THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

Convolution

The convolution of two functions f and g is defined as:

$$(f*g)(x,y)=\sum_{v=-\infty}^{\infty}\sum_{u=-\infty}^{\infty} f(u,v)g(x-u,y-v)$$

Where $f(x,y)$ is a function that represents the image and $g(x,y)$ is a function that represents the kernel.

In practice, the kernel is only defined over a finite set of points, so we can modify the definition to:

$$(f*g)(x,y)=\sum_{v=y-h}^{y+h}\sum_{u=x-w}^{x+w} f(u,v)g(x-u,y-v)$$

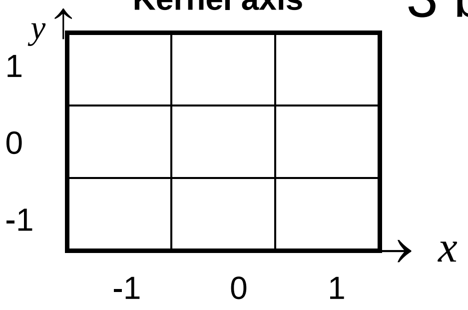
Where $2w+1$ is the width of the kernel and $2h+1$ is the height of the kernel.

g is defined only over the points $[-w,w]\times[-h,h]$

33

3 by 3 convolution

Kernel axis



$$\begin{pmatrix} \alpha_{-11} & \alpha_{01} & \alpha_{11} \\ \alpha_{-10} & \alpha_{00} & \alpha_{10} \\ \alpha_{-1-1} & \alpha_{0-1} & \alpha_{1-1} \end{pmatrix}$$

- Consider the above 3 by 3 kernel with weights α_{ij} .
- We can write the convolution Image I_old by the above kernel as:

$$I\_new(x, y) = \sum_{j=-1}^{1} \sum_{i=-1}^{1} \alpha_{ij} I\_old(x - i, y - j)$$

If all α_{ij} are positive we can normalise the kernel.

$$I\_new\_normalized(x, y) = \frac{1}{\sum_{j=-1}^{1} \sum_{i=-1}^{1} \alpha_{ij}} \sum_{j=-1}^{1} \sum_{i=-1}^{1} \alpha_{ij} I\_old(x - i, y - j)$$

Why normalizing is important ?

34

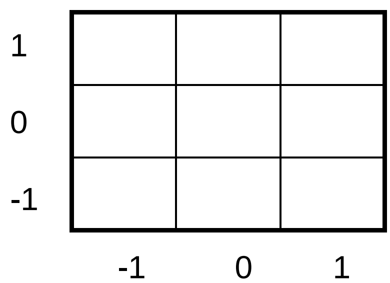
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

Convolution Pseudocode

Pseudocode for the convolution of an image f(x,y) with a kernel k(x,y) (2w+1 columns, 2h+1 lines) to produce a new image g(x,y):

```
for y = 0 to ImageHeight do
    for x = 0 to ImageWidth do
        sum = 0
        for i= -h to h do
            for j = -w to w do
                sum = sum + k(j,i) * f(x - j, y - i)
            end for
        end for
        g(x,y) = sum
    end for
end for
```

Kernel axis



Convolution equation for a 3 by 3 kernel

The pixel value p(x,y) of image f after convolution with a 3 by 3 kernel k is:

$$
\begin{aligned}
p(x,y) &= \sum_{i=-1}^{i=+1}\sum_{j=-1}^{j=+1} k(j,i)\, f(x-j, y-i) \\
&= k(-1,-1)\, f(x+1, y+1) + \\
&= k(0,-1)\, f(x, y+1) + \\
&= k(1,-1)\, f(x-1, y+1) + \\
&= k(-1,0)\, f(x+1, y) + \\
&= k(0,0)\, f(x, y) + \\
&= k(1,0)\, f(x-1, y) + \\
&= k(-1,1)\, f(x+1, y-1) + \\
&= k(0,1)\, f(x, y-1) + \\
&= k(1,1)\, f(x-1, y-1)
\end{aligned}
$$

Kernel k

y \ x	-1	0	1
1	$k(-1,1)$	$k(0,1)$	$k(1,1)$
0	$k(-1,0)$	$k(0,0)$	$k(1,0)$
-1	$k(-1,-1)$	$k(0,-1)$	$k(1,-1)$

36

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

Convolution examples

Do the convolution of the following kernel k with the image I:

Kernel

	-1	0	1
1	1	0	-1
0	0	1	0
-1	-1	0	1

Original image (x horizontal, y vertical)

15	20	25	25	15	10
20	15	50	30	20	15
20	50	55	60	30	20
20	15	65	30	15	30
15	20	30	20	25	30
20	25	15	20	10	15

P(x,y)= -1*30+
0*50+
1*15+
0*60+
1*55+
0*50+
1*30+
0*65+
-1*15
=55

k*I

		55			

37

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

Convolution – Potential Problems

Summation over a neighbourhood might exceed the range and/or sign permitted in the image format:

- The data may need to be temporarily stored in a 16 – 32 bit integer representation.
- Then normalised back to the appropriate range (0-255 for an 8 bit image).

Another issue is how to deal with image borders:

- Convolution is not possible if part of the kernel lies outside the image.
- What is the size of image window which is processed normally when performing a Convolution of size m x n on an original image of size M x N ?

38

THE UNIVERSITY OF AUCKLAND NEW ZEALAND

Convolution Border Issues

How to deal with convolution at image borders:

1) **Extend image limits with 0s (Zero padding)**
2) **Extend image limits with own image values**
3) **Generate specific filters to take care of the borders**

Find the corner and border specific kernel for:

1	1	1
1	1	1
1	1	1

Image top left corner filter:

Kernel center (in red)

1	1
1	1

-1	-1	-1
-1	8	-1
-1	-1	-1

Image left most column filter:

Kernel center (in red)

-1	-1
-1	5
-1	-1

-1	0	1
-1	0	1
-1	0	1

Top row filter:

Kernel center (in red)

-1	0	1

39