

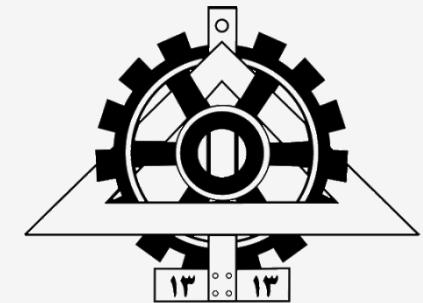


Hyperspectral Remote Sensing

Target Detection



University of Tehran
College of Engineering
Faculty of Surveying and Geospatial engineering
Remote Sensing



Dr. Hasanlou

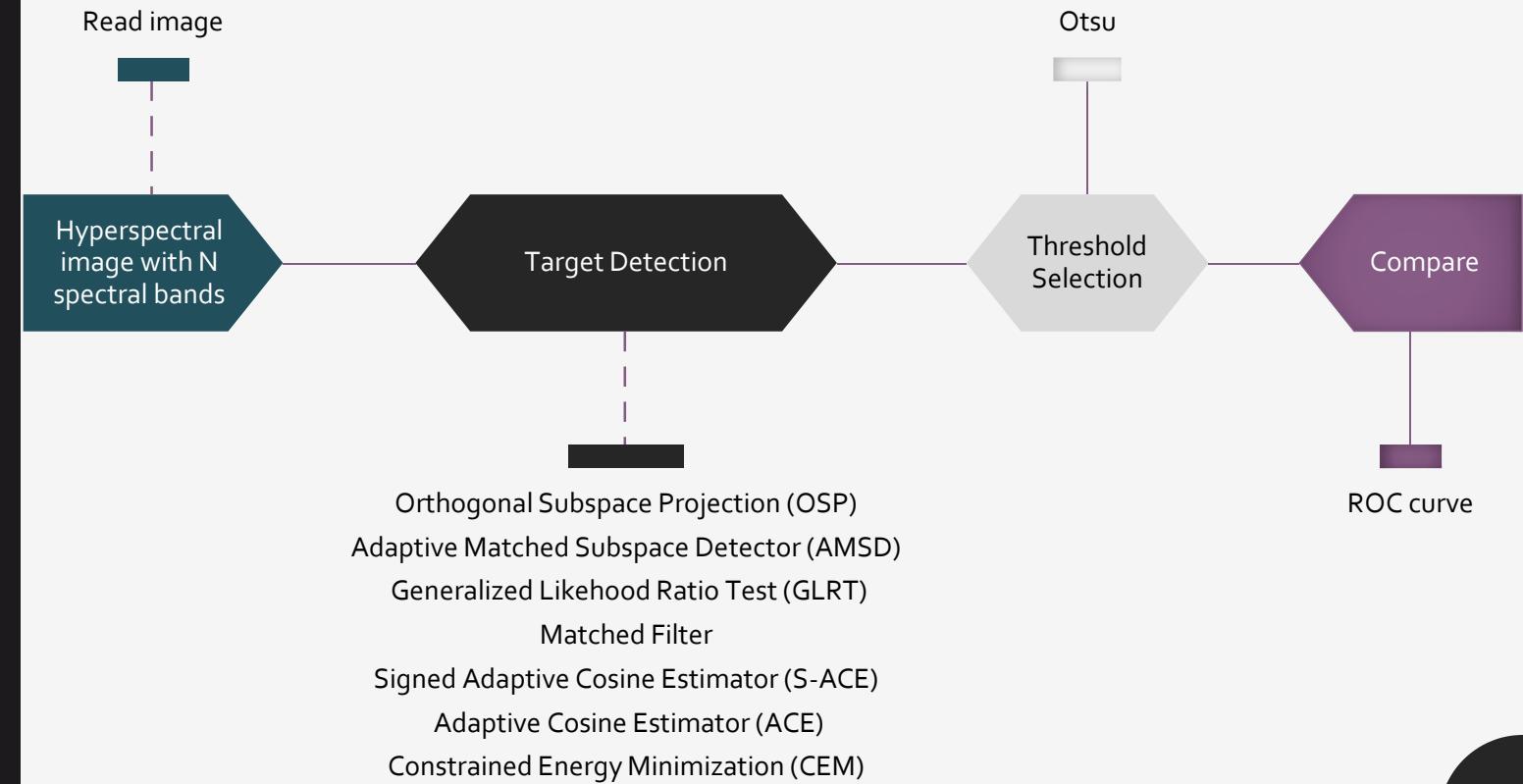
Faezeh Zamiri

810399040

aghdam.zamiri.fa@ut.ac.ir

General scheme of Target Detection

Method



Target Detection

Structured methods for target detection



Constrained Energy
Minimization
(CEM)



Adaptive Cosine
Estimator
(ACE)



Signed Adaptive
Cosine Estimator
(S-ACE)



Matched Filter



Generalized Likelihood
Ratio Test
(GLRT)

Target Detection

Unstructured methods for target detection

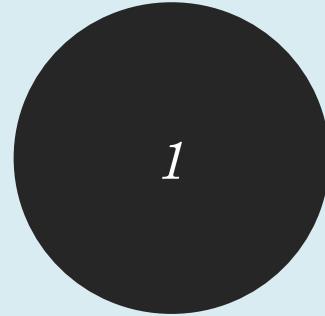


Adaptive Matched Subspace Detector (AMSD)



Orthogonal Subspace Projection (OSP)

Threshold Selection

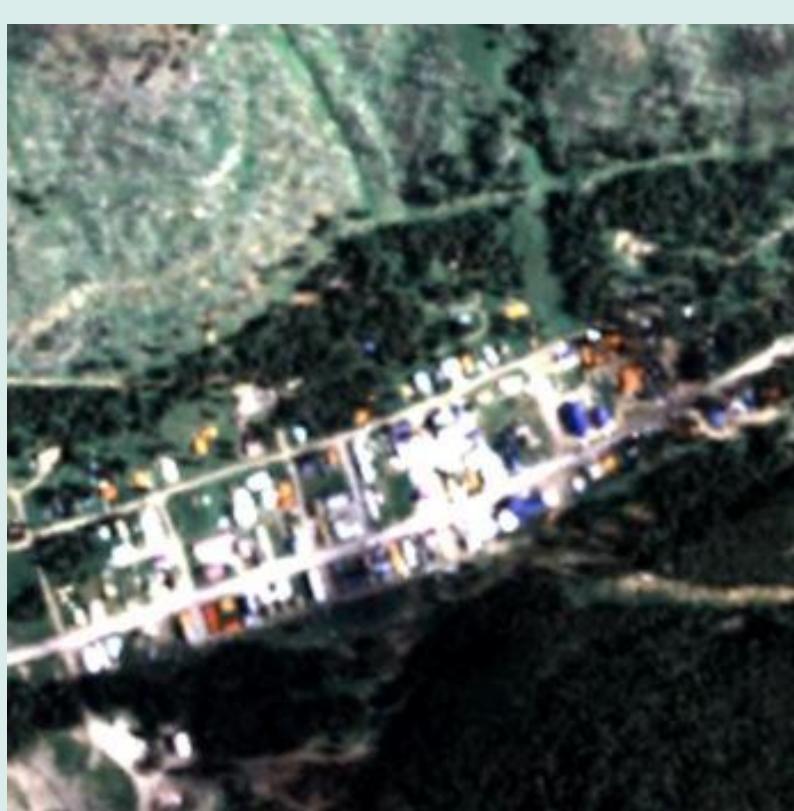


Otsu

Otsu's method chooses a threshold that minimizes the intraclass variance of the thresholded black and white pixels.

Data

Data that I used in this exercise.

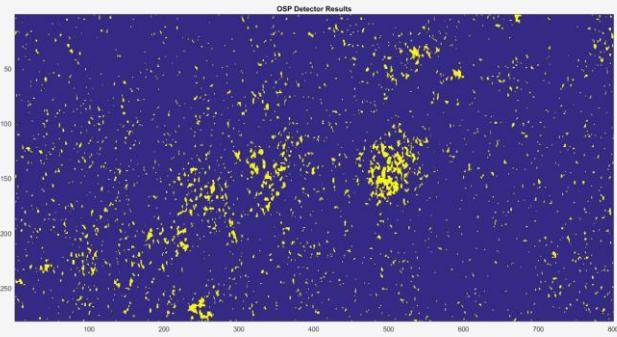


The Cooke City Dataset

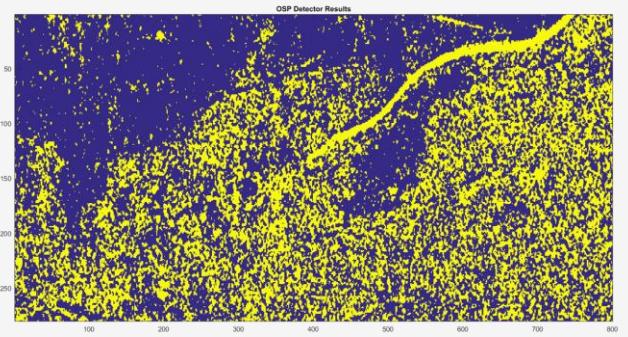
(280*800) *126

Result of OTSU Threshold Selection

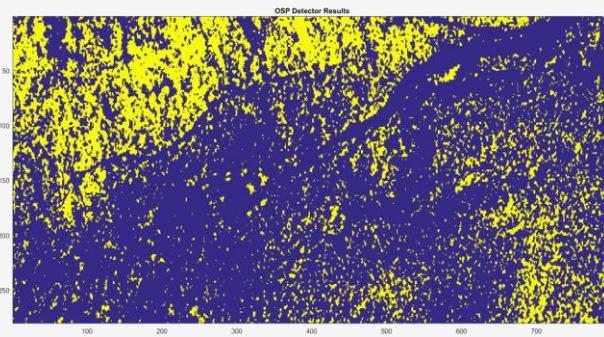
Method	Fabric 1	Fabric 2	Fabric 3	Fabric 4	Vehicle 1	Vehicle 2	Vehicle 3
CEM	0.0010	0.0028	0.0035	0.0011	0.0032	0.0090	0.0012
ACE	0.0155	0.0182	0.0183	0.0159	0.0150	0.0185	0.0164
S-ACE	0.0135	0.0166	0.0155	0.0136	0.0122	0.0159	0.0134
MF	0.0010	0.0027	0.0035	0.0011	0.0032	0.0089	0.0012
GLRT	0.0154	0.0180	0.0181	0.0157	0.0149	0.0183	0.0163
OSP	0.0026	0.0064	0.0011	0.0028	0.0068	0.0154	0.0021
AMSD	0.0981	0.0783	0.0648	0.0934	0.0903	0.1622	0.0647



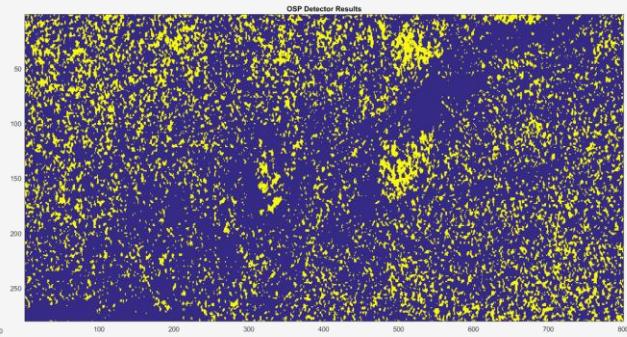
Fabric 1
 $T = 0.0026$



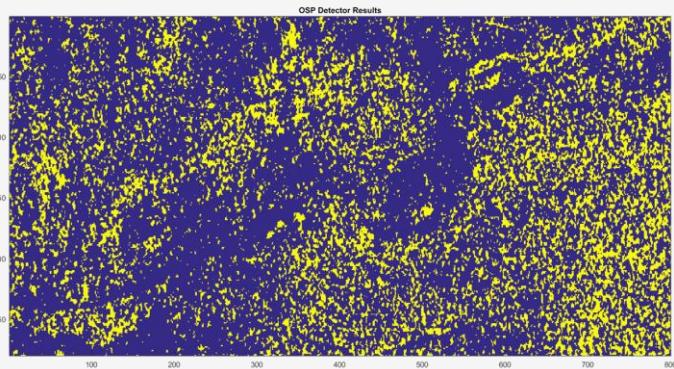
Fabric 2
 $T = 0.0064$



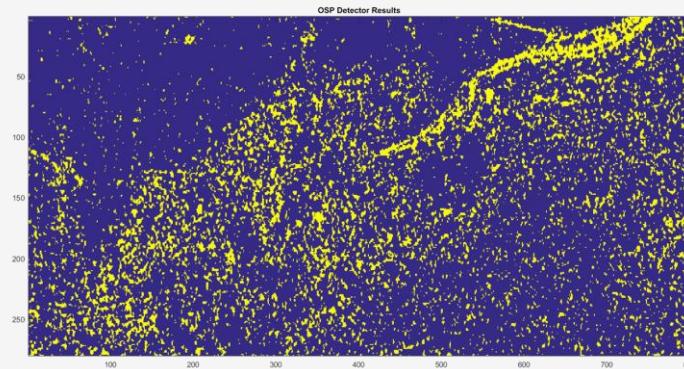
Fabric 3
 $T = 0.0011$



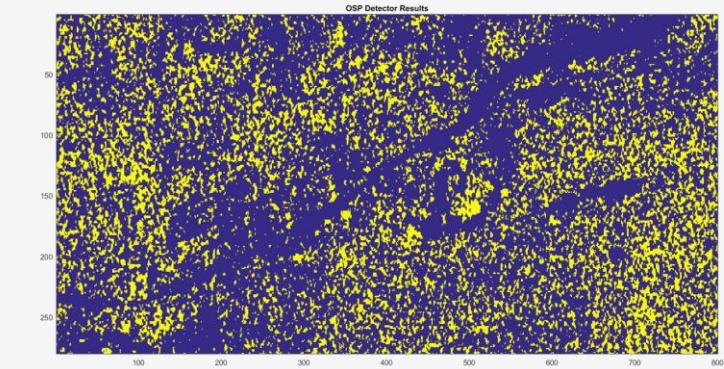
Fabric 4
 $T = 0.0028$



Vehicle 1
 $T = 0.0068$



Vehicle 2
 $T = 0.0154$



Vehicle 3
 $T = 0.0021$

Threshold Selection Result on OSP Method

9

- This algorithm seems to work optimistically or because the amount of target data is small, this algorithm has low accuracy

CEM

Constrained Energy Minimization

- HYPERCEM Performs constrained energy minimization (CEM) algorithm
- Performs the constrained energy minimization algorithm for target detection.

Usage

[results] = hyperCem(M, target)

Inputs

M - 2d matrix of HSI data ($p \times N$) → 126*224000

target - target of interest ($p \times 1$) → 126*1

Outputs

results - vector of detector output ($N \times 1$) → 224000*1

ACE

Adaptive Cosine Estimator

- HYPERACE Performs the adaptive cosin/coherent estimator algorithm
- Performs the adaptive cosin/coherent estimator algorithm for target detection.

Usage

[results] = hyperAce(M, S)

Inputs

M - 2d matrix of HSI data ($p \times N$) → 126*224000

S - 2d matrix of target endmembers ($p \times 1$) → 126*1

Outputs

results - vector of detector output ($N \times 1$) → 224000*1

GLRT

Generalized Likelihood Ratio Test

- HYPERGLRT Performs the generalized likelihood test ratio algorithm
- Performs the generalized likelihood test ratio algorithm for target detection.

Usage

[results] = hyperGlrt(M, U, target)

Inputs

M - 2d matrix of HSI data ($p \times N$) → 126*224000

t - target of interest ($p \times 1$) → 126*1

Outputs

results - vector of detector output ($N \times 1$) → 224000*1

S-ACE

Signed Adaptive Cosine Estimator (S-ACE)

- HYPERACE Performs the adaptive cosin/coherent estimator algorithm
- Performs the adaptive cosin/coherent estimator algorithm for target detection.

Usage

[results] = hyperSignedAce(M, S)

Inputs

M - 2d matrix of HSI data ($p \times N$) → 126*224000

S - target of interest ($p \times 1$) → 126*1

Outputs

results - vector of detector output ($N \times 1$) → 224000*1

MF

Matched Filter

- HYPERACE Performs Matched Filter estimator algorithm
- Performs the estimator algorithm for target detection.

Usage

[results] = hyperMatchedFilter(M, S)

Inputs

M - 2d matrix of HSI data ($p \times N$) → 126*224000

S - target of interest ($p \times 1$) → 126*1

Outputs

results - vector of detector output ($N \times 1$) → 224000*1

OSP

Orthogonal Subspace Projection

- HYPEROSP Performs the orthogonal subspace projection (OSP) algorithm
- Performs the orthogonal subspace projection algorithm for target detection.

Usage

```
[results] = hyperOsp(M, U, target)
```

Inputs

M - 2d matrix of HSI data ($p \times N$)

U - 2d matrix of background endmembers ($p \times q$)

target - target of interest ($p \times 1$)

Outputs

results - vector of detector output ($N \times 1$)

AMSD

Adaptive Matched Subspace Detector

2 of 2 Unstructured Method

- HYPERAMSD Adaptive matched subspace detector (AMSD) algorithm
- Performs the adaptive matched subspace detector (AMSD) algorithm for target detection

Usage

```
[results] = hyperAmsd(M, U, target)
```

Inputs

M - 2d matrix of HSI data ($p \times N$)

B - 2d matrix of background endmembers ($p \times q$)

target - target of interest ($p \times 1$)

Outputs

results - vector of detector output ($N \times 1$)

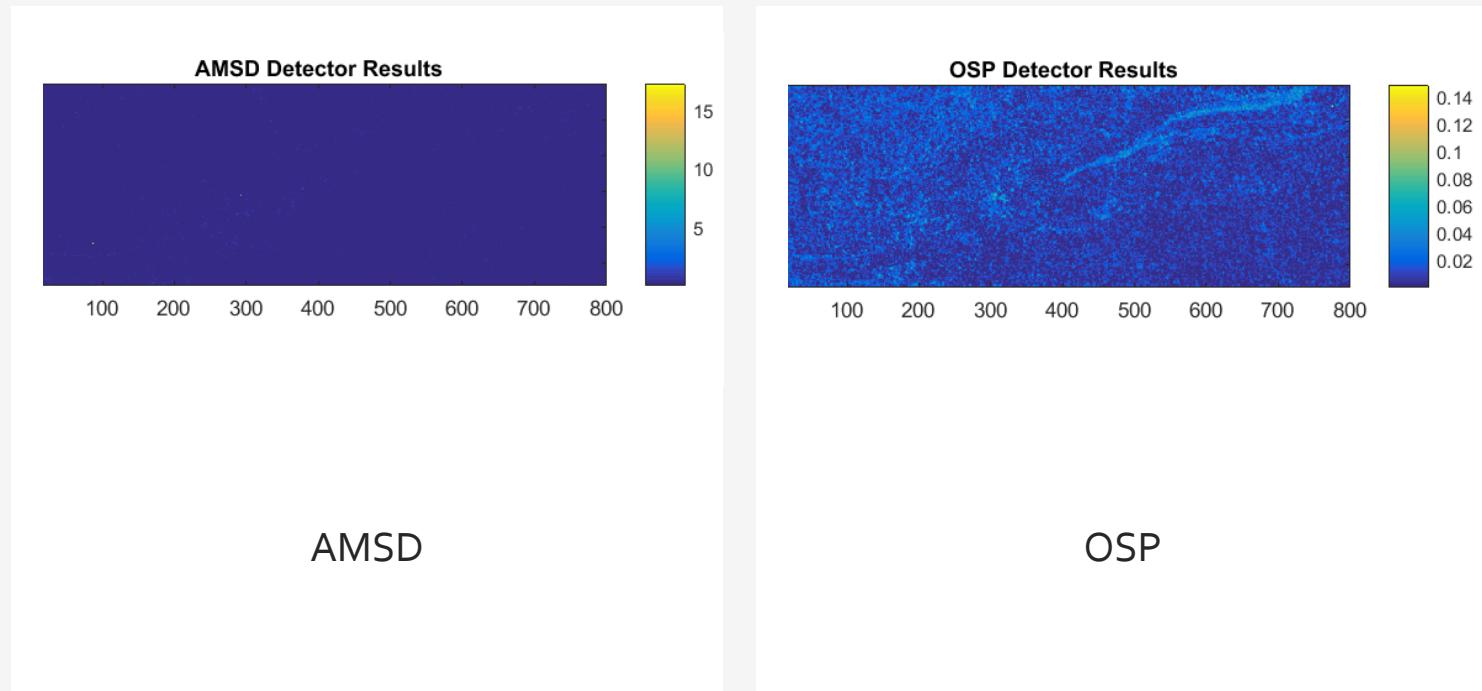
Fabric 1

Result of Structured Method
Detector Map



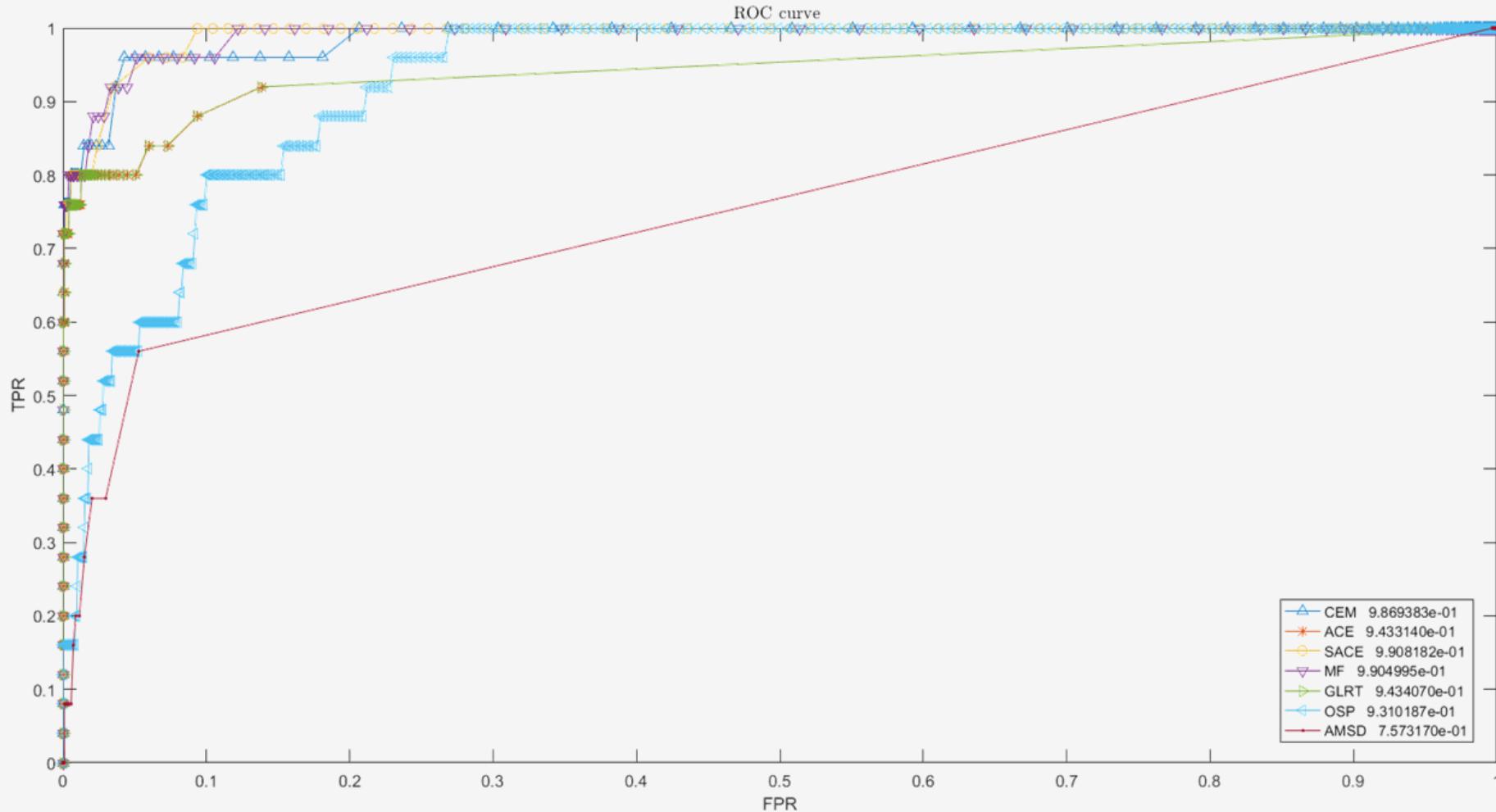
Fabric 1

Result of Unstructured Method
Detector Map



In Unstructured method, first of all estimate number of endmembers with HFC ,that is 36 , then calculate matrix of background endmembers with ATGP and finally use AMSD and OSP to target detection.

Fabric 1



- ROC curve of Fabric 1 data for 1000 different thresholds
- The AUC of each curve calculated

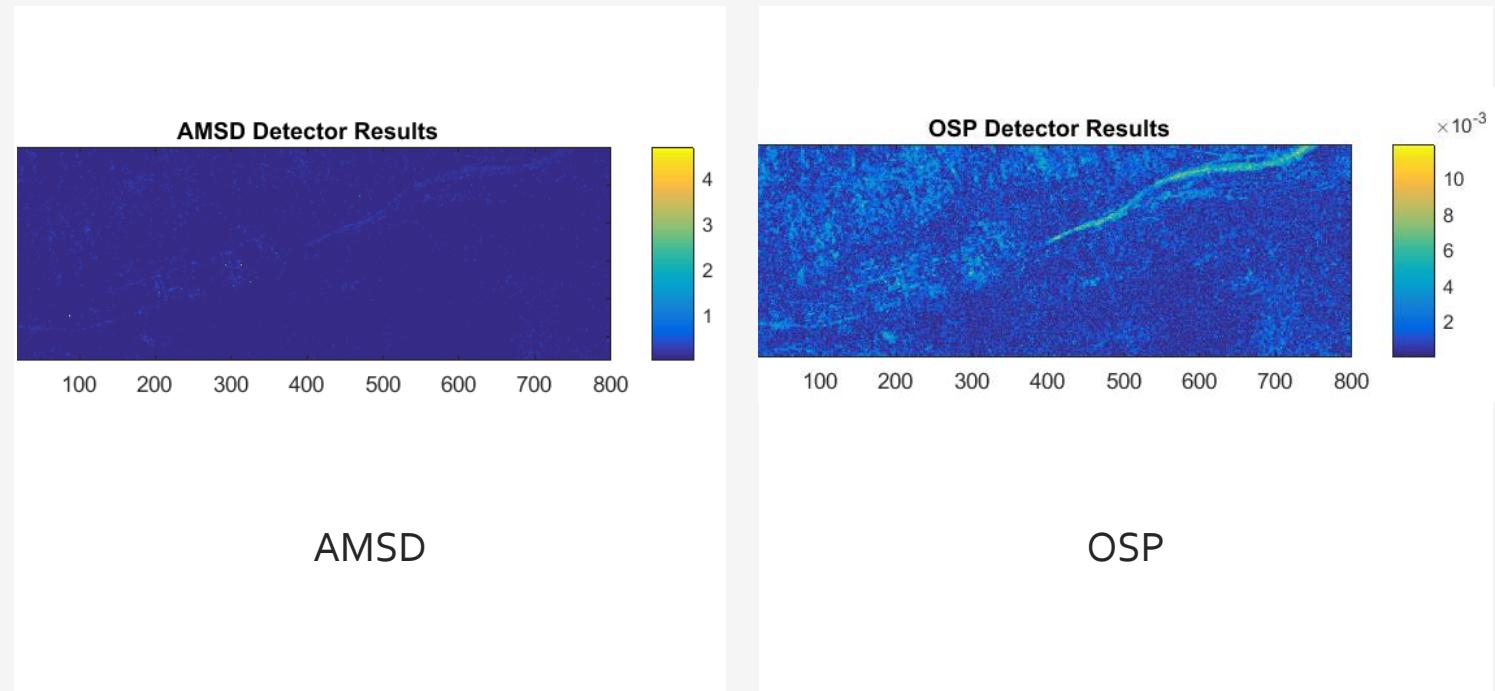
Fabric 2

Result of Structured Method
Detector Map



Fabric 2

Result of Unstructured Method
Detector Map

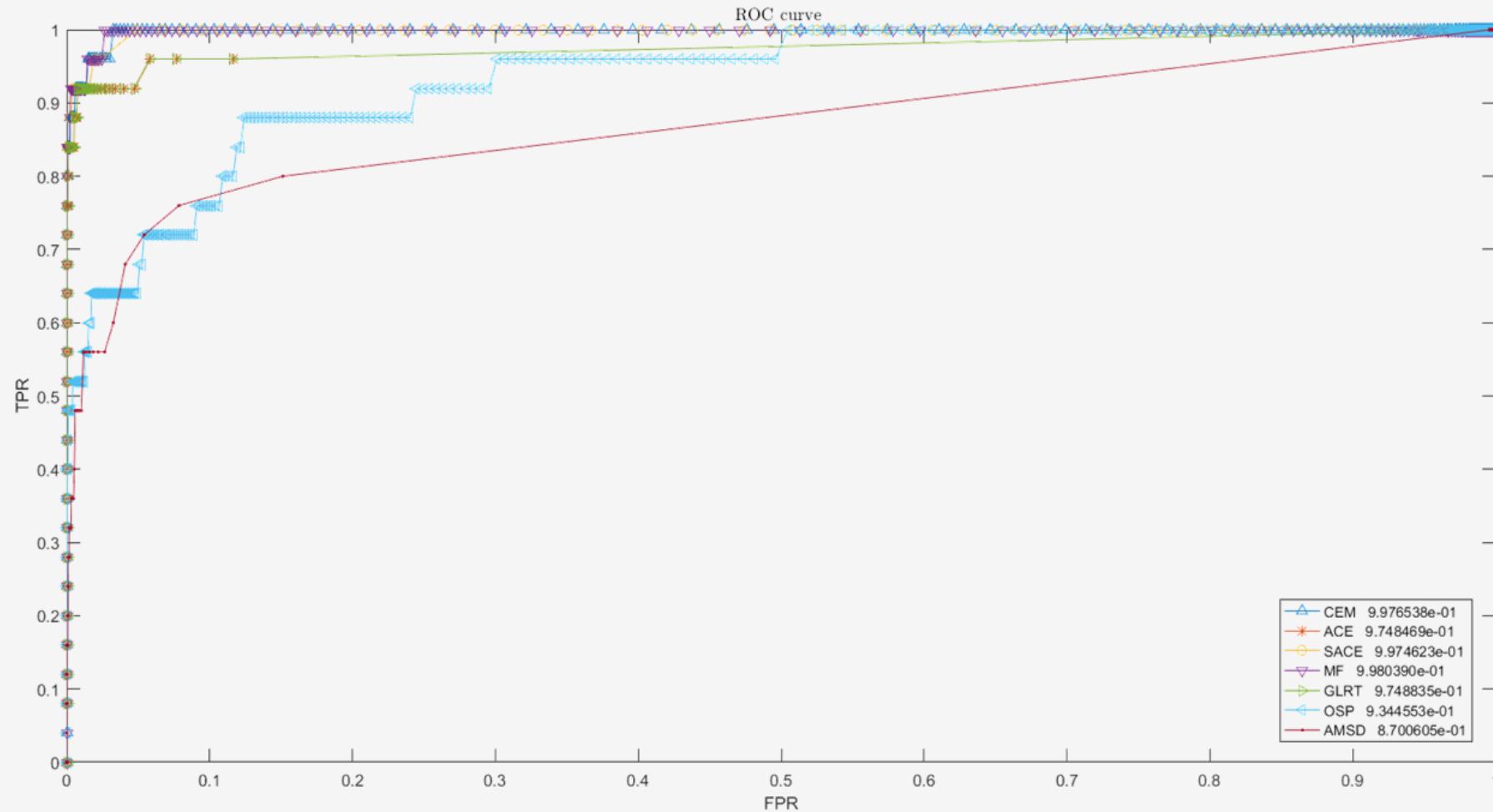


AMSD

OSP

In Unstructured method, first of all estimate number of endmembers with HFC ,that is 36 , then calculate matrix of background endmembers with ATGP and finally use AMSD and OSP to target detection.

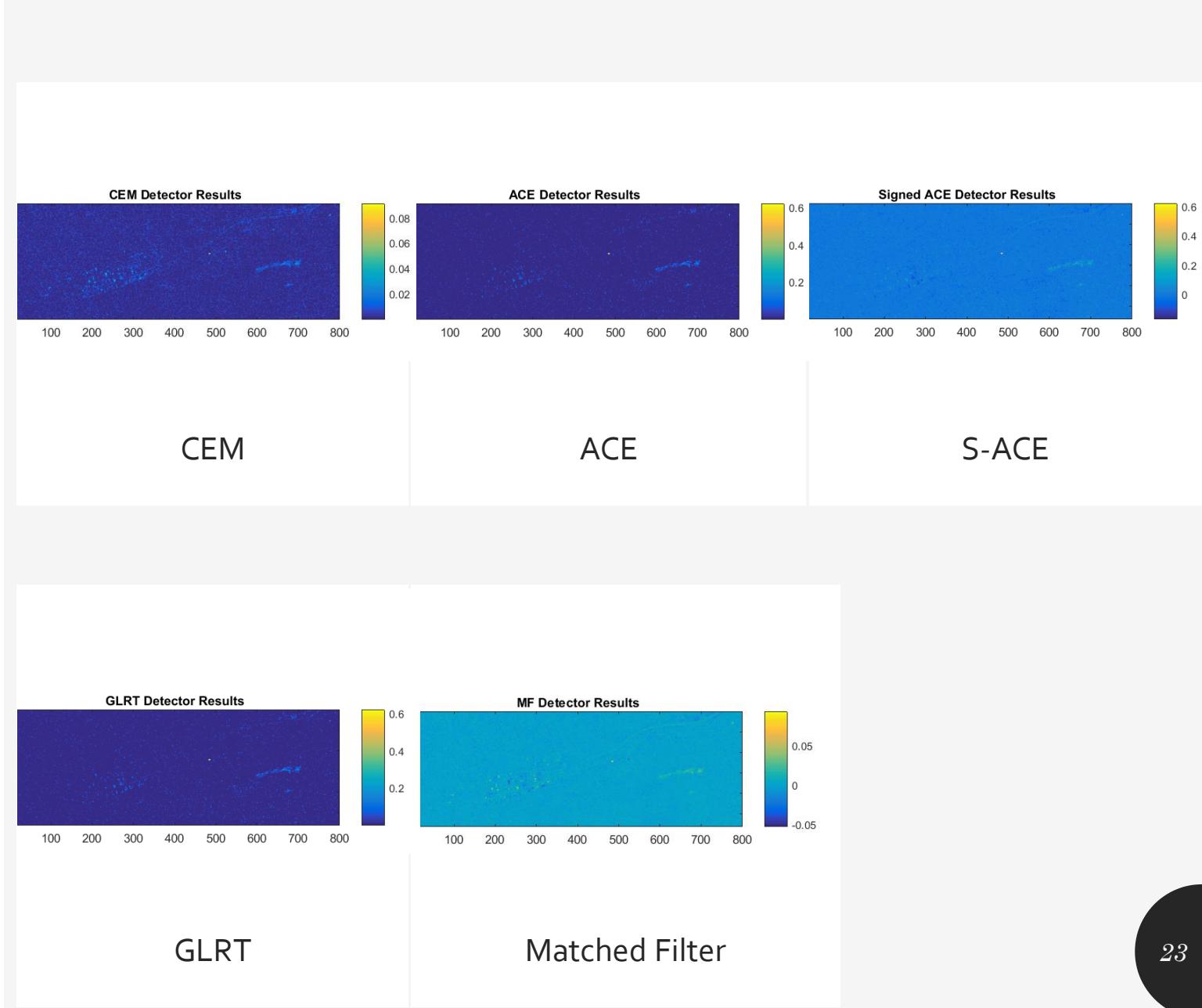
Fabric 2



- ROC curve of Fabric 2 data for 1000 different thresholds
- The AUC of each curve calculated

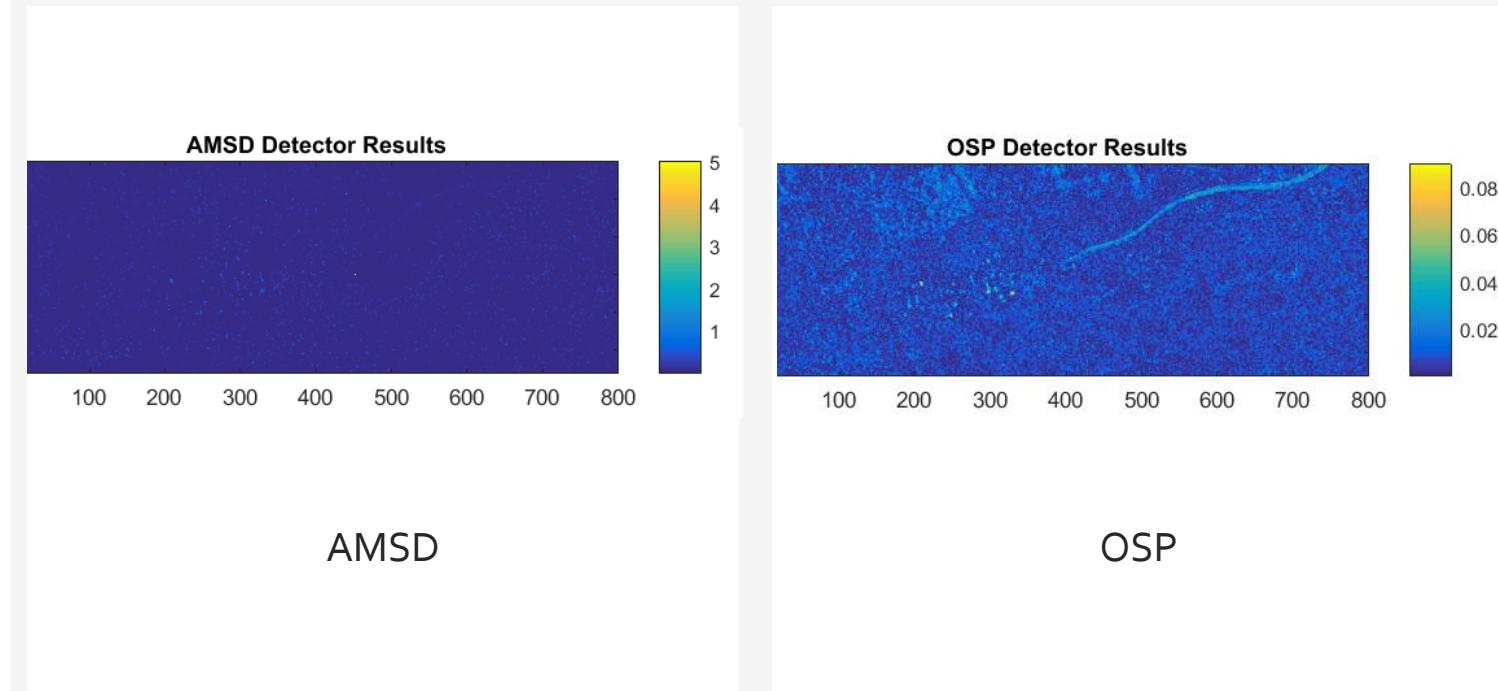
Fabric 3

Result of Structured Method
Detector Map



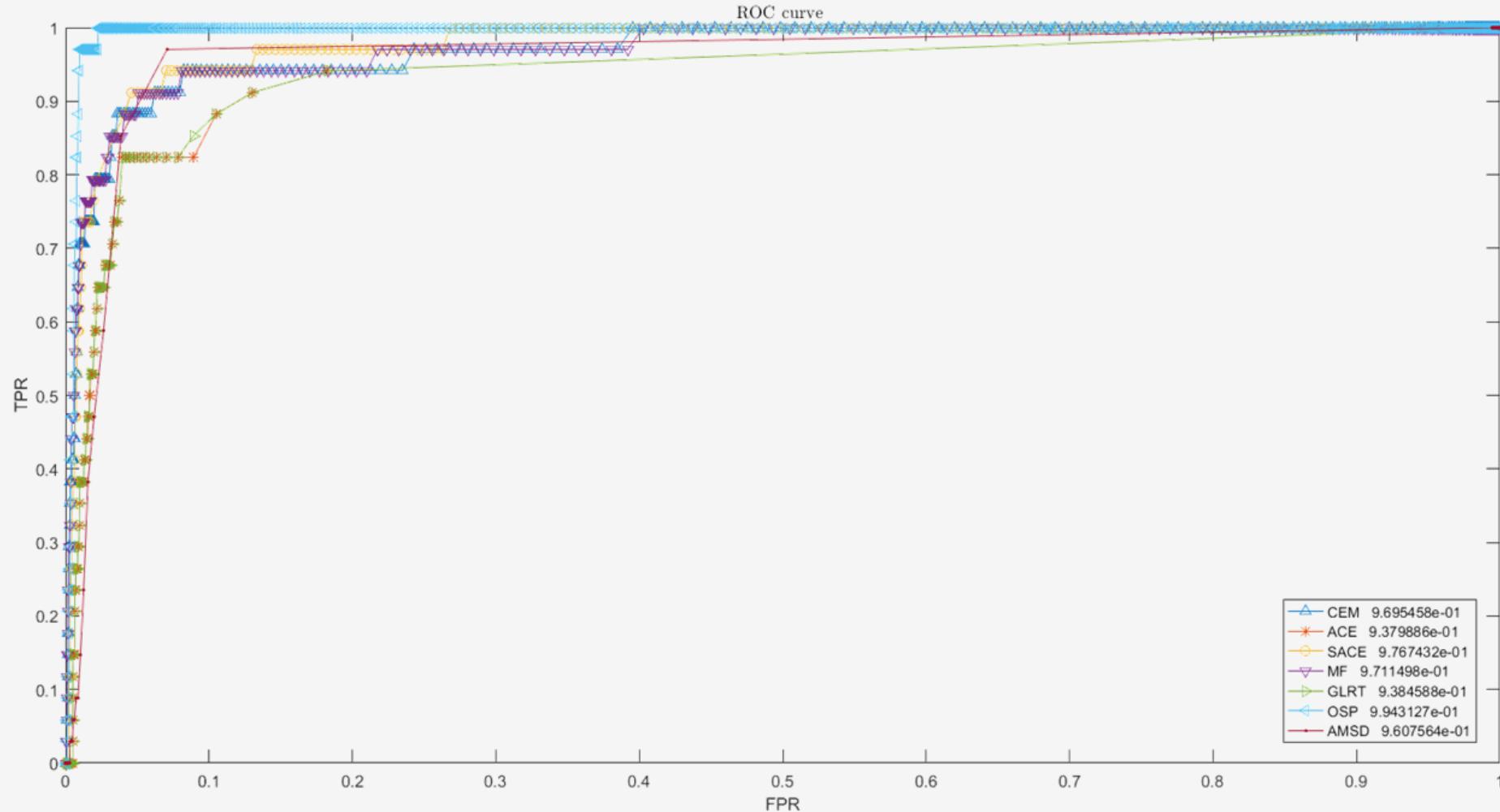
Fabric 3

Result of Unstructured Method
Detector Map



In Unstructured method, first of all estimate number of endmembers with HFC ,that is 36 , then calculate matrix of background endmembers with ATGP and finally use AMSD and OSP to target detection.

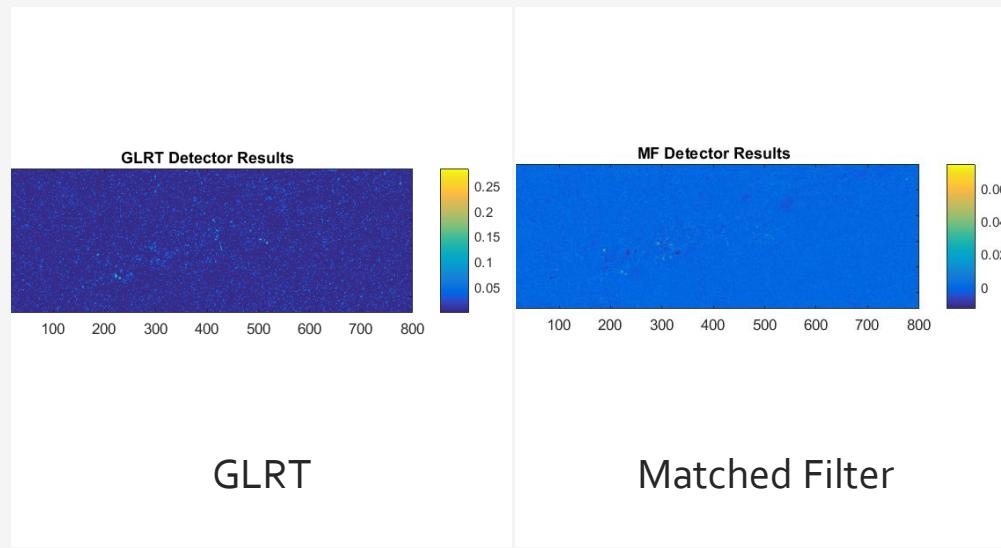
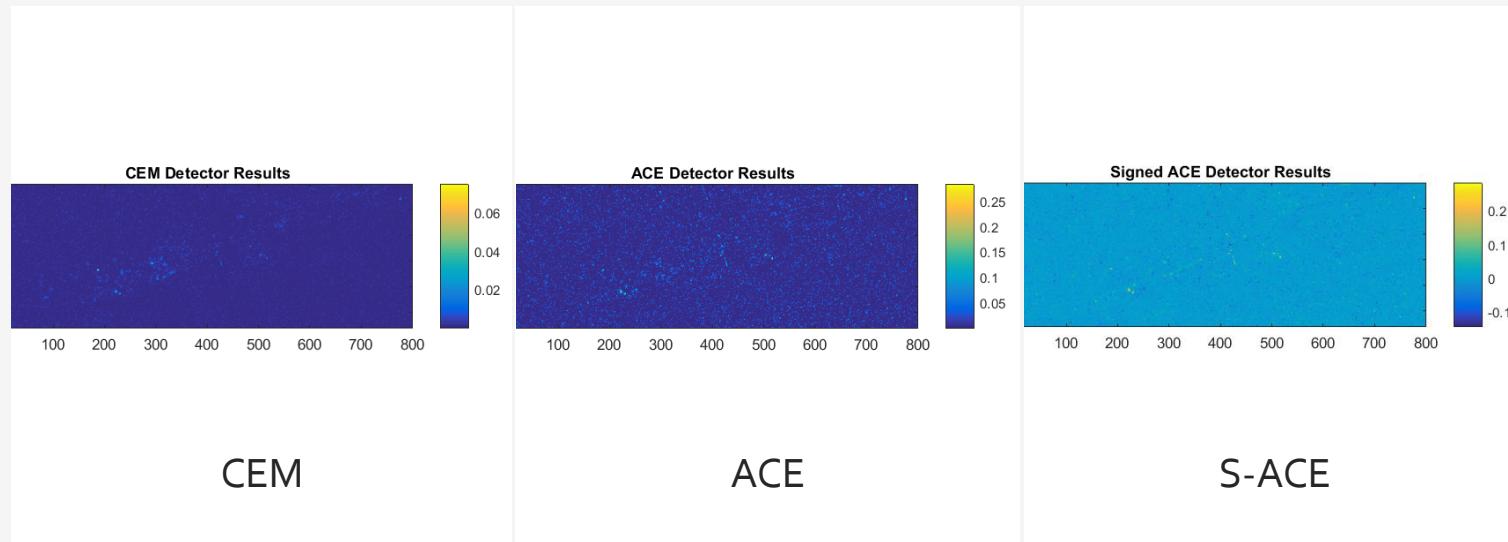
Fabric 3



- ROC curve of Fabric 3 data for 1000 different thresholds
- The AUC of each curve calculated

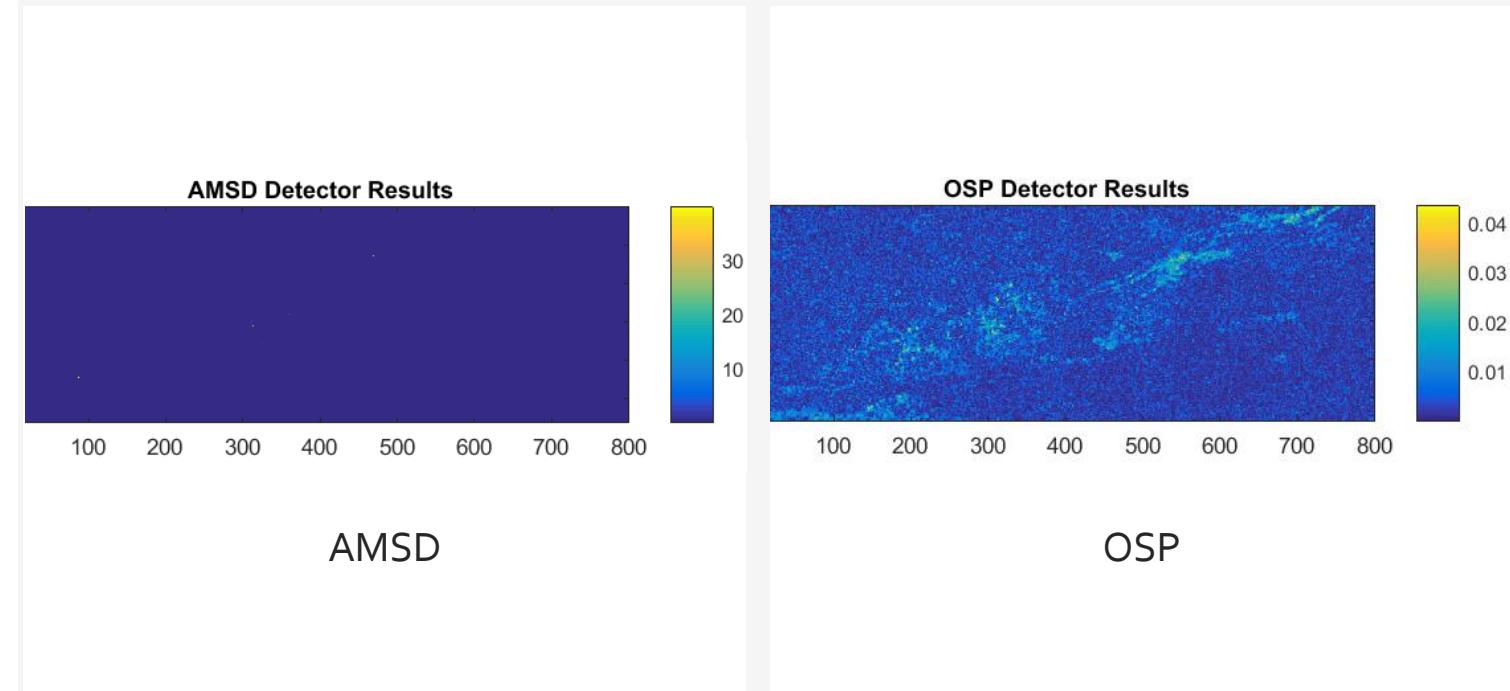
Fabric 4

Result of Structured Method
Detector Map



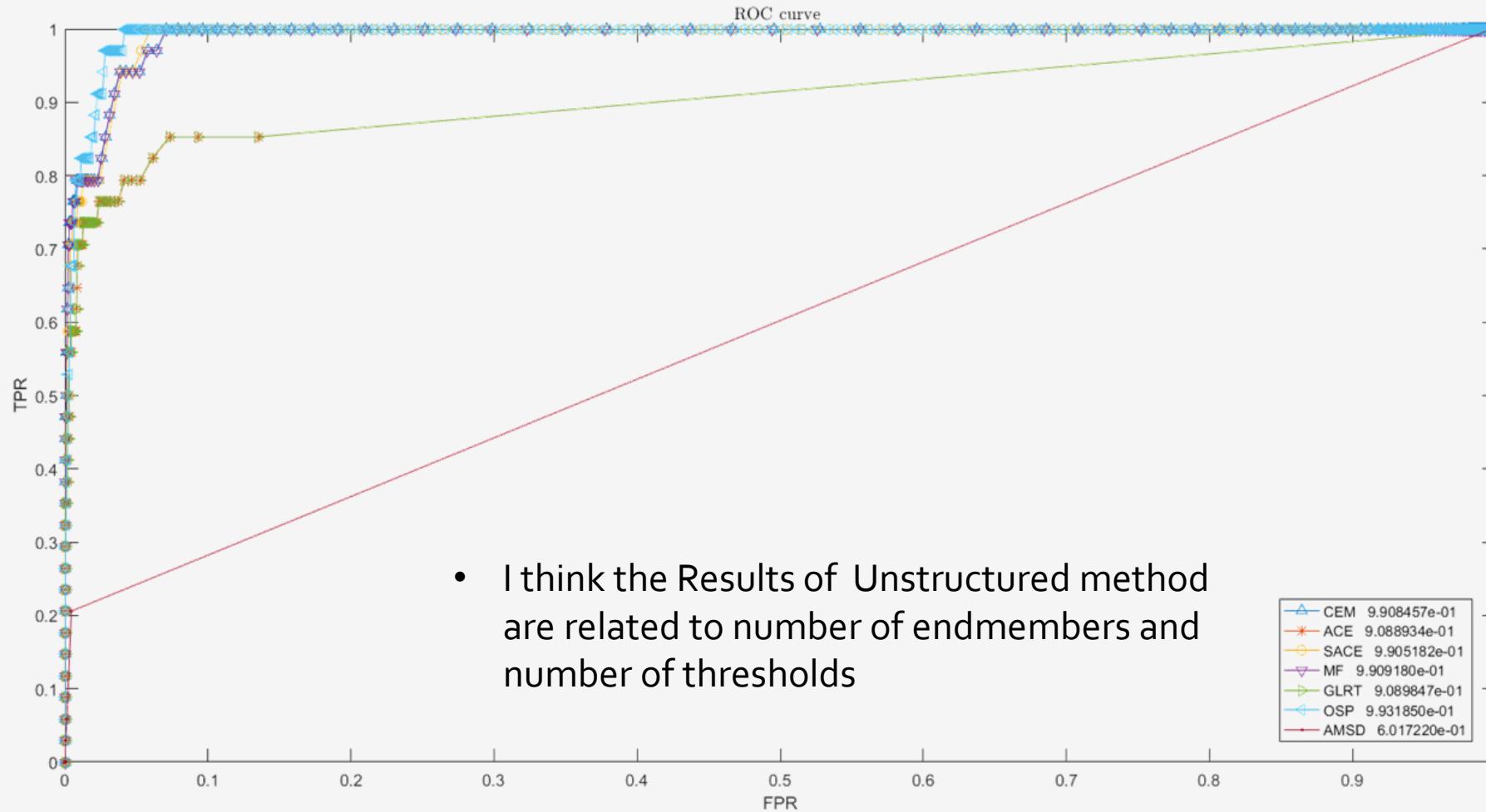
Fabric 4

Result of Unstructured Method
Detector Map



In Unstructured method, first of all estimate number of endmembers with HFC ,that is 36 , then calculate matrix of background endmembers with ATGP and finally use AMSD and OSP to target detection.

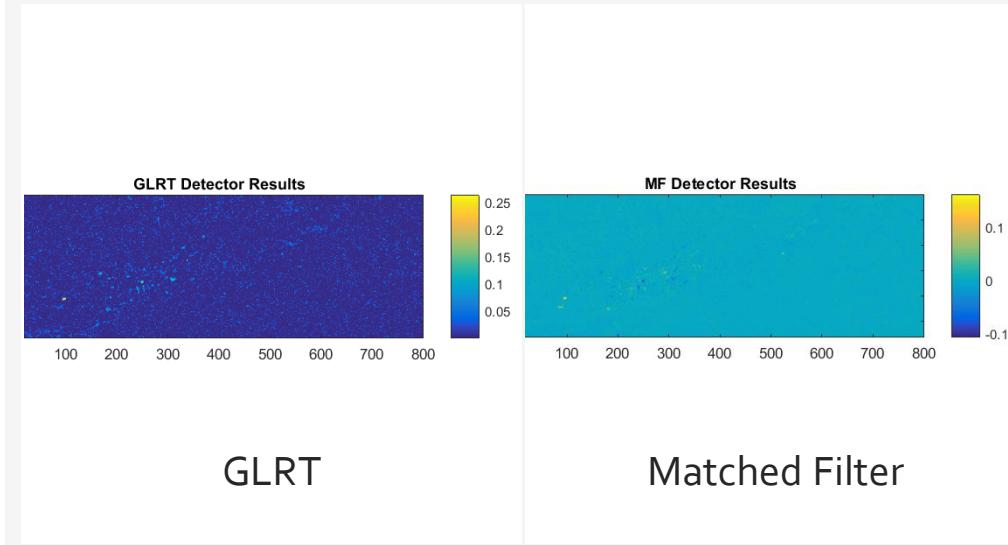
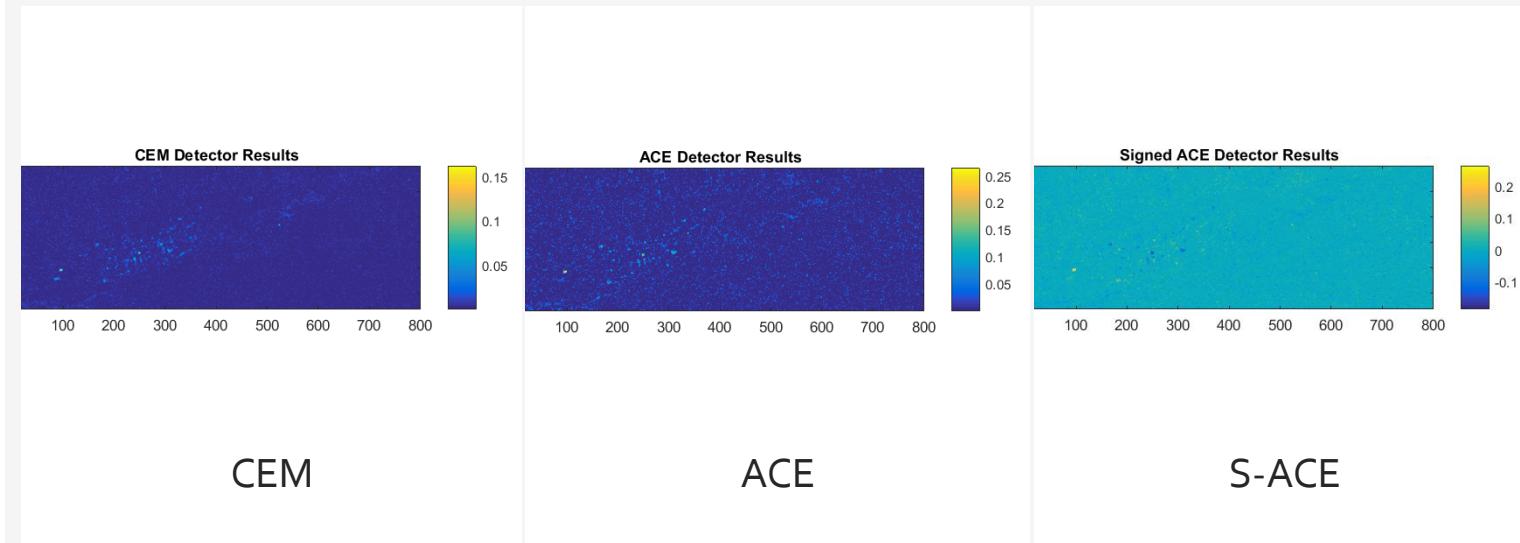
Fabric 4



- ROC curve of Fabric 4 data for 1000 different thresholds
- The AUC of each curve calculated

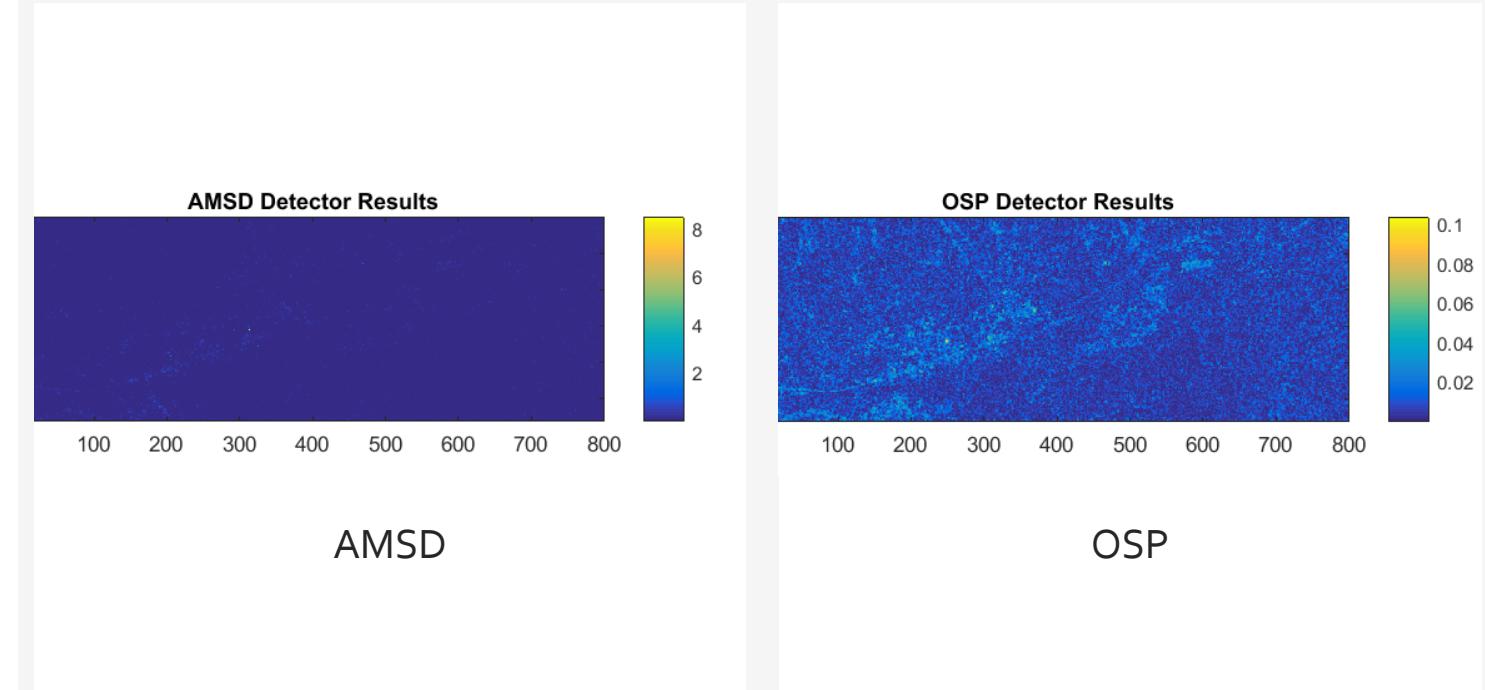
Vehicle 1

Result of Structured Method
Detector Map



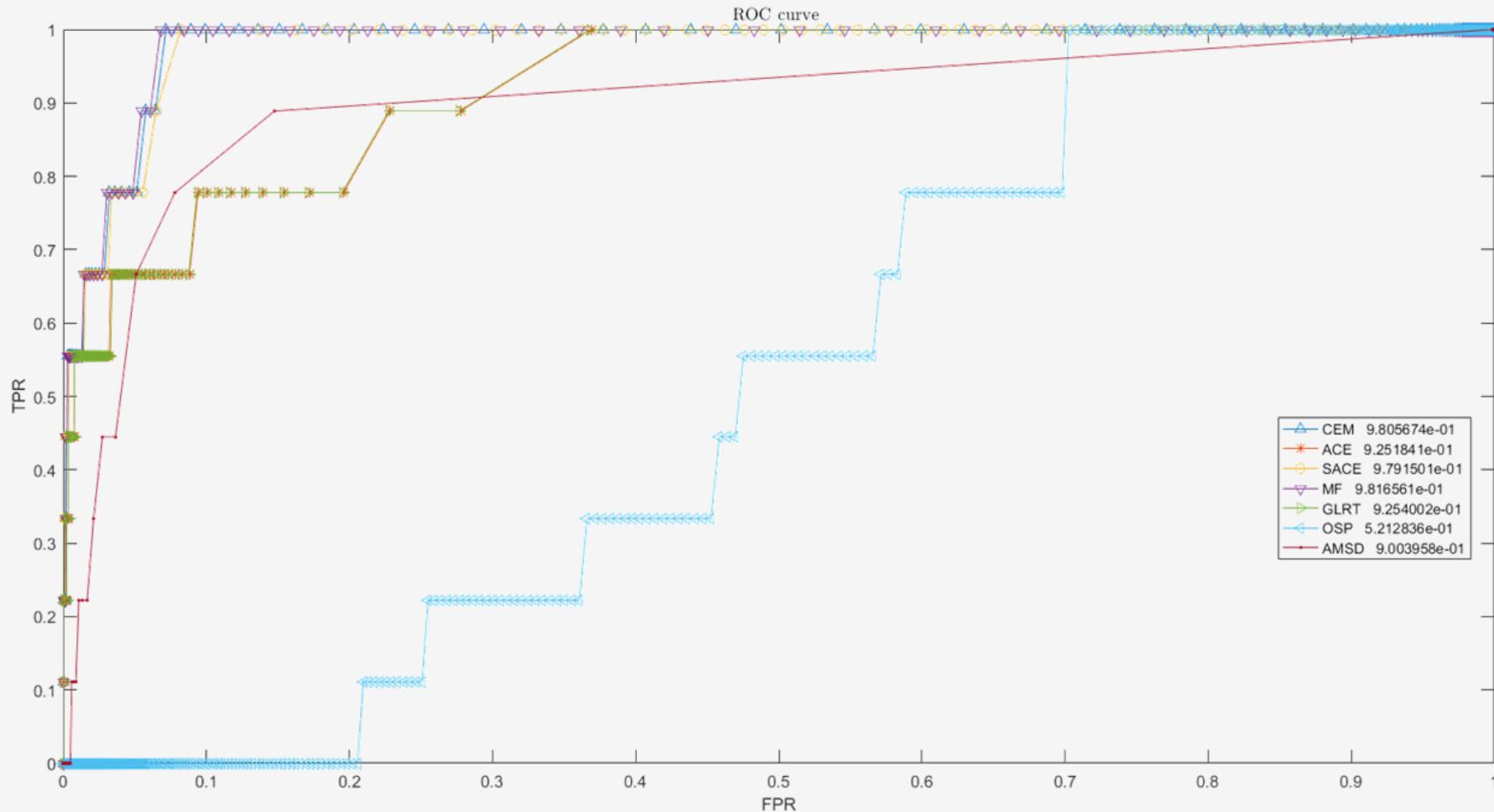
Vehicle 1

Result of Unstructured Method
Detector Map



In Unstructured method, first of all estimate number of endmembers with HFC ,that is 36 , then calculate matrix of background endmembers with ATGP and finally use AMSD and OSP to target detection.

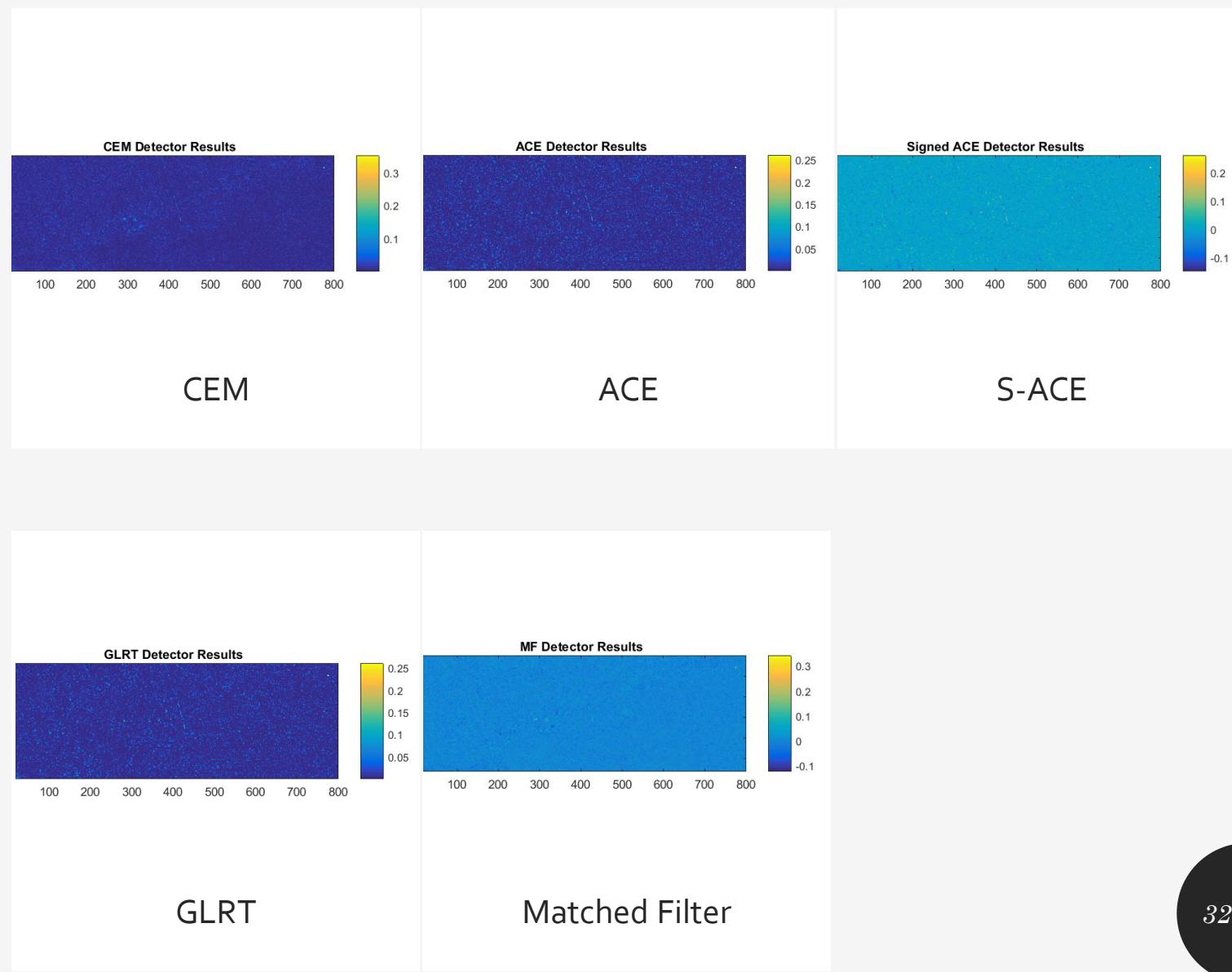
Vehicle 1



- ROC curve of Vehicle 1 data for 1000 different thresholds
- The AUC of each curve calculated

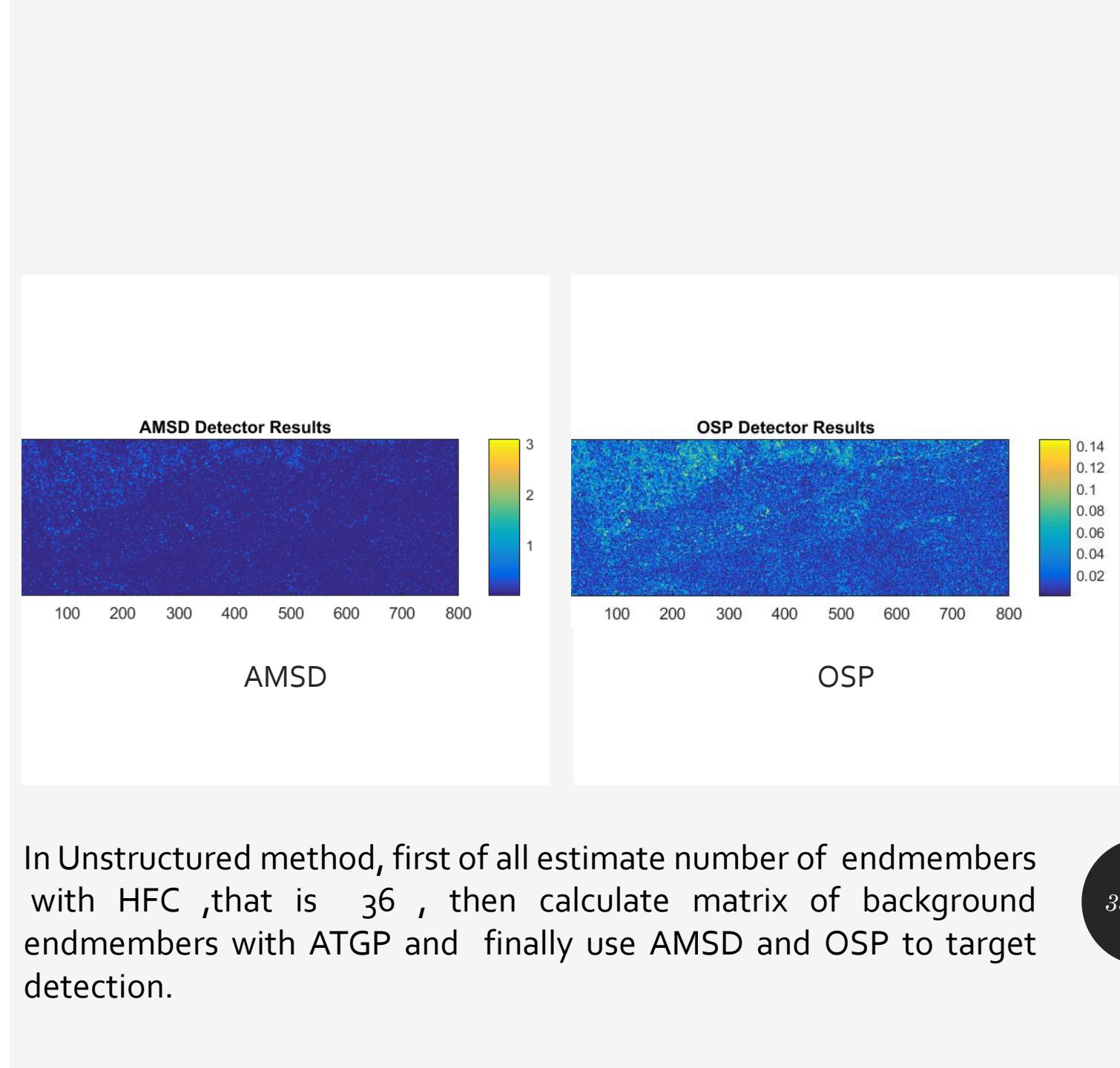
Vehicle 2

Result of Structured Method
Detector Map



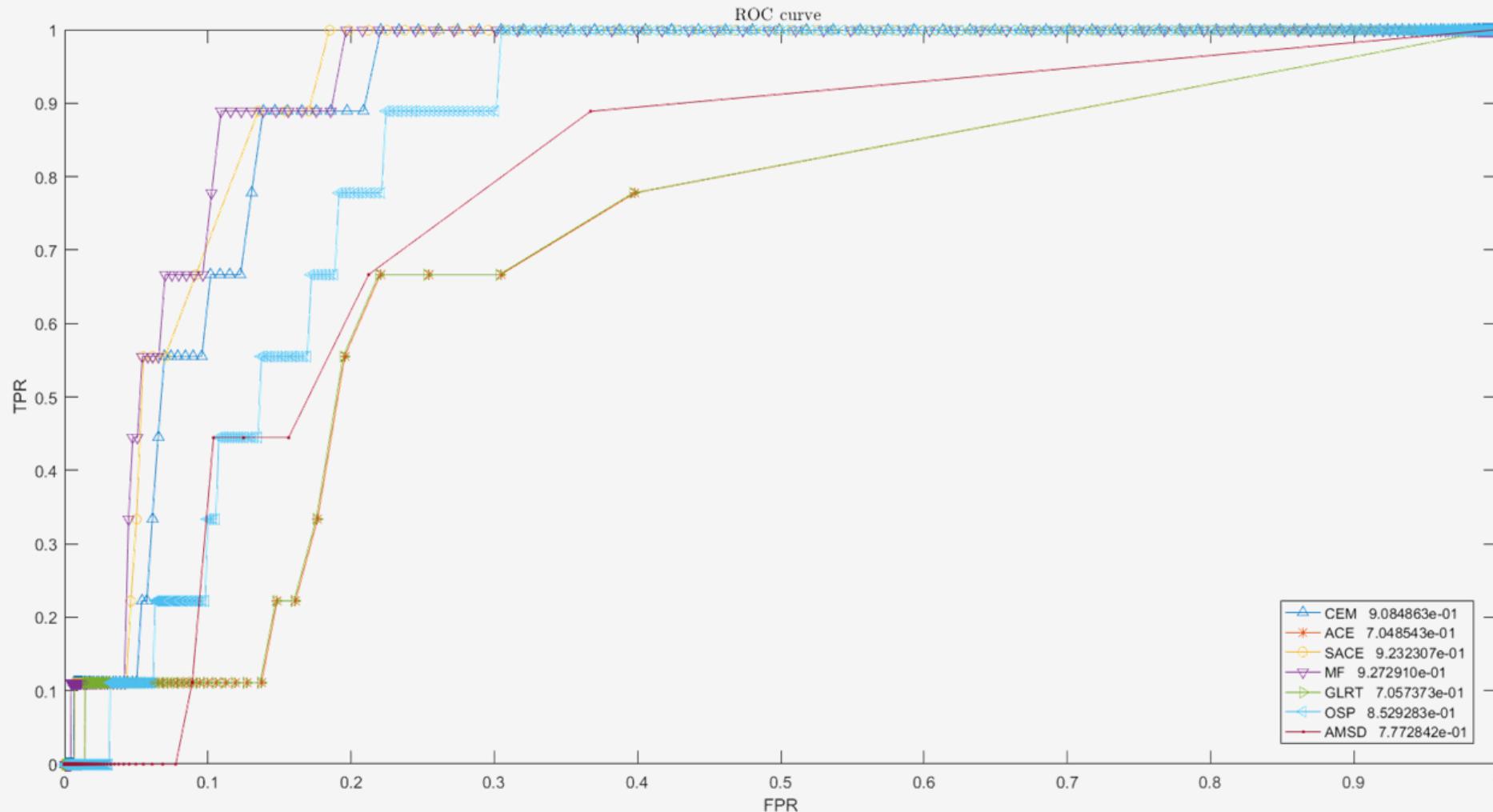
Vehicle 2

Result of Unstructured Method
Detector Map



In Unstructured method, first of all estimate number of endmembers with HFC ,that is 36 , then calculate matrix of background endmembers with ATGP and finally use AMSD and OSP to target detection.

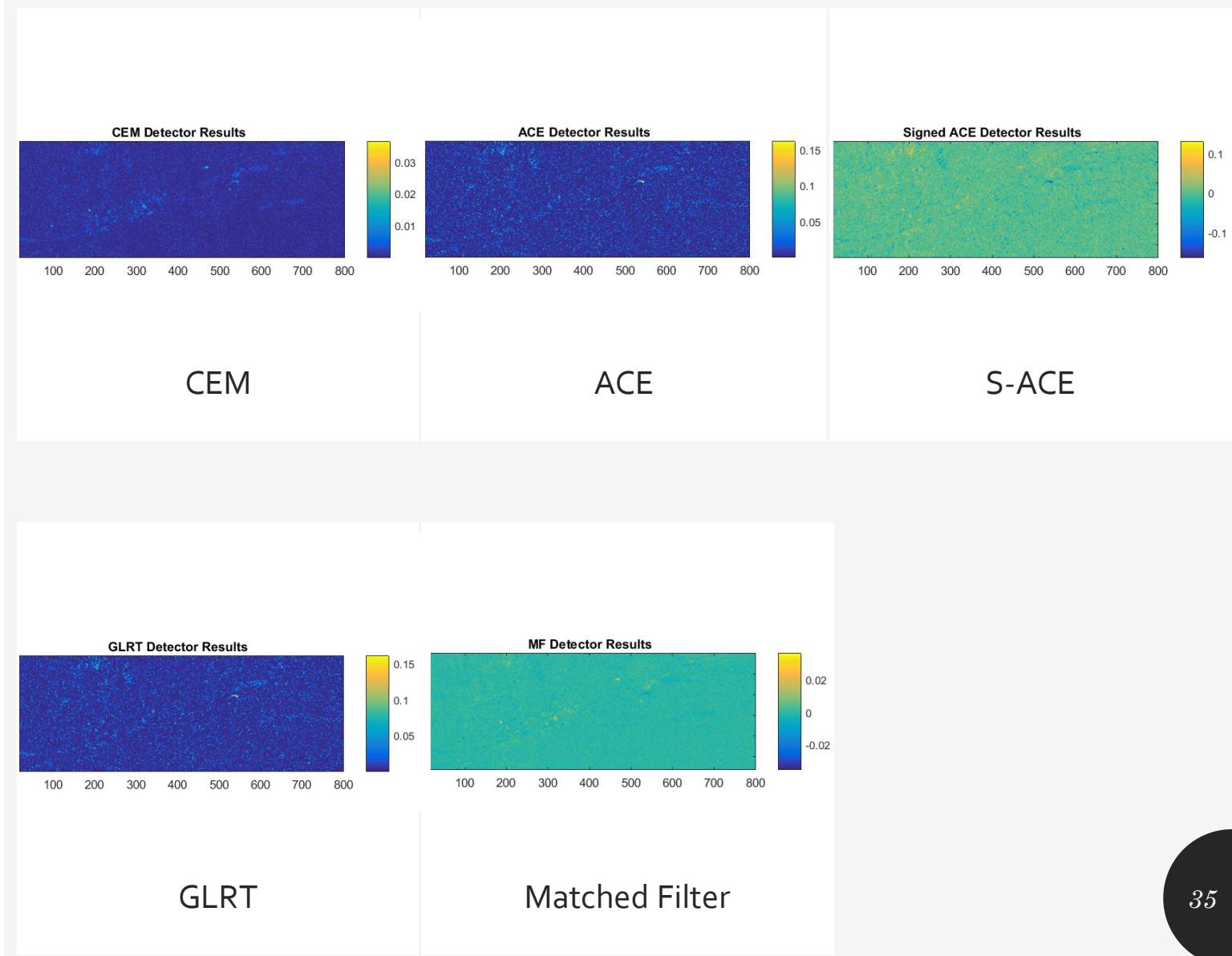
Vehicle 2



- ROC curve of Vehicle 2 data for 1000 different thresholds
- The AUC of each curve calculated

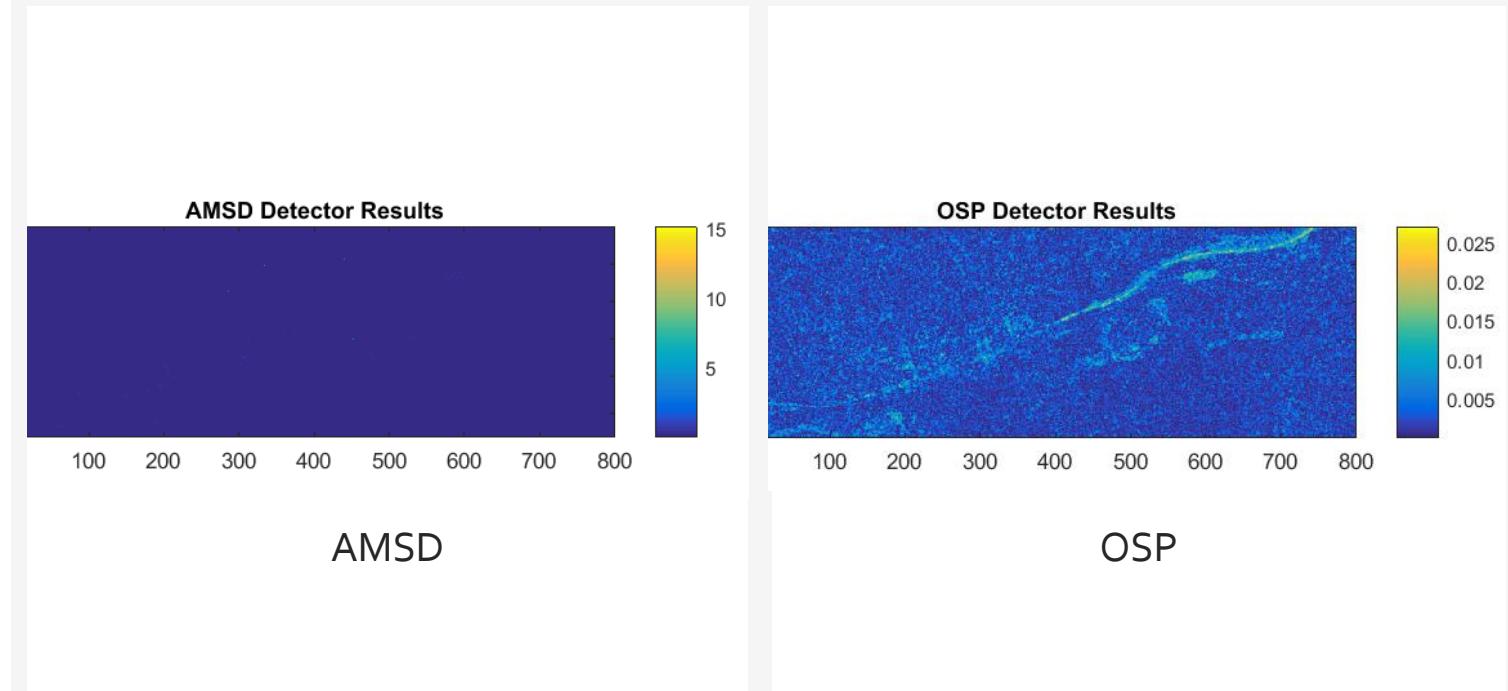
Vehicle 3

Result of Structured Method
Detector Map



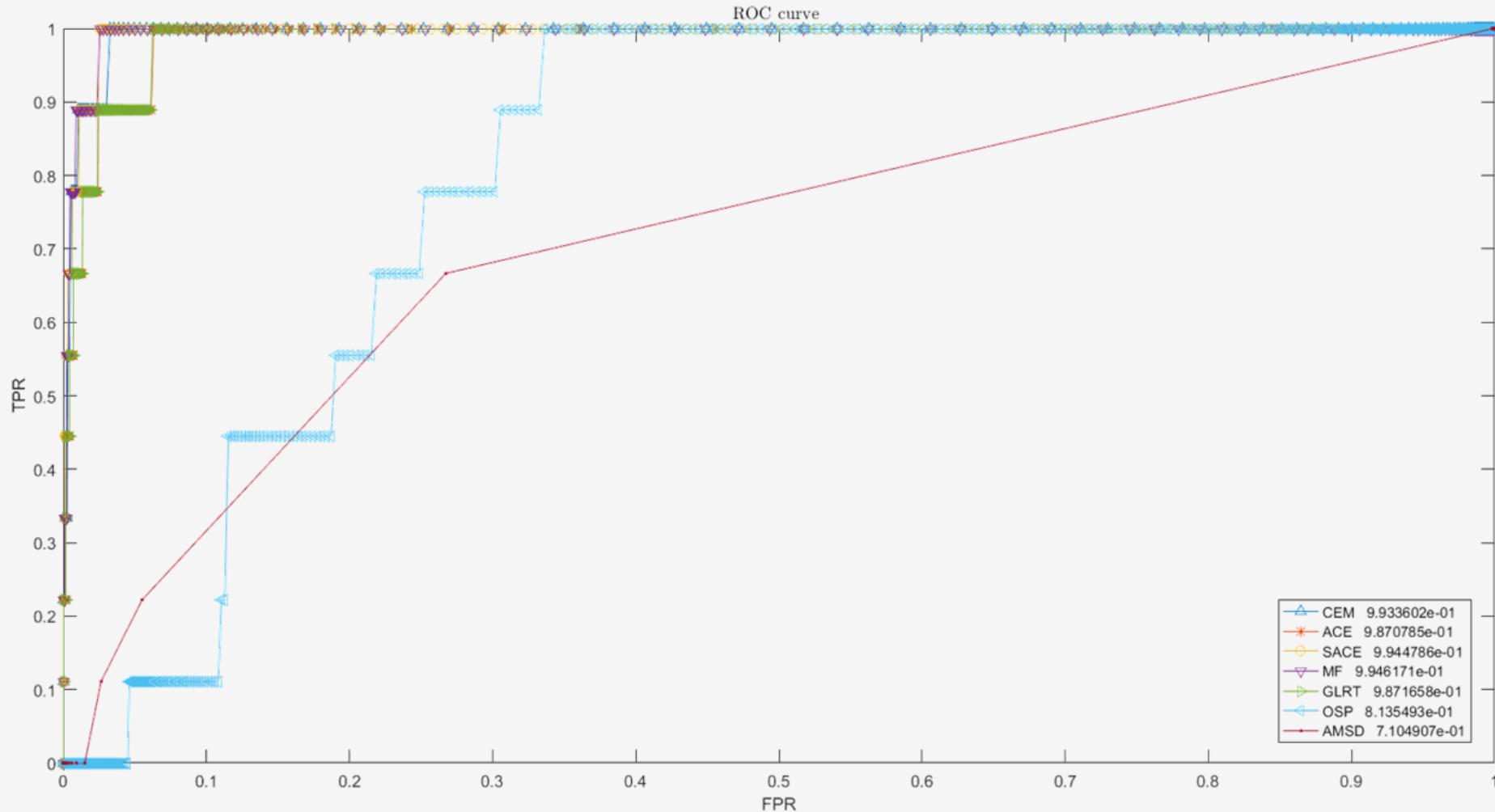
Vehicle 3

Result of Unstructured Method
Detector Map



In Unstructured method, first of all estimate number of endmembers with HFC ,that is 36 , then calculate matrix of background endmembers with ATGP and finally use AMSD and OSP to target detection.

Vehicle 3



- ROC curve of Vehicle 3 data for 1000 different thresholds
- The AUC of each curve calculated

RX

Anomaly Detection

- HYPERRX RX anomaly detector
- hyperRxDetector performs the RX anomaly detector

Usage

[result] = hyperRxDetector(M)

Inputs

M - 2D data matrix ($p \times N$) → 126×224000

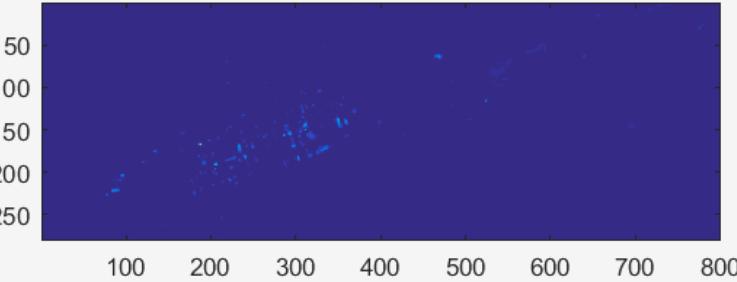
Outputs

result - Detector output ($1 \times N$) → 1×224000

sigma - Covariance matrix ($p \times p$) → 126×126

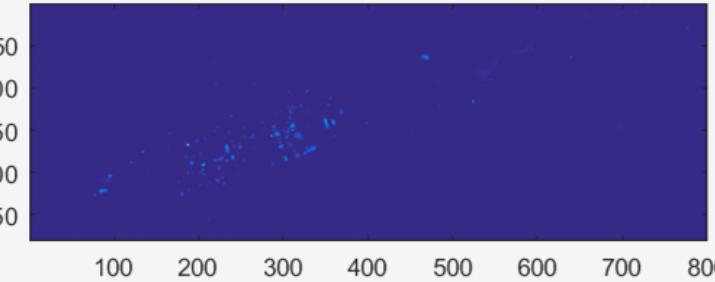
sigmalnv - Inverse of covariance matrix ($p \times p$) → 126×126

RX Detector Results



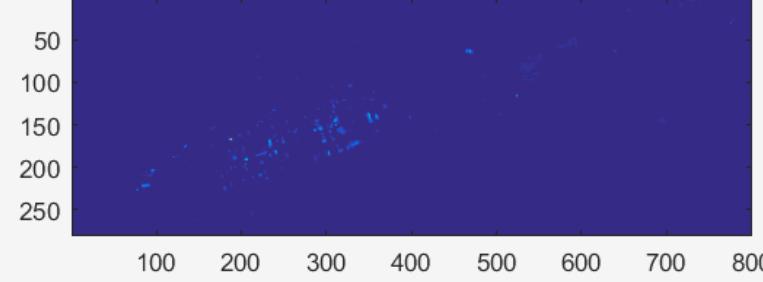
Fabric 1

RX Detector Results



Fabric 2

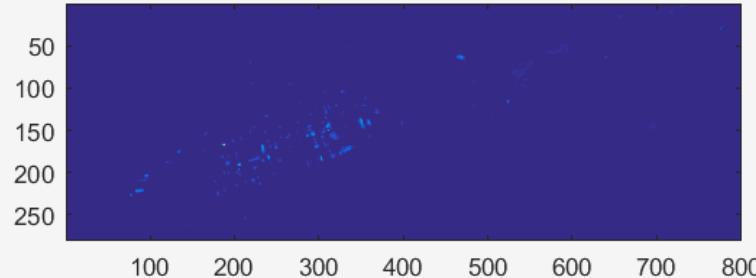
RX Detector Results



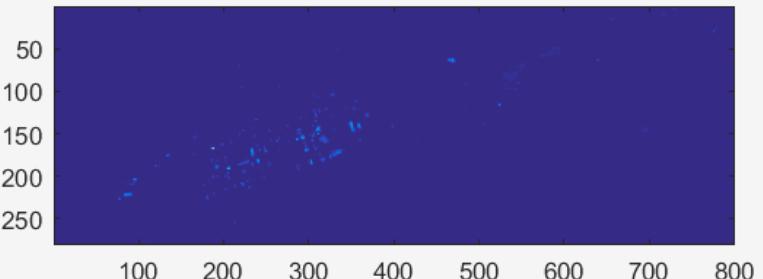
Fabric 3

Fabric 4

RX Detector Results

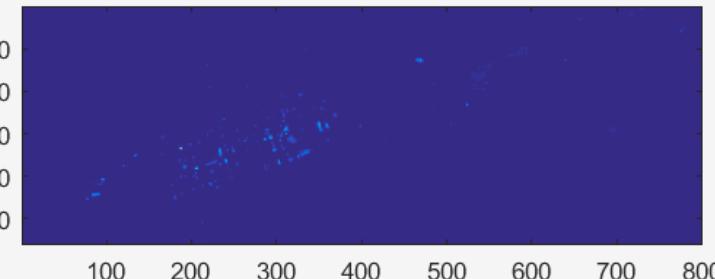


RX Detector Results



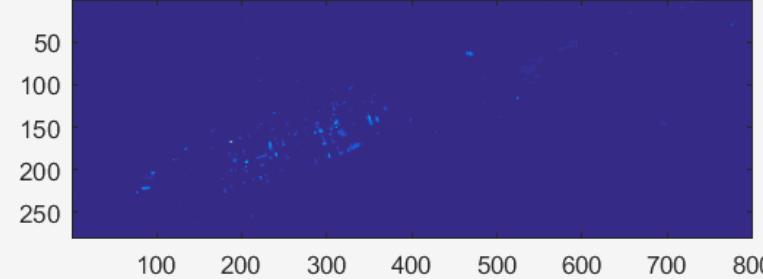
Vehicle 1

RX Detector Results



Vehicle 2

RX Detector Results



Vehicle 3

Thank You

 Faezeh Zamiri

 aghdam.zamiri.fa@ut.ac.ir

 [Linkedin.com/in/faezeh-Zamiri-1b51081bb](https://www.linkedin.com/in/faezeh-Zamiri-1b51081bb)

 09224015584

The End

References for codes and all of results:

<https://github.com/FaezehZamiri/TargetDetection>