In [2]:
```
import sdm as sdmlib
import matplotlib.pyplot as plt
%matplotlib inline
```

In [3]:
```
def gen_sdm(scanner_type, bits=1000, sample=1000000):
    address_space = sdmlib.AddressSpace.init_from_b64_file('sdm-10000w.as')
    counter = sdmlib.Counter.load_file('sdm-10000w')
    sdm = sdmlib.SDM(address_space, counter, 451, scanner_type)
    return sdm
```

In [4]:
```
sdm = gen_sdm(sdmlib.SDM_SCANNER_OPENCL)
```

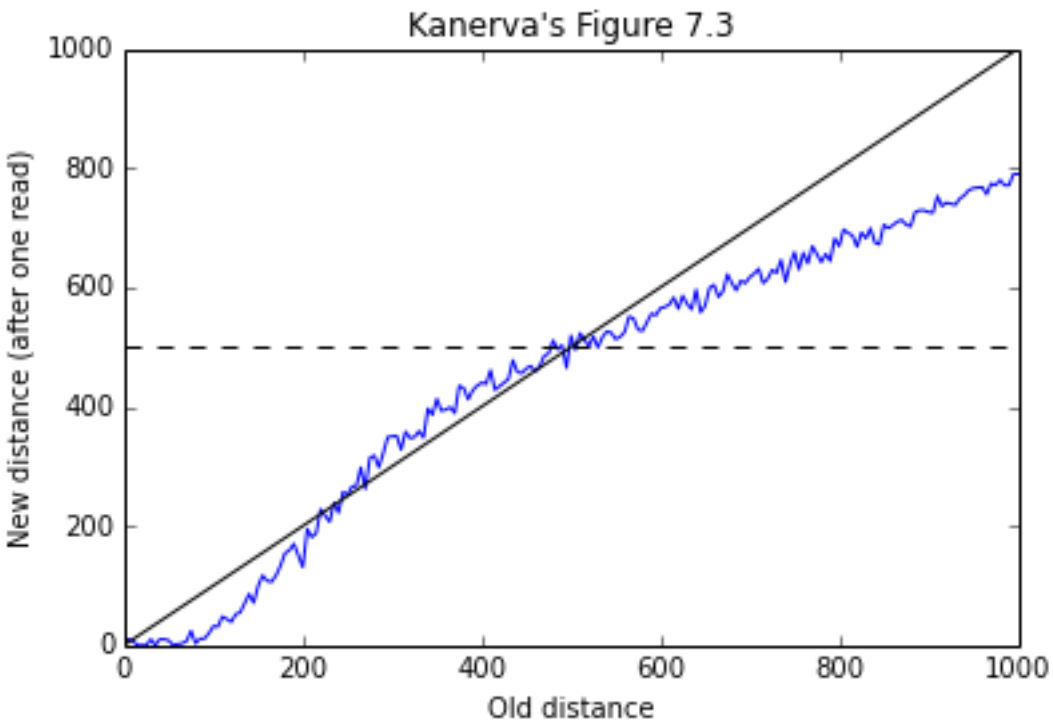
In [5]:
```
b = sdmlib.Bitstring.init_random(1000)
sdm.write(b, b)
```

In [6]:
```
from IPython.display import clear_output

distances = []
x = range(0, 1001, 5)
for i, dist in enumerate(x):
    clear_output(wait=True)
    print 'Distance: {:4d} ({:.2f}%)'.format(dist, 100.*(i+1)/len(x))
    c = sdmlib.Bitstring.init_from_bitstring(b)
    c.flip_random_bits(dist)
    assert c.distance_to(b) == dist
    d = sdm.read(c)
    distances.append(d.distance_to(b))
print 'Done!'
```

```
Distance: 1000 (100.00%)
Done!
```

In [29]:
```
plt.figure()
plt.plot(x, distances)
plt.plot(x, x, 'k')
plt.plot(x, [500]*len(x), 'k--')
plt.title('Kanerva\'s Figure 7.3')
plt.ylabel('New distance (after one read)')
plt.xlabel('Old distance')
plt.axis([0, 1000, 0, 1000]);
```


Kanerva's Figure 7.3
New distance (after one read)
Old distance
0
200
400
600
800
1000