ACRI

A Fast Iterative Method for Removing Sparse Noise from Sparse Signals

S. Sadrizadeh[1], N. Zarmehi[1], F. Marvasti[1] and S. Gazor[2]

[1]Advanced Communications Research Institute (ACRI),

Electrical Engineering Dept., Sharif University of Technology, Tehran, Iran

[2]Electrical and Computer Engineering Dept., Queen's University at Kingston, Ontario, Canada

Contents

Itroduction

Applications:

- **Sparse Noise Removal From Sparse Signals:**
  - Impulsive Noise Removal from Images/Videos
  - Clicks and Pops Removal from Audios
  - Text/Image Separation
  - Dictionary Learning with Random Missing Samples

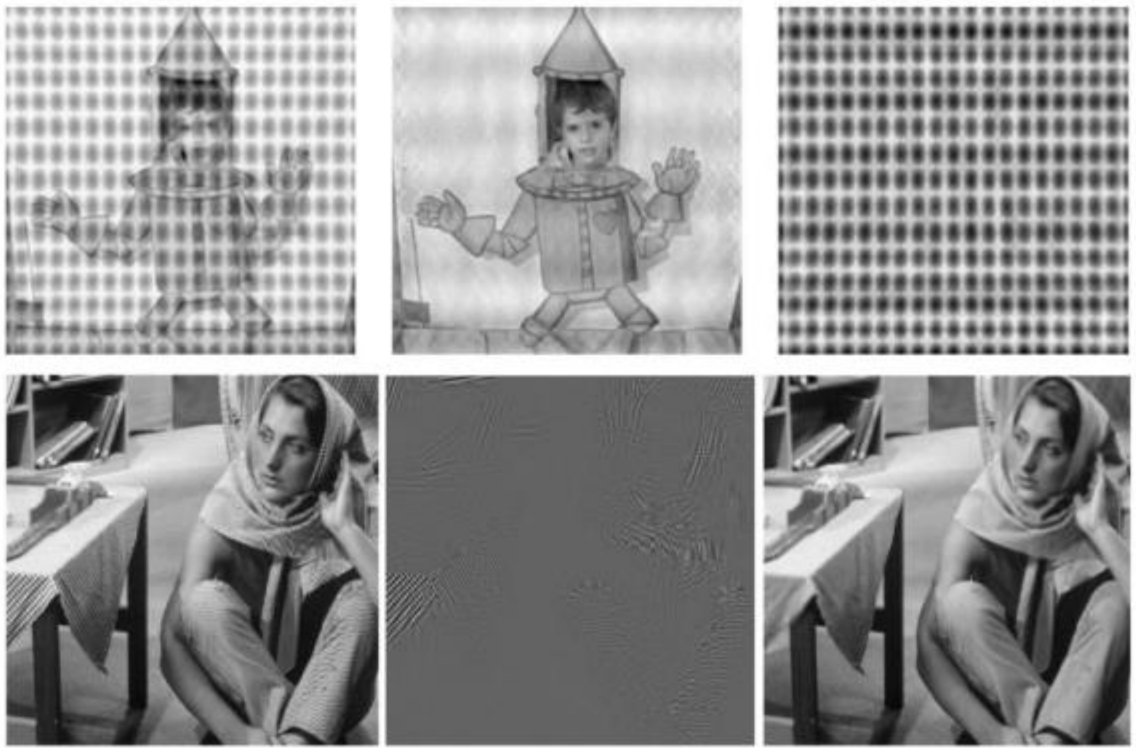
Itroduction

Applications:

- **Morphological Component Analysis:**

  Decomposing Images:

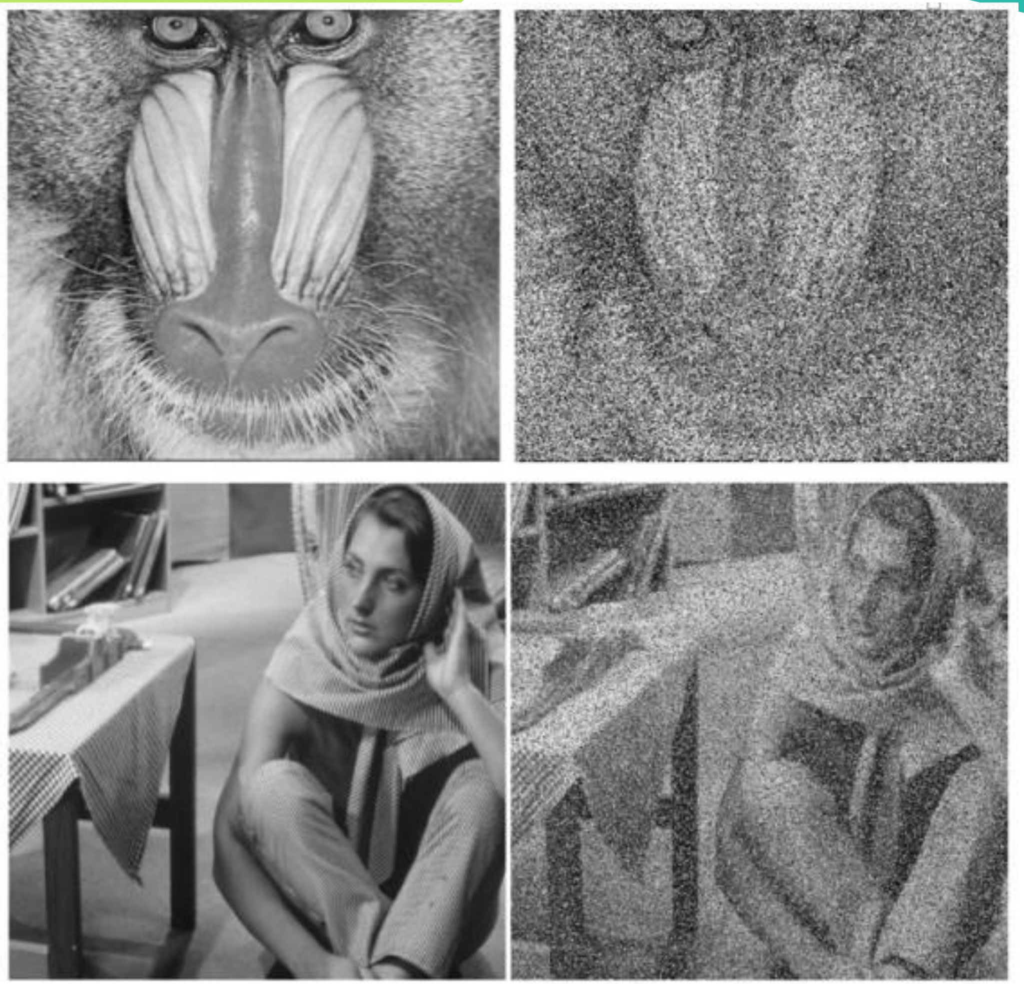
  Texture + Cartoon Part (piecewise smooth)



Itroduction

Impulsive Noise in Images:

- Salt-and-Pepper Noise (SPN)
- Random Valued Impulsive Noise (RVIN)



IDT Method

Problem Formulation:

$$\mathbf{Y} = \mathcal{D}^{-1}(\mathbf{X_0}) + \mathbf{N_0} \quad \mathrm{W} \triangleq \{(\mathbf{X}, \mathbf{N}) | \mathbf{Y} = \mathcal{D}^{-1}(\mathbf{X}) + \mathbf{N}\}$$

$$\underset{(\mathbf{X},\mathbf{N})}{\operatorname{argmin}} \|\mathrm{vec}(\mathbf{X})\|_0 + \|\mathrm{vec}(\mathbf{N})\|_0, \quad \mathrm{s.t.} \quad (\mathbf{X}, \mathbf{N}) \in \mathrm{W} \longrightarrow$$ **Np-Hard**

$$f_\lambda(\mathbf{X}, \mathbf{N}, \mathbf{T_1}, \mathbf{T_2}) \triangleq \|(\mathbf{1} - \mathbf{T_1}) \odot \mathbf{X}\|_F^2 + \|(\mathbf{1} - \mathbf{T_2}) \odot \mathbf{N}\|_F^2 + \lambda(\|\mathrm{vec}(\mathbf{T_1})\|_1 + \|\mathrm{vec}(\mathbf{T_2})\|_1).$$

$$\longrightarrow \underset{(\mathbf{X},\mathbf{N}),(\mathbf{T_1},\mathbf{T_2})}{\operatorname{argmin}} f_\lambda(\mathbf{X}, \mathbf{N}, \mathbf{T_1}, \mathbf{T_2}) \quad \mathrm{s.t.} \ (\mathbf{X}, \mathbf{N}) \in \mathrm{W}$$

IDT Method

Algorithm:

Algorithm 1 IDT

Input:
  Observed matrix: $\mathbf{Y} \in \mathbb{R}^{m \times n}$
  Four constants: $\alpha_1, \beta_1, \alpha_2, \beta_2$
  Standard deviation of the gaussian filter: σ $(*)$
  Maximum number of iterations: K
  Stopping threshold: δ
Output:
  Recovered estimate of the signal: $\hat{\mathbf{X}} \in \mathbb{R}^{m \times n}$
  Recovered estimate of the noise: $\hat{\mathbf{N}} \in \mathbb{R}^{m \times n}$
procedure
  $\mathbf{X}^0 \leftarrow \mathcal{D}(\mathbf{Y}), \quad \mathbf{N}^0 \leftarrow 0, \quad k \leftarrow 0$
  while $e > \delta \,\&\, k \leq K$ **do**
    $\mathbf{X}^k \leftarrow \text{threshold}(|\mathbf{X}^k_{i,j}|^{m,n}_{i=1,j=1}, \beta_1 e^{-\alpha_1 k})$
    $\mathbf{X}^k \leftarrow \text{clip}(\mathcal{D}^{-1}\left(\mathbf{X}^k\right))$ $(*)$
    $\mathbf{X}^k \leftarrow \text{gaussian-filter}(\mathbf{X}^k\ , \sigma)$ $(*)$
    $\mathbf{N}^{k+1} \leftarrow \mathbf{Y} - \mathbf{X}^k$
    $\mathbf{N}^{k+1} \leftarrow \text{threshold}(|\mathbf{N}^{k+1}_{i,j}|^{m,n}_{i=1,j=1}, \beta_2 e^{-\alpha_2 k})$
    $\mathbf{X}^{k+1} \leftarrow \mathcal{D}\left(\mathbf{Y} - \mathbf{N}^{k+1}\right)$
    $e \leftarrow \|\mathbf{N}^{k+1} - \mathbf{N}^k\|_F$
    $k \leftarrow k + 1$
  end while
  return $\hat{\mathbf{X}} \leftarrow \mathbf{X}^k, \hat{\mathbf{N}} \leftarrow \mathbf{N}^k$
end procedure

IDT Method

Theorems:

- **Theorem 1:** Under sufficient condition, the solution of the proposed optimization problem is the sparsest member of set W.

- **Theorem 2:** Under sufficient condition, the sparsest member of set W is unique.

IDT Method

Simulation Results:

- Artificial Sparse Signals

ρ_x \ ρ_n	**10%**	**20%**	**30%**
10%	316.5 (100%)	313.5 (100%)	311.6 (100%)
20%	315.9 (100%)	312.6 (100%)	310.4 (100%)
30%	314.9 (100%)	311.4 (100%)	224.082 (73%)

SNR (Success Rate)

IDT Method

Simulation Results:

- Salt-and-Pepper Noise

		PSNR					SSIM				
Noise Densities		10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
Lena	AMF	38.27	35.9	33.56	31.87	30.39	0.9628	0.9545	0.9389	0.9174	0.8908
	TPFF	35.78	35.06	32.79	30.98	29.71	NOT AVAILABLE				
	WESNR	35.91	35.56	35.11	34.52	33.61	0.9151	0.9134	0.9099	0.9045	0.8963
	IDT	**44.54**	**41.19**	**38.78**	**36.85**	**34.74**	**0.9939**	**0.9831**	**0.9712**	**0.9572**	**0.9379**
Peppers	AMF	36.06	33.98	32.17	30.67	29.23	0.9482	0.9410	0.9257	0.9021	0.8733
	TPFF	35.80	33.45	31.27	29.21	28.00	NOT AVAILABLE				
	WESNR	35.01	34.59	34.08	**33.34**	**32.49**	0.8842	0.8834	0.8818	0.8785	0.8727
	IDT	**40.20**	**37.03**	**35.26**	32.46	31.50	**0.9848**	**0.9669**	**0.9450**	**0.9187**	**0.8920**
F-16	AMF	35.87	32.97	31.05	29.43	27.77	0.9780	0.9699	0.9560	0.9377	0.9118
	TPFF	35.78	32.83	30.72	29.18	28.01	NOT AVAILABLE				
	WESNR	35.27	34.64	33.96	**32.80**	**31.85**	0.9382	0.9361	0.9322	0.9264	0.9187
	IDT	**42.79**	**38.99**	**35.96**	32.79	31.40	**0.9979**	**0.9911**	**0.9816**	**0.9678**	**0.9504**
Baboon	AMF	26.95	25.73	24.53	23.29	22.14	0.8922	0.8717	0.8351	0.7866	0.7264
	TPFF	30.96	27.90	26.34	25.15	23.87	NOT AVAILABLE				
	WESNR	26.44	26.17	25.70	24.93	24.11	0.7982	0.7938	0.7784	0.7529	0.7170
	IDT	**33.04**	**30.05**	**27.90**	**26.35**	**25.06**	**0.9777**	**0.9486**	**0.9117**	**0.8701**	**0.8140**
Boat	AMF	33.94	32.05	30.25	28.74	27.16	0.9345	0.9215	0.8986	0.8679	0.8274
	TPFF	NOT AVAILABLE									
	WESNR	32.78	32.40	31.84	31.13	30.08	0.8659	0.8635	0.8567	0.8478	0.8317
	IDT	**39.03**	**36.06**	**34.10**	**32.06**	**30.36**	**0.9832**	**0.9621**	**0.9382**	**0.9120**	**0.8780**

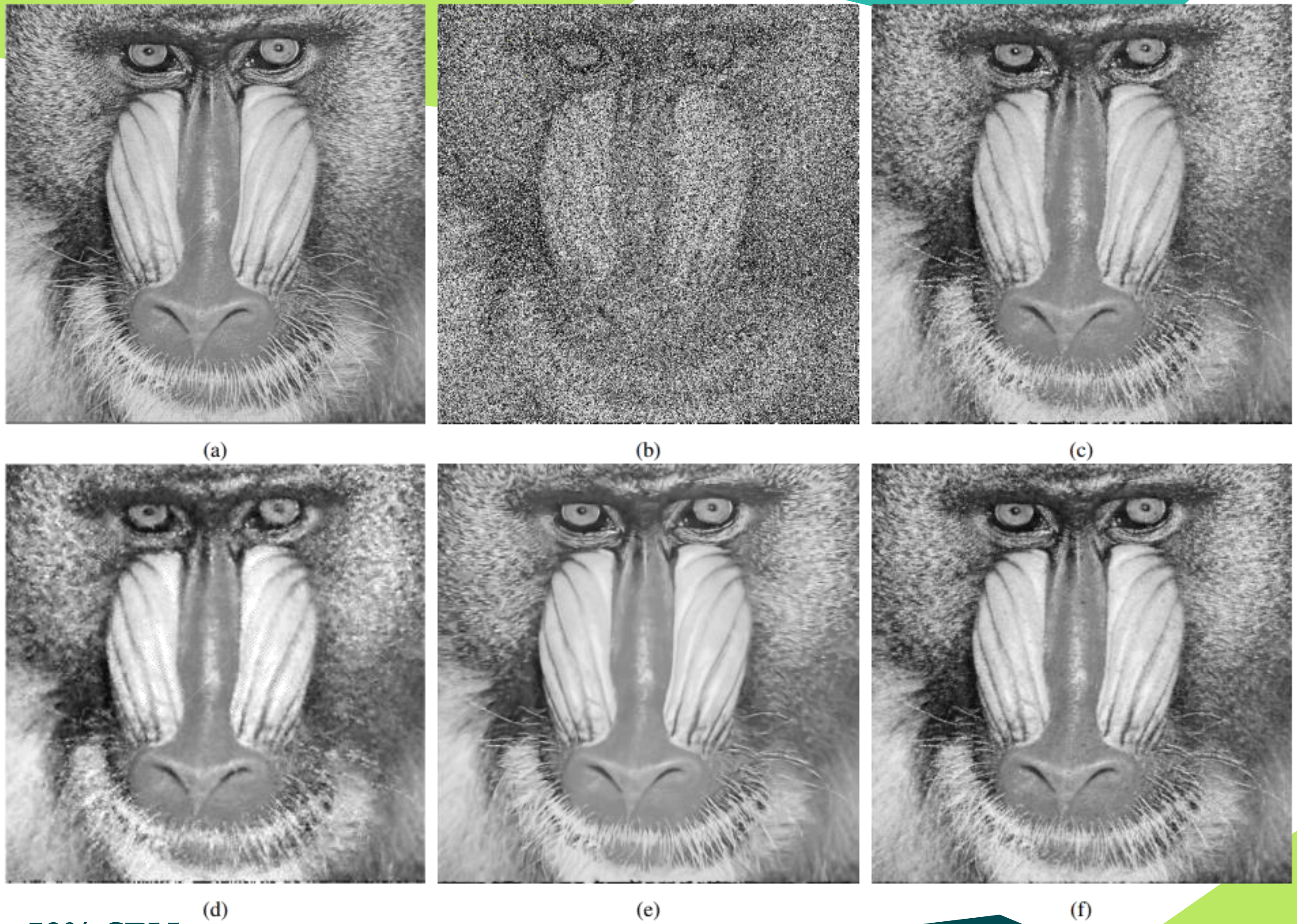
IDT Method

Simulation Results:

- **Salt-and-Pepper Noise**

AMF	TPFF	WESNR	IDT
1.13	1.75	32.26	**1.65**

Run-Time for *Lena* corrupted with 30% SPN



50% SPN

IDT Method

Simulation Results:

- Random-Valued Impulsive Noise

		PSNR						SSIM					
Noise Densities		5%	10%	20%	30%	40%	50%	5%	10%	20%	30%	40%	50%
Lena	ACWMF	38.53	35.32	31.61	28.76	26.15	23.52	0.9669	0.9325	0.8562	0.7592	0.6459	0.5133
	WESNR	36.83	36.30	35.10	33.00	30.91	28.18	0.9271	0.9245	0.9165	0.8972	0.8619	0.7880
	SAFE	35.92	34.97	34.79	33.33	31.97	30.43	0.9635	0.9587	0.9551	0.9407	**0.9208**	**0.8930**
	ALOHA	40.79	38.81	35.32	32.66	30.62	26.69	0.9821	0.9695	0.9403	0.9023	0.8752	0.7729
	IDT	**41.32**	**38.97**	**36.10**	**34.19**	**32.56**	**30.95**	**0.9877**	**0.9770**	**0.9554**	**0.9331**	0.9061	0.8750
Peppers	ACWMF	36.73	34.30	30.56	27.85	25.06	22.23	0.9659	0.9311	0.8494	0.7469	0.6201	0.4819
	WESNR	35.31	34.68	**33.61**	31.83	29.13	26.01	0.8955	0.8949	0.8888	0.8752	0.8342	0.7398
	SAFE	30.80	30.56	30.40	29.82	28.97	28.40	0.9537	0.9492	**0.9423**	0.9151	0.8812	**0.8742**
	ALOHA	37.55	35.91	33.00	31.16	28.50	25.02	0.9585	0.9310	0.8759	0.8337	0.7767	0.6824
	IDT	**38.62**	**36.42**	33.32	**32.22**	**30.75**	**29.14**	**0.9808**	**0.9691**	0.9341	**0.9189**	**0.8888**	0.8499
F-16	ACWMF	36.78	33.62	29.95	27.28	24.57	21.72	0.9700	0.9351	0.8561	0.7576	0.6220	0.4784
	WESNR	35.75	34.80	32.85	30.72	28.37	25.44	0.9453	0.9420	0.9321	0.9115	0.8601	0.7511
	SAFE	28.81	28.64	28.29	28.02	27.31	26.47	0.9525	0.9447	0.9498	0.9188	0.8898	**0.8804**
	ALOHA	36.17	34.93	32.26	28.33	27.76	24.68	0.9410	0.9255	0.9072	0.8788	0.8633	0.7503
	IDT	**39.50**	**36.99**	**33.97**	**31.87**	**30.03**	**28.27**	**0.9891**	**0.9809**	**0.9633**	**0.9420**	**0.9151**	0.8779
Baboon	ACWMF	27.63	26.31	24.37	22.64	21.28	19.83	0.9311	0.8955	0.8214	0.7299	0.6329	0.5171
	WESNR	26.90	26.12	24.86	23.76	22.62	21.42	0.8420	0.8251	0.7852	0.7402	0.6761	0.5931
	SAFE	24.57	24.26	24.02	23.35	22.24	21.19	0.8558	0.8382	0.8217	**0.7827**	**0.7126**	**0.6305**
	ALOHA	29.37	28.59	25.56	22.66	22.33	20.56	0.7721	0.7716	0.7015	0.6495	0.6433	0.5792
	IDT	**31.58**	**28.98**	**26.22**	**24.55**	**23.22**	**22.15**	**0.9474**	**0.9089**	**0.8368**	0.7632	0.6828	0.6097
Boat	ACWMF	34.65	32.29	29.15	26.86	24.64	22.31	0.9633	0.9274	0.8476	0.7545	0.6451	0.5193
	WESNR	33.13	32.55	31.24	29.58	27.85	25.72	0.8892	0.8835	0.8675	0.8387	0.7945	0.7161
	SAFE	31.33	30.65	30.19	29.26	28.04	26.62	0.9377	0.9187	0.9171	**0.8922**	**0.8560**	**0.8051**
	ALOHA	35.88	**34.53**	31.11	29.47	26.99	24.29	0.9574	0.9536	0.8869	0.8571	0.7594	0.6818
	IDT	**36.24**	34.23	**31.8**	**30.13**	**28.87**	**27.10**	**0.9726**	**0.9543**	**0.9186**	0.8819	0.8409	0.7914

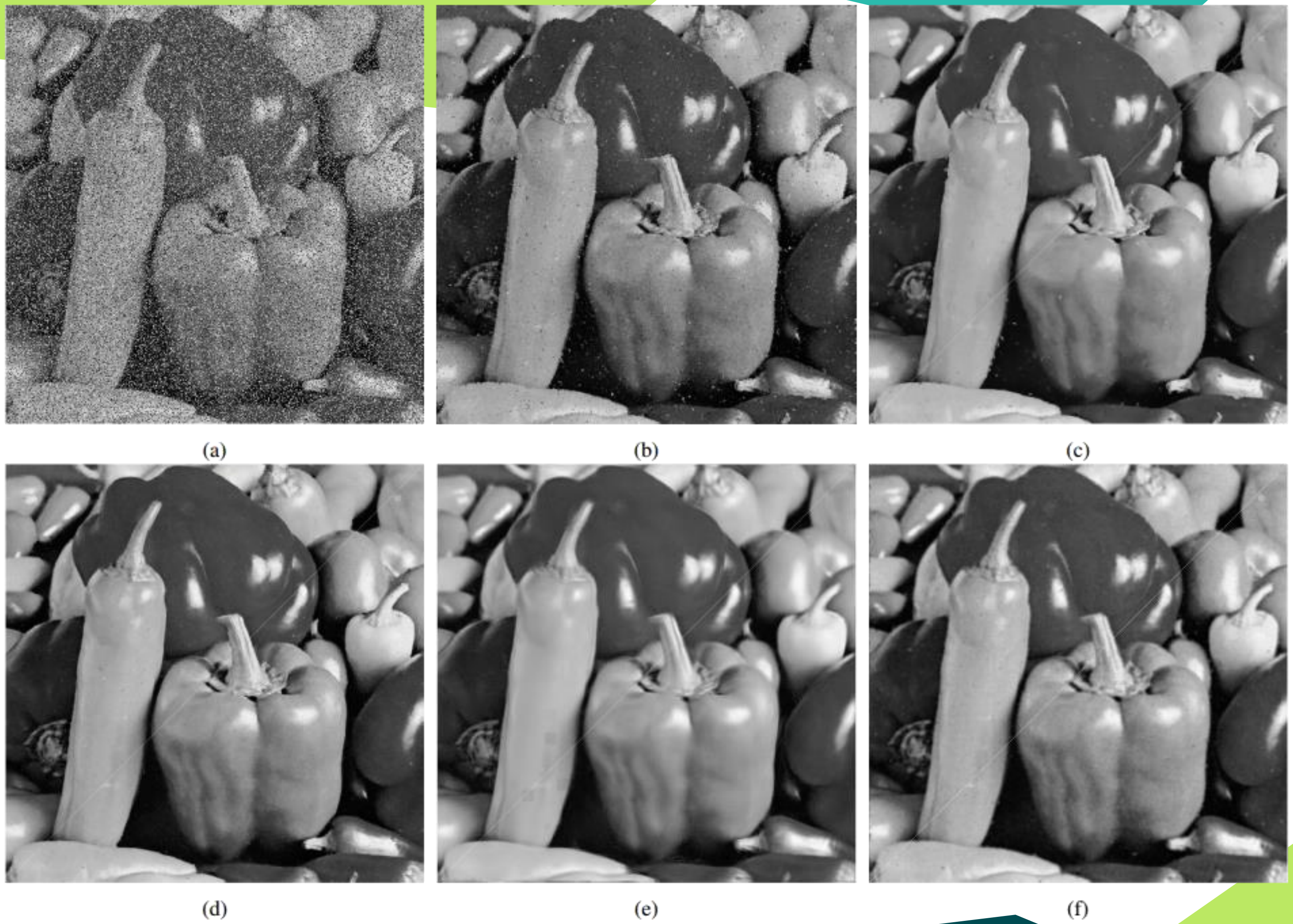
IDT Method

Simulation Results:

- Random-Valued Impulsive Noise

ACWMF	WESNR	SAFE (on GPU)	ALOHA on GPU	IDT
1.94	34.32	763.48 (21.07)	1368.2	**1.71**

Run-Time for *Lena* corrupted with 30% RVIN



40% RVIN

Contributions

- A General Framework for Separating Two Sparse Signal
- Fast Algorithm with Low Complexity
- Comparable Reconstruction Quality

Future Works

- Impulsive Noise Removal from Videos
- Block Sparse Noise Removal
- Mixed Noise (Gaussian + Impulsive) Removal

Thank You

For Your Attention ☺