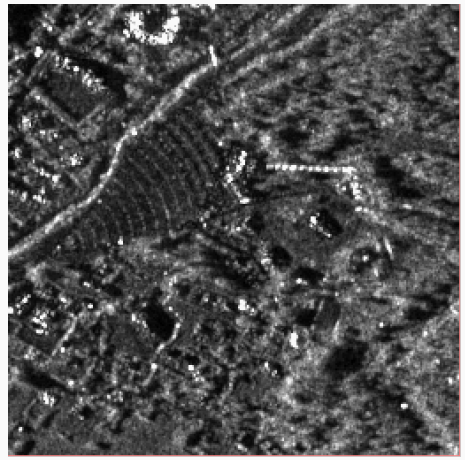
Patches and non-local filters

Spatial filtering – Looking for other views



T noisy observations $y^{(t)}$



Estimation $\hat{x}$ of the unknown signal x

- Sample averaging of T noisy values:

$$\mathbb{E}[\hat{x}_i] = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} y_i^{(t)}\right] = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[y_i^{(t)}] = \frac{1}{T}\sum_{t=1}^{T} x_i = x_i \qquad \text{(unbiased)}$$

$$\text{and} \quad \mathrm{Var}[\hat{x}_i] = \mathrm{Var}\left[\frac{1}{T}\sum_{t=1}^{T} y_i^{(t)}\right] = \frac{1}{T^2}\sum_{t=1}^{T}\mathrm{Var}[y_i^{(t)}] = \frac{1}{T^2}\sum_{t=1}^{T}\sigma^2 = \frac{\sigma^2}{T}$$

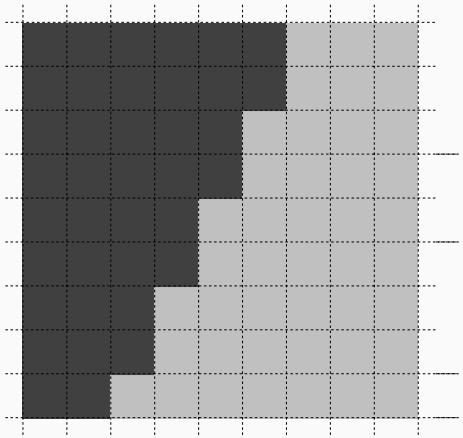
(reduce noise)

- . . . only if the selected values are iid.

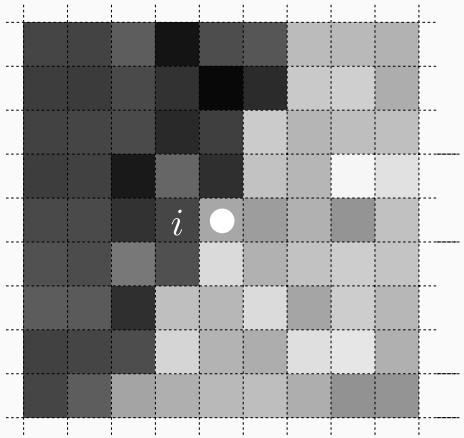
similar = close to be iid

→ How can we select them on a single image?

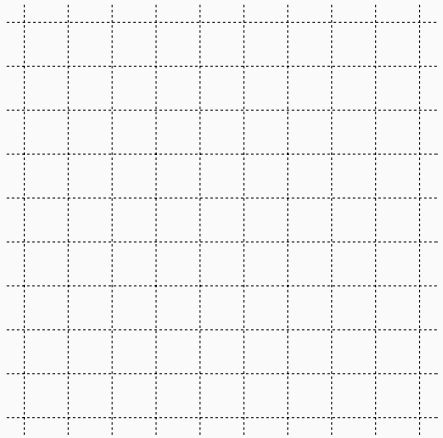
Spatial filtering – Selection-based filtering



Unknown noise-free image x



Input noisy image y

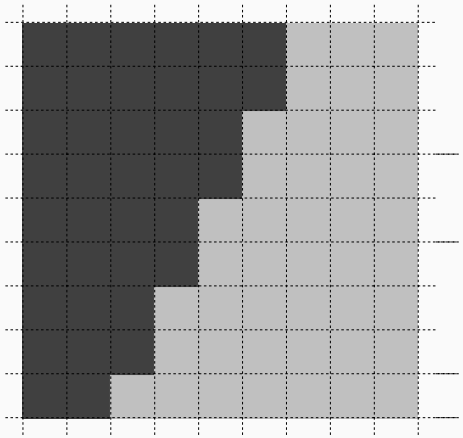


Output denoised image $\hat{x}$

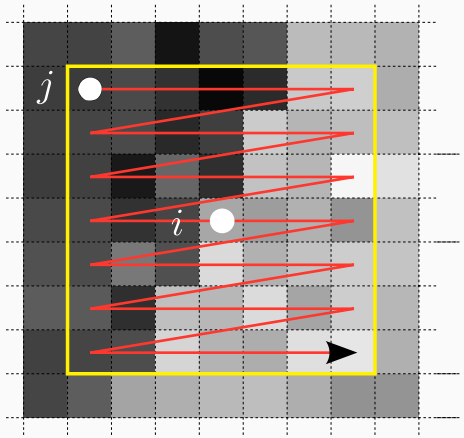
General idea

- Goal: estimate the image x from the noisy image y
- Choose a pixel i to denoise

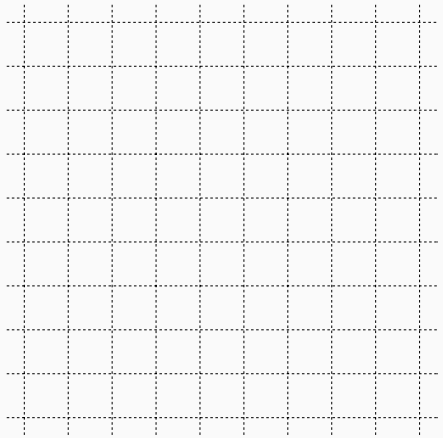
Spatial filtering – Selection-based filtering



Unknown noise-free image x



Input noisy image y

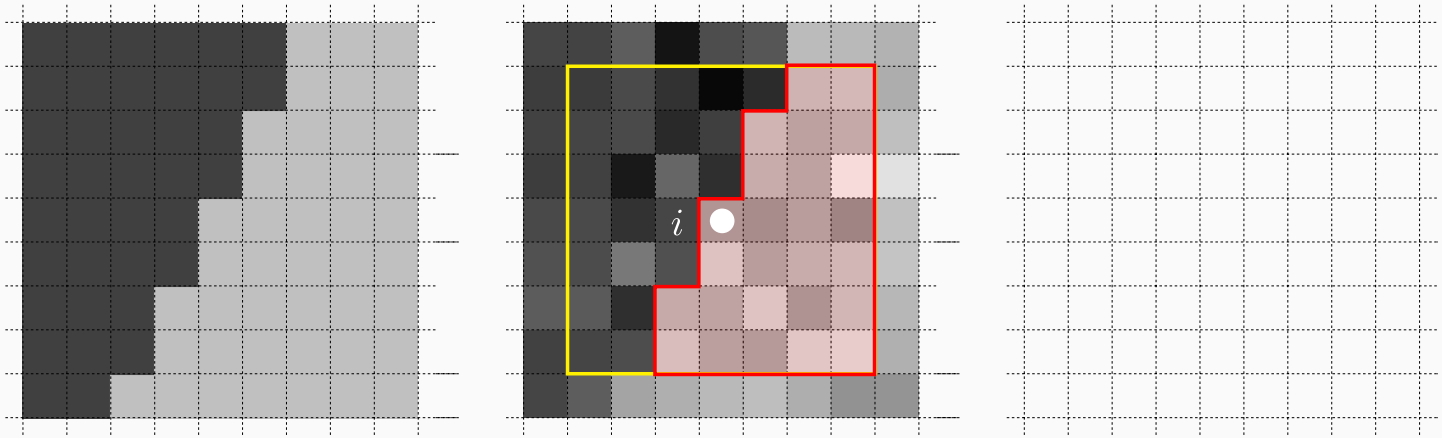


Output denoised image $\hat{x}$

General idea

- Goal: estimate the image x from the noisy image y
- 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

Spatial filtering – Selection-based filtering

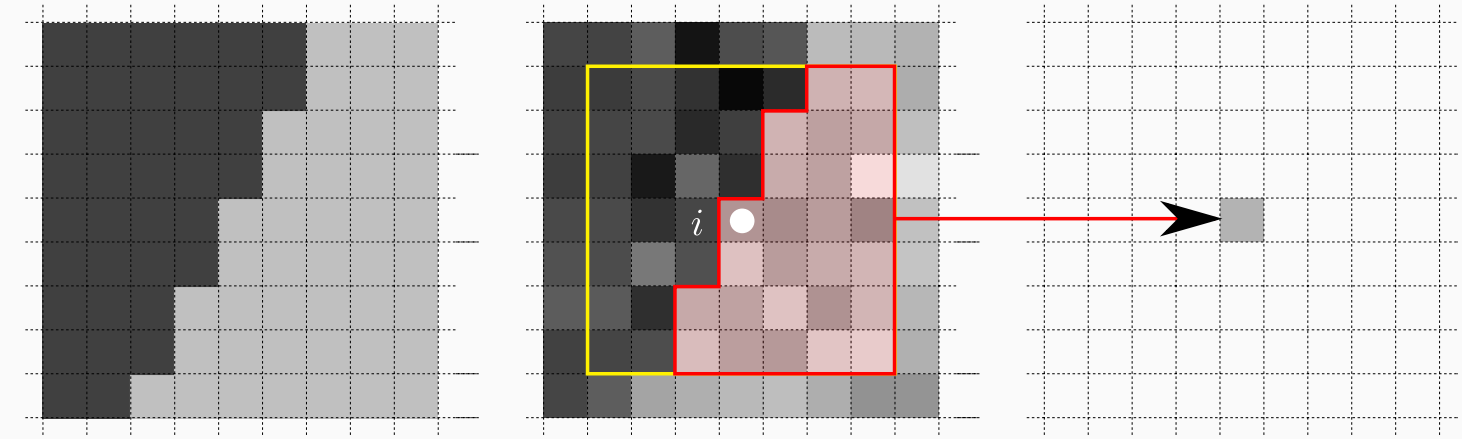


Unknown noise-free image x | Input noisy image y | Output denoised image $\hat{x}$

General idea

- Goal: estimate the image x from the noisy image y
- 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

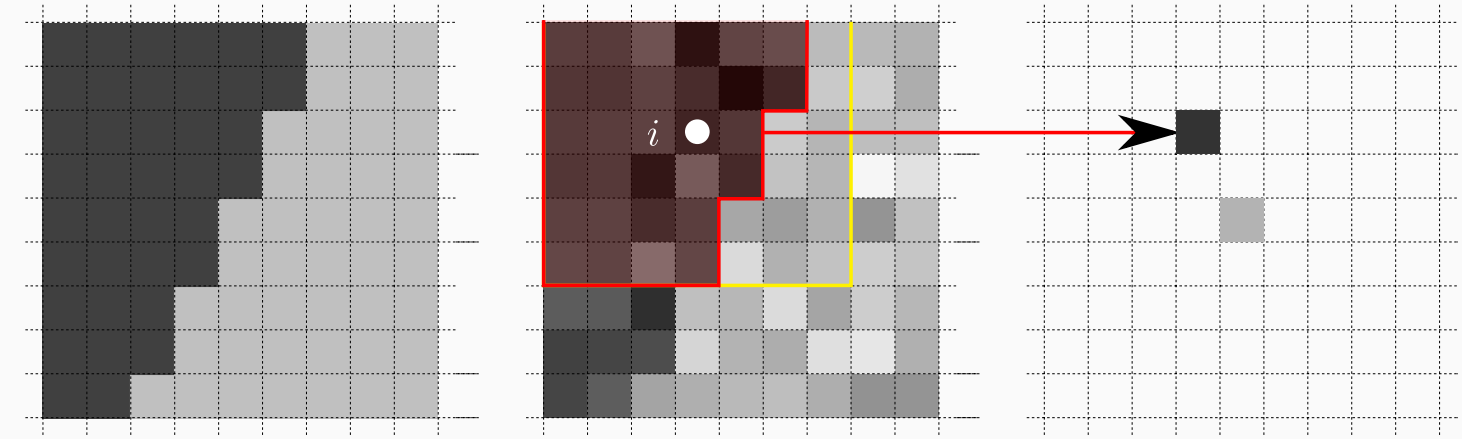
Spatial filtering – Selection-based filtering



General idea

- Goal: estimate the image x from the noisy image y
- 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

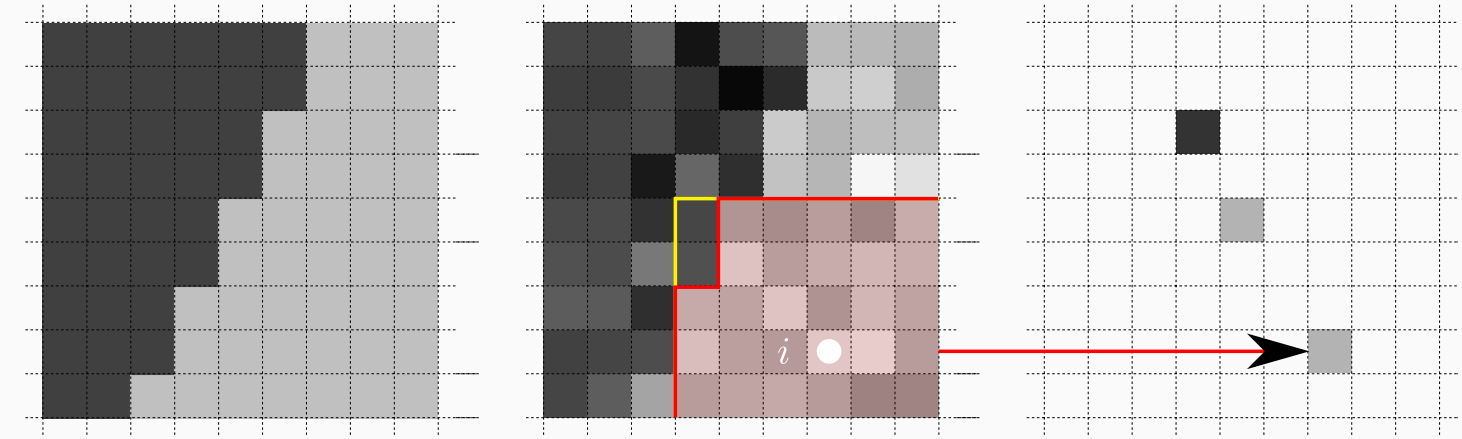
Spatial filtering – Selection-based filtering



General idea

- Goal: estimate the image x from the noisy image y
- 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

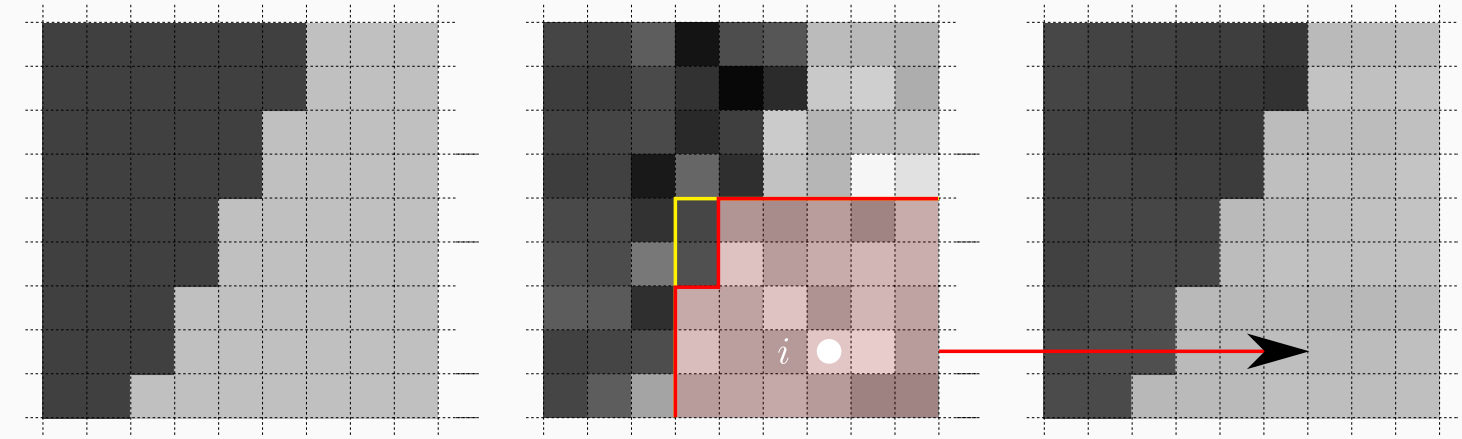
Spatial filtering – Selection-based filtering



General idea

- Goal: estimate the image x from the noisy image y
- 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

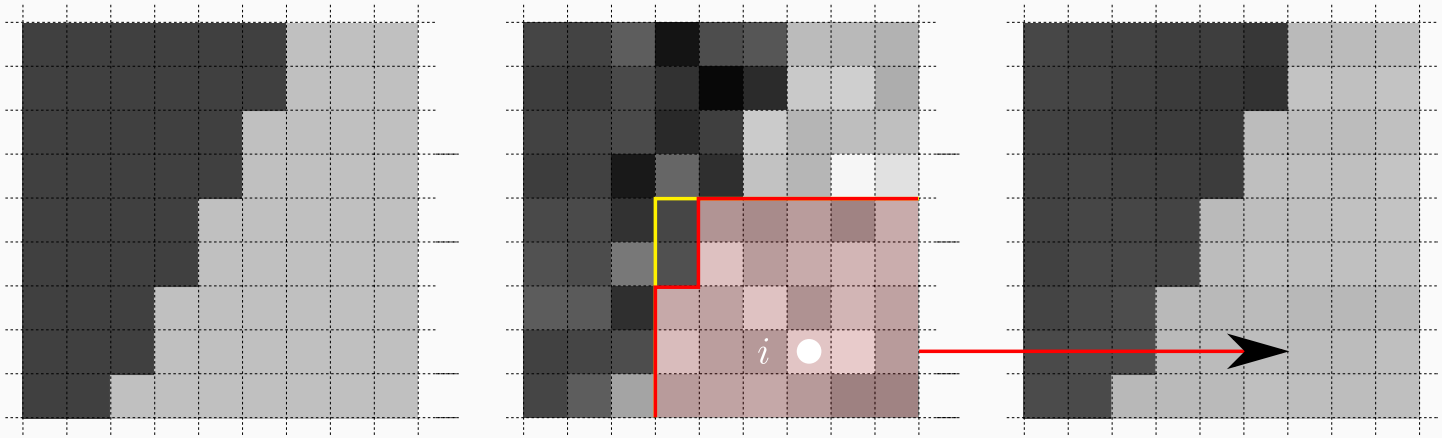
Spatial filtering – Selection-based filtering



General idea

- Goal: estimate the image x from the noisy image y
- 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

Spatial filtering – Selection-based filtering



Unknown noise-free image x | Input noisy image y | Output denoised image $\hat{x}$

Selection rules

$$\hat{x}_i = \frac{\sum_j w_{i,j} y_j}{\sum_j w_{i,j}} \quad \text{where}$$

$$w_{i,j} = \begin{cases} 1 & \text{if } \boxed{\|s_i - s_j\| \leqslant \tau} \\ 0 & \text{otherwise} \end{cases} \leftarrow \text{Moving average}$$

$$w_{i,j} = \begin{cases} 1 & \text{if } \boxed{\|y_i - y_j\| \leqslant \tau} \\ 0 & \text{otherwise} \end{cases} \leftarrow \text{Sigma filter}$$

$$w_{i,j} = \begin{cases} 1 & \text{if } \boxed{x_i = x_j} \\ 0 & \text{otherwise} \end{cases} \leftarrow \text{Oracle}$$

How to choose suitable pixels j to combine?

Spatial filtering – 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

$\rightarrow$ trade-off:
size $\nearrow$ $\Rightarrow$ more distinctive/informative
size $\searrow$ $\Rightarrow$ easier to model/learn/match

---

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

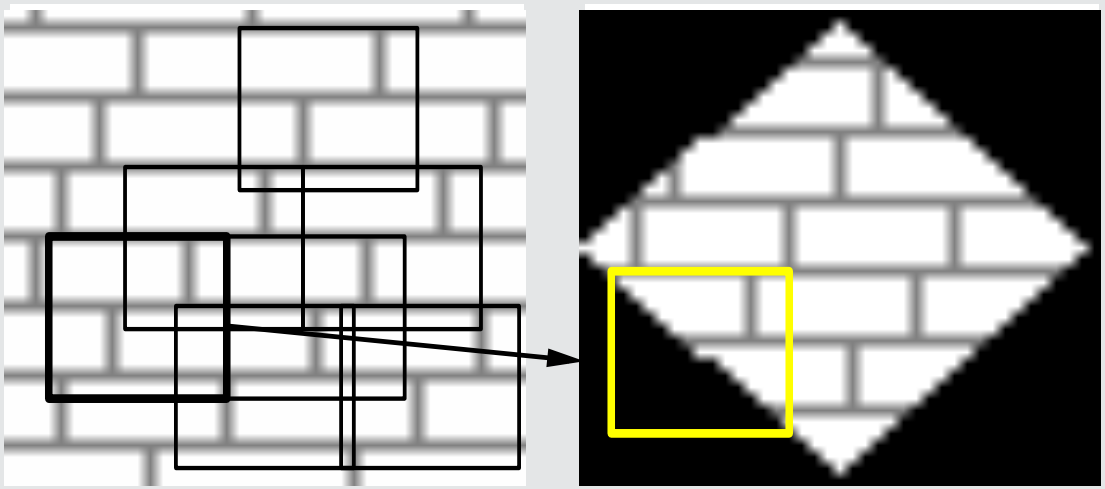
patches capture local context: geometry and texture

Copying/pasting similar patches yields impressive texture synthesis:

Texture synthesis method by Efros and Leung (1999)

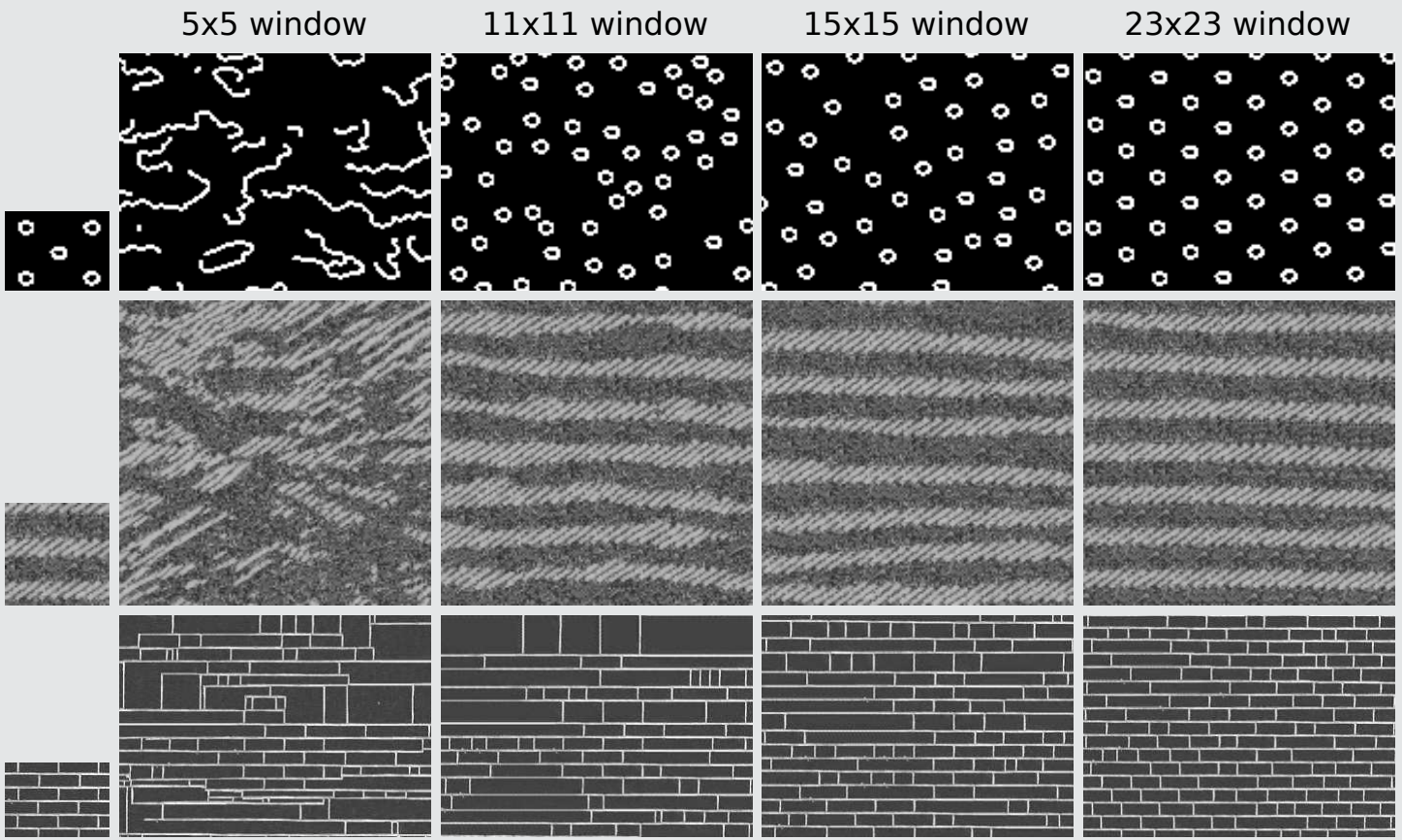
To generate a new pixel value:

- extract the surrounding patch (yellow)
- find similar patches in the reference image
- randomly pick one of them
- use the value of the central pixel of that patch



Copying/pasting similar patches yields impressive texture synthesis:

Texture synthesis method by Efros and Leung (1999)



Spatial filtering – Non-local means

Bilateral filter [Tomasi & Manduchi, 1998]

$$\hat{x}_i = \frac{\sum_{j \in \mathcal{N}_i} w_{i,j} y_j}{\sum_{j \in \mathcal{N}_i} w_{i,j}} \quad \text{with} \quad w_{i,j} = \varphi_{\text{space}}(\|s_i - s_j\|_2^2) \times \varphi_{\text{color}}(\|y_i - y_j\|_2^2)$$

weights depend on the distance between **pixel positions** and **pixel values**

Non-local means [Buades at al, 2005, Awate et al, 2005]

$$\hat{x}_i = \frac{\sum_{j \in \mathcal{N}_i} w_{i,j} y_j}{\sum_{j \in \mathcal{N}_i} w_{i,j}} \quad \text{with} \quad w_{i,j} = \varphi(\|\mathcal{P}_i y - \mathcal{P}_j y\|_2^2)$$

- $\mathcal{N}_i$: large neighborhood of i, called search window (typically 21×21)
- $\mathcal{P}_i$: operator extracting a small window, *patch*, at i (typically 7×7)

weights in a **large search windows** depend on the distance between **patches**

Spatial filtering – Non-local means

Remarks

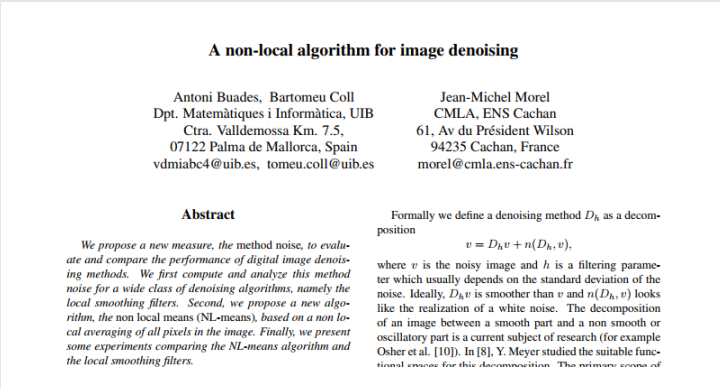
The term *non-local* refers to that disconnected pixels are mixed together.

The Sigma, Yaroslavsky and Bilateral filters are then also non-local.

But *Non-Local means* always refers to the one using patches.

(or *NL-means*)

A similar algorithm was concurrently proposed under the name UINTA.



A non-local algorithm for image denoising

Antoni Buades, Bartomeu Coll
Dpt. Matemàtiques i Informàtica, UIB
Ctra. Valldemossa Km. 7.5,
07122 Palma de Mallorca, Spain
vdmiabc4@uib.es, tomeu.coll@uib.es

Jean-Michel Morel
CMLA, ENS Cachan
61, Av du Président Wilson
94235 Cachan, France
morel@cmla.ens-cachan.fr

Abstract

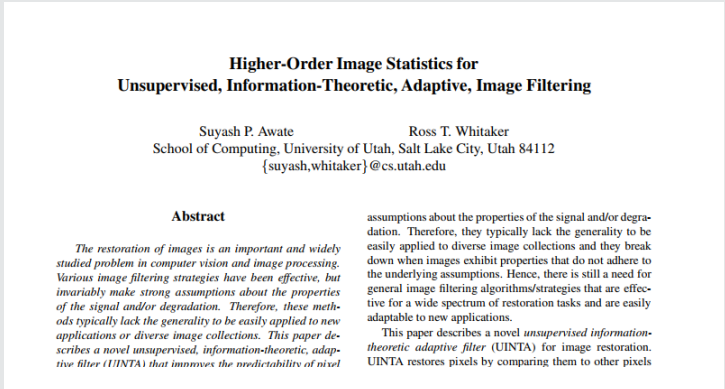
We propose a new measure, the method noise, *to evaluate and compare the performance of digital image denoising methods. We first compute and analyze this method noise for a wide class of denoising algorithms, namely the local smoothing filters. Second, we propose a new algorithm, the* non local means (NL-means), *based on a non local averaging of all pixels in the image. Finally, we present some experiments comparing the NL-means algorithm and the local smoothing filters.*

Formally we define a denoising method D_h as a decomposition

$$v = D_h v + n(D_h, v),$$

where v is the noisy image and h is a filtering parameter which usually depends on the standard deviation of the noise. Ideally, $D_h v$ is smoother than v and $n(D_h, v)$ looks like the realization of a white noise. The decomposition of an image between a smooth part and a non smooth or oscillatory part is a current subject of research (for example Osher et al. [10]). In [8], Y. Meyer studied the suitable functional spaces for this decomposition. The primary scope of

NL-means [Buades et al, CVPR 2005]



Higher-Order Image Statistics for Unsupervised, Information-Theoretic, Adaptive, Image Filtering

Suyash P. Awate Ross T. Whitaker
School of Computing, University of Utah, Salt Lake City, Utah 84112
{suyash,whitaker}@cs.utah.edu

Abstract

The restoration of images is an important and widely studied problem in computer vision and image processing. Various image filtering strategies have been effective, but invariably make strong assumptions about the properties of the signal and/or degradation. Therefore, these methods typically lack the generality to be easily applied to new applications or diverse image collections. This paper describes a novel unsupervised, information-theoretic, adaptive filter (UINTA) that improves the predictability of pixel

assumptions about the properties of the signal and/or degradation. Therefore, they typically lack the generality to be easily applied to diverse image collections and they break down when images exhibit properties that do not adhere to the underlying assumptions. Hence, there is still a need for general image filtering algorithms/strategies that are effective for a wide spectrum of restoration tasks and are easily adaptable to new applications.

This paper describes a novel *unsupervised information-theoretic adaptive filter* (UINTA) for image restoration. UINTA restores pixels by comparing them to other pixels

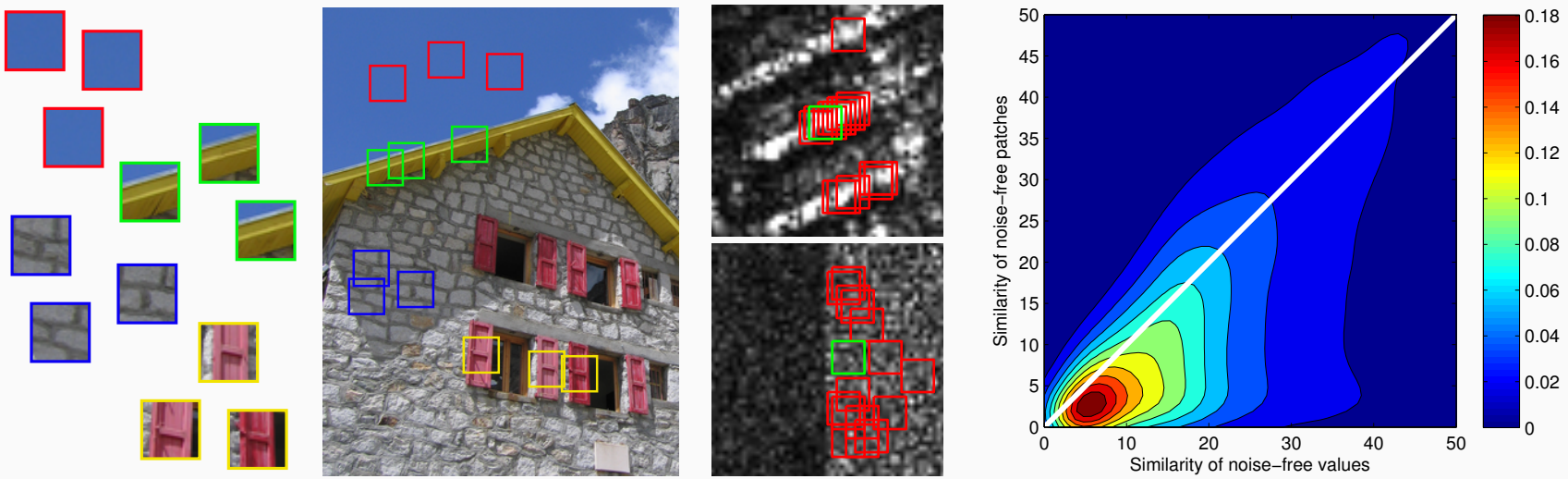
UINTA [Awate et al, CVPR 2005]

Spatial filtering – Non-local means

Non-local approach [Buades at al, 2005, Awate et al, 2005]

- Local filters: average neighborhood pixels
- Non-local filters: average pixels being in a similar context

$$\hat{x}_i = \frac{\sum_j w_{i,j} y_j}{\sum_j w_{i,j}}$$



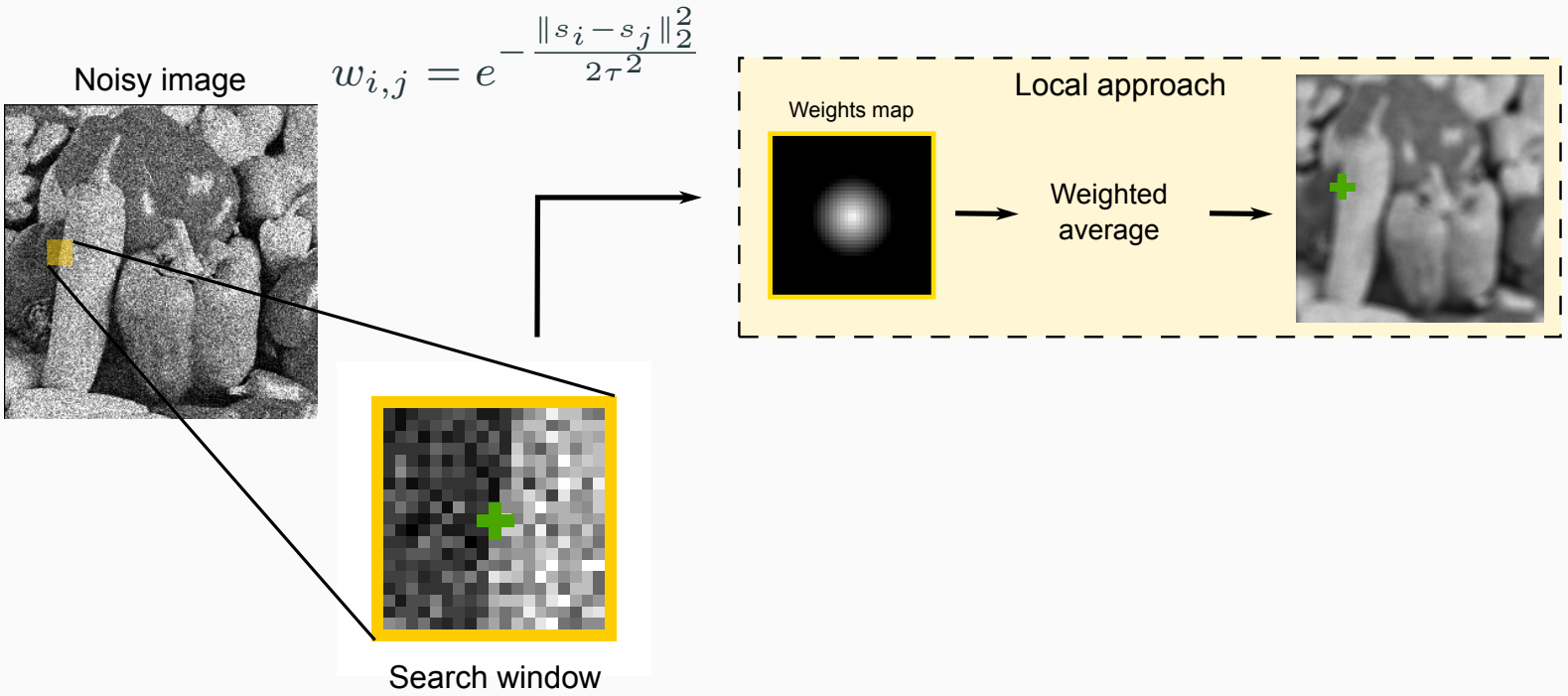
Patches are redundant in most types of images (large noise reduction)
and **similar ones tend to share the same underlying noise-free values** (unbiasedness)

Spatial filtering – Non-local means

Non-local approach [Buades at al, 2005, Awate et al, 2005]

- Local filters: average neighborhood pixels
- Non-local filters: average pixels being in a similar context

$$\hat{x}_i = \frac{\sum_j w_{i,j} y_j}{\sum_j w_{i,j}}$$

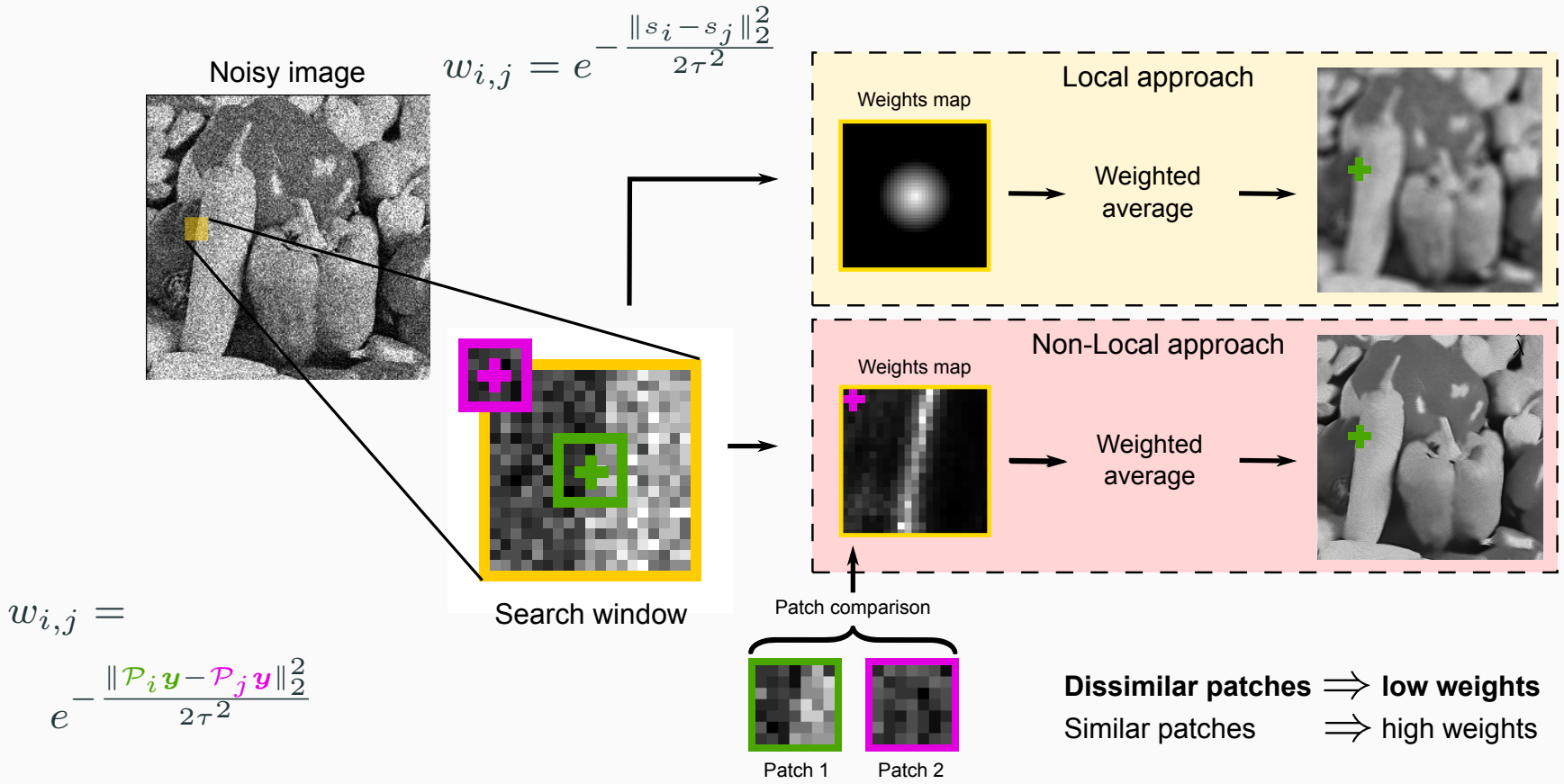


Spatial filtering – Non-local means

Non-local approach [Buades at al, 2005, Awate et al, 2005]

- Local filters: average neighborhood pixels
- Non-local filters: average pixels being in a similar context

$$\hat{x}_i = \frac{\sum_j w_{i,j} y_j}{\sum_j w_{i,j}}$$

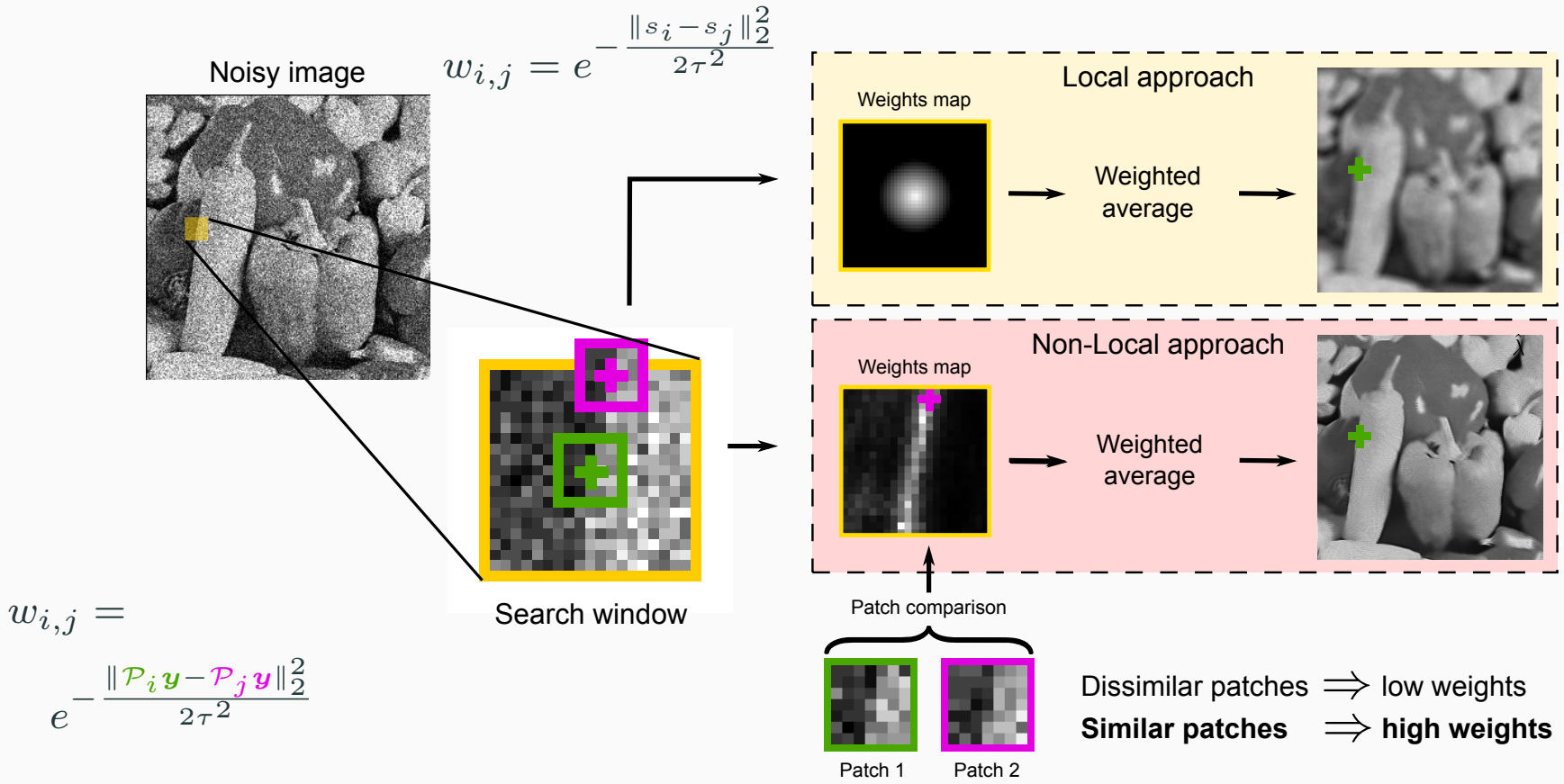


Spatial filtering – Non-local means

Non-local approach [Buades at al, 2005, Awate et al, 2005]

- Local filters: average neighborhood pixels
- Non-local filters: average pixels being in a similar context

$$\hat{x}_i = \frac{\sum_j w_{i,j} y_j}{\sum_j w_{i,j}}$$



Example (Map of non-local weights)

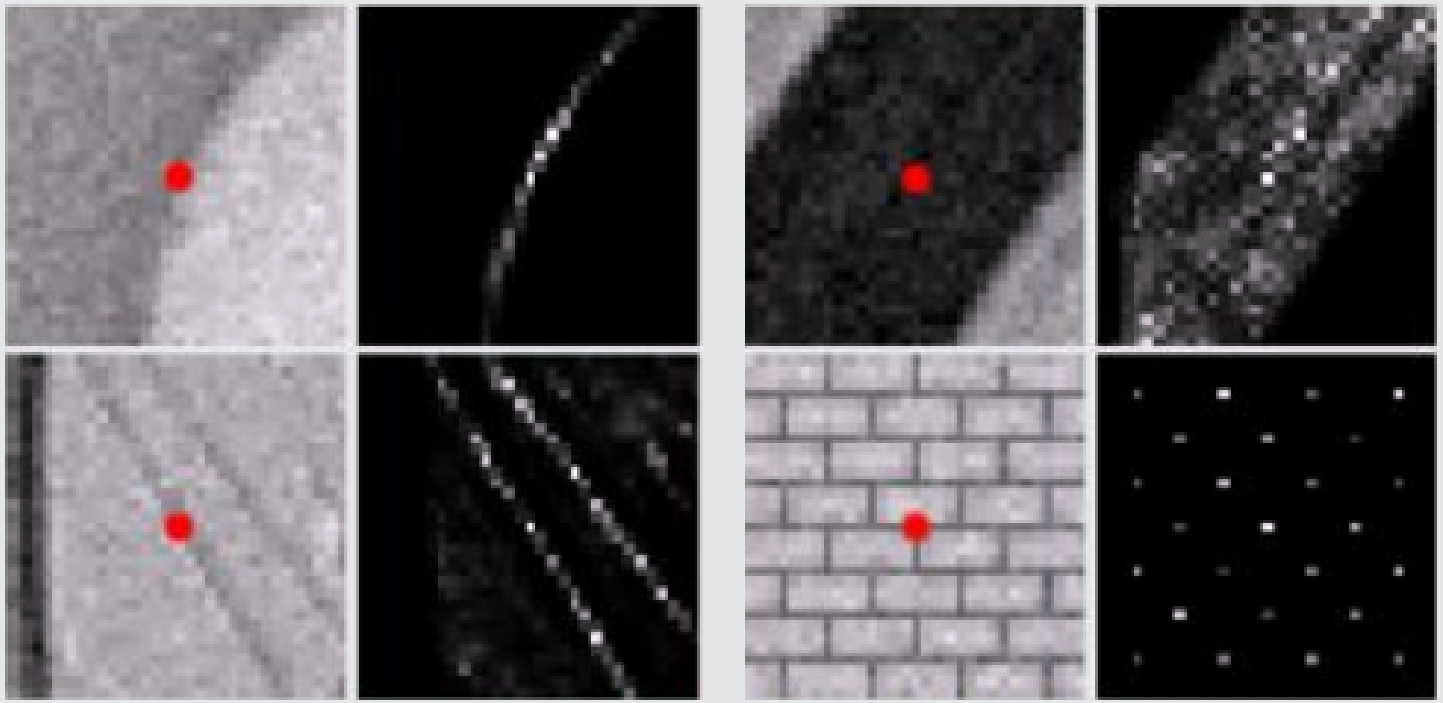
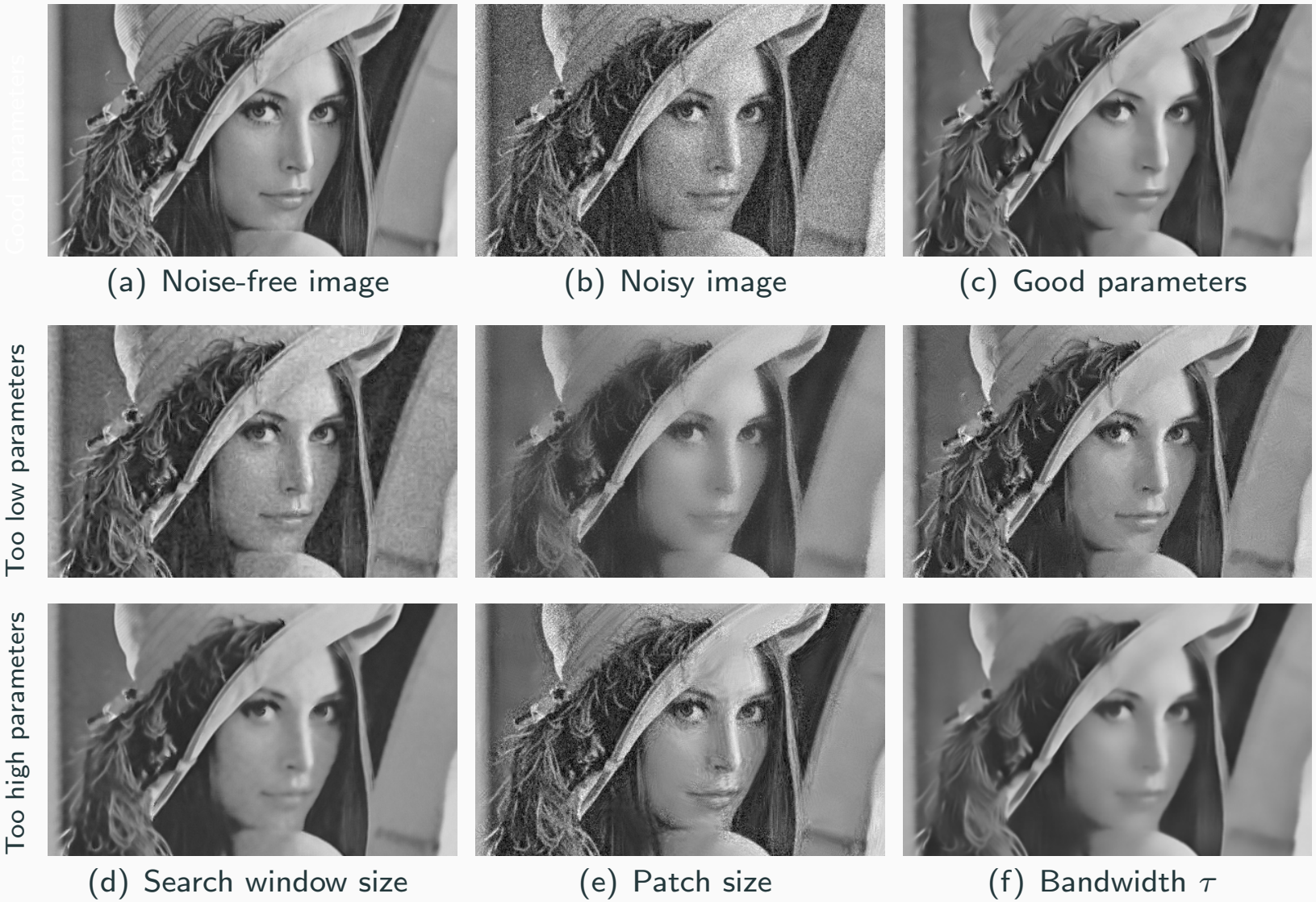


Figure 11 – Image extracted from [Buades et al., 2005]

Spatial filtering – Non-local means



(a) Noise-free image (b) Noisy image (c) Good parameters

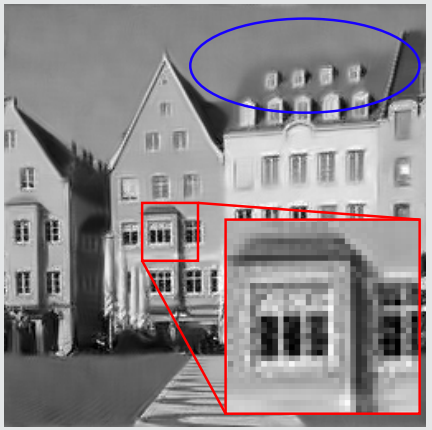
(d) Search window size (e) Patch size (f) Bandwidth τ

Figure 12 – Influence of the three main parameters of the NL means on the solution.

Limitations of NL-means

- ☺ Respect edges
- ☺ Good for texture
- ☹ Remaining noise around rare patches
- ☹ Lose/blur details with low SNR

(a) Noisy image

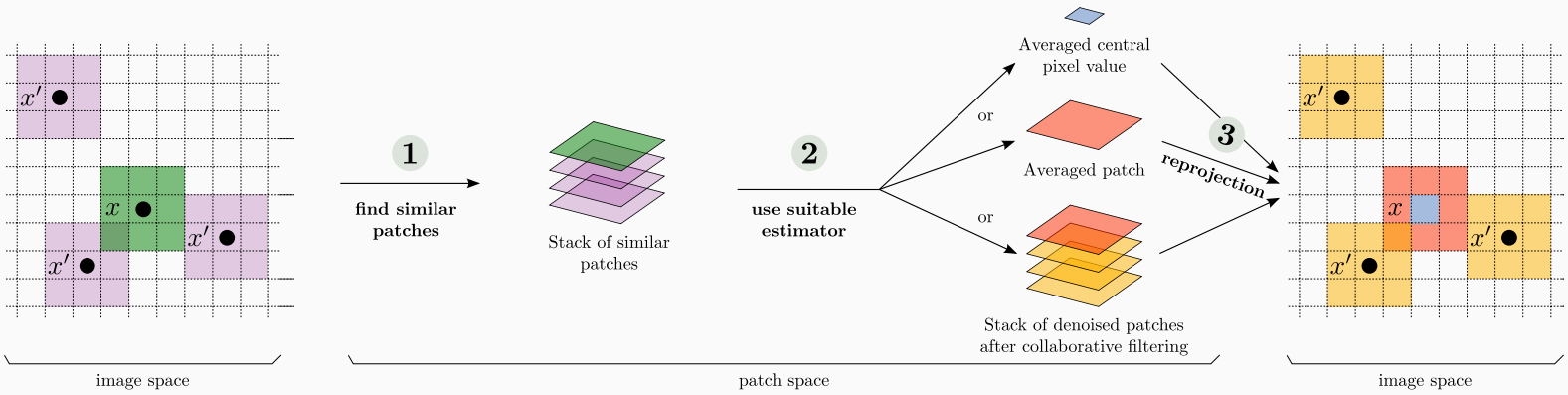


(b) NL-means

(c) BM3D

- ☹ Naive implementation: $O(n|\mathcal{N}||\mathcal{P}|)$ ($\sim$ 1 minute for 256×256 image)
- ☺ Using integral tables: $O(n|\mathcal{N}|)$ (few seconds for 256×256 image)
- ☺ Or FFT: $O(n|\mathcal{N}| \log |\mathcal{N}|)$

Spatial filtering – Extensions of non-local means



More elaborate schemes mostly rely on patches and use more sophisticated estimators than the average

But we will need to study some more of the basics first...