In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import *
%matplotlib
```

```
Using matplotlib backend: MacOSX
```

In [3]:

```
x = np.array([0, 1, 2, 3, 4, 5, 6, 7])*2.5e-2+45e-2
W = np.array([1820, 1661, 1480, 1312, 1100, 917, 744, 602])*1e-6/2
V = np.array([1827, 1689, 1491, 1281, 1099, 929, 759.5, 590])*1e-6/2
l = 532e-9
w0 = 50e-6
x0 = 70e-2
w1 = w0*np.sqrt(1+((x-x0)/(np.pi*w0**2/l))**2)
plt.scatter(x, W)
plt.scatter(x, V)
plt.ylim(250e-6, 1000e-6)
plt.plot(x, w1)
plt.show()
```

In [4]:

```
def airy(x, w0, x0):
    return w0*np.sqrt(1+((x-x0)/(np.pi*w0**2/l))**2)
```

In [5]:

```
popt1, pcov1 = curve_fit(airy, x, W,  bounds=([30e-6, 65e-2], [70e-6, 80e-2]))
popt2, pcov2 = curve_fit(airy, x, V,  bounds=([30e-6, 65e-2], [70e-6, 80e-2]))

plt.scatter(x, W, marker=".", label='original data of W')
plt.scatter(x, V, marker=".", label='original data of V')

plt.plot(x, airy(x, *popt1), label='fit of W: w0 = %4.2f $\mu m$, z0 = %4.2f
cm' % tuple(popt1*(10**6, 10**2)))
plt.plot(x, airy(x, *popt2), label='fit of V: w0 = %4.2f $\mu m$, z0 = %4.2f
cm' % tuple(popt2*(10**6, 10**2)))

plt.xlabel('Distance relative to FI(m)')
plt.ylabel('Waist size(m)')
#plt.title('fit of mode cleaner transmission')
plt.ylim([250e-6, 1000e-6])
plt.xlim([0.45, 0.625])
plt.legend()

plt.grid()
plt.show()
# residual sum of squares
ss_res = np.sum((W - airy(x, *popt1)) ** 2)

# total sum of squares
ss_tot = np.sum((W - np.mean(airy(x, *popt1))) ** 2)

# r-squared
r2 = 1 - (ss_res / ss_tot)

print(r2)
wa = airy(0.4, *popt1)
```

```
0.9946990625616797
```

In [8]:

```
def myFunction(z):
    z0 = np.array(z[0])
    w0 = np.array(z[1])

    l = 532e-9 #wave length
    w1 = (popt1[0]+popt1[0])/2 #waist size after lens
    x1 = (popt1[1]+popt1[1])/2-0.4 #waist position relative to lens
    f = 0.1 #focal length

    F = np.empty((2))
    F[0] = w0*np.sqrt(1+(z0/(np.pi*w0**2/l))**2)-wa
    F[1] = z0*[1+((np.pi*w0**2/l)/z0)**2]-x1*(1+(np.pi*w1**2/l/x1)**2)+1/f

    return F
```

In [9]:

```
zGuess = np.array([-1,0.001])
```

In [11]:

```
z = fsolve(myFunction, zGuess)
```

In [12]:

```
print(z*((1,10**6)))
```

```
[ -3.16929507 877.15597017]
```