A robust separable image denoising based on relative intersection of confidence intervals rule

Damir Seršić and Ana Sović (ana.sovic@fer.hr)
University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia

Overview

- Problem
- Intersection of confidence intervals rule
- Relative intersection of confidence intervals rule
- Separable image denoising
- Examples
- Conclusion

Problem

- Input signal (image) corrupted with noise

$$y(n) = x(n) + w(n)$$

- Goal: find a good estimate $\overline{x}(n)$ of input signal that accurately restores smooth parts and preserves its edges and discontinuities

Moving average

- Zero-order estimate of the input signal

$$\bar{x}_{h_k}(n) = \frac{1}{h_k} \sum_{i=\langle h_k \rangle} y(i)$$

- Large windows h_k
  - good estimates of low frequency components
  - unwanted smoothing of jumps and edges

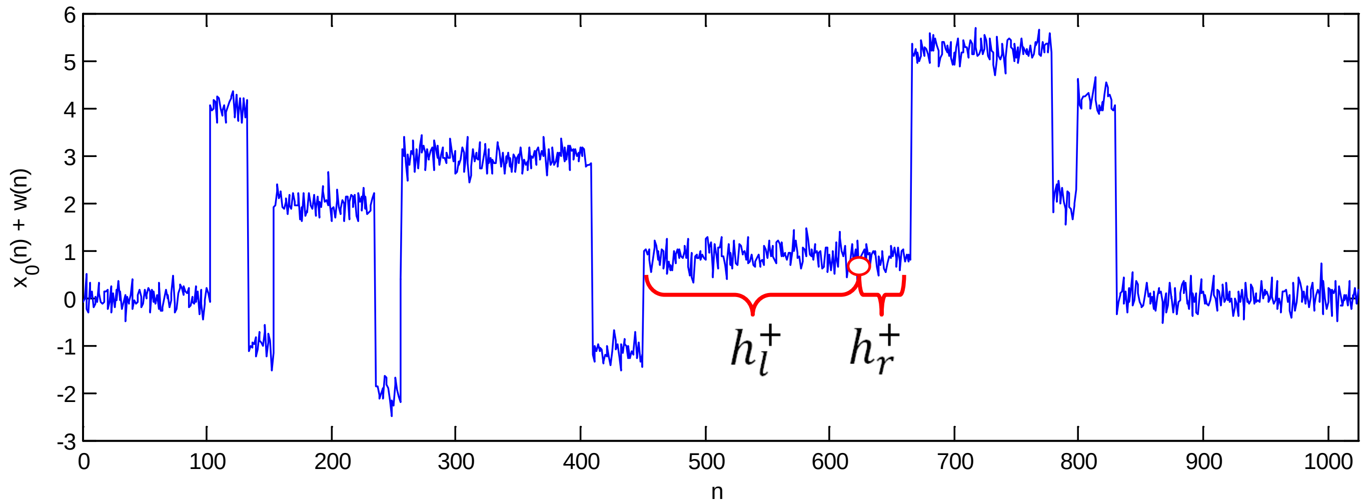
- Optimal window length h^* at some sample n → the smoothest zero-order estimate that does not span across the edges

Intersection of confidence intervals

- Growing set of estimators of window lengths

$$h_1 < h_2 < \cdots < h_k < \cdots < h_K$$

- Increase h_k until we reach the optimal window length h^* for each observed sample n



Intersection of confidence intervals

- Confidence interval

$$D_k(n) = [L_k, U_k], \qquad 1 \leq k \leq K$$

- Lower and upper limits

$$L_k = \bar{x}_{h_k}(n) - \Gamma \cdot \sigma_{h_k}(n)$$
$$U_k = \bar{x}_{h_k}(n) + \Gamma \cdot \sigma_{h_k}(n)$$

$$\sigma_{h_k}(n) = \frac{\sigma_w}{\sqrt{h_k}}$$

- The smallest upper and the largest lower confidence interval limits

$$\overline{L}_k(n) = \max_{i=1,\dots,k} L_i(n)$$
$$\underline{U}_k(n) = \min_{i=1,\dots,k} U_i(n)$$

ICI rule

- Optimal window length $h^+ = h_k$ is the largest window length for which the condition

$$\overline{L}_k(n) \leq \underline{U}_k(n)$$

  is satisfied

- Close to ideal h^*

- Repeat for each sample of $y(n)$

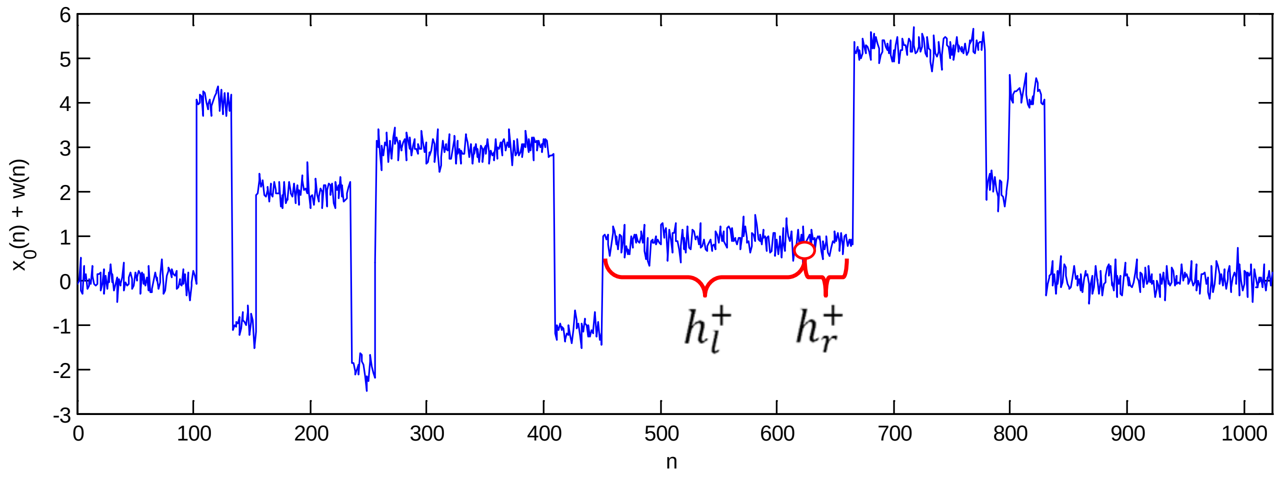
Relative intersection of confidence intervals (RICI)

- chosen window length is the largest $h^+ = h_k$

for which is still satisfied

$$\frac{\underline{U}_k(n) - \overline{L}_k(n)}{U_k(n) - L_k(n)} \geq r_c$$

Estimate of the input signal



$$\bar{x}_{h^+}(n) = \text{mean}\,\{y(n), \dots, y(n + h^+(n) - 2), y(n + h^+(n) - 1)\}$$

- In asymptotical sense → mean can be replaced by median

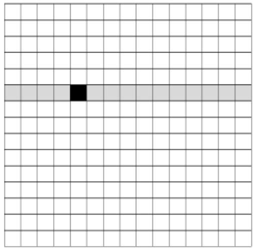
$$\bar{x}_{h^+}(n) = \text{median}\,\{y(n), \dots, y(n + h^+(n) - 2), y(n + h^+(n) - 1)\}$$

Why median

- results are more robust (even if the window lengths were not perfectly determined)
- more accurate results for almost any noise distribution
- significantly less sensitive to choice of parameters
- less sensitive to outliers

Image denoising algorithm – 1st step

1. For each pixel *y*(*n*,*m*) observe its ROW
2. Find the widest window length on the left side and on the right side from *y*(*n*,*m*) using ICI or RICI rule
3. Join all pixels on the intervals – use median – intermediate image $\bar{x}_r(n,m)$



4. For each pixel $\bar{x}_r(n,m)$ observe its COLUMN
5. Find the widest window length on the upper side and on the down side from $\bar{x}_r(n,m)$ using ICI or RICI rule
6. Join all pixels on the intervals – use median – intermediate image $\bar{x}_{rc}(n,m)$

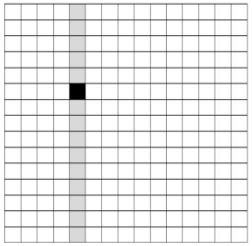
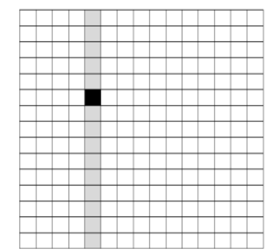


Image denoising algorithm – 2nd step

1. For each pixel *y*(*n*,*m*) observe its COLUMN
2. Find the widest window length on the upper side and on the down side from *y*(*n*,*m*) using ICI or RICI rule
3. Join all pixels on the intervals – use median – intermediate image $\bar{x}_c(n,m)$



4. For each pixel $\bar{x}_c(n,m)$ observe its ROW
5. Find the widest window length on the left side and on the right side from $\bar{x}_c(n,m)$ using ICI or RICI rule
6. Join all pixels on the intervals – use median – intermediate image $\bar{x}_{cr}(n,m)$

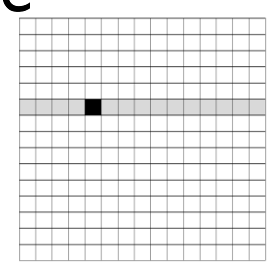


Image denoising algorithm – 3rd step

- Combine estimated images $\bar{x}_{rc}(n,m)$ and $\bar{x}_{cr}(n,m)$ to the final denoised result
  - MRICI method with fixed weights

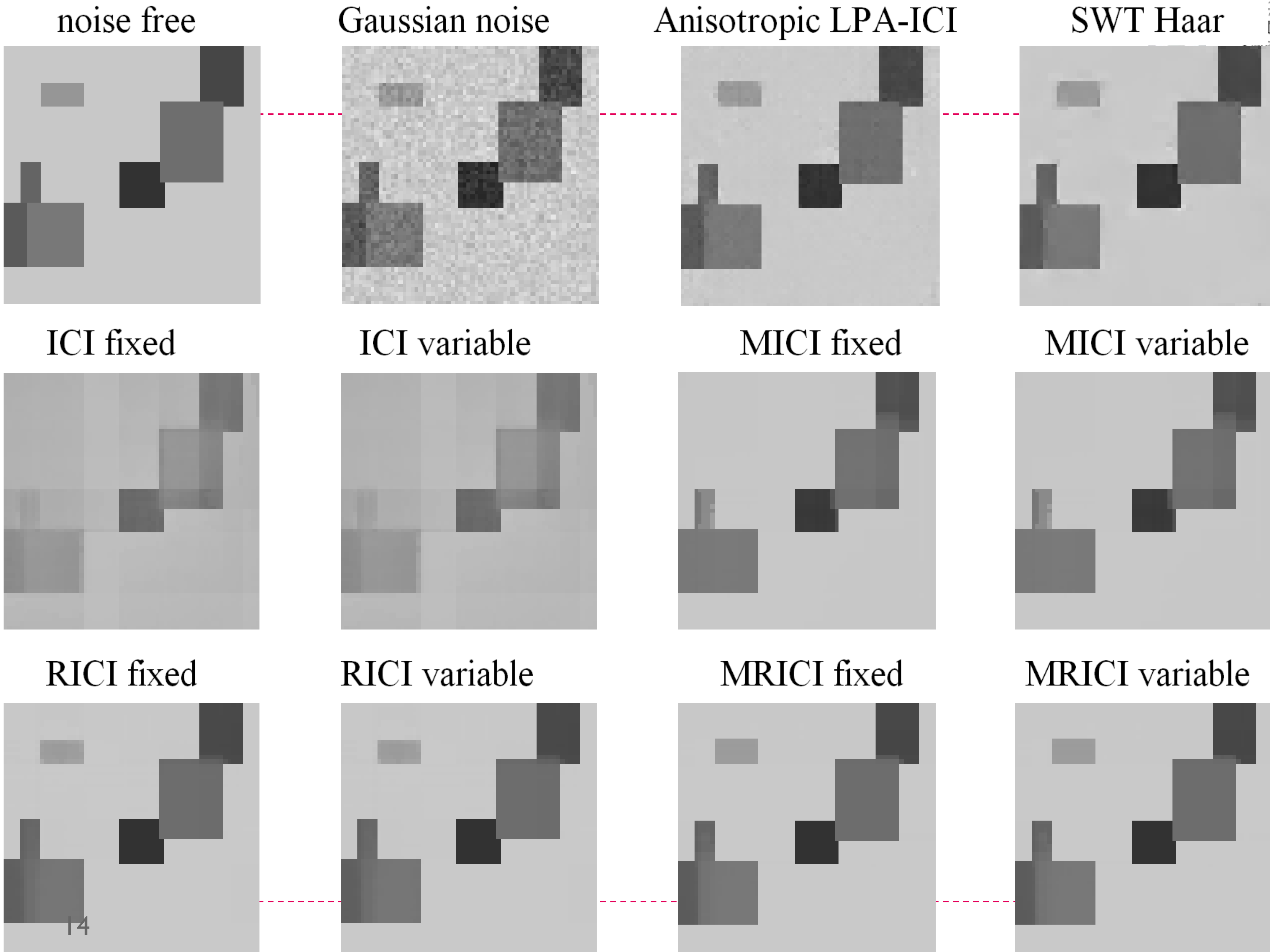
$$\bar{x}(n,m) = \frac{\bar{x}_{rc}(n,m) + \bar{x}_{cr}(n,m)}{2}$$

  - MRICI method with the variable weights

$$\omega_{rc}(n,m) = h^{+}_{rcL}(n,m) + h^{+}_{rcR}(n,m) + h^{+}_{rcU}(n,m) + h^{+}_{rcD}(n,m)$$

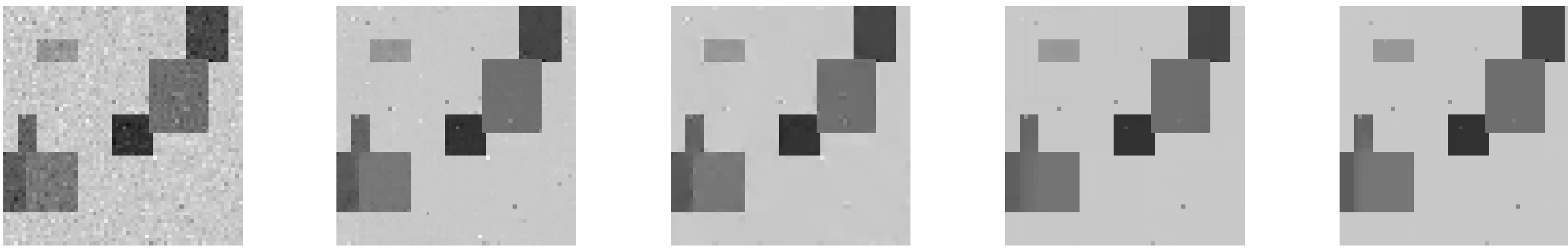
$$\omega_{cr}(n,m) = h^{+}_{crL}(n,m) + h^{+}_{crR}(n,m) + h^{+}_{crU}(n,m) + h^{+}_{crD}(n,m)$$

$$\bar{x}(n,m) = \frac{\omega_{rc}(n,m) \cdot \bar{x}_{rc}(n,m) + \omega_{cr}(n,m) \cdot \bar{x}_{cr}(n,m)}{\omega_{rc}(n,m) + \omega_{cr}(n,m)}$$

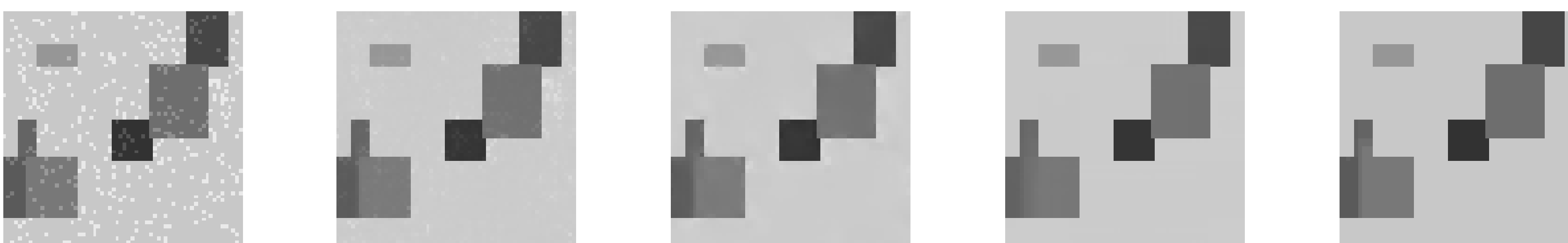


noise free
Gaussian noise
Anisotropic LPA-ICI
SWT Haar
ICI fixed
ICI variable
MICI fixed
MICI variable
RICI fixed
RICI variable
MRICI fixed
MRICI variable

Laplacian noise | Anisotropic LPA-ICI | SWT Haar | RICI fixed | MRICI fixed



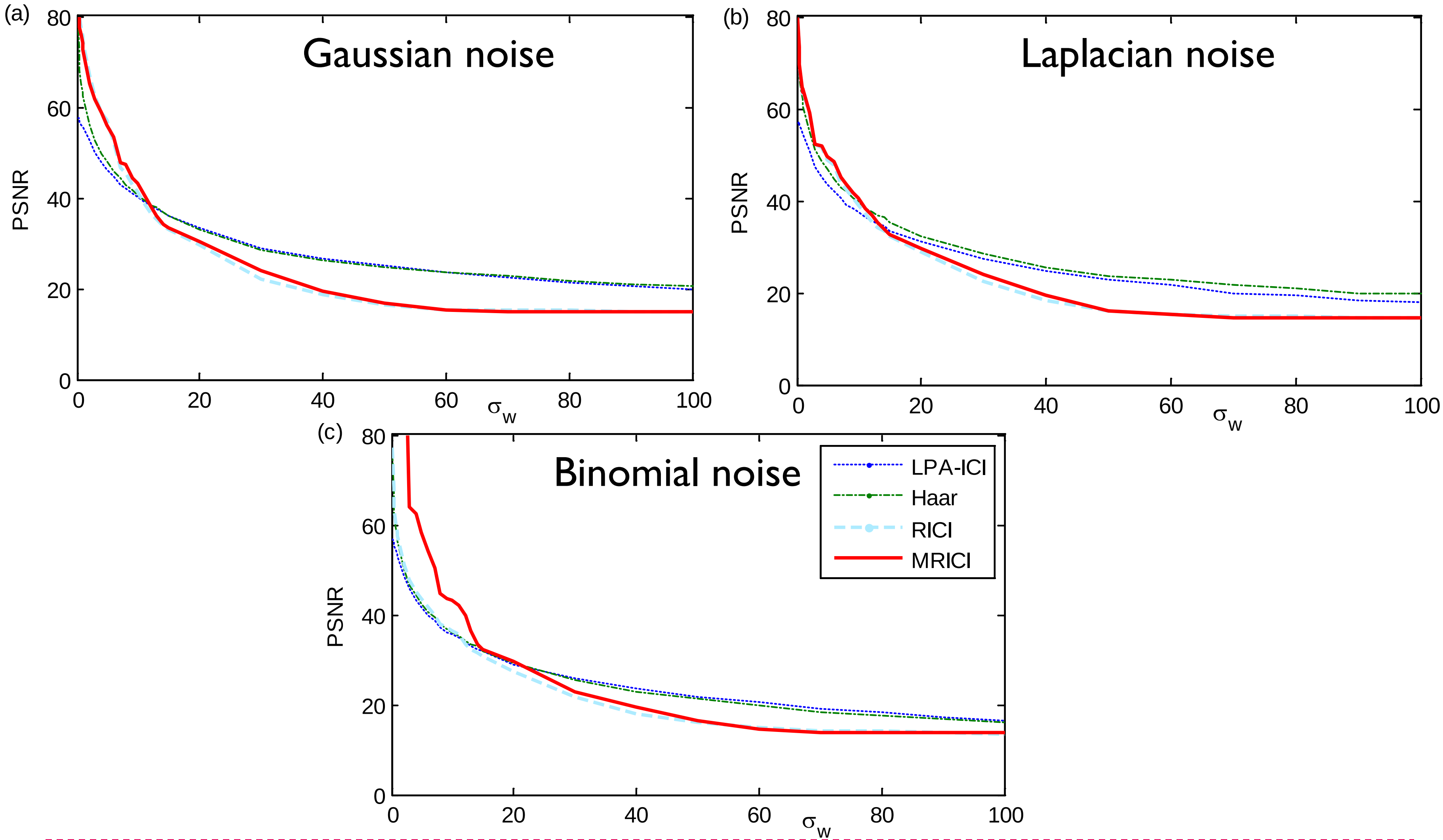
Binomial noise | Anisotropic LPA-ICI | SWT Haar | RICI fixed | MRICI fixed



PSNR for 30 realizations of the test image

	Gaussian noise, dB	Laplacian noise, dB	Binomial noise, dB
noisy-image	28.1372	28.1263	27.7333
ICI, fixed weight	19.5834	19.5659	19.7396
ICI, variable weight	19.5706	19.5530	19.7269
MICI, fixed weight	28.1511	28.1373	27.7886
MICI, variable weight	28.1523	28.1405	27.7762
RICI, fixed weight	40.8962	38.9303	36.3624
RICI, variable weight	40.7430	39.0041	36.2947
MRICI, fixed weight	**42.1992**	40.1020	**42.6739**
MRICI, variable weight	41.8532	**40.1671**	42.3026
anisotropic LPA-ICI	39.9478	37.3952	35.5632
Haar wavelet	40.4542	39.7961	35.9131

Image + noise (different noise deviations)



Conclusion

- novel method for image denoising based on the relative intersection of confidence interval rule
- separable three steps method
- median based estimators

- smooth parts are accurately restored
- edges and discontinuities are reserved

Thank you for your attention!