Reprojection for Patch-Based Denoising

Joseph Salmon [1] and Yann Strozecki [1]

[1]Université Paris Diderot - Paris 7

May 2010 - Statistical Models for Images
Luminy

Contents

Introduction and Framework

Framework

- Estimate an image $I^\star$ thanks to a noisy version I corrupted by an additive noise

Introduction and Framework

Framework

- Estimate an image $I^\star$ thanks to a noisy version I corrupted by an additive noise

Historical

- Classical (Old) solution : averaging locally pixels values
- « Patches » approach : using pixels neighborhoods instead of using only pixels
- NL-Means (aka Non Local Means) : Gaussian smoothing in the space of patches

Introduction and Framework

Framework

- Estimate an image $I^\star$ thanks to a noisy version I corrupted by an additive noise

Historical

- Classical (Old) solution : averaging locally pixels values
- « Patches » approach : using pixels neighborhoods instead of using only pixels
- NL-Means (aka Non Local Means) : Gaussian smoothing in the space of patches

Introduction and Framework

Framework

- Estimate an image $I^\star$ thanks to a noisy version I corrupted by an additive noise

Historical

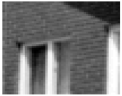
- Classical (Old) solution : averaging locally pixels values
- « Patches » approach : using pixels neighborhoods instead of using only pixels
- NL-Means (aka Non Local Means) : Gaussian smoothing in the space of patches

Images, Noise and Estimation

$I^\star$: Ideal Image $N \times N$

- For a pixel (of interest) $i = (i_1, i_2) \in [\![1, N]\!]^2 = \Omega$
- Grayscale model : $I^\star(i) \in \mathbb{R}$
- Quadratic Loss L_2, associated norm : $\| \cdot \|$

Images, Noise and Estimation



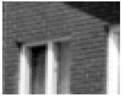
$I^\star$: Ideal Image $N \times N$

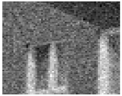
- For a pixel (of interest) $i = (i_1, i_2) \in [\![1, N]\!]^2 = \Omega$
- Grayscale model : $I^\star(i) \in \mathbb{R}$
- Quadratic Loss L_2, associated norm : $\| \cdot \|$

Noisy observation : Regression Model

- $I(i) = I^\star(i) + \sigma\mathcal{E}(i)$
- $\mathcal{E}$ i.i.d. standard Gaussian noise and known σ^2
- Other noises possible...

Images, Noise and Estimation





$I^\star$: Ideal Image $N \times N$

- For a pixel (of interest) $i = (i_1, i_2) \in [\![1, N]\!]^2 = \Omega$
- Grayscale model : $I^\star(i) \in \mathbb{R}$
- Quadratic Loss L_2, associated norm : $\|\cdot\|$

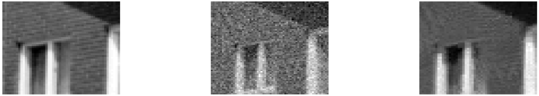
Noisy observation : Regression Model

- $I(i) = I^\star(i) + \sigma\mathcal{E}(i)$
- $\mathcal{E}$ i.i.d. standard Gaussian noise and <u>known σ^2</u>
- Other noises possible...

Estimation

- Estimate $I^\star(i)$ by $\widehat{I}(i)$ thanks to I

Images, Noise and Estimation



$I^\star$: Ideal Image $N \times N$

- For a pixel (of interest) $i = (i_1, i_2) \in [\![1, N]\!]^2 = \Omega$
- Grayscale model : $I^\star(i) \in \mathbb{R}$
- Quadratic Loss L_2, associated norm : $\|\cdot\|$

Noisy observation : Regression Model

- $I(i) = I^\star(i) + \sigma\mathcal{E}(i)$
- $\mathcal{E}$ i.i.d. standard Gaussian noise and known σ^2
- Other noises possible...

Estimation

- Estimate $I^\star(i)$ by $\widehat{I}(i)$ thanks to I

Patch Based Methods

Self - Similarity

- NL-Means : Use weights taking into account patches similarity

- Patch to denoise

Patch Based Methods

Self - Similarity

- NL-Means : Use weights taking into account patches similarity

- Patch to denoise
- Similar Patches :
  Useful for denoising

Patch Based Methods

Self - Similarity

- NL-Means : Use weights taking into account patches similarity

- Patch to denoise
- Similar Patches : Useful for denoising
- Less Similar Patchs : less useful

Patch Based Methods

Self - Similarity

- NL-Means : Use weights taking into account patches similarity

- Patch to denoise
- Similar Patches : Useful for denoising
- Less Similar Patchs : less useful
- Very Different Patches : Useless ! ! !

Patch Based Methods

Self - Similarity

- NL-Means : Use weights taking into account patches similarity

- Patch to denoise
- Similar Patches : Useful for denoising
- Less Similar Patchs : less useful
- Very Different Patches : Useless ! ! !

Denoising with Patches in Three Steps

Patches

- Patch P_i^I : Square Sub-Image "centered" on i, with width W :

$$P_i^I = \left(P_i^I[\delta]\right)_{\delta \in V_W} = \left(I(i+\delta)\right)_{\delta \in V_W}$$

with $\qquad V_W = \{-W_- \leq \delta_1, \delta_2 \leq W_+\}$

- Link patch/pixel : $I(i) = P_i^I[0]$

Denoising with Patches in Three Steps

Patches

- Patch P_i^I : Square Sub-Image "centered" on i, with width W :

$$P_i^I = \left(P_i^I[\delta]\right)_{\delta \in V_W} = \left(I(i+\delta)\right)_{\delta \in V_W}$$

with $$V_W = \{-W_- \leq \delta_1, \delta_2 \leq W_+\}$$

- Link patch/pixel : $I(i) = P_i^I[0]$

The Three Steps

Denoising with Patches in Three Steps

Patches

- Patch P_i^I : Square Sub-Image "centered" on i, with width W :

$$P_i^I = \left(P_i^I[\delta]\right)_{\delta \in V_W} = \left(I(i+\delta)\right)_{\delta \in V_W}$$

with $\qquad V_W = \{-W_- \leq \delta_1, \delta_2 \leq W_+\}$

- Link patch/pixel : $I(i) = P_i^I[0]$

The Three Steps

1. **Patchization** : create a collection of patches from the image $\left(I(i)\right)_{i \in \Omega} \mapsto \left(P_i^I\right)_{i \in \Omega}$

Denoising with Patches in Three Steps

Patches

- Patch P_i^I : Square Sub-Image "centered" on i, with width W :

$$P_i^I = \left(P_i^I[\delta]\right)_{\delta \in V_W} = \left(I(i+\delta)\right)_{\delta \in V_W}$$

with $$V_W = \{-W_- \leq \delta_1, \delta_2 \leq W_+\}$$

- Link patch/pixel : $I(i) = P_i^I[0]$

The Three Steps

1. **Patchization** : create a collection of patches from the image $\left(I(i)\right)_{i\in\Omega} \mapsto \left(P_i^I\right)_{i\in\Omega}$
2. **Denoising** : use patches redundancy to denoise this collection $\left(P_i^I\right)_{i\in\Omega} \mapsto \left(\widehat{P_i^I}\right)_{i\in\Omega}$

Denoising with Patches in Three Steps

Patches

- Patch P_i^I : Square Sub-Image "centered" on i, with width W :

$$P_i^I = \left(P_i^I[\delta]\right)_{\delta \in V_W} = \left(I(i+\delta)\right)_{\delta \in V_W}$$

with $$V_W = \{-W_- \leq \delta_1, \delta_2 \leq W_+\}$$

- Link patch/pixel : $I(i) = P_i^I[0]$

The Three Steps

1. **Patchization** : create a collection of patches from the image $\left(I(i)\right)_{i \in \Omega} \mapsto \left(P_i^I\right)_{i \in \Omega}$
2. **Denoising** : use patches redundancy to denoise this collection $\left(P_i^I\right)_{i \in \Omega} \mapsto \left(\widehat{P_i^I}\right)_{i \in \Omega}$
3. **Reprojection** : from the patches collection to an image : $\left(\widehat{P_i^I}\right)_{i \in \Omega} \mapsto \left(\tilde{I}(i)\right)_{i \in \Omega}$

Denoising with Patches in Three Steps

Patches

- Patch P_i^I : Square Sub-Image "centered" on i, with width W :

$$P_i^I = \left(P_i^I[\delta]\right)_{\delta \in V_W} = \left(I(i+\delta)\right)_{\delta \in V_W}$$

with $$V_W = \{-W_- \leq \delta_1, \delta_2 \leq W_+\}$$

- Link patch/pixel : $I(i) = P_i^I[0]$

The Three Steps

1. **Patchization** : create a collection of patches from the image $\left(I(i)\right)_{i\in\Omega} \mapsto \left(P_i^I\right)_{i\in\Omega}$
2. **Denoising** : use patches redundancy to denoise this collection $\left(P_i^I\right)_{i\in\Omega} \mapsto \left(\widehat{P_i^I}\right)_{i\in\Omega}$
3. **Reprojection** : from the patches collection to an image : $\left(\widehat{P_i^I}\right)_{i\in\Omega} \mapsto \left(\tilde{I}(i)\right)_{i\in\Omega}$

The three steps

Literature

1. **Patchization** :
   - Most convenient=Square neighborhood, see NL-Means [BCM05], BM3D [DFKE07] , Julien Mairal's Method [MBP+09]
   - “Funny shapes“ Shape Adaptive BM3D [DFKE08]

The three steps

Literature

1. **Patchization** :
   - Most convenient=Square neighborhood, see NL-Means [BCM05], BM3D [DFKE07] , Julien Mairal's Method [MBP$^+$09]
   - "Funny shapes" Shape Adaptive BM3D [DFKE08]
2. **Denoising** :
   - Averaging similar patches : NL-Means [BCM05]
   - Wavelet thresholding on a block of similar patches [DFKE07]
   - Dictionary learning on a block of similar patches, through Lasso based techniques [MBP$^+$09] (cf. next talk)

The three steps

Literature

1. **Patchization** :
   - Most convenient=Square neighborhood, see NL-Means [BCM05], BM3D [DFKE07] , Julien Mairal's Method [MBP$^+$09]
   - "Funny shapes" Shape Adaptive BM3D [DFKE08]
2. **Denoising** :
   - Averaging similar patches : NL-Means [BCM05]
   - Wavelet thresholding on a block of similar patches [DFKE07]
   - Dictionary learning on a block of similar patches, through Lasso based techniques [MBP$^+$09] (cf. next talk)
3. **Reprojection** : Main subject of the present talk

The three steps

Literature

1. **Patchization** :
   - Most convenient=Square neighborhood, see NL-Means [BCM05], BM3D [DFKE07] , Julien Mairal's Method [MBP$^+$09]
   - "Funny shapes" Shape Adaptive BM3D [DFKE08]
2. **Denoising** :
   - Averaging similar patches : NL-Means [BCM05]
   - Wavelet thresholding on a block of similar patches [DFKE07]
   - Dictionary learning on a block of similar patches, through Lasso based techniques [MBP$^+$09] (cf. next talk)
3. **Reprojection** : Main subject of the present talk

Non Local Means (NL-Means)

Averaging Patches

- Estimate a patch (of interest) by averaging similar patches :

$$\widehat{P_i^I} = \sum_{k \in \Omega} \theta_{i,k} P_k^I$$

- Patchs useless if $\theta_{i,k}$ does not depend on P_k^I and P_i^I !

Non Local Means (NL-Means)

Averaging Patches

- Estimate a patch (of interest) by averaging similar patches :

$$\widehat{P_i^I} = \sum_{k \in \Omega} \theta_{i,k} P_k^I$$

- Patchs useless if $\theta_{i,k}$ does not depend on P_k^I and P_i^I !

"Limited Range" [Tas09] : Searching/Influence Zone

Average in a "small" searching zone Ω_R (if enough time take $\Omega_R = \Omega$) :

$$\widehat{P_i^I} = \sum_{k \in \Omega_R} \theta_{i,k} P_k^I$$

Non Local Means (NL-Means)

Averaging Patches

- Estimate a patch (of interest) by averaging similar patches :

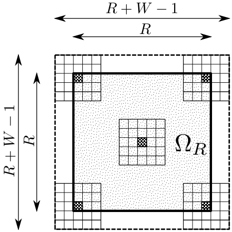
$$\widehat{P_i^I} = \sum_{k \in \Omega} \theta_{i,k} P_k^I$$

- Patchs useless if $\theta_{i,k}$ does not depend on P_k^I and P_i^I !

"Limited Range" [Tas09] : Searching/Influence Zone

Average in a "small" searching zone Ω_R (if enough time take $\Omega_R = \Omega$) :

$$\widehat{P_i^I} = \sum_{k \in \Omega_R} \theta_{i,k} P_k^I$$



NL-Means in 3 parameters : Patch (W), Searching (R), Bandwidth (h)

Easy to implement and to explain : few parameters !

NL-Means [BCM05]

- Use weights defined by $\theta_{i,k} = \frac{K_h(\|P_i^I - P_k^I\|)}{\sum_{k'} K_h(\|P_i^I - P_{k'}^I\|)}$
- Gaussian Kernel $K_h(x) = \exp(-x^2/h^2)$ with bandwidth h

NL-Means in 3 parameters : Patch (W), Searching (R), Bandwidth (h)

Easy to implement and to explain : few parameters !

NL-Means [BCM05]

- Use weights defined by $\theta_{i,k} = \frac{K_h(\|P_i^I - P_k^I\|)}{\sum_{k'} K_h(\|P_i^I - P_{k'}^I\|)}$
- Gaussian Kernel $K_h(x) = \exp(-x^2/h^2)$ with bandwidth h

Common parameters choices for 512×512 images

- Patch width : $1 \leq W \leq 16$ (W increases with σ)
- Searching zone width : $10 \leq R \leq \cdots$ ($R = 21$ [BCM05])
- Kernel bandwidth h : harder to choose, increases with σ, often $h \propto \sigma$

NL-Means in 3 parameters : Patch (W), Searching (R), Bandwidth (h)

Easy to implement and to explain : few parameters !

NL-Means [BCM05]

- Use weights defined by $\theta_{i,k} = \frac{K_h(\|P_i^I - P_k^I\|)}{\sum_{k'} K_h(\|P_i^I - P_{k'}^I\|)}$
- Gaussian Kernel $K_h(x) = \exp(-x^2/h^2)$ with bandwidth h

Common parameters choices for 512×512 images

- Patch width : $1 \leq W \leq 16$ (W increases with σ)
- Searching zone width : $10 \leq R \leq \cdots$ ($R = 21$ [BCM05])
- Kernel bandwidth h : harder to choose, increases with σ, often $h \propto \sigma$

"Limited Range NL-Means" : R

The Larger R the longer the algorithm

- Time limit : Algorithmic Complexity $O(N^2R^2)$, leading to choose a small searching zone Ω_R

"Limited Range NL-Means" : R

The Larger R the longer the algorithm

- Time limit : Algorithmic Complexity $O(N^2R^2)$, leading to choose a small searching zone Ω_R

The Larger R the better the algorithm ?

- Performance limit : too many small weights damage the procedure...

"Limited Range NL-Means" : R

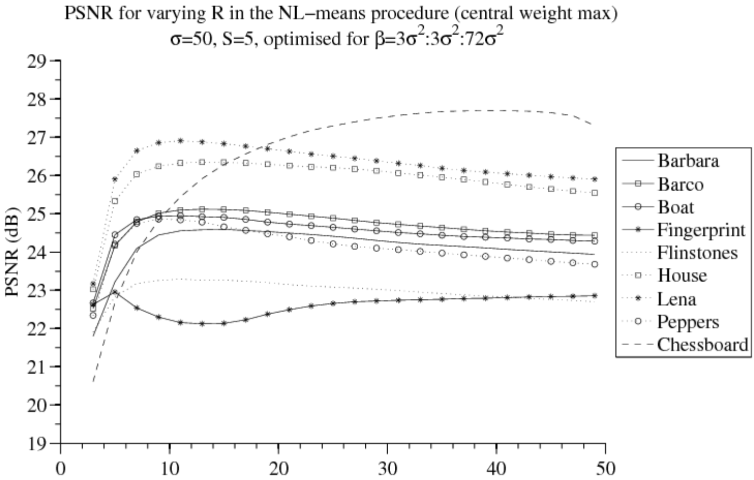
The Larger R the longer the algorithm

- Time limit : Algorithmic Complexity $O(N^2 R^2)$, leading to choose a small searching zone Ω_R

The Larger R the better the algorithm ?

- Performance limit : too many small weights damage the procedure...

Influence of R



$\sigma = 50$, $W = 5$, oracle h (optimized)

Kernel choice K

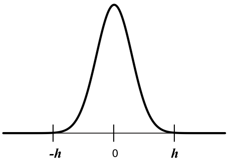
Usual Suspects $\cdots$

Kernel choice K

Usual Suspects $\cdots$

Kernel choice K

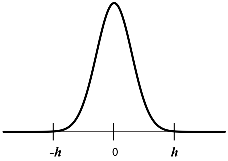
Usual Suspects $\cdots$



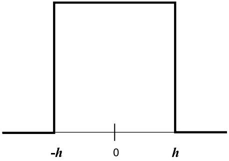
(a) Gaussian Kernel

Kernel choice K

Usual Suspects $\cdots$



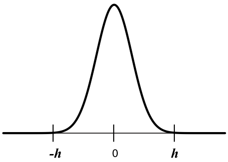
(a) Gaussian Kernel



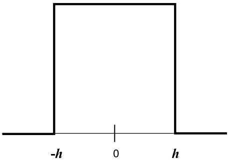
(b) Flat Kernel

Kernel choice K

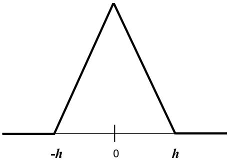
Usual Suspects $\cdots$



(a) Gaussian Kernel



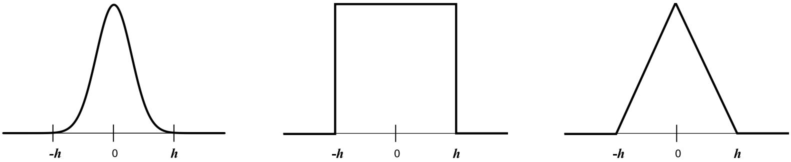
(b) Flat Kernel



(c) Triangular Kernel

Kernel choice K

Usual Suspects $\cdots$



(a) Gaussian Kernel (b) Flat Kernel (c) Triangular Kernel

Previous slide : rather use a **compact** kernel since small weights degrade the performance

Patch of Interest's weight(s)

Comparing a patch with itself : "Central Weight Problem"

- Maximum weight $\theta_{i,j}$ reached for $i = j$, with K symmetric and non-increasing on $\mathbb{R}^+$
- Overestimate the patch of interest weight

Patch of Interest's weight(s)

Comparing a patch with itself : "Central Weight Problem"

- Maximum weight $\theta_{i,j}$ reached for $i = j$, with K symmetric and non-increasing on $\mathbb{R}^+$
- Overestimate the patch of interest weight

Experiment :

Patch of Interest's weight(s)

Comparing a patch with itself : "Central Weight Problem"

- Maximum weight $\theta_{i,j}$ reached for $i = j$, with K symmetric and non-increasing on $\mathbb{R}^+$
- Overestimate the patch of interest weight

Experiment :

Central weight (before normalisation)

- "Max" : $\theta_{i,i} = \max_{j \neq i} \theta_{i,j}$ [BCM05]
- Unbiased Risk Estimate Weights [Sal10] :

$$\mathbb{E}(\|P_j^I - P_i^I\|^2 - 2\sigma^2 W^2) = \|P_j^{I^\star} - P_i^{I^\star}\|^2$$

  With Gaussian kernel : Substitute $\theta_{i,i} = K_h(2W^2\sigma^2)$ to $K_h(0)$, others unchanged
- Normal Weights : no weight changed
- Zero Weights : $\theta_{i,i} = 0$ (standard measurement)

Patch of Interest's weight(s)

Comparing a patch with itself : "Central Weight Problem"

- Maximum weight $\theta_{i,j}$ reached for $i = j$, with K symmetric and non-increasing on $\mathbb{R}^+$
- Overestimate the patch of interest weight

Experiment :

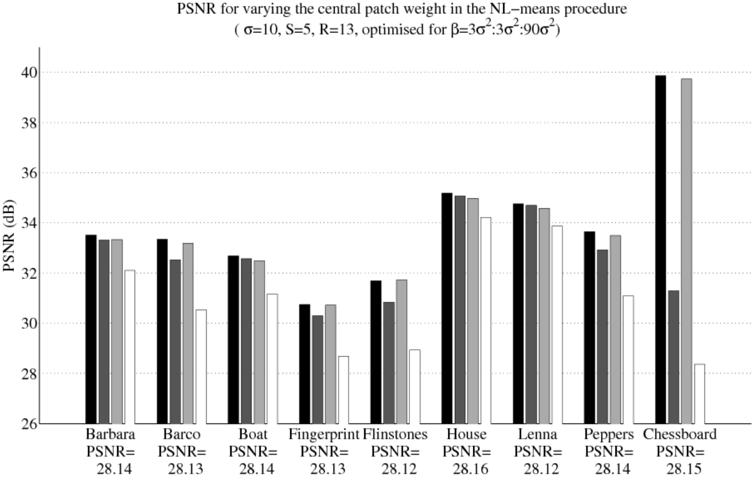
Central weight (before normalisation)

- "Max" : $\theta_{i,i} = \max_{j \neq i} \theta_{i,j}$ [BCM05]
- Unbiased Risk Estimate Weights [Sal10] :

$$\mathbb{E}(\|P_j^I - P_i^I\|^2 - 2\sigma^2 W^2) = \|P_j^{I^\star} - P_i^{I^\star}\|^2$$

  With Gaussian kernel : Substitute $\theta_{i,i} = K_h(2W^2\sigma^2)$ to $K_h(0)$, others unchanged
- Normal Weights : no weight changed
- Zero Weights : $\theta_{i,i} = 0$ (standard measurement)

Central Weight and PSNR



$\sigma = 10$

Flat Kernel Advantages

Previous slide : use a kernel constant around 0 to avoid the Central Weight problem

Keep or Kill the patches

- $K_h(x) = \mathbb{1}_{[-1,1]}(\frac{x}{h})$
- Same weights for all selected patches

Flat Kernel Advantages

Previous slide : use a kernel constant around 0 to avoid the Central Weight problem

Keep or Kill the patches

- $K_h(x) = \mathbb{1}_{[-1,1]}(\frac{x}{h})$
- Same weights for all selected patches

Selecting the bandwidth h with hypothesis testing

- H_0 :"$P_i^{I^\star} = P_j^{I^\star}$" vs H_1 :"$P_i^{I^\star} \neq P_j^{I^\star}$"
- Under H_0, $\|P_i^I - P_j^I\|^2/(2\sigma^2) \sim \chi^2(W^2)$
- Choose $h^2 = 2\sigma^2 q_\alpha^{W^2}$ for a confidence level α (e.g $\alpha = 99\%$)

Flat Kernel Advantages

Previous slide : use a kernel constant around 0 to avoid the Central Weight problem

Keep or Kill the patches

- $K_h(x) = \mathbb{1}_{[-1,1]}(\frac{x}{h})$
- Same weights for all selected patches

Selecting the bandwidth h with hypothesis testing

- H_0 :"$P_i^{I^\star} = P_j^{I^\star}$" vs H_1 :"$P_i^{I^\star} \neq P_j^{I^\star}$"
- Under H_0, $\|P_i^I - P_j^I\|^2/(2\sigma^2) \sim \chi^2(W^2)$
- Choose $h^2 = 2\sigma^2 q_\alpha^{W^2}$ for a confidence level α (e.g $\alpha = 99\%$)

Other Advantages...

- Easy to interpret. $N_i = \sum_k K_h(\|P_i^I - P_k^I\|)$ number of selected patches =weights normalization
- Easy to implement ...

Flat Kernel Advantages

Previous slide : use a kernel constant around 0 to avoid the Central Weight problem

Keep or Kill the patches

- $K_h(x) = \mathbb{1}_{[-1,1]}(\frac{x}{h})$
- Same weights for all selected patches

Selecting the bandwidth h with hypothesis testing

- H_0 :"$P_i^{I^\star} = P_j^{I^\star}$" vs H_1 :"$P_i^{I^\star} \neq P_j^{I^\star}$"
- Under H_0, $\|P_i^I - P_j^I\|^2/(2\sigma^2) \sim \chi^2(W^2)$
- Choose $h^2 = 2\sigma^2 q_\alpha^{W^2}$ for a confidence level α (e.g $\alpha = 99\%$)

Other Advantages...

- Easy to interpret. $N_i = \sum_k K_h(\|P_i^I - P_k^I\|)$ number of selected patches =weights normalization
- Easy to implement ...

Seminal Reprojection I : Central

Central Reprojection

- Define a center for each patch : $P_i^I[0]$

(a) Odd width (b) Even width

- Use the same weights measuring the similarity of patches for the similarity of the centers :

$$\widehat{I}(i) = \sum_{k \in \Omega_R} \theta_{i,k} P_k^I[0] = \widehat{P_i^I}[0]$$

Central Reprojection in Practice

(a) Original

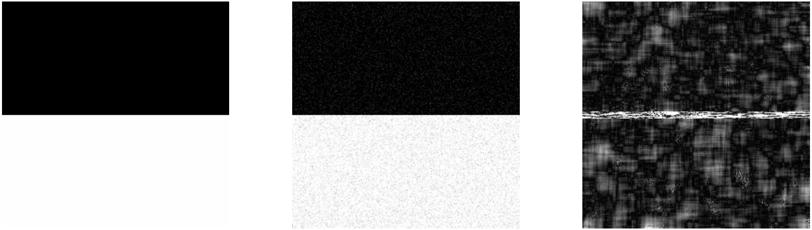
(b) Noisy PSNR $= 22.07$

(c) PSNR $= 27.60$

FIG.: (a) Original image, (b) Noisy image with $\sigma = 20$, (c) Denoised image (**Flat kernel being used until the end**)

Parameters : $R = 9$, $W = 9$, $h^2 = 2\sigma^2 q_{0.99}^{W^2}$.

Edges Artifacts on a Toy Example



(a) Original (b) Noisy PSNR $= 22.07$ (c) PSNR $= 45.78$

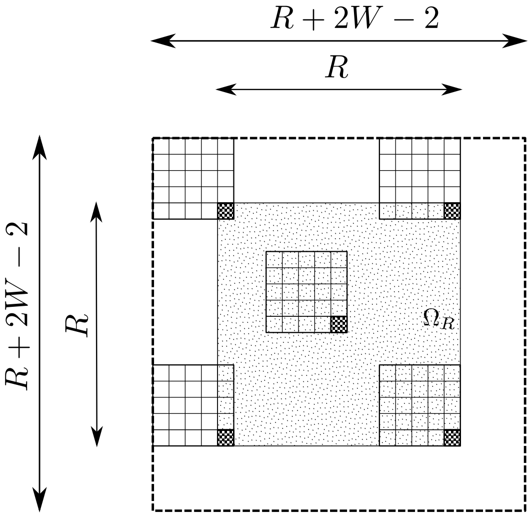
FIG.: (a) Original image, (b) Noisy image with $\sigma = 20$, (c) Absolute difference between the denoised image and the original image (the whiter the worse)

Parameters : $R = 21,\ W = 9, h^2 = 2\sigma^2 q_{0.99}^{W^2}$.

Edges Artifacts on Natural Images

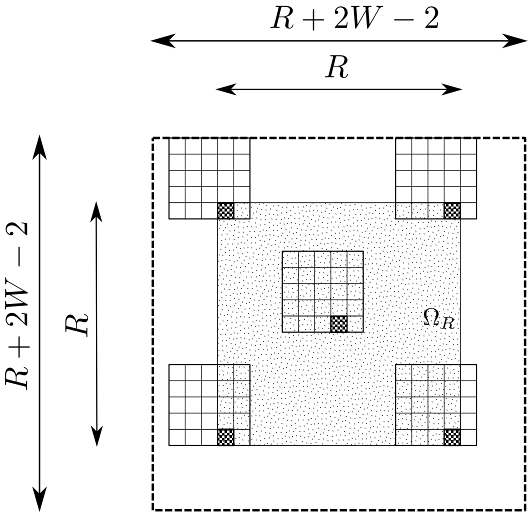
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



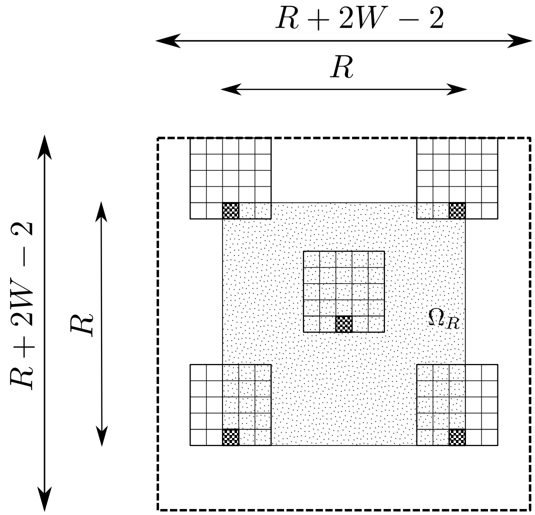
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



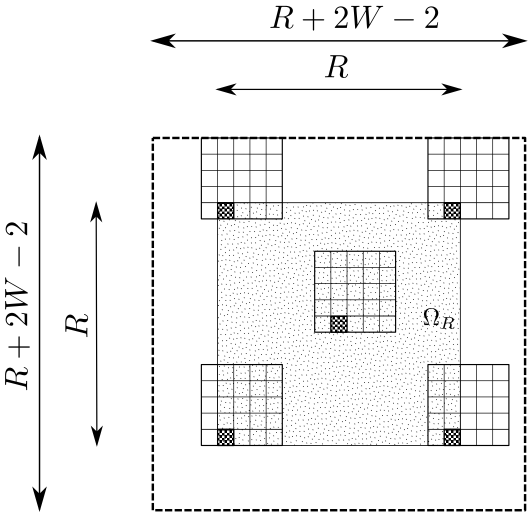
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



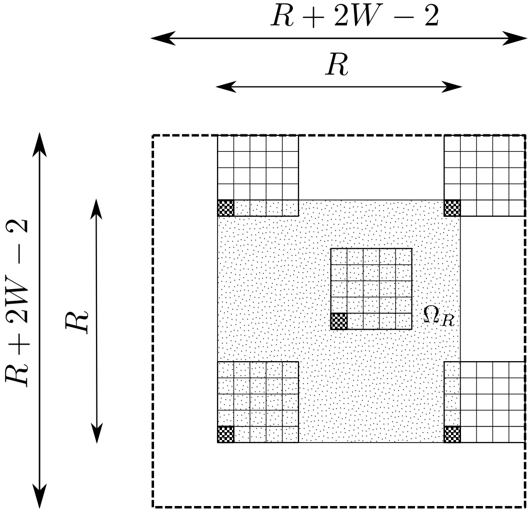
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



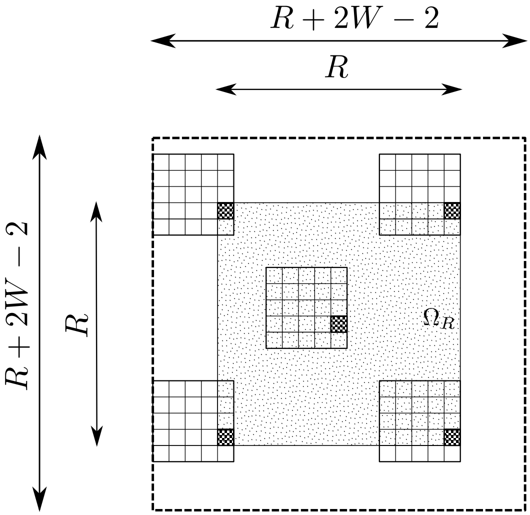
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



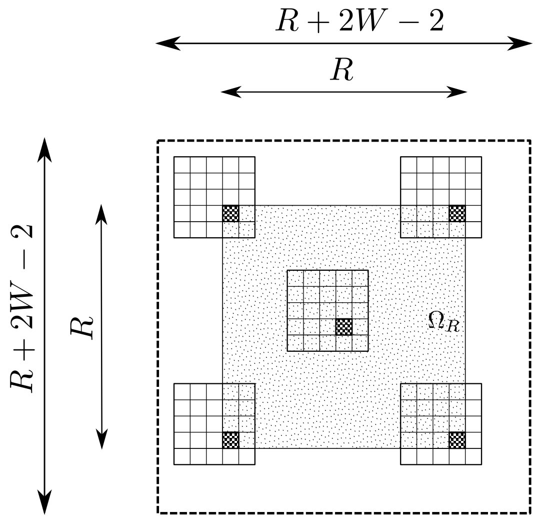
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



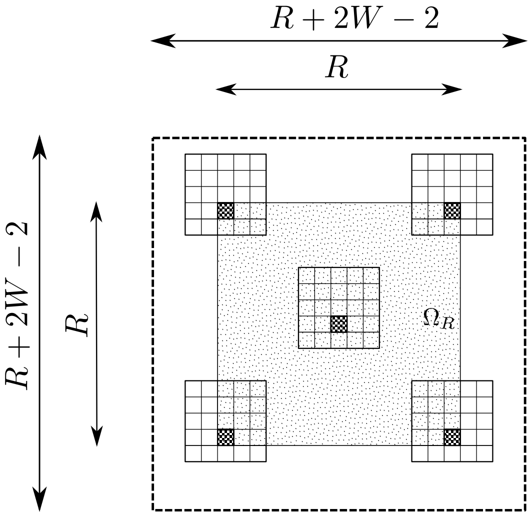
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



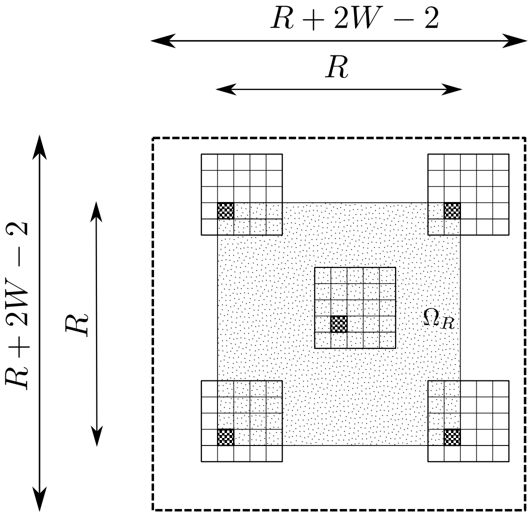
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



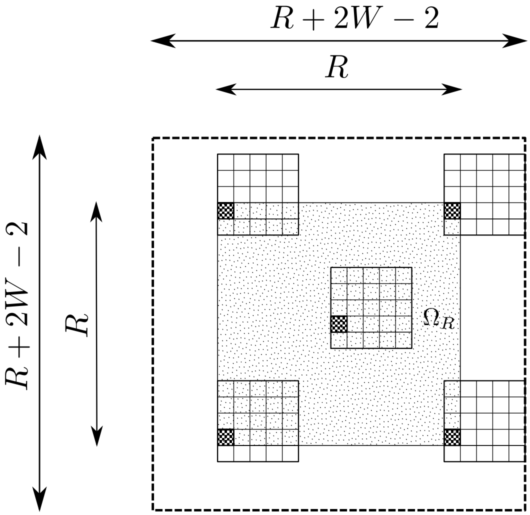
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



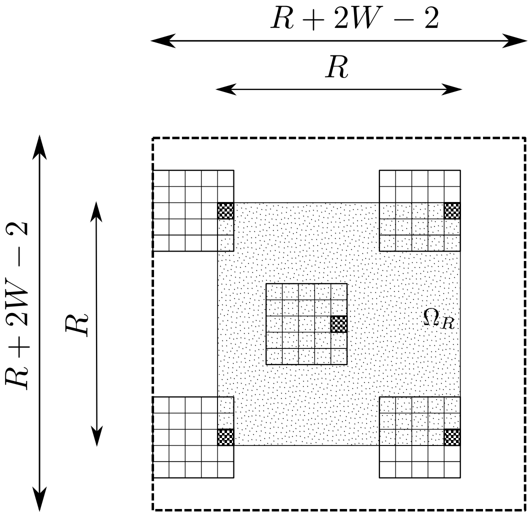
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



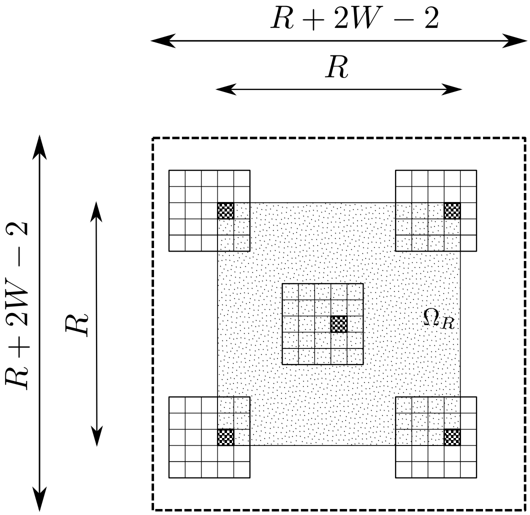
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



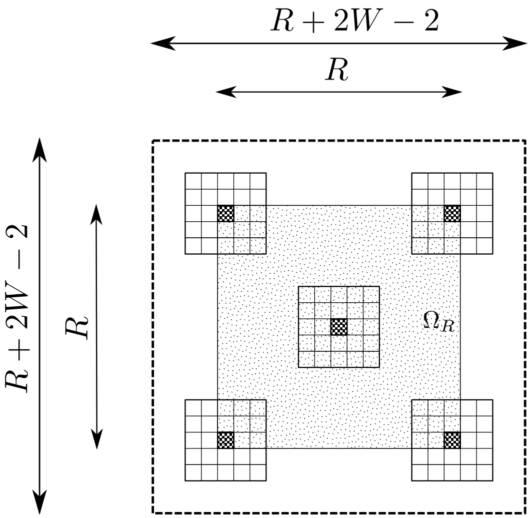
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



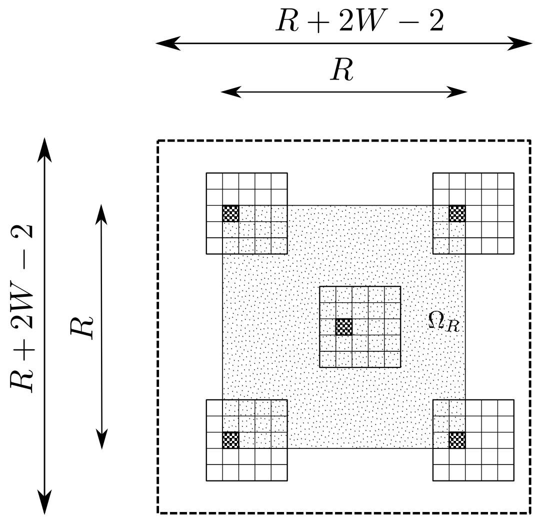
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



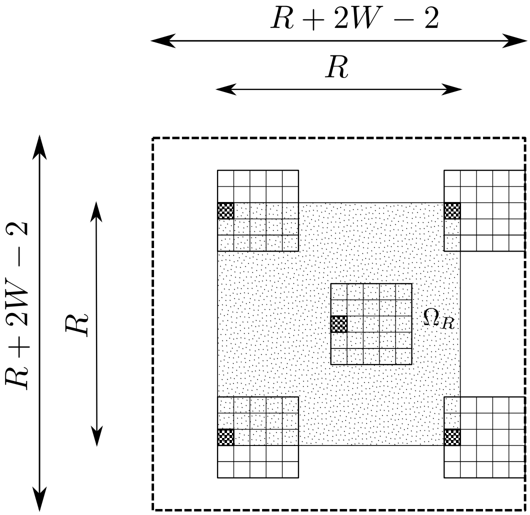
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



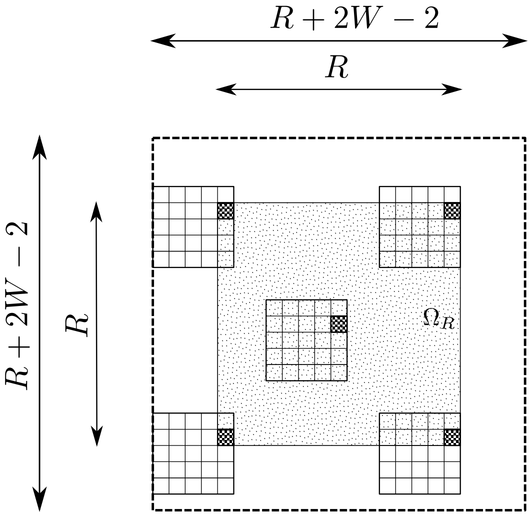
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



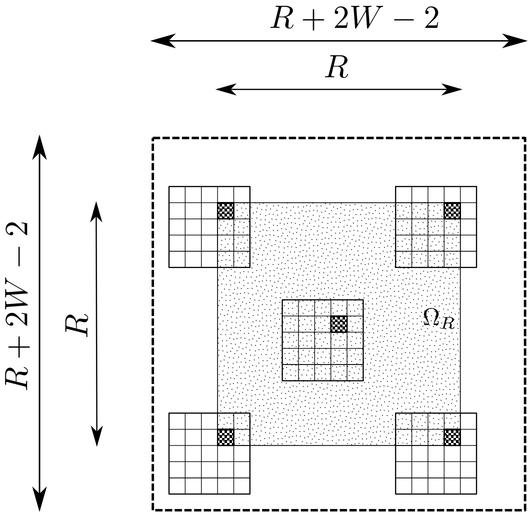
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



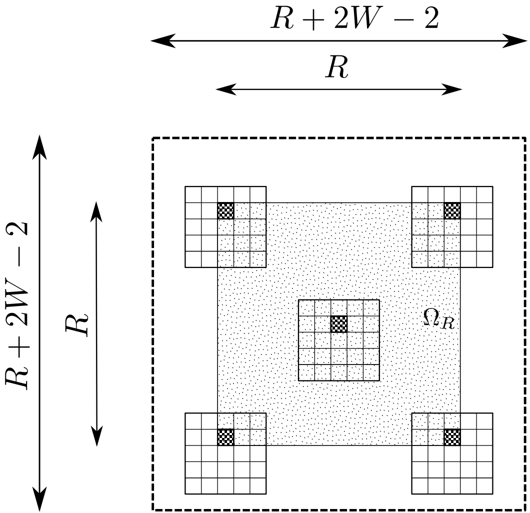
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



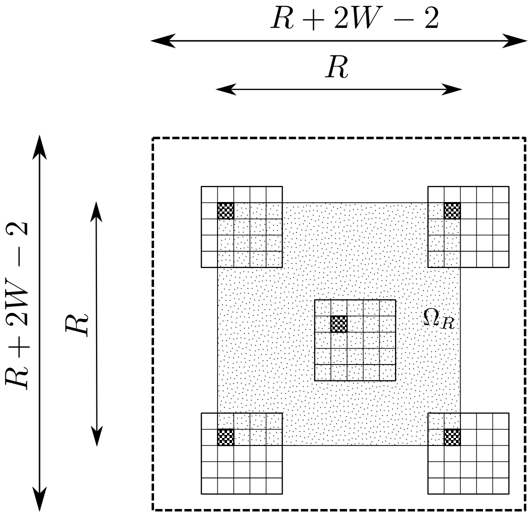
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



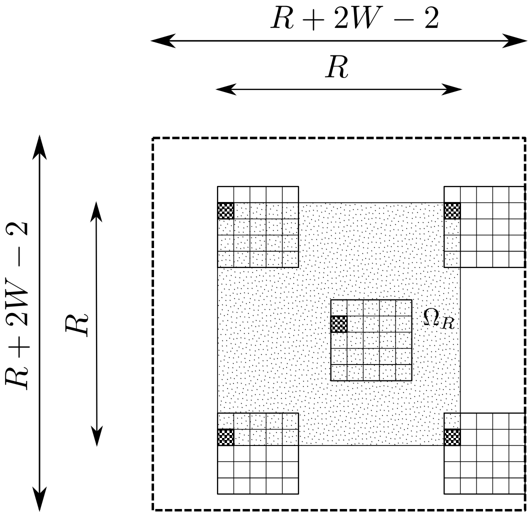
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



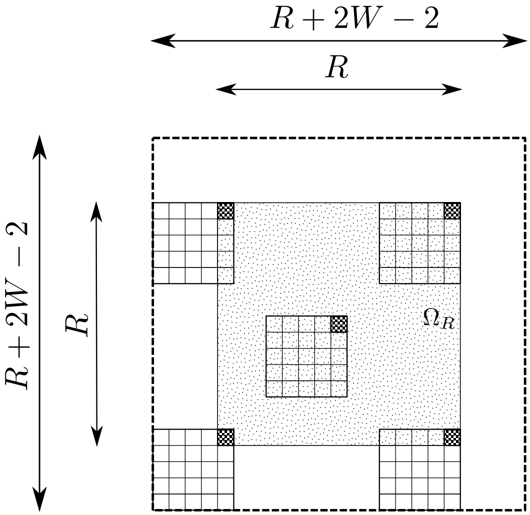
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



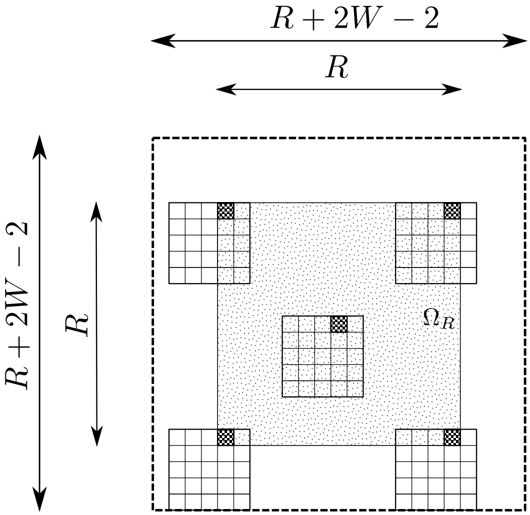
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



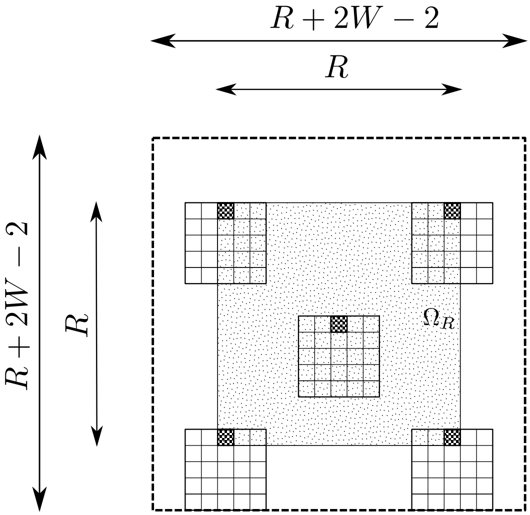
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



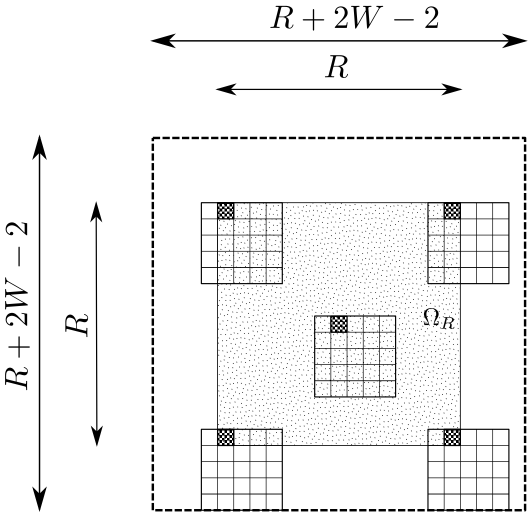
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



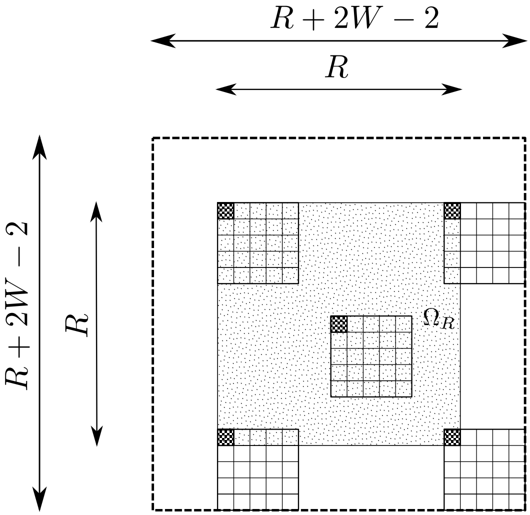
Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



Searching Zone, Influence Zone and Patches

Patches width $W = 5$, searching zone width $R = 15$



Seminal Reprojection II : Uniform Average of Estimators (Uae)

Uae Reprojection [BCM05]

- Each pixel belongs to W^2 patches $\Rightarrow$ each pixel has W^2 estimators :

$$\forall \delta \in V_W, I(i) = P^I_{i-\delta}[\delta]$$

- Average those estimators with uniform weights

$$\widehat{I}_{\mathrm{Uae}}(i) = \frac{1}{|V_W|} \sum_{\delta \in V_W} \widehat{P^I_{i-\delta}}[\delta]$$

- In practice : important PSNR improvement

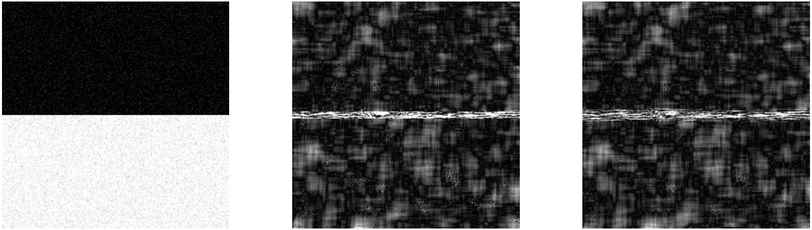
Uae : Natural Images

(a) Noisy (b) Central PSNR $= 27.60$ (c) Uae PSNR $= 28.65$

FIG.: (a) Noisy image ($\sigma = 20$), (b) Central Reprojection, (c) Uae Reprojection.

Parameters : $R = 9,\ W = 9, h^2 = 2\sigma^2 q_{0.99}^{W^2}$.

Uae : Edges Artifacts on a Toy Image



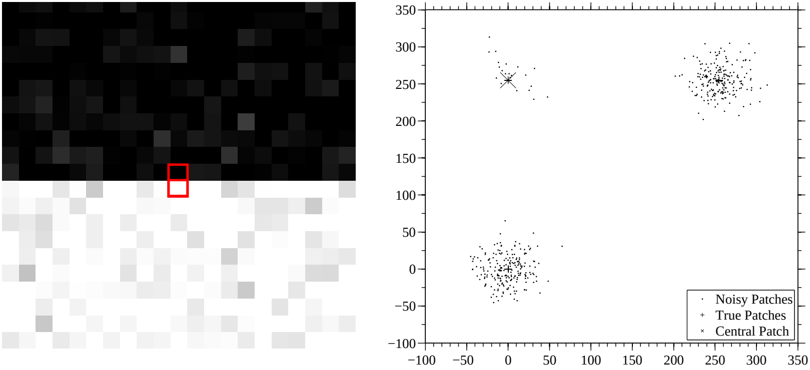
(a) Noisy (b) Central PSNR $= 45.78$ (c) Uae PSNR $= 46,51$

FIG.: (a) Noisy image ($\sigma = 20$), (b) Central Reprojection, (c) Uae Reprojection.

Parameters : $R = 21,\ W = 9, h^2 = 2\sigma^2 q_{0.99}^{W^2}$.

Uae : Edges Artifacts on Natural Images

Artifacts Explanation



(a) Toy Image Close Up (b) Clusters of Patches

FIG.: (a) Searching zone width $R = 21$, $\sigma = 20$, (b) Patches distribution in the searching zone (a). Patches are vertical with size 2×1

Minimizing Variance Reprojection

Naive Idea : assume the bias ≈ 0, choose the estimate with minimum variance

Minimum Variance

$$\widehat{I}_{\mathrm{Min}}(i) = \widehat{P^I_{i-\hat{\delta}}}[\hat{\delta}]$$

$$\text{with} \qquad \hat{\delta} = \arg\min_{\delta \in V_M} \mathrm{Var}\left(\widehat{P^I_{i-\delta}}[\delta]\right)$$

- Rough approximation of the variance [KB06] :

$$\mathrm{Var}\left(\widehat{P^I_{i-\delta}}[\delta]\right) \approx \frac{\sum_k \theta^2_{i-\delta,k}}{(\sum_k \theta_{i-\delta,k})^2}\sigma^2$$

- Flat Kernel case : Choose the estimator selecting the maximum number of patches (candidates)

$$\mathrm{Var}\left(\widehat{P^I_{i-\delta}}[\delta]\right) \approx \frac{\sigma^2}{N_{i-\delta}}$$

Minimizing Variance Reprojection

Naive Idea : assume the bias ≈ 0, choose the estimate with minimum variance

Minimum Variance

$$\widehat{I}_{\mathrm{Min}}(i) = \widehat{P^I_{i-\hat{\delta}}}[\hat{\delta}]$$

with $$\hat{\delta} = \arg\min_{\delta \in V_M} \mathrm{Var}\left(\widehat{P^I_{i-\delta}}[\delta]\right)$$

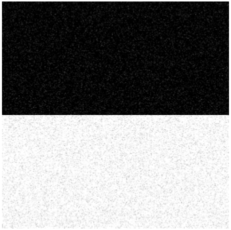
- Rough approximation of the variance [KB06] :

$$\mathrm{Var}\left(\widehat{P^I_{i-\delta}}[\delta]\right) \approx \frac{\sum_k \theta^2_{i-\delta,k}}{\left(\sum_k \theta_{i-\delta,k}\right)^2}\sigma^2$$

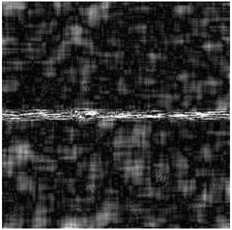
- Flat Kernel case : Choose the estimator selecting the maximum number of patches (candidates)

$$\mathrm{Var}\left(\widehat{P^I_{i-\delta}}[\delta]\right) \approx \frac{\sigma^2}{N_{i-\delta}}$$

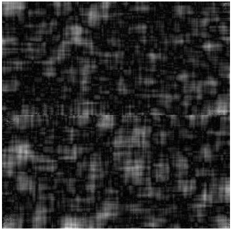
Minimizing Variance Reprojection in Practice



(a) Noisy



(b) Uae :PSNR $= 46, 51$



(c) Min : PSNR $= 48.10$

FIG.: (a) Noisy image ($\sigma = 20$), (b) Uae Reprojection , (c) Min Reprojection.

Parameters : $R = 21,\ W = 9, h^2 = 2\sigma^2 q_{0.99}^{W^2}$.

Minimizing Variance Reprojection in Practice

(a) Noisy (b) Uae : $\underline{\text{PSNR} = 28.65}$ (c) Min : $\underline{\text{PSNR} = 27.09}$

FIG.: (a) Noisy image ($\sigma = 20$), (b) Uae Reprojection , (c) Min Reprojection.

Parameters : $R = 9,\ W = 9, h^2 = 2\sigma^2 q_{0.99}^{W^2}$.

Crenelated Edges

Weighted Averaged Reprojection (Wav)

Naive Idea II : assume bias ≈ 0, and remember that an average of unbiased is unbiased. Thus take the weighted average minimizing the variance :

Wav Reprojection

$$\widehat{I}_{\mathrm{Wav}}(i) = \sum_{\delta \in V_W} \beta^\star_\delta \widehat{P^I_{i-\delta}}[\delta]$$

$$\text{where} \quad (\beta^\star_\delta)_{\delta \in V_W} = \underset{\beta \in \mathbb{R}^{W^2}}{\arg\min} \operatorname{Var}\left(\sum_{\delta \in V_W} \beta_\delta \widehat{P^I_{i-\delta}}[\delta] \right)$$

s.t. $$\sum_{\delta \in V_W} \beta_\delta = 1$$

Weighted Averaged Reprojection (Wav)

Naive Idea II : assume bias ≈ 0, and remember that an average of unbiased is unbiased. Thus take the weighted average minimizing the variance :

Wav Reprojection

$$\widehat{I}_{\mathrm{Wav}}(i) = \sum_{\delta \in V_W} \beta^\star_\delta \widehat{P^I_{i-\delta}}[\delta]$$

$$\text{where} \quad (\beta^\star_\delta)_{\delta \in V_W} = \underset{\beta \in \mathbb{R}^{W^2}}{\arg\min} \operatorname{Var}\left(\sum_{\delta \in V_W} \beta_\delta \widehat{P^I_{i-\delta}}[\delta]\right)$$

s.t.

$$\sum_{\delta \in V_W} \beta_\delta = 1$$

Weighted Averaged Reprojection

Lagrangian Optimization

$$\beta_\delta^\star = \left[\mathrm{Var}\left(\widehat{P_{i-\delta}^I}[\delta]\right)\right]^{-1} / \sum_{\delta \in V_W} \left[\mathrm{Var}\left(\widehat{P_{i-\delta}^I}[\delta]\right)\right]^{-1}$$

- Rough approximation of the variance :
$$\mathrm{Var}\left(\widehat{P_{i-\delta}^I}[\delta]\right) \approx \frac{\sum_{\Omega_R} \theta_{i,k}^2}{\left(\sum_{\Omega_R} \theta_{i,k}\right)^2} \sigma^2$$
- Flat Kernel : weights are proportional to $N_{i-\delta}$, the number of patches (candidates) selected

Weighted Averaged Reprojection

Lagrangian Optimization

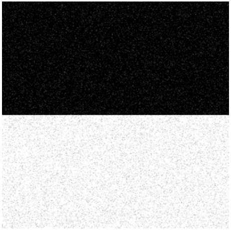
$$\beta_\delta^\star = \left[\mathrm{Var}\left(\widehat{P_{i-\delta}^I}[\delta]\right)\right]^{-1} / \sum_{\delta \in V_W} \left[\mathrm{Var}\left(\widehat{P_{i-\delta}^I}[\delta]\right)\right]^{-1}$$

- Rough approximation of the variance :

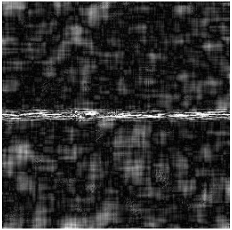
$$\mathrm{Var}\left(\widehat{P_{i-\delta}^I}[\delta]\right) \approx \frac{\sum_{\Omega_R} \theta_{i,k}^2}{\left(\sum_{\Omega_R} \theta_{i,k}\right)^2} \sigma^2$$

- Flat Kernel : weights are proportional to $N_{i-\delta}$, the number of patches (candidates) selected

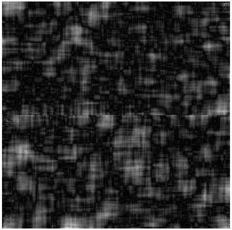
Weighted Averaged Reprojection in Practice



(a) Noisy



(b) Uae :PSNR $= 46, 51$



(c) Min : PSNR $= 47, 77$

(a) Noisy

(b) Uae : PSNR $= 28.65$

(c) Wav : PSNR $= 29.08$

FIG.: (a) Noisy image ($\sigma = 20$), (b) Uae Reprojection , (c) Wav Reprojection.
Parameters : $R = 21, 9$ (up, down), $W = 9, h^2 = 2\sigma^2 q_{0.99}^{W^2}$.

Uae

Wav

Conclusion

Improvements with Wav Reprojection

- Numerical (PSNR)
- Visual : edges better preserved

Conclusion

Improvements with Wav Reprojection

- Numerical (PSNR)
- Visual : edges better preserved

Work in progress...

- Other reprojections
- Last obvious drawback : small region with no redundancy, patch size should change according to local redundancy
- Theoretical results (cf. Erwan's Talk)

Conclusion

Improvements with Wav Reprojection

- Numerical (PSNR)
- Visual : edges better preserved

Work in progress...

- Other reprojections
- Last obvious drawback : small region with no redundancy, patch size should change according to local redundancy
- Theoretical results (cf. Erwan's Talk)

References I

- A. Buades, B. Coll, and J-M. Morel.
  A review of image denoising algorithms, with a new one.
  Multiscale Model. Simul., 4(2) :490–530, 2005.
- K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian.
  Image denoising by sparse 3-D transform-domain collaborative filtering.
  IEEE Trans. Image Process., 16(8), 2007.
- K. Dabov, A. Foi, V. Katkovnik, and K. Egiaziarian.
  A non-local and shape-adaptive transform-domain collaborative filtering.
  In *LNLA*, pages 179–186, 2008.
- Ch. Kervrann and J. Boulanger.
  Optimal spatial adaptation for patch-based image denoising.
  IEEE Trans. Image Process., 15(10), 2006.
- J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman.
  Non-local sparse models for image restoration.
  ICCV, 2009.

References II

- J. Salmon.

  On two parameters for denoising with Non-Local Means.

  IEEE Signal Process. Lett., 17 :269–272, 2010.

- T. Tasdizen.

  Principal neighborhood dictionaries for nonlocal means image denoising.

  IEEE Trans. Image Process., 18(12) :2649–2660, 2009.