Coded Aperture Design by Motion Estimation Using Sparse Representation in Adaptive Compressed Spectral Video Sensing

PhD(c) N. Diaz[1], MS(c) C. Noriega [1]
Ph.D A. Basarab [3], Ph.D J-Y. Tourneret [3]
Advisor: Ph.D H. Arguello [2]

[1]Department of Electrical and Computer Engineering,
[2]Department of Computer Science
Universidad Industrial de Santander, Bucaramanga, Colombia.
[3]University of Toulouse, Toulouse, Francia.

March 5, 2022

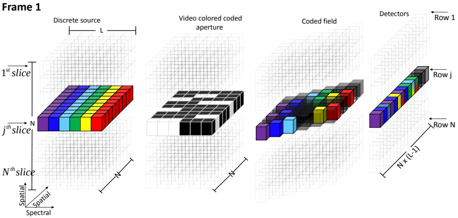
Universidad Industrial de Santander

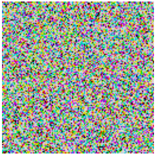
TéSA Telecommunications for Space and Aeronautics

Outline

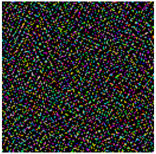
1. Introduction
   - Challenges in Video-CSI
2. Methods
   - Motion Estimation in Spectral Imaging
   - Sparse regularization term
   - Dictionary Filters and Coefficient Maps
   - Adaptive Video Colored Coded Aperture Design
3. Results
   - Simulation Parameters
   - Quality of Image Reconstruction
4. Conclusions
   - Video Motion Estimation Algorithm

Challenges in Video-CSI





Random color aperture

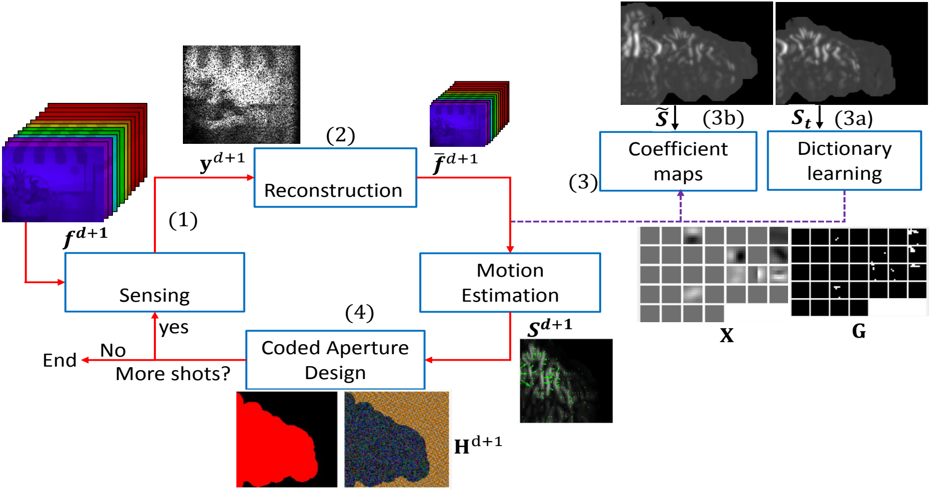


Blue noise aperture[1].

- Traditionally, coded apertures for video-CSI are designed randomly, ignoring the redundancy of the static and dynamic scene.
- Optimal approaches for sampling CSI could be extended to Video-CSI, however, those approaches promote complementary coded apertures, ignoring the motion between a couple of frames.

[1] Correa, Claudia, 2016 [1]

Proposed Adaptive Coded Aperture Design



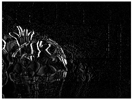
Proposed Spectral Video Motion Estimation

- A pair of successive frames $\mathbf{F}_H^{d-1}$ and $\mathbf{F}_H^{d}$ (of $\mathbb{R}^{M\times N\times L}$) from a spectral video acquired at time instants $d-1$ and d
- Denote as $\mathbf{S}^d_{(\ell,x)}$ and $\mathbf{S}^d_{(\ell,y)} \in \mathbb{R}^{M\times N\times L}$ the video motions for the frame d along the x and y axes [2].
- The motion estimation field is formulated as the minimization of a cost function with energy $E_{\text{data}}(\mathbf{S}^d, \mathbf{F}_H^d, \mathbf{F}_H^{d-1})$ penalized by spatial and sparse regularizations, i.e.,

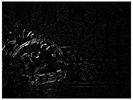
$$\underset{\mathbf{X},\mathbf{S}^d}{\operatorname{argmin}}\left\{E_{\text{data}}(\mathbf{S}^d, \mathbf{F}_H^d, \mathbf{F}_H^{d-1}) + \lambda_s E_{\text{spatial}}(\mathbf{S}^d) + \lambda_p E_{\text{sparse}}(\mathbf{S}^d, \mathbf{X})\right\} \tag{1}$$

$\mathbf{F}^{d-1}$ spectral video sequence.

$\mathbf{F}^{d}$ spectral video sequence.



Horizontal motion $\mathbf{S}^d_{(\ell,x)}$.



Vertical motion $\mathbf{S}^d_{(\ell,y)}$.

[2] Note that the displacement vectors components along x and y are estimated independently for simplicity, i.e., $\mathbf{S}^d = \mathbf{S}^d_{(\ell,x)}$ or $\mathbf{S}^d = \mathbf{S}^d_{(\ell,y)}$

Data Fidelity and Spatial Regularization

Optical flow assumes brightness constancy and temporal consistency, leading to the following optical flow equation

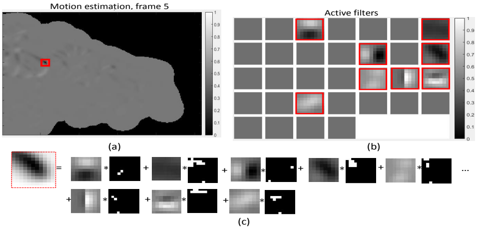
$$\partial_t \mathbf{f}_H^d + \nabla \mathbf{f}_H^T \mathbf{s}^d = 0 \tag{2}$$

where $\mathbf{s}^d \in \mathbb{R}^{NM}$ represents the flow field such that $\mathbf{s}_\ell^d$ is the vectorized video motion $\mathbf{S}_\ell$, $\partial_t \mathbf{f}_H^d$ denotes the temporal derivative and $\nabla \mathbf{f}_H^T$ is the spatial gradient of the brightness. The data fidelity term resulting from optical flow is

$$E_{\text{data}}(\mathbf{s}^d, \mathbf{f}_H^d, \mathbf{f}_H^{d-1}) = \left\| \partial_t \mathbf{f}_H^d + \nabla \mathbf{f}_H^T \mathbf{s}^d \right\|_2^2 \tag{3}$$

where $\|.\|_2^2$ is the squared ℓ_2 norm. The first regularization term promotes smooth variations in the video motion field by using a standard total variation function, $E_{\text{spatial}}(\mathbf{S}^d) = \|\nabla \mathbf{S}^d\|_2^2$

Sparse Regularization Term



$\mathbf{S}^d$ is modeled as a convolution between the coefficient maps $\mathbf{X}_v$ and a set of V filters $\mathbf{G}_v$ [2], $\mathbf{S}^d \approx \sum_{v=1}^{V} \mathbf{G}_v * \mathbf{X}_v$

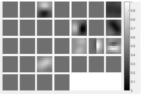
The second regularization term promotes sparsity of the motion vectors in a dictionary of representative motions. It decomposes the video motion $\mathbf{S}^d$ as a convolution between V sparse coefficient maps $\mathbf{X}_v$ and a set of V filters $\mathbf{G}_v$, i.e.,

$$E_{\text{sparse}}(\mathbf{S}^d, \mathbf{X}) = \left\| \mathbf{S}^d - \sum_{v=1}^{V} \mathbf{G}_v * \mathbf{X}_v \right\|_2^2 \quad (4)$$

where $*$ denotes convolution.

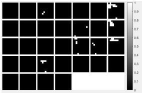
Dictionary Filters and Coefficients Maps

The dictionary learning is performed by solving the following problem (where $\widetilde{S}_d$ denotes the training video sequence which was obtained using Horn-Schunck optical flow estimation)



$$\underset{\mathbf{G}_v, \mathbf{X}_{d,v}}{\operatorname{argmin}} \frac{1}{2} \sum_d \left\| \sum_v \mathbf{X}_{d,v} * \mathbf{G}_v - \widetilde{\mathbf{S}}^d \right\|_2^2 + \lambda \sum_{v=1}^{V} \sum_d \|\mathbf{X}_{d,v}\|_1 \quad \text{s.t.} \quad \|\mathbf{G}_v\| = 1 \; \forall v = 1, \ldots, V. \tag{5}$$

Once the dictionary $\mathbf{G}_v$ has been determined, the coefficient maps of a sequence of test images denoted as $\mathbf{S}_t^d$ are obtained by solving the following optimization problem



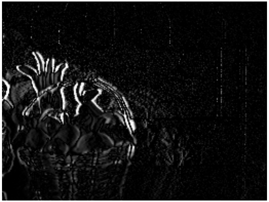
$$\underset{X_v}{\operatorname{argmin}} \frac{1}{2} \left\| \sum_{v=1}^{V} \mathbf{X}_v * \mathbf{G}_v - \mathbf{S}_t^d \right\|_2^2 + \lambda \sum_{v=1}^{V} \|\mathbf{X}_v\|_1 \tag{6}$$

which can again be replicated using the ADMM algorithm.

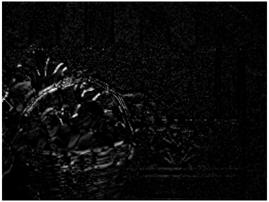
Spectral Video Motion Estimation

$$\underset{\mathbf{S}_{\ell}^{d}}{\operatorname{argmin}}\Big\{E_{\mathrm{data}}(\hat{\mathbf{F}}_{H}^{d-1},\hat{\mathbf{F}}_{H}^{d},\mathbf{S}_{\ell}^{d-1})+\lambda_s\|\nabla\mathbf{S}_{\ell}^{d-1}\|_2^2+$$
$$\lambda_p(k)\|\mathbf{S}_{\ell}^{d-1}-\sum_{v}\mathbf{G}_v*\mathbf{X}_v\|_2^2\Big\}\ \text{s.t.}\ \|\mathbf{G}_v\|=1\ \forall v$$

$\mathbf{f}^d=\mathbf{\Psi}^d\boldsymbol{\theta}^d$ spectral video.

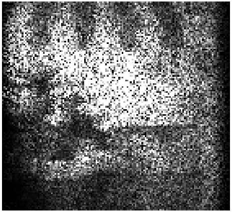


Horizontal motion $\mathbf{S}_{(\ell,x)}^{d}$.

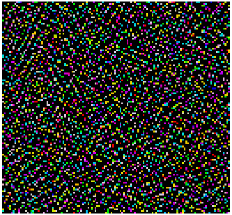


Vertical motion $\mathbf{S}_{(\ell,y)}^{d}$.

Compressive Spectral Video Sensing

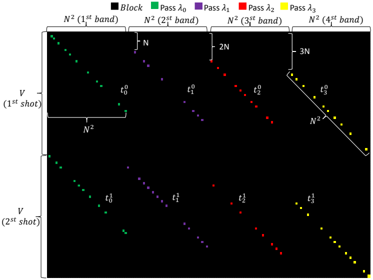


$\mathbf{y}^d = \mathbf{H}^d \mathbf{f}^d$ Video spectral sequence.



Rows of $\mathbf{H}^d$ represent the coded aperture.

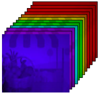
$\mathbf{f}^d = \mathbf{\Psi}^d \boldsymbol{\theta}^d$ Video spectral sequence.



Sensing matrix $\mathbf{H}^d$

Low Resolution Reconstruction and Interpolation

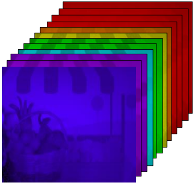
The low resolution datacube is computed by



$$\hat{\mathbf{f}}_L^{d-1} = \boldsymbol{\Psi}_L^{-1}(\underset{\boldsymbol{\theta}_L}{\text{argmin}}\|\mathbf{y}^{d-1} - \mathbf{H}_L^{d-1}\boldsymbol{\Psi}_L^{d-1}\boldsymbol{\theta}_L^{d-1}\|_2^2 + \tau\|\boldsymbol{\theta}_L^{d-1}\|_1)$$

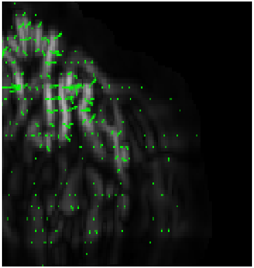
$$\hat{\mathbf{f}}_L^{d} = \boldsymbol{\Psi}_L^{-1}(\underset{\boldsymbol{\theta}_L}{\text{argmin}}\|\mathbf{y}^{d} - \mathbf{H}_L^{d}\boldsymbol{\Psi}_L^{d}\boldsymbol{\theta}_L^{d}\|_2^2 + \tau\|\boldsymbol{\theta}_L^{d}\|_1)$$

where $\mathbf{H}_L^0$ is the LR sensing matrix, $\boldsymbol{\Psi}_L^d$ is the LR representation basis, and $\boldsymbol{\theta}_L^d$ is the vectorization of a sparse vector for the LR reconstruction.

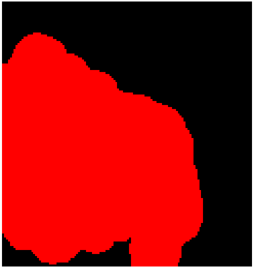


The LR datacube is interpolated using $P(.)$ a bilinear interpolator $\hat{\mathbf{f}}_H^{\mathbf{d-1}} \leftarrow \mathbf{P}(\hat{\mathbf{f}}_L^{d-1})$, and $\hat{\mathbf{f}}_H^{\mathbf{d}} \leftarrow \mathbf{P}(\hat{\mathbf{f}}_L^{d})$.

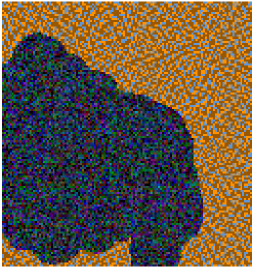
Design of Video Adaptive Colored Coded Aperture (VA-CCA)



Motion estimation
$\sqrt{(\mathbf{S}^d_{(\ell,x)})^2 + (\mathbf{S}^d_{(\ell,y)})^2}.$

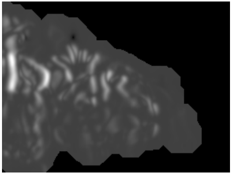


Thresholding
$\mathbf{Q}^d_\ell \leftarrow (\mathbf{S}^{d-1}_\ell, \mathbf{S}^d_\ell)$

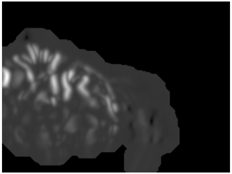


Next coded aperture
$\mathbf{r}^d_\ell \leftarrow \mathbf{q}^d_\ell \odot \mathbf{b}^d_\ell + (\mathbf{1} - \mathbf{q}^d_\ell) \odot \hat{\mathbf{b}}^d_\ell$

Simulation Parameters



Training spectral motion sequence $\widetilde{\mathbf{S}}^d$



Test spectral motion sequence $\mathbf{S}_t^d$

Step	Parameters	Values
Dictionary learning	Database 23 frames	Peasant woman 1 [3]
	Filter size	8×8
	Filters number	$M = 32$
	Sparsity term	$\lambda = 0.05$
	Number of iteration	500
Sparse coding	Database 23 frames	Peasant woman 2 [3]
	Number of iteration	500
Video motion estimation	Regularization parameter	$\lambda_s = 0.75$
	Sparsity term (video)	$\lambda_d = \{1 \times 10^{-6} \times 10^{-3}\}$

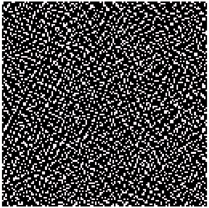
Comparison of Quality of Image Reconstruction

Average PSNR=20.4296 dB.

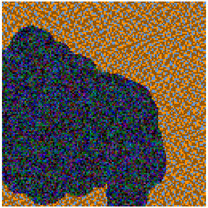
Average PSNR=21.1145 dB.

Average PSNR=19.6498 dB.

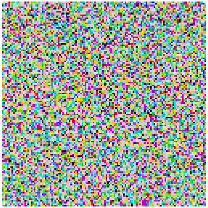
Average PSNR=18.7091dB.



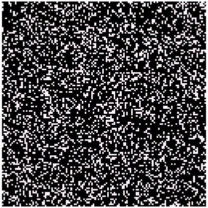
Blue noise non adaptive (BNA).



Video colored coded aperture (V-CCA).



Random colored coded aperture (R-CCA).

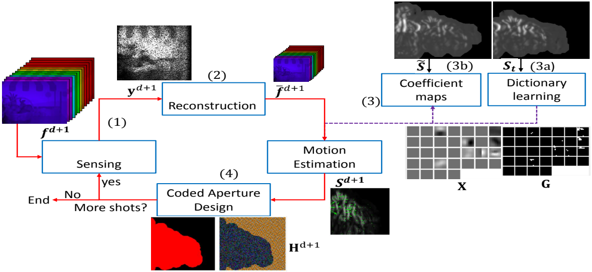


Block-unblock coded aperture (BUA).

Size of $\mathbf{F}^d \in \mathbb{R}^{128\times128\times12}$ and 23 frames.

Conclusions

- A new design of adaptive colored coded apertures (VA-CCA) for spectral video.
- The approach provides a motion estimation between frames to sample the static and the dynamic scene differently.
- The proposed approach overcomes, block-unblock CA (2.4 dB), random-colored CA (1.46 dB), non-adaptive blue noise (0.68 dB).



Questions?

References

C. V. Correa, H. Arguello, and G. R. Arce, "Spatiotemporal blue noise coded aperture design for multi-shot compressive spectral imaging," *J. Opt. Soc. Am. A*, vol. 33, no. 12, pp. 2312–2322, Dec 2016. [Online]. Available: http://josaa.osa.org/abstract.cfm?URI=josaa-33-12-2312

B. Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 301–315, Jan 2016.

K. M. León-López, L. V. Galvis Carreño, and H. Arguello Fuentes, "Temporal colored coded aperture design in compressive spectral video sensing," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 253–264, Jan 2019.

Video Motion Estimation Algorithm (Initialization)

Input: $\lambda_s, \lambda_p, K, D, \lambda, \rho,\ \widetilde{\mathbf{S}}, \mathbf{S}_t$: Training/test video motions
Output: $\mathbf{S}_\ell^d$

function CODED APERTURE DESIGN USING VIDEO MOTION ESTIMATION $(\mathbf{y}^0, \mathbf{y}^1, \lambda_s, \lambda_p, K, J, \lambda, \rho, \widetilde{\mathbf{S}}, \mathbf{S}_t)$
- $\mathbf{G}_v \leftarrow$ Computes the dictionary by solving (5)
- $\mathbf{X}_v \leftarrow$ Computes the coefficient maps by solving (6)
- $\mathbf{y}^0 \leftarrow \mathbf{H}^0\mathbf{f}$ ▷ First snapshot
- $\hat{\mathbf{f}}_L^0 \leftarrow \boldsymbol{\Psi}_L^{-1}(\text{argmin}_{\boldsymbol{\theta}_L} \|\mathbf{y}^0 - \mathbf{H}_L^0\boldsymbol{\Psi}_L^d\boldsymbol{\theta}_L^d\|_2^2 + \tau\|\boldsymbol{\theta}_L^d\|_1)$
- ▷ Low-resolution
- $\hat{\mathbf{f}}_H^{\mathbf{0}} \leftarrow \mathbf{P}(\hat{\mathbf{f}}_L^0)$ ▷ Interpolation
- $\hat{\mathbf{F}}_H^0 \leftarrow rearrange(\hat{\mathbf{f}}_H^0)$ ▷ Rearrange
- $\hat{\mathbf{f}} \leftarrow \boldsymbol{\Psi}^{-1}(\text{argmin}_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{H}\boldsymbol{\Psi}\boldsymbol{\theta}\|_2^2 + \tau\|\boldsymbol{\theta}\|_1)$
- Motion estimation and Adaptive Coded Aperture Desing
- **return** $\mathbf{S}_\ell^d$ ▷ (Estimated motion field)

Motion estimation and Adaptive Coded Aperture Design

for $k \leftarrow 1, K$ **do**

  for $d \leftarrow 1, D$ **do**

  $\hat{\mathbf{f}}_L^d \leftarrow \boldsymbol{\Psi}_L^{-1}(\text{argmin}_{\boldsymbol{\theta}_L} \|\mathbf{y}^d - \mathbf{H}_L^d \boldsymbol{\Psi}_L^d \boldsymbol{\theta}_L^d\|_2^2 + \tau \|\boldsymbol{\theta}_L^d\|_1)$

  ▷ Low-resolution

    $\hat{\mathbf{f}}_H^{\mathbf{d}} \leftarrow \mathbf{P}(\hat{\mathbf{f}}_L^d)$ ▷ Interpolation

    $\hat{\mathbf{F}}_H^d \leftarrow \textit{rearrange}(\hat{\mathbf{f}}_H^d)$ ▷ Rearrange

    for $\ell \leftarrow 1, L$ **do**

      $\text{argmin}_{\mathbf{S}_\ell^d} \{ E_{\text{data}}(\hat{\mathbf{F}}_H^{d-1}, \hat{\mathbf{F}}_H^d, \mathbf{S}_\ell^{d-1}) +$
      $\lambda_s \|\nabla \mathbf{S}_\ell^{d-1}\|_2^2 + \lambda_p(k) \|\mathbf{S}_\ell^{d-1} - \sum_v \mathbf{G}_v * \mathbf{X}_v\|_2^2 \}$
      s.t. $\|\mathbf{G}_v\| = 1 \; \forall v$ ▷ Video motion estimation

      $\mathbf{Q}_\ell^d \leftarrow (\mathbf{S}_\ell^{d-1}, \mathbf{S}_\ell^d)$ ▷ Thresholding motion

      $\mathbf{q}_\ell^d \leftarrow \textit{vec}(\mathbf{Q}_\ell^d)$ ▷ Vectorized motion areas

    $\mathbf{r}_\ell^d \leftarrow \mathbf{q}_\ell^d \odot \mathbf{b}_\ell^d + (\mathbf{1} - \mathbf{q}_\ell^d) \odot \hat{\mathbf{b}}_\ell^d$ ▷ Next code

      $\mathbf{H}_\ell^d \leftarrow \textit{rearrange}(\mathbf{r}_\ell^d)$ ▷ Rearrange

    $\mathbf{y}^d \leftarrow \mathbf{H}^d \mathbf{f}$ ▷ Next snapshot

Parameters λ and ρ

	$\rho_0 = 25$	$\rho_1 = 50$	$\rho_2 = 100$	$\rho_3 = 150$
$\lambda_0 = 0.0005$	30.9600	30.8260	30.7236	31.1311
$\lambda_1 = 0.0001$	38.1156	37.7255	38.6955	39.9385
$\lambda_2 = 0.00005$	33.5062	32.0604	**40.3777**	38.4921
$\lambda_3 = 0.00001$	29.6564	30.5600	30.0820	29.5902
$\lambda_4 = 0.000001$	34.5530	28.1833	33.5431	30.4982

Table 1: Image quality (PSNR) with 100 iteration of (CBPDN) for different choices of λ and ρ by using the motion horizontal ground-truth.

	$\rho_0 = 25$	$\rho_1 = 50$	$\rho_2 = 100$	$\rho_3 = 150$
$\lambda_0 = 0.0005$	37.7552	38.2876	38.5107	38.6064
$\lambda_1 = 0.0001$	41.3404	39.6111	39.7874	40.1052
$\lambda_2 = 0.00005$	41.6682	39.8263	40.6523	42.0653
$\lambda_3 = 0.00001$	42.1040	39.8192	40.1028	**49.2796**
$\lambda_4 = 0.000001$	41.0436	39.6918	40.9101	38.4830

Image quality (PSNR) with 100 iteration of (CBPDN) for different choices of λ and ρ by using the motion vertical ground-truth.