Computational Photography Tutorial 1.1

Convolution in 1d

```
Convolution:
|   kernel    | * |    signal    |
[-1/2, 1, 1/2] * [0, 0, 1, 0, 0]

Location 0: [0, 0, 1] * [1/2, 1, -1/2] > [0*(1/2) + 0*(1) + 1*(-1/2)] = -1/2
Location 1: [0, 1, 0] * [1/2, 1, -1/2] > [0*(1/2) + 1*(1) + 0*(-1/2)] = 1
Location 2: [1, 0, 0] * [1/2, 1, -1/2] > [1*(1/2) + 0*(1) + 0*(-1/2)] = 1/2

output = [-1/2, 1, 1/2]
```

In [89]:
```
kernel = np.array([-0.5, 1, 0.5])
signal = np.array([0, 0, 1, 0, 0])

output = np.zeros(1 + signal.shape[0] - kernel.shape[0])
kernel_reversed = kernel[::-1]

for i in range(0, output.shape[0]):
    signal_patch = signal[i:i+len(kernel)]
    output[i] = (kernel_reversed * signal_patch).sum()

print output
```

```
[-0.5  1.   0.5]
```