

利用OpenCV建立視差圖像

努比 3D視覺初學者 今天

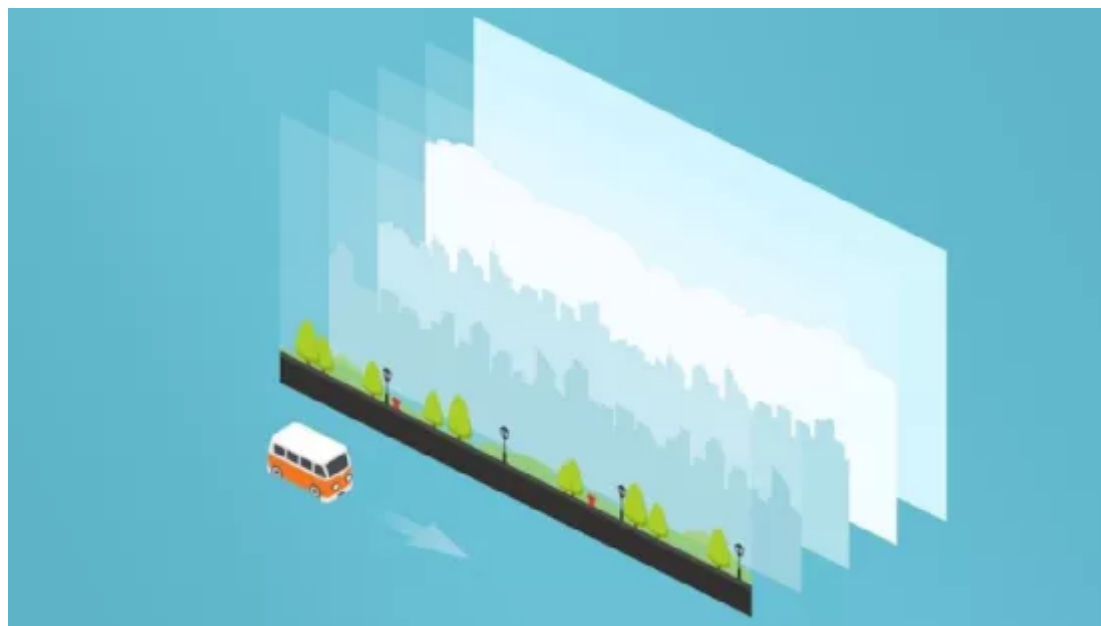
點擊上方

重磅乾貨，第一時間送達

本文轉自 | 小白學視覺

我們都看過3D電影，他們看起來都很酷，這給了我們一個想法，使用一些工具通過改變看圖像視角，模擬觀眾的頭部移動。

效果如何？我們都熟悉



視差

因此，如果我們能在2D圖像中獲得與不同圖像層的相同效果，那麼我們可以在這些圖像中產生類似的感覺，並產生我們想要的酷效果。

讓我們分解一下這個過程



深度圖

因此，首先，我們需要將圖像分解為不同的圖層，並且，我們需要一個2D圖像的深度圖。深度圖只是一個黑白圖像，其中圖像的白色顯示對象與鏡頭的接近度。獲得基本圖層後，我們需要從每個圖層中畫出缺失的部分。最後，我們將單個圖像分解成不同的圖層。現在，我們可以顯示不同的圖層，這些圖層看起來與原始圖像相同。現在，我們可以使用相機

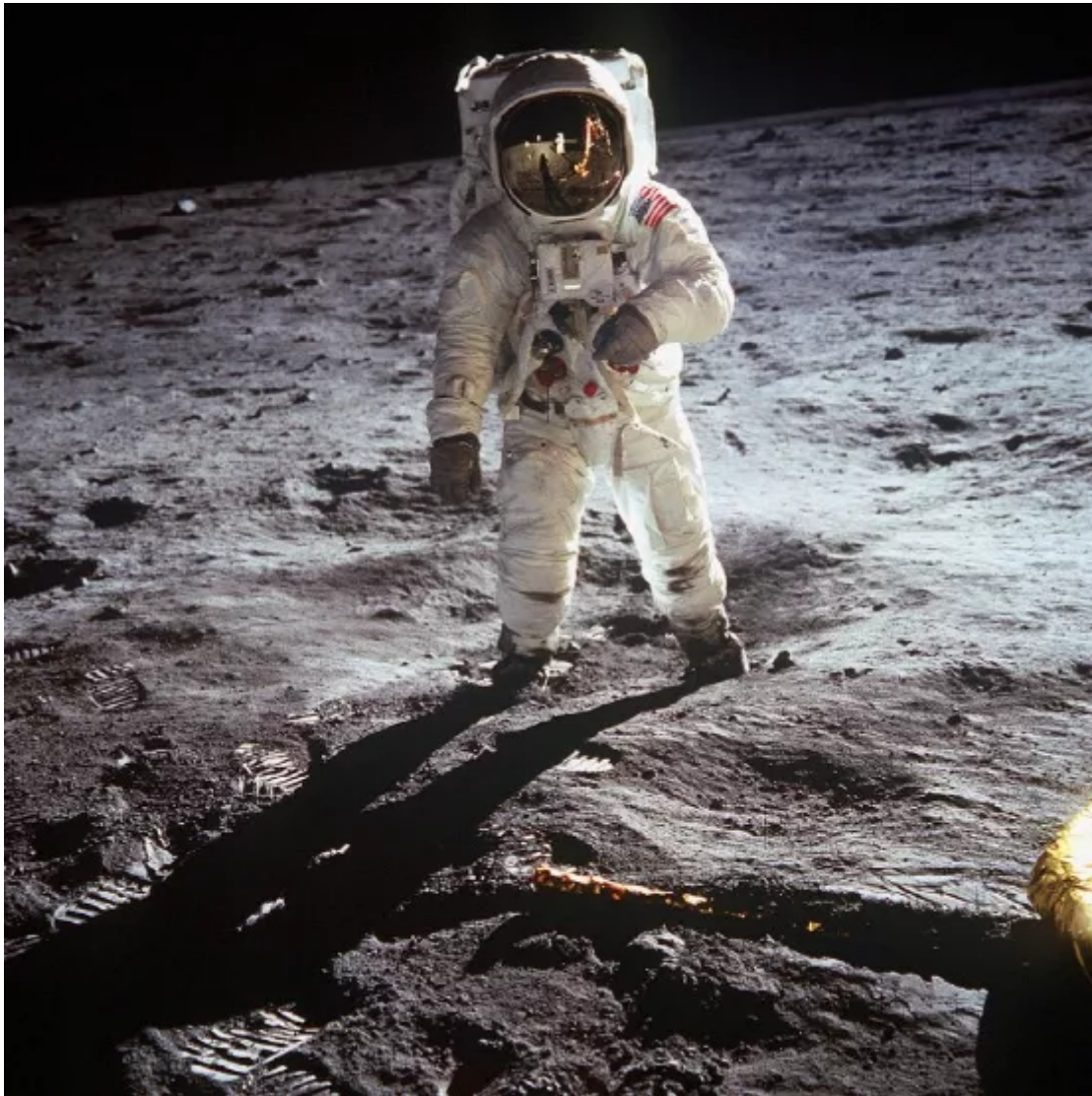
讓我們看看如何編寫此工具的代碼

因此，首先，我們需要導入一些文件，建議使用版本4.1.0.25 的OpenCV。

```
1 import os, sys
2 import numpy as np
3 import pygame as pg
4 import cv2
```

現在，我們需要加載圖像和深度 現在，我們將提供一個深度圖、代碼，但你可以生成自己的模型MiDaS。

```
1 img = cv2.imread('moon.jpg', flags=cv2.CV_8UC4)
2 depth_map = cv2.imread('moon_depth_map.png')
3 depth_map = cv2.cvtColor(depth_map, cv2.COLOR_RGB2GRAY)
4 img = cv2.resize(img, depth_map.shape[:2])
```





現在，在加載深度貼圖後，我們可以通過按不同閾值對深度貼圖不同圖層創建蒙版。在製作一個圖層時，我們需要兩個蒙版，一個是該圖層，另一個是上一層的第二個蒙版，用於畫上一個圖層的缺失部分。我們將在循環之外取最後一個圖層，以便提取此層中的所有剩餘部分。

```
1  
2 layers = []
```

```
3 prev_thres = 255
4 div=30
5
6 for thres in range(255 - div, 0, -div):
7     ret, mask = cv2.threshold(depth_map, thres, 255,          cv2.THRESH_BINARY)
8
9     ret, prev_mask = cv2.threshold(depth_map, prev_thres, 255, cv2.THRESH_BINARY)
10
11     prev_thres = thres
12     inpaint_img = cv2.inpaint(img, prev_mask, 10, cv2.INPAINT_NS)
13     layer = cv2.bitwise_and(inpaint_img, inpaint_img, mask = mask)
14     layers.append(conv_cv_alpha(layer, mask))
15
16 # adding Last Layer
17
18 mask = np.zeros(depth_map.shape, np.uint8)
19 mask[:, :] = 255
20
21 ret, prev_mask = cv2.threshold(depth_map, prev_thres, 255, cv2.THRESH_BINARY)
22
23 inpaint_img = cv2.inpaint(img, prev_mask, 10, cv2.INPAINT_NS)    layer = cv2.bitwise_and(inpaint_img, inpaint_img
24 layers.append(conv_cv_alpha(layer, mask))
25
26 layers = layers[::-1]
```

我們已經反轉了圖層，因此我們可以按最後一個層 當我們將圖層添加到列表中時，我們使用的是函數

```
1 def conv_cv_alpha(cv_image, mask):
2     b, g, r = cv2.split(cv_image)
3     rgba = [r, g, b, mask]
4     cv_image = cv2.merge(rgba,4)
5
6     return cv_image
```

現在來了人臉檢測和顯示圖像的部分。對於人臉檢測，我們將使用哈卡卡德。現在，我們將加載haar 級聯進行人臉檢測，並創建一個函數，該函數將從圖像中返回人臉。

```
1 face_cascade = cv2.CascadeClassifier( 'haarcascade_frontalface_default.xml')
2
3 def get_face_rect(img):
4     gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
5     face_rects = face_cascade.detectMultiScale(gray_img, 1.3, 5)
6     if len(face_rects) == 0:
7         return ()
8     return face_rects[0]
```

現在，我們必須顯示圖像，將根據用戶的頭移動。我們將使用 為了計算每一層的移位，我們將計算從框架中心的頭部移位，然後縮小頭移位以獲得一個小的移位值。之後，我們將每個圖層的索引值相乘，以獲得相應圖層的移位值，還可以乘以一些常量值，以獲得更好的結果。

我們將創建一個比原始圖像稍小的Pygame 窗口並加載相機。我們使用了

```
1 scale = 1
2 off_set = 20
```

```

3 width, height = layers[0].get_width(), layers[0].get_height()      win = pg.display.set_mode((int((width - off_s
4 pg.display.set_caption('Parallax_image')
5 scaled_layers = []
6 for layer in layers:
7     scaled_layers.append(pg.transform.scale(layer, (int(width*scale), int(height*scale))))
8 cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)

```

我們將設置一些常量。你可以玩這些常量來獲得不同的結果。

```

1 x_transform = True      # allow shift in x-axis
2 y_transform = False     # allow shift in y-axis
3 sens = 50              # the amount of scale down of shift value
4 show_cam = False       # show your face cam
5 shift_x = 0
6 shift_y = 0
7 run = True

```

最後，主循環渲染所有圖層。

```

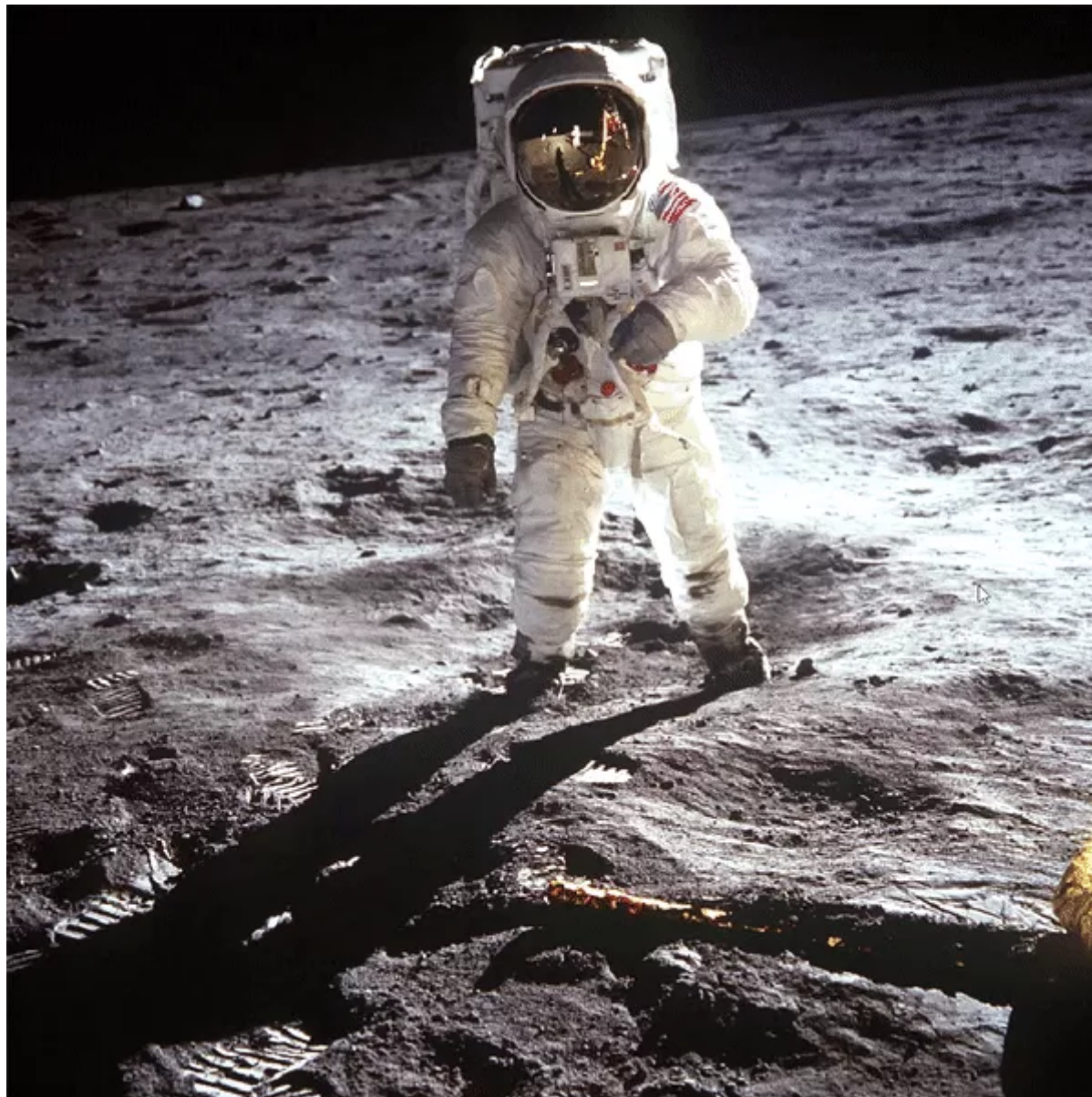
1 while run:
2     for event in pg.event.get():
3         if event.type==pg.QUIT:
4             run = False
5     ret, frame = cap.read()
6     frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
7     initial_pos = (frame.shape[0]/2, frame.shape[1]/2)

```



```
8     face_rect = get_face_rect(frame)
9     if len(face_rect) != 0:
10         x,y,w,h, = face_rect
11         face_rect_frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (255,255,0), 3)
12         shift_x = (initial_pos[0] - (x + w/2))/(sens*scale)
13         shift_y = (initial_pos[1] - (y + h/2))/(sens*scale)
14     win.fill((255, 255, 255))
15
16     for i, layer in enumerate(scaled_layers):
17         new_x = -off_set/2
18         new_y = -off_set/2
19         if x_transform:
20             new_x = 0 + shift_x*i
21         if y_transform:
22             new_y = 0 + shift_y*i
23         win.blit(layer, (new_x, new_y))
24
25     face_rect_frame = cv2.resize(face_rect_frame, (100, 100))
26     if show_cam:
27         win.blit(conv_cv_pygame(face_rect_frame), (0, 0))
28     pg.display.update()
29     cap.release()
30     cv2.destroyAllWindows()
31     pg.quit()
```

就是這裡，最終結果。



最終結果



不同圖像的演示

我已經創建了一個更高級版本的這個工具，你可以只選擇圖像，它會自動創建視差圖像，深度地圖將自動生成。

代碼鏈接：



微信搜一搜



3D视觉初学者

喜歡此內容的人還喜歡

灰燼般的粒子分散照片特效PS圖層樣式

GG素材閣



用Enscape做出“照片級”效果圖，4個技巧必須拿捏！（轉載）

Enscape設計



讓佈光不再單調，Blender燈源節點玩法解析

壹零社

