



北京大学数字媒体研究所
INSTITUTE OF DIGITAL MEDIA, PEKING UNIVERSITY



J0071: Non-local Structure-based Filter with Integer Operation

Peking University

DJI

Apr,2018

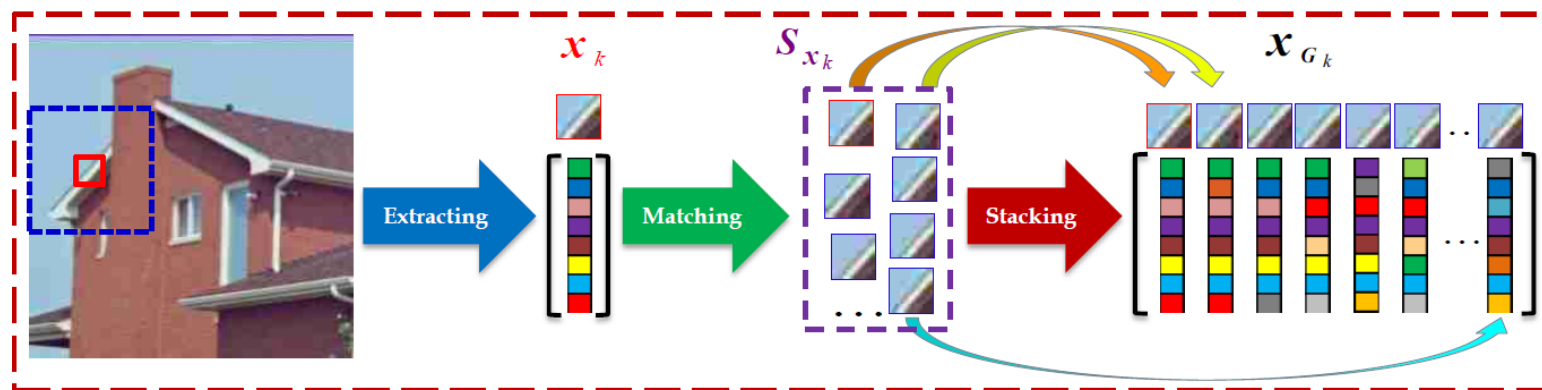
Outline

- Brief Review of NLSF in J0011
- Implementation Details of Proposed Integer NLSF
- Performance
- Conclusion



NLSF in J0011

□ Block matching and group construction



□ Group-based filtering

- Decomposed each group $\mathbf{x}_{G_k}$ by SVD

$$\begin{array}{c} \mathbf{x}_{G_k} \\ \left[\begin{array}{cccccc} \text{img} & \text{img} & \text{img} & \text{img} & \text{img} & \dots & \text{img} \\ \text{col1} & \text{col1} & \text{col1} & \text{col1} & \text{col1} & \dots & \text{col1} \\ \text{col2} & \text{col2} & \text{col2} & \text{col2} & \text{col2} & \dots & \text{col2} \\ \text{col3} & \text{col3} & \text{col3} & \text{col3} & \text{col3} & \dots & \text{col3} \\ \text{col4} & \text{col4} & \text{col4} & \text{col4} & \text{col4} & \dots & \text{col4} \\ \text{col5} & \text{col5} & \text{col5} & \text{col5} & \text{col5} & \dots & \text{col5} \\ \text{col6} & \text{col6} & \text{col6} & \text{col6} & \text{col6} & \dots & \text{col6} \end{array} \right] \end{array} = U_{G_k} \Sigma_{G_k} V_{G_k}^T = \sum_{i=1}^m \gamma_{G_k \otimes i} (u_{G_k \otimes i} v_{G_k \otimes i}^T)$$

NLSF in J0011

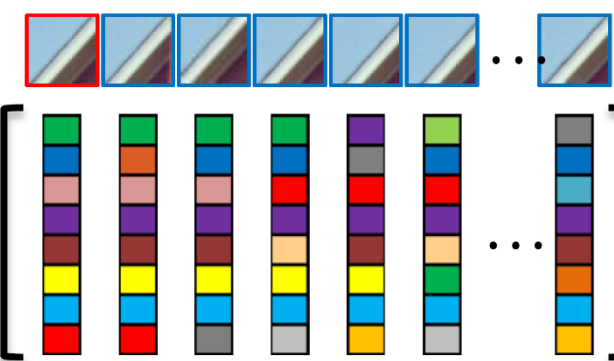
□ Group-based filtering

- Apply hard threshold operation to singular value γ_{G_k}

$$\alpha_{G_k} = \text{hard}(\gamma_{G_k}, c\tau)$$

□ Image reconstruction

- Reconstruct group $\hat{x}_{G_k}$ from the shrunk singular values α_{G_k}

$$\sum_{i=1}^m \alpha_{G_k \otimes i} (u_{G_k \otimes i} v_{G_k \otimes i}^T) = \begin{bmatrix} \begin{matrix} \text{[Colorful blocks]} \end{matrix} \\ \dots \end{bmatrix} \hat{x}_{G_k}$$


Proposed integer NLSF

□ Motivation

- The decimal operations during SVD and threshold estimation are not good for hardware realization
- Hardware implementation of decimal operations has high complexity
- Different processors might deal with decimals in different ways

□ Integer NLSF sub-module

- SVD
- Threshold estimation



Proposed integer NLSF

□ Overall description

■ Power method based iterative SVD

□ *Singular_value_decom*

➤ Obtain largest singular value and its left/right singular vectors

□ Calculate the difference between X_{G_k} and $\hat{X}_{G_k}$

□ Feed the difference into *Singular_value_decom*

Algorithm 1: Power Method based Singular Value Decomposition (SVD)

Input: X_{G_i} , p , q , τ

Output: Collaborative filtered similar block group.

```
1  $r = 0$ ;  
2  $sing\_val = \text{singular\_value\_decom}(X_{G_i}, p, q, U_{i,G_i}, V_{i,G_i}^T)$ ;  
3 while  $sing\_val > \tau$  AND  $r \leq K$  do  
4    $\hat{X}_{G_i} = U_{G_i} \cdot sing\_val \cdot V_{G_i}^T$ ;  
5    $X_{G_i} = X_{G_i} - \hat{X}_{G_i}$ ;  
6    $sing\_val = \text{singular\_value\_decom}(X_{G_i}, p, q, U_{i,G_i}, V_{i,G_i}^T)$ ;  
7    $r = r + 1$ ;  
8 end
```

Proposed integer NLSF

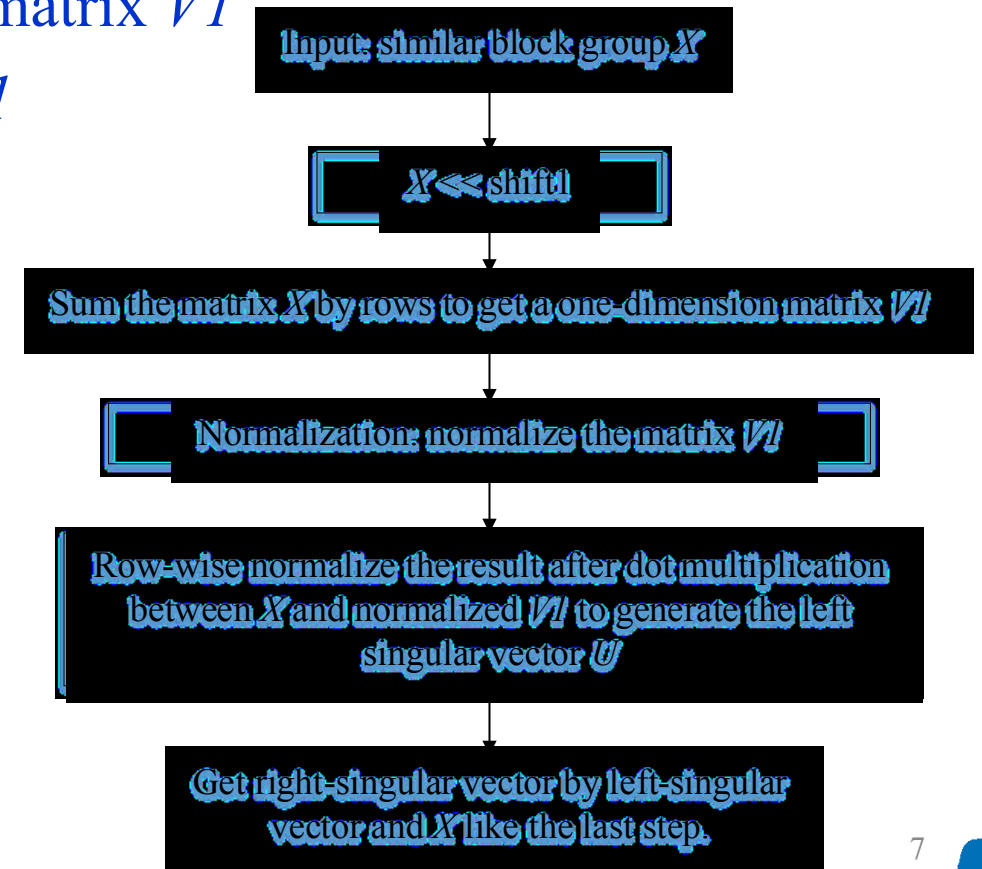
□ SVD

- Left shifting X by $shift1$ to convert it from double-type to integer-type
- Sum X by rows and get a one-dimension matrix $V1$
- Calculate normalization factor $norm$ of $V1$
 - Integer square root

$$norm = \sqrt{\sum_{j=1}^q v1_j^2}$$

- Normalization operation

$$v1'_i = \frac{v1_i}{(norm \ll shift2)}$$



Proposed integer NLSF

□ Threshold estimation

■ Look-up table $Qstep_T$

$$Qstep_T_i = 2^{\frac{i-4}{6}} \times (1 \ll 12)$$

■ i ($0 \leq i \leq 51$) represents the i th element of $Qstep_T$.



Performance

□ General performance compared to JEM7.0

	Constraint Set 1					
	Proposed Over JEM7.0			NLSF in JVET-J0011		
	BD-rate(Y)	BD-rate(U)	BD-rate(V)	BD-rate(Y)	BD-rate(U)	BD-rate(V)
Average SDR-A	-0.53%	0.73%	-0.04%	-0.91%	0.03%	-0.80%
Average SDR-B	-1.18%	0.15%	-0.11%	-1.58%	-0.75%	-1.85%
Average all	-0.86%	0.44%	-0.07%	-1.25%	-0.36%	-1.33%
Enc Time[%]	97%			101%		
Dec Time[%]	316%			397%		
	Constraint Set 2					
	Proposed Over JEM7.0			NLSF in JVET-J0011		
	BD-rate(Y)	BD-rate(U)	BD-rate(V)	BD-rate(Y)	BD-rate(U)	BD-rate(V)
Average SDR-B	-1.64%	-0.61%	-2.35%	-1.92%	0.36%	-3.15%
Enc Time[%]	99%			103%		
Dec Time[%]	537%			607%		

Conclusion

- The proposed integer NLSF can achieve hardware-friendly design
- The proposed integer NLSF has minor coding loss but coding performance improvement is still substantial



Thanks!

