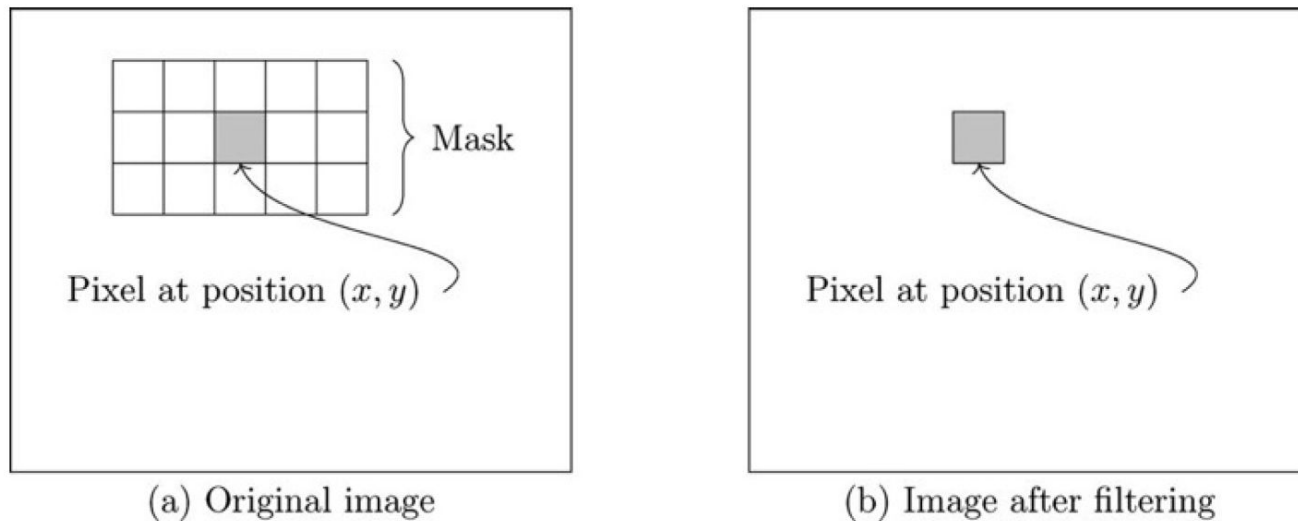


臨域處理

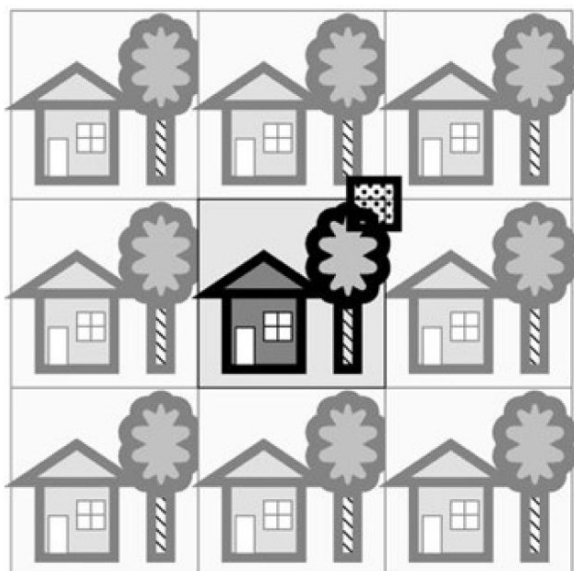
使用遮罩 (mask) 以及遮罩上的 function 所產生的 filter 來對影像進行轉換, 通常 mask 的 row 跟 column 都會是奇數個

Figure 5.1: Using a spatial mask on an image

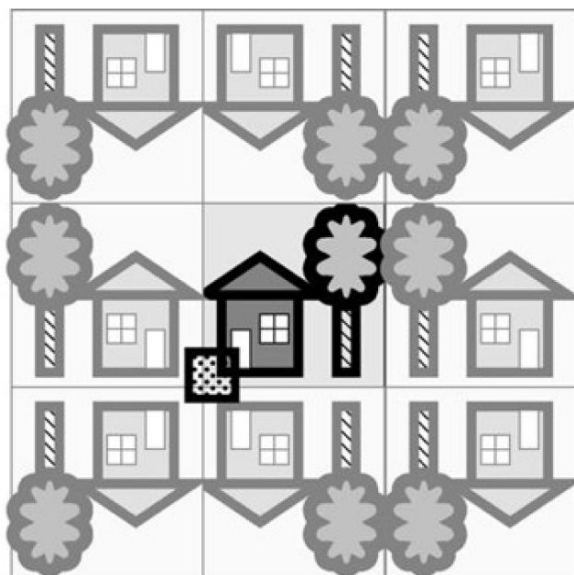


基本上就是把 mask 上的係數乘上 pixel 的灰階值再加起來, 然後把這個結果當作中間那格的新灰階值
那邊界該如何處理? 有下面四種方法

1. Ignore the edges: 遮罩只會計算他可以完全覆蓋的地方, 邊界會直接忽略, 這也代表著新的圖片會比舊的還小
2. Pad with zeros: 把那些遮罩無法計算的地方都補 0, 這樣的話新的圖片大小會跟原本的一樣, 但是會多出 unwanted black borders.
3. Repeat the image: 把圖片重複延展, 但這樣的話邊界會跟原本的不一樣
4. Reflect the image: 把圖片鏡射延展, 這樣是最適合的作法, 因為邊界都會是差不多的



Repetition



Reflection

實作

首先創造 4 個 standard average filter

```
f1 = fspecial("average") % 創造 3*3 filter
```

```
f1 = 3x3
    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111
```

```
f2 = fspecial("average", [3, 11]); % 創造 3 * 11 filter
```

```
f3 = fspecial("average", 11); % 創造 11 * 11 filter
```

```
f4 = fspecial("average", 25); % 創造 25 * 25 filter
```

```
img = imread("Image\cameraman.png");
```

```
p1 = imfilter(img, f1, "symmetric"); % reflect the image
```

```
p2 = imfilter(img, f2, "circular"); % repeat the image
```

```
p3 = imfilter(img, f3, "replicate"); % 缺的地方用邊界值來補
```

```
p4 = imfilter(img, f4, 0); % pad with zero
```

```
subplot(2, 2, 1)
```

```
imshow(p1)
```

```
subplot(2, 2, 2)
```

```
imshow(p2)
```

```
subplot(2, 2, 3)
```

```
imshow(p3)
```

```
subplot(2, 2, 4)
```

```
imshow(p4)
```



可以發現當 average filter 的 size 越大時, 新的影像會變得比較模糊, 這是因為新的單點會被原本的多個點平均掉, 導致新的點互相會比較接近。另外也可以注意到右下角用 pad with zero 方法的 image 產生了 black boarder

高低通濾波器

在一張圖片中, 顏色變化快速的地方為高頻, 顏色一致的地方為低頻。而高通濾波器 (High Pass Filter) 意思就是能讓高頻通過、保留高頻。低通濾波器 (Low Pass Filter) 則是能讓低頻通過、保留低頻。

通常在 high pass filter 裡, mask 的係數要接近 0, 這樣才可以把相近的顏色濾掉。而在 low pass filter 裡, mask 的係數要接近 1, 這樣才可以保留原本相近的顏色

```
% 創造一個 Laplacian filter (High Pass Filter)
f1 = fspecial("laplacian")
```

```
f1 = 3x3
    0.1667    0.6667    0.1667
    0.6667   -3.3333    0.6667
    0.1667    0.6667    0.1667
```

```
p1 = imfilter(img, f1, "symmetric");
```

```
% 創造一個 gaussian filter (Low Pass Filter)
f2 = fspecial("gaussian", [25, 25], 10)
```

```
f2 = 25x25
    0.0006    0.0007    0.0008    0.0008    0.0009    0.0010    0.0010    0.0011 ...
    0.0007    0.0008    0.0008    0.0009    0.0010    0.0011    0.0012    0.0012
    0.0008    0.0008    0.0009    0.0010    0.0011    0.0012    0.0013    0.0014
```

0.0008	0.0009	0.0010	0.0011	0.0012	0.0013	0.0014	0.0015
0.0009	0.0010	0.0011	0.0012	0.0013	0.0015	0.0016	0.0016
0.0010	0.0011	0.0012	0.0013	0.0015	0.0016	0.0017	0.0018
0.0010	0.0012	0.0013	0.0014	0.0016	0.0017	0.0018	0.0019
0.0011	0.0012	0.0014	0.0015	0.0016	0.0018	0.0019	0.0020
0.0011	0.0013	0.0014	0.0016	0.0017	0.0018	0.0020	0.0021
0.0012	0.0013	0.0015	0.0016	0.0018	0.0019	0.0020	0.0022
⋮							

```
p2 = imfilter(img, f2, "symmetric");

subplot(1, 2, 1)
imshow(p1)
subplot(1, 2, 2)
imshow(p2)
```



可以看到左邊的 high pass filter 保留了原本圖片的邊界, 也就是顏色變化快速的地方。而右邊的 low pass filter 則是讓圖片變模糊, 原本的邊界變得很難看出來。