

Склейка изображений

Гауссовская пирамида

В данном задании были рассмотрены три различные комбинации параметров (1-ое число - параметр σ - среднеквадратичное отклонение, 2-ое число количество слоев):

(1, 15) (2, 10) (4, 5)

```
def gaussian_pyramid(sigma, n_layers, img):  
    out_imgs = []  
    sz = round(2 * sigma)  
    kr = 1 / (2 * pi * sigma ** 2) * e**(-(mgrid[-sz:sz+1,-sz:sz+1]**2).sum(axis=0) / (2 * sigma ** 2))  
    out_img = img  
    out_imgs.append(out_img[n_layers * sz : - n_layers * sz, n_layers * sz : - n_layers * sz])  
    for i in range(n_layers):  
        out_img = convolve2d(out_img, kr / kr.sum(), mode='valid')  
        out_imgs.append(out_img[(n_layers - i - 1) * sz : out_img.shape[0]-(n_layers - i - 1) * sz, (n_layers - i - 1) * sz : out_img.shape[1]-(n_layers - i - 1) * sz])  
    return out_imgs
```



(a) 0-слой

(b) 1-слой

(c) 2-слой

(d) 15-слой

Figure 1: Результат свертки с n - слоев при (1, 15)



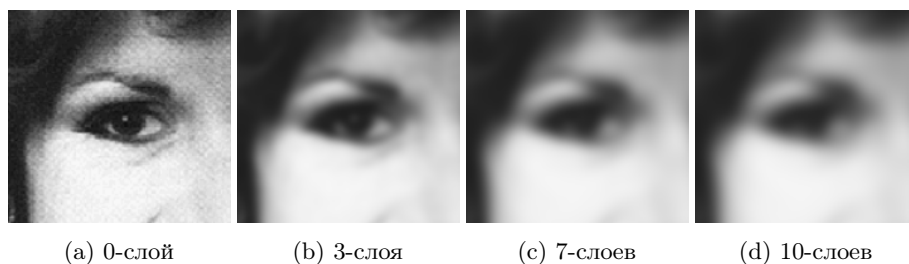
(a) 0-слой

(b) 1-слой

(c) 2-слой

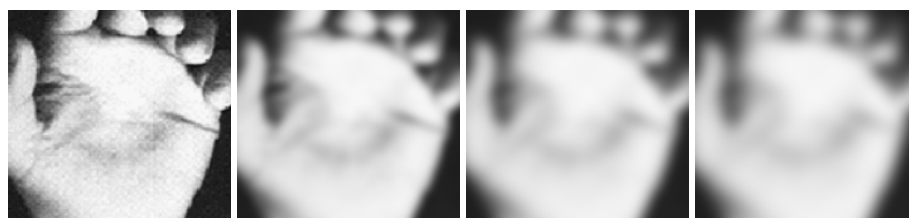
(d) 15-слой

Figure 2: Результат свертки с n - слоев при (1, 15)



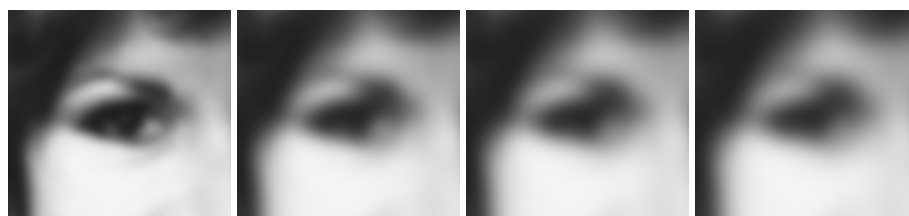
(a) 0-слой (b) 3-слой (c) 7-слоев (d) 10-слоев

Figure 3: Результат свертки с n - слоев при (2, 10)



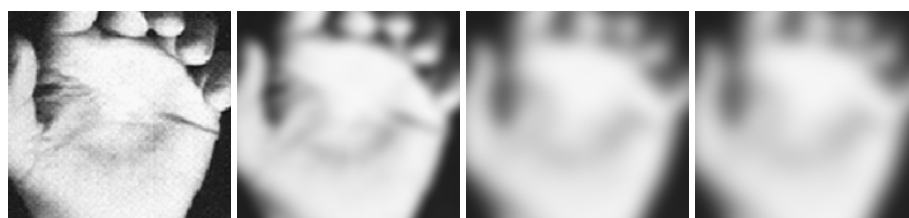
(a) 0-слой (b) 3-слой (c) 7-слоев (d) 10-слоев

Figure 4: Результат свертки с n - слоев при (2, 10)



(a) 1-слой (b) 3-слой (c) 4-слоев (d) 5-слоев

Figure 5: Результат свертки с n - слоев при (4, 5)



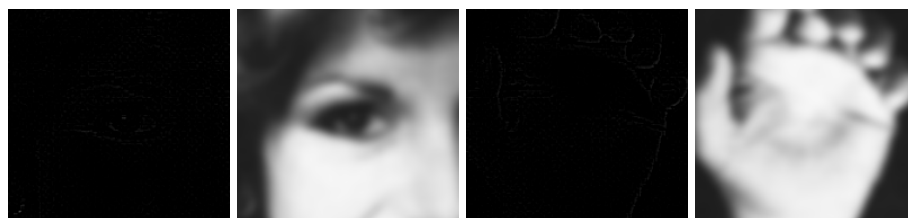
(a) 1-слой (b) 3-слой (c) 4-слоев (d) 5-слоев

Figure 6: Результат свертки с n - слоев при (4, 5)

Лапласовская пирамида

При тех же значениях параметров строим лапласовскую пирамиду. Лапласовская пирамида осуществляет разбиение на полосы частот. Соответственно - первые слои пирамиды отвечают высоким частотам, последний слой - содержит только низкие и совпадает с последним элементом гауссовской пирамидой. В первом случае перепады малы, и поэтому невооруженным взглядом отличить не просто от черного квадрата.

```
def laplasian_pyramid(sigma, n_layers, img):  
    gp = gaussian_pyramid(sigma, n_layers, img)  
    lp = [gp[i] - gp[i+1] for i in range(len(gp) - 1)]  
    lp.append(gp[n_layers-1])  
    return lp
```



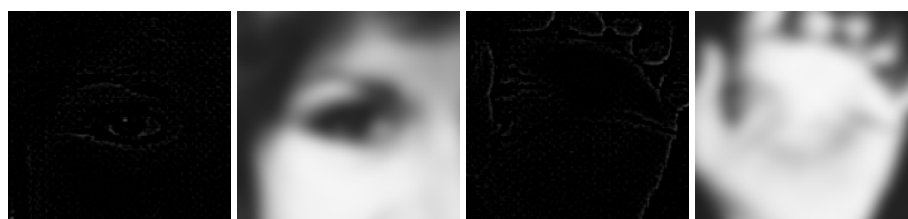
(a) 1-слой

(b) 15-слой

(c) 1-слой

(d) 15-слой

Figure 1: Лапласовская пирамида при (1, 15)



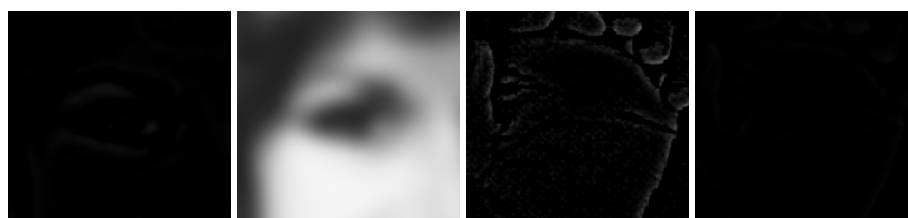
(a) 1-слой

(b) 10-слой

(c) 1-слой

(d) 10-слой

Figure 2: Результат свертки с n - слоев при (2, 10)



(a) 2-слой

(b) 5-слой

(c) 1-слой

(d) 3-слой

Figure 3: Результат свертки с n - слоев при (4, 5)

Склейка

Проведем склейку при приведенных выше значениях параметров и еще несколько коллажей, полученных аналогичным способом. В данном случае размер изображения уменьшен по сравнению с исходными, что можно было в принципе преодолеть, используя зеркалирование границы.

```
def merge(lpa, lpb, mask, sigma, n_layers):
    gsm = gaussian_pyramid(sigma, n_layers, mask)
    s = zeros(gsm[0].shape)
    steps = []
    for i in range(n_layers, -1, -1):
        s = s + lpa[i] * gsm[i]
        s = s + lpb[i] * (ones(gsm[0].shape) - gsm[i])
        steps.append(s)
    return steps
```



(a) (1, 15)



(b) (2, 10)



(c) (4, 5)

Figure 1: Склейка при различных значениях параметров



(a)

(b)

(c)

Figure 2: Склейка ночного неба и фрагмента обложки альбома



(a)

(b)

(c)

Figure 3: Склейка ночного неба и дневного неба



(a)



(b)



(c)

Figure 4: Склейка груши и Трампа