CS 512 HW6 Report
SUDIPTA SWARNKAR

Estimate Optical Flow Vector Using Horn-Schunck Algorithm

Problem Statement 1:
Find optical flow vectors using Horn-Schunck method

Sol:
The optical flow equation can be calculated using the Taylor series expansion or by using the differential chain rule. In short we will get the equation according to the chain rule: If the image is represented by the brightness value E (x, y, t) at x, y coordinates and at time t, the brightness value does not change as the pattern moves. Optical flow rate in the x direction u=dx and in y direction v=dy. So we obtain the optical flow equation as $Ex * u + Ey * v + Et = 0.$
Here our goal is to minimize the value of E(v).

Calculate SobelX and SobelY for first image,

```
Dx = cv2.Sobel(prev, cv2.CV_32F, 1, 0, ksize=1)
Dy = cv2.Sobel(prev, cv2.CV_32F, 0, 1, ksize=1)
```

Calculate Optical Flow for second image using SobelX and SobelY values for first image,

Dt = curr – prev

```
for i in range(itr):
    uAvg = ndimage.convolve(flow[:, :, 0], kernel_1, mode='constant', cval=0.0)
    vAvg = ndimage.convolve(flow[:, :, 1], kernel_1, mode='constant', cval=0.0)
    Y = alpha * alpha + np.multiply(Dx, Dx) + np.multiply(Dy, Dy)
    dyv = np.multiply(Dy, vAvg)
    dxu = np.multiply(Dx, uAvg)
    flow[:, :, 0] = uAvg - (Dx * (dxu + dyv + Dt)) / Y
    flow[:, :, 1] = vAvg - (Dy * (dxu + dyv + Dt)) / Y
return flow
```

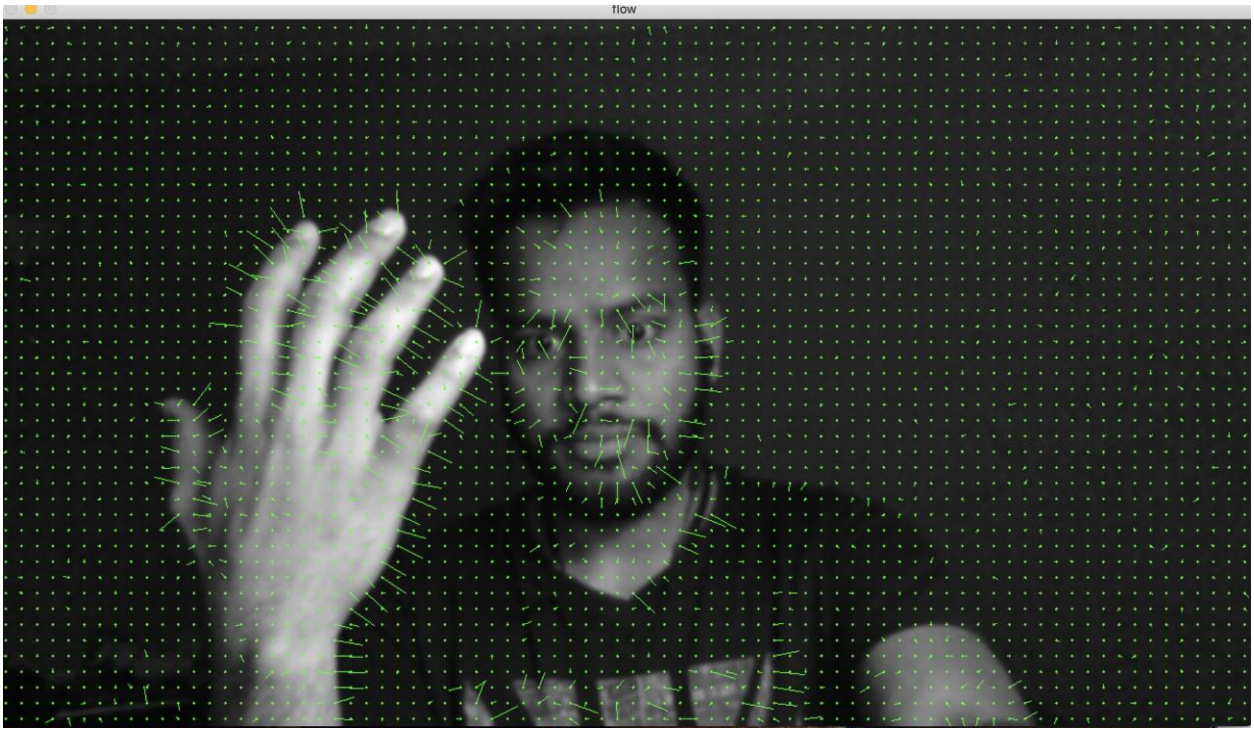
Problem Statement 2:
Display Computed optical flow vectors on second image

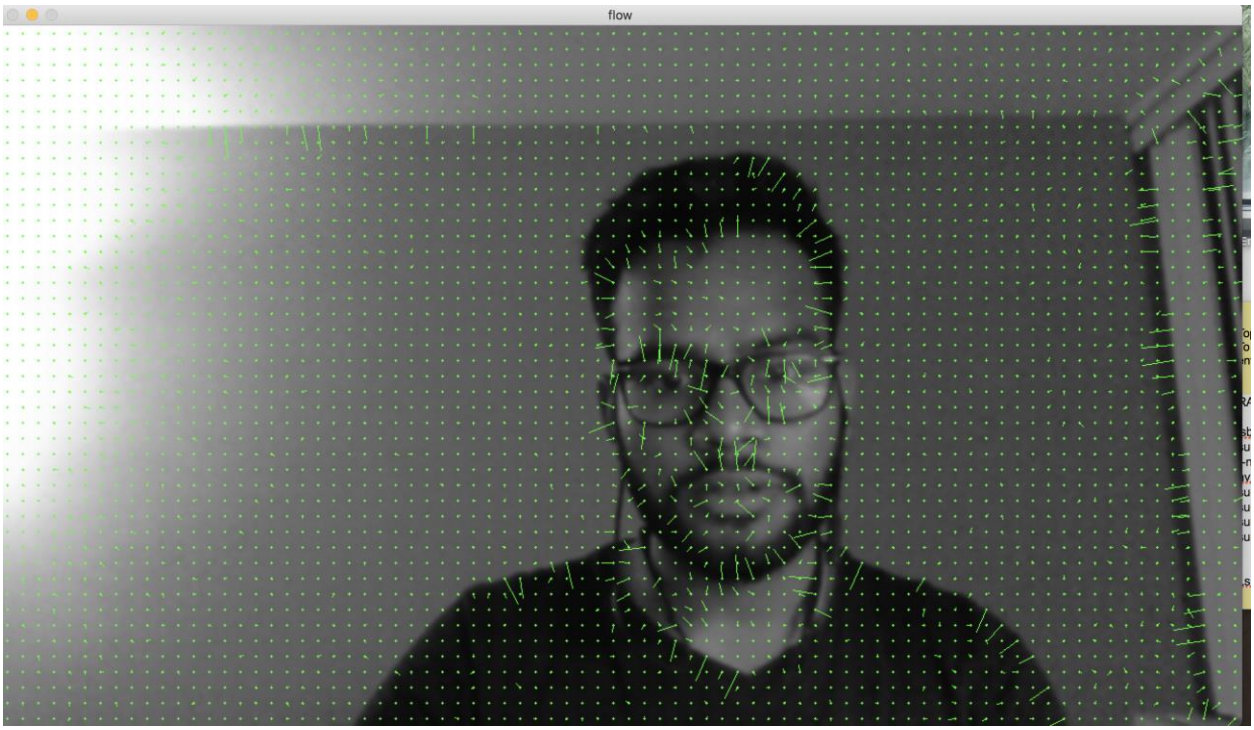
Sol:
I am displaying calculated optical vectors on second image using below computation,

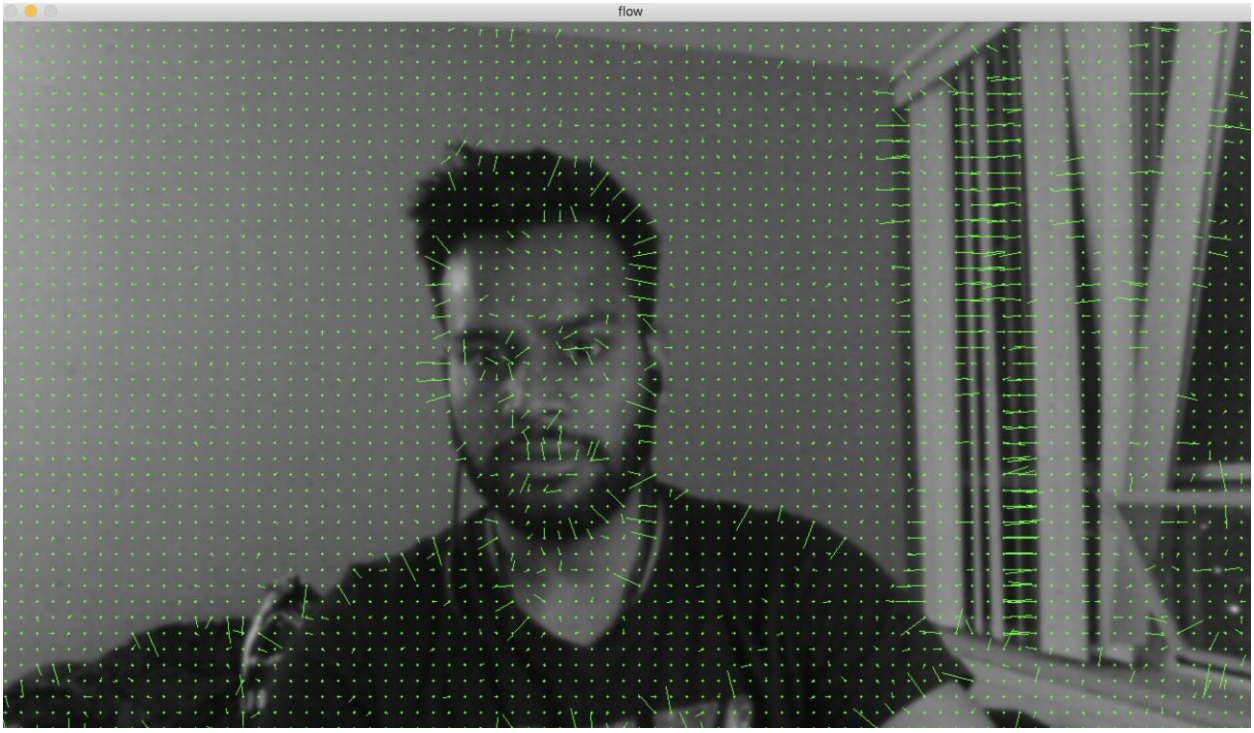
```
h, w = img.shape[:2]
y, x = np.mgrid[step / 2:h:step, step / 2:w:step].reshape(2, -1)
fx, fy = flow[y, x].T
lines = np.vstack([x, y, x + fx, y + fy]).T.reshape(-1, 2, 2)
lines = np.int32(lines + 0.5)
```

```
vis = cv2.cvtColor(img, cv2.COLOR_GRAY2BGR)
cv2.polylines(vis, lines, 0, (0, 255, 0))
for (x1, y1), (x2, y2) in lines:
    cv2.circle(vis, (x1, y1), 1, (0, 255, 0), -1)
return vis
```

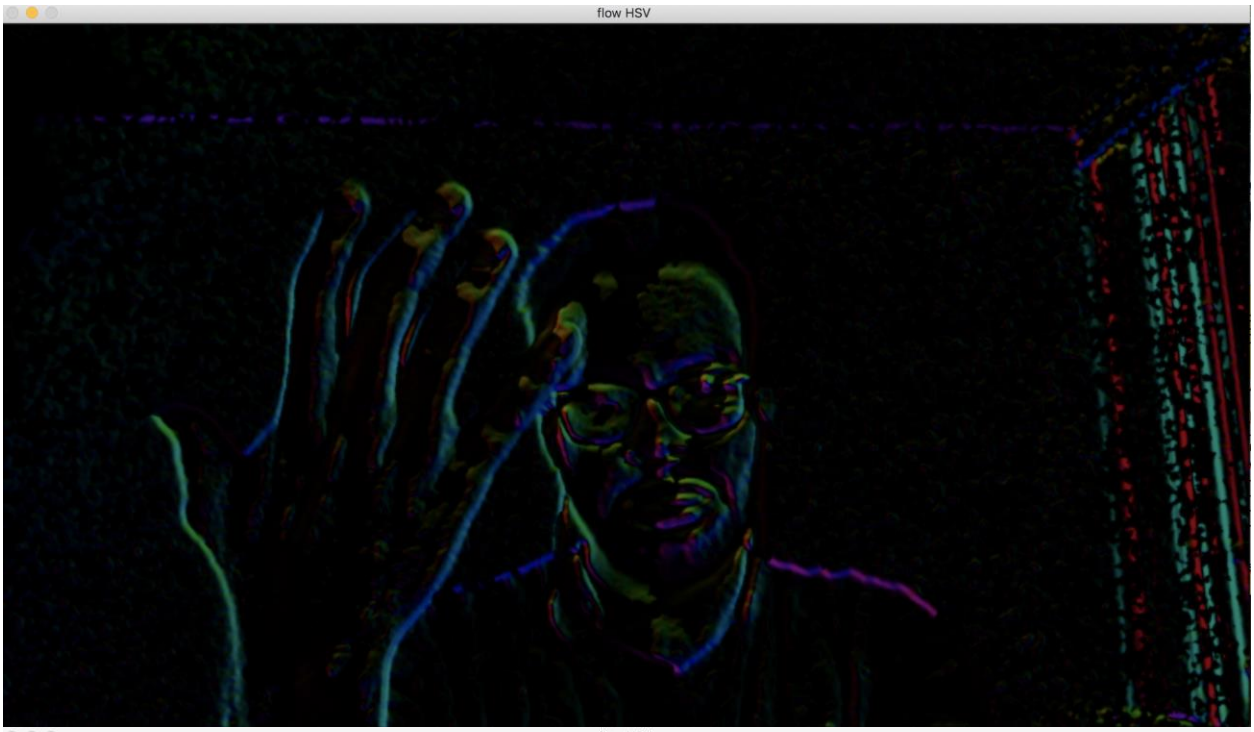
out0puts:

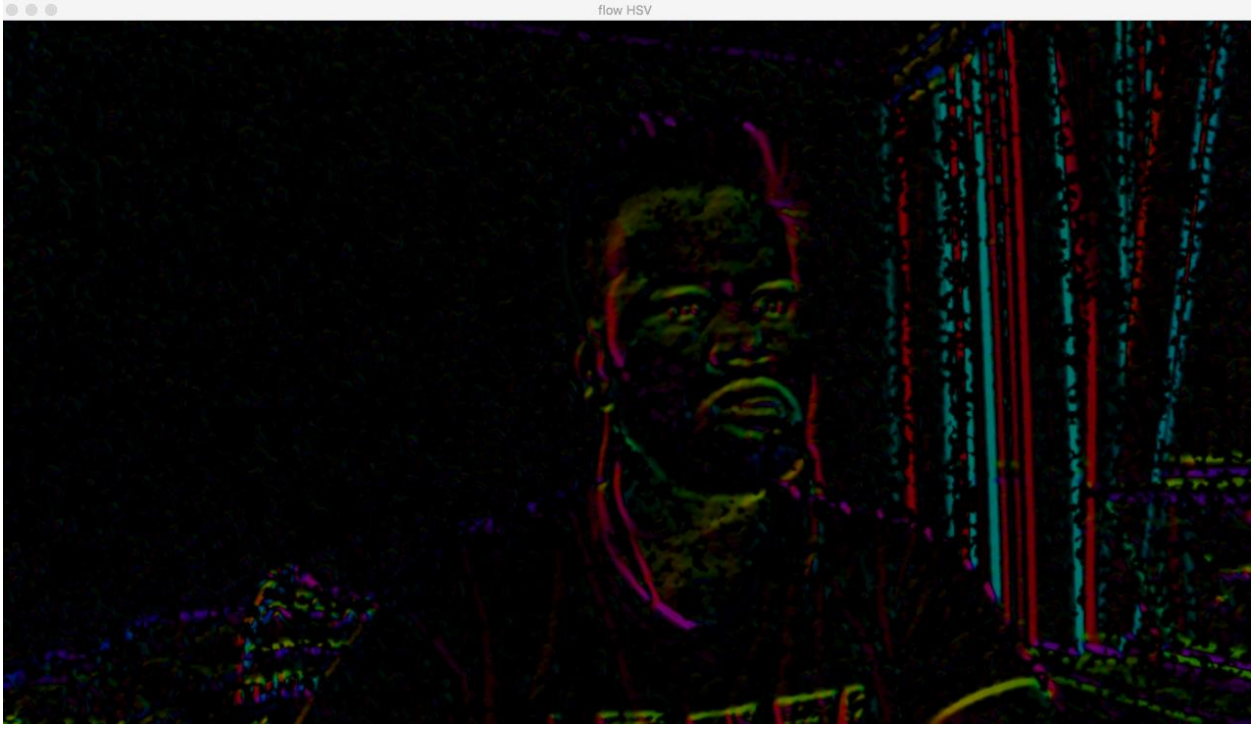


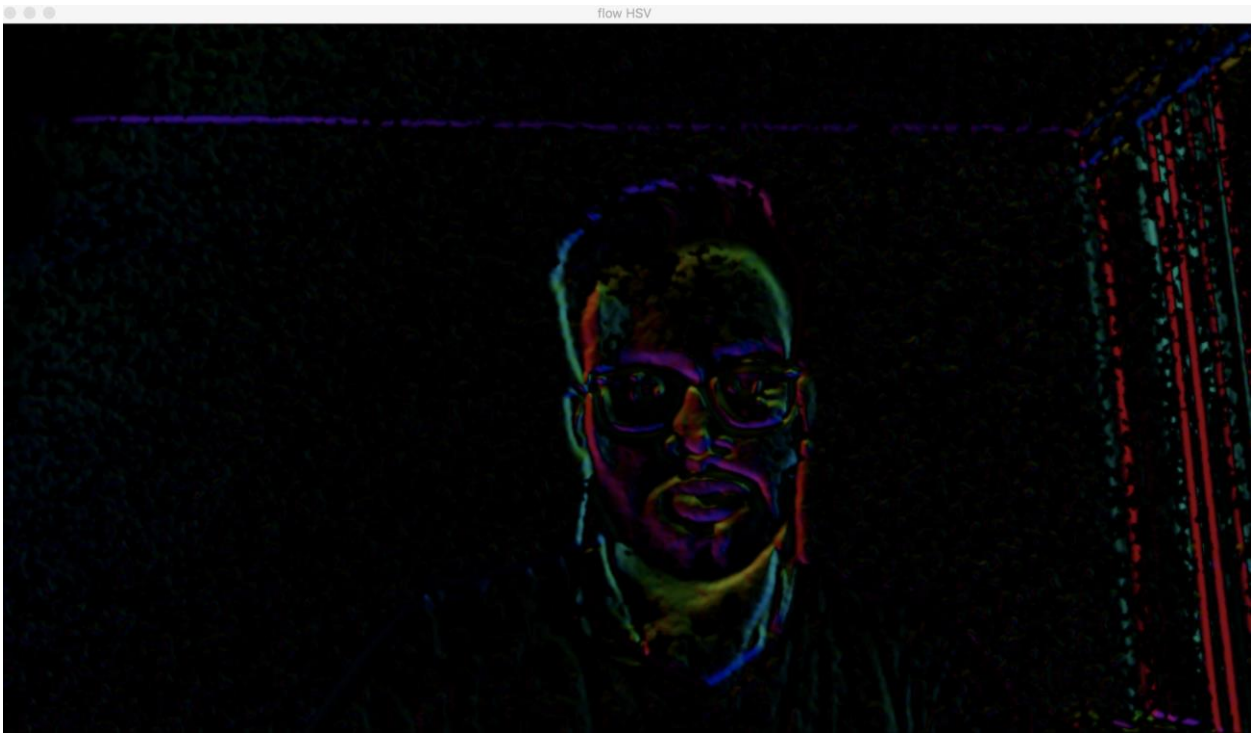




Draw in HSV:


flow HSV


flow HSV



Optical Flow Vectors:

```
[[[ -4.96901572e-02   3.06734303e-03]
  [ -1.90224200e-01  -4.26285341e-02]
  [ -7.20838904e-01  -1.51631907e-01]
  ...,
  [ -8.98584557e+00   3.50813651e+00]
  [ -1.44968805e+01   4.86215878e+00]
  [ -4.01232815e+00   3.88504219e+00]]

 [[ -9.11525637e-02   6.79815281e-03]
  [ -3.07220191e-01  -8.61748755e-02]
  [ -6.61985636e-01  -4.84748721e-01]
  ...,
  [ -5.37082624e+00   8.87943363e+00]
  [ -1.22314348e+01   1.26981640e+01]
  [ -4.27148533e+00   1.33277502e+01]]

 [[ -1.44073799e-01  -1.00271806e-01]
  [ -5.04925430e-01  -2.90254742e-01]
  [ -1.23163974e+00  -9.25036311e-01]
  ...,
  [ -1.10097659e+00   4.39938498e+00]
  [ -2.28316689e+00   6.53564930e+00]
  [ -2.28173280e+00   1.08712206e+01]]
```

```
 ...,
 [[  4.58463430e+00  -4.89333630e+00]
  [  1.14470873e+01  -9.59115505e+00]
  [  4.86986923e+00  -3.90872359e+00]
  ...,
  [  4.45142221e+00  -9.75689411e-01]
  [  1.92745876e+00  -1.07273459e-03]
  [  4.29151535e-01   7.99394906e-01]]

 [[  4.51266670e+00  -1.58883953e+00]
  [  1.17531128e+01  -3.30922842e+00]
  [  5.86915255e+00  -2.46384239e+00]
  ...,
  [  4.63470268e+00   1.45003963e+00]
  [  2.72518682e+00   1.33367360e+00]
  [  9.96586442e-01   3.29234153e-01]]

 [[  2.97119570e+00  -5.46549618e-01]
  [  8.96633530e+00  -9.23205554e-01]
  [  9.15011883e+00  -1.02890778e+00]
  ...,
  [  2.30910492e+00   8.64236474e-01]
  [  3.75607848e+00   4.90782559e-01]
  [  9.12280679e-01   2.07572132e-01]]]
```

Respective Spatio-Temporal Vector:

```
(array([[0, 0, 0, ..., 0, 0, 7],
       [0, 0, 0, ..., 0, 0, 7],
       [0, 0, 0, ..., 0, 0, 8],
       ...,
       [0, 1, 0, ..., 0, 0, 0],
       [1, 0, 0, ..., 2, 0, 0],
       [2, 0, 0, ..., 3, 0, 2]], dtype=uint8), array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 2, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [2, 2, 0, ..., 0, 0, 0],
       [1, 1, 0, ..., 0, 2, 2],
       [0, 0, 0, ..., 0, 0, 4]], dtype=uint8), array([[252, 252, 252, ...,  66,  66,  66],
       [252, 252, 252, ...,  66,  66,  66],
       [250, 250, 250, ...,  66,  66,  66],
       ...,
       [124, 124, 124, ..., 106, 107, 107],
       [123, 123, 123, ..., 107, 107, 108],
       [123, 123, 123, ..., 107, 108, 109]], dtype=uint8))
```

Optical Flow Vectors:

```
[[[ 0.         0.       ]
  [ 0.         0.       ]
  [ 0.         0.       ]
  ...,
  [ 1.37598097  0.52671123]
  [ 1.05019701  0.42316407]
  [ 0.29298598  0.22212455]]

 [[ 0.         0.       ]
  [ 0.         0.       ]
  [ 0.         0.       ]
  ...,
  [ 1.5002774   1.16660511]
  [ 1.27341557  1.5806241 ]
  [ 0.38650697  0.39896852]]

 [[ 0.         0.       ]
  [ 0.         0.       ]
  [ 0.         0.       ]
  ...,
  [ 1.92695379  1.94344282]
  [ 0.97203112  1.09826553]
  [ 0.38781738  0.48821715]]

 ...,
 [[ 0.19354999  2.1975379 ]
  [ 0.43020207  2.78213406]
  [ 0.63014114  2.20458841]
  ...,
  [ 1.36675048 -0.10566779]
  [ 0.89528346 -0.11243546]
  [ 0.31473327 -0.10677158]]

 [[ 0.06998201  1.4172833 ]
  [ 0.15218669  1.86204028]
  [ 0.25255132  1.76487327]
  ...,
  [ 1.28163147 -0.04988059]
  [ 0.79163206 -0.07713379]
  [ 0.27503654 -0.06535903]]
```

```
 [[ 0.0185141   0.43119812]
  [ 0.0421498   0.64193642]
  [ 0.06762357  0.62906098]
  ...,
  [ 0.93669379 -0.01880838]
  [ 0.55609369 -0.02393929]
  [ 0.17122537 -0.01844733]]]
```

Respective Spatio-Temporal Vector:

```
(array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 1, 0, 0],
       [0, 0, 0, ..., 1, 0, 0],
       [0, 0, 0, ..., 1, 1, 0]], dtype=uint8), array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=uint8), array([[216, 216, 216, ..., 148, 148, 148],
       [216, 216, 216, ..., 148, 148, 148],
       [216, 216, 215, ..., 148, 148, 147],
       ...,
       [114, 114, 114, ..., 100, 100, 100],
       [114, 114, 114, ...,  98,  97,  96],
       [114, 114, 114, ...,  96,  95,  95]], dtype=uint8))
```