```python
#!/usr/bin/env python
#### v3dedge a stand-alone python edge detector program for 3D

import sys
from numpy import *
from v4 import vx
import math

# Sobel separable matrix: h and h'
h = [1, 2, 1]
h_p = [1, 0, -1]

# Initialize all mask to be of 3x3x3 zero arrays
mask_x = [[[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]]]
mask_y = [[[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]]]
mask_z = [[[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]]]

# Create sobel kernel using loop
for i in range(3):
    for j in range(3):
        for k in range (3):
            mask_x[i][j][k] = h_p[i] * h[j] * h[k]
            mask_y[i][j][k] = h[i] * h_p[j] * h[k]
            mask_z[i][j][k] = h[i] * h[j] * h_p[k]
#print(mask_x)
#print(mask_y)
#print(mask_z)

# compute 3 x 3 x 3 edge detector
def v3dedge ( img ) :
 im = img.i
 tmimage = vx.Vx( img )
 tmimage.embedim((1,1,1,1,1,1))
 tm = tmimage.i

 magnitude = 0

 for z in range(im.shape[0]):
    for y in range(im.shape[1]):
        for x in range(im.shape[1]):
            sum_x = 0
            sum_y = 0
            sum_z = 0
            for zz in (0, 1, 2):
                for yy in (0, 1, 2):
                   for xx in (0, 1, 2):
                      sum_x += mask_x[zz][yy][xx] * tm[z + zz][y + yy][x+xx]
```

```
                        sum_y += mask_y[zz][yy][xx] * tm[z + zz][y + yy][x+xx]
                        sum_z += mask_z[zz][yy][xx] * tm[z + zz][y + yy][x+xx]
                sum_x = sum_x / 27
                sum_y = sum_y / 27
                sum_z = sum_z / 27
                magnitude = math.sqrt(sum_x*sum_x + sum_y*sum_y + sum_z*sum_z)
                # print(magnitude)
                if magnitude > 25:
                    im[z,y,x] = 200
                else:
                    im[z,y,x] = 0

## stand-alone wrapper
of=' '
vxif=' '
clist = vx.vxparse(sys.argv,  "if= of= -v  - ")
exec (clist )

if 'OPT' in locals():
   print ("v3dedge V4 3D python example program")
   print ("if= input file")
   print ("of= output file")
   print ("[-v] verbose mode")
   exit(0)
optv = 'OPTv' in locals()

ximage = vx.Vx( vxif )   ;#read image
v3dedge(ximage)          ;#process image
if optv:
   print (ximage.i)      ;# for very small images
ximage.write(of)         ;# Write the result file

```