作業二

1.作業簡介

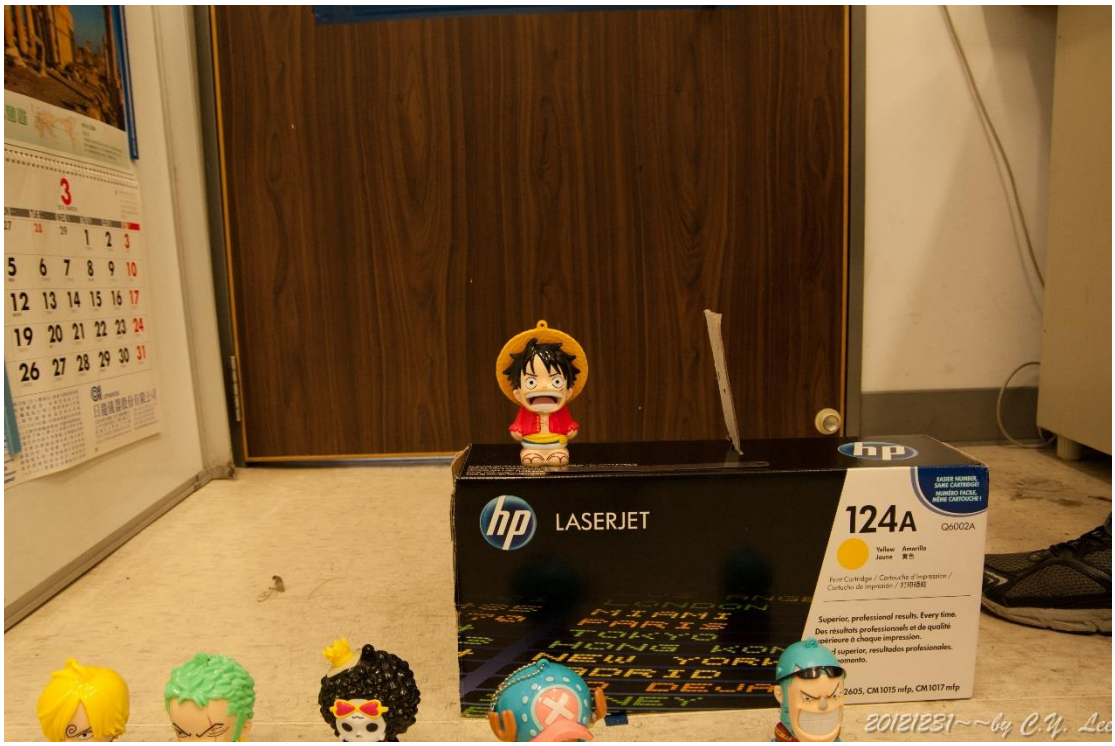
本作業照片有三種焦距 18mm，53mm，135mm，三種拍攝距離 600mm，1200mm，1800mm 共 9 種拍攝參數，每種參數有 5 張圖片分別位移距離為 0，1mm，5mm，10mm，20mm，共計 36 張圖片。

要做的目標有：

1. 相機的校正，即計算物體位移（#mm/像素）
2. 用角度去計算水平的視野（以角度為單位）。
3. 計算理論值（FOV）並與測量值進行比較。

實現方法：

1. 選取其中位移為 0 的圖像，設定參考點為路飛的第一個紐扣，如圖



相關參數

手動標定其坐標，從 18mm_600mm 到 135mm_1800mm 分別為：[2252, 1738], [2324, 1653], [2317, 1620], [2220, 1859], [2362, 1757], [2382, 1398], [2115, 1980], [2427, 1820], [2269,

1308]。

選取的塊大小 block_size，從 18mm 到 135mm 分別為[50, 150, 250]。

搜尋範圍 search range，焦距 18mm 時，1mm-20mm 位移距離下分別為[15, 50, 80, 120]，焦距 53mm 時，分別爲[30, 100, 150, 300], 焦距 135mm 時, 分別爲[100, 200, 300, 500]。

2. 使用作業 1 的方法計算運動向量（MV），其中爲了增加精確度，更改絕對值差匹配（SAD）為平方差匹配（SQD），演算法如下圖:

```
if 0 <= x1 < width - block_size and 0 <= y1 < height - block_size:
    block_a = trucka[y:y + block_size, x:x + block_size]
    block_b = truckb[y1:y1 + block_size, x1:x1 + block_size]
    # Sum of Absolute Differences (SAD)
    # sad = np.sum(np.abs(block_a - block_b))
    sad = np.sum(np.square(block_a-block_b))
    if sad < best_sad:
        best_sad = sad
        best_dx, best_dy = dx, dy
```

3. 根據 MV 計算 FOV

$$\text{FOV} = 2 \times \arctan\left(\frac{Image\ width(in\ pixel) \times \text{mm per pixel}}{2 \times object\ distance}\right) \times \frac{180}{\pi}$$

與理論值比較

$$\mathrm{FOV} = 2 \times \arctan\left(\frac{sensor\ width}{2 \times f}\right) \times \frac{180}{\pi}$$

算法如下：

```python
def FOV_theoretical():
    FOV_the = []
    for i in range(3):
        FOV_the.append(2 * np.arctan(sensor_width/(2*f[i]))*180/pi)
    return FOV_the
1 usage
def FOV_measured():
    FOV_value = []
    for j in range(3):
        for i in range(4):
            FOV_value.append(2 * np.arctan((image_size[0] * (float(mm_moved[i+1]) / motion_vectors[i+j*12][0])) / (2*object_distance[0])
            FOV_value.append(2 * np.arctan(
                (image_size[0] * (float(mm_moved[i + 1]) / motion_vectors[i+4+j*12][0])) / (2 * object_distance[1])) * 180 / pi)
            FOV_value.append(2 * np.arctan(
                (image_size[0] * (float(mm_moved[i + 1]) / motion_vectors[i + 8+j*12][0])) / (2 * object_distance[2])) * 180 / pi)
    return FOV_value
fov_theoretical = FOV_theoretical()
fov_measured = FOV_measured()
```

2.作業結果

生成圖表如下

相機焦距(mm)	物體距離(mm)	物體實際位移(mm)	物體位移(pixels)	mm/pixel	FOV估計值	FOV理論值
18	600	1	7	0.142857	58.16479	66.047735
18	600	5	26	0.192308	88.45156	66.047735
18	600	10	54	0.185185	46.785903	66.047735
18	600	20	102	0.196078	73.645714	66.047735
18	1200	1	2	0.5	62.627087	66.047735
18	1200	5	16	0.3125	71.582277	66.047735
18	1200	10	30	0.333333	71.582277	66.047735
18	1200	20	55	0.363636	65.958182	66.047735
18	1800	1	3	0.333333	68.669351	66.047735
18	1800	5	9	0.555556	74.716134	66.047735
18	1800	10	19	0.526316	70.587735	66.047735
18	1800	20	45	0.444444	59.951847	66.047735
53	600	1	22	0.045455	20.071421	24.897316
53	600	5	63	0.079365	22.031618	24.897316
53	600	10	129	0.077519	12.344885	24.897316
53	600	20	263	0.076046	34.341745	24.897316
53	1200	1	10	0.1	24.94729	24.897316
53	1200	5	44	0.113636	19.894225	24.897316
53	1200	10	78	0.128205	33.588551	24.897316
53	1200	20	161	0.124224	28.026392	24.897316
53	1800	1	12	0.083333	25.224916	24.897316
53	1800	5	37	0.135135	32.984985	24.897316
53	1800	10	58	0.172414	27.188794	24.897316
53	1800	20	99	0.20202	29.382013	24.897316
135	600	1	59	0.016949	7.550801	9.906515
135	600	5	160	0.03125	-6.553773	9.906515
135	600	10	264	0.037879	35.950734	9.906515
135	600	20	500	0.04	13.873784	9.906515
135	1200	1	-34	-0.029412	17.844002	9.906515
135	1200	5	62	0.080645	22.248635	9.906515
135	1200	10	58	0.172414	16.778422	9.906515
135	1200	20	-443	-0.045147	37.10682	9.906515
135	1800	1	4	0.25	16.778422	9.906515
135	1800	5	33	0.151515	17.703519	9.906515
135	1800	10	88	0.113636	-10.045142	9.906515
135	1800	20	208	0.096154	14.225923	9.906515

可以看到除了個別 FOV 估計值與實際值有較大差別，大部分估計值與實際值類似。

運動向量繪圖可以作證:

如圖可以看到運動向量準確指向了移動后的參考點（第一個紐扣）。

（所有運動向量繪圖詳見文件夾 Move Vector）

由於繪圖原因程序執行較慢，可以注解掉繪圖部分代碼。