

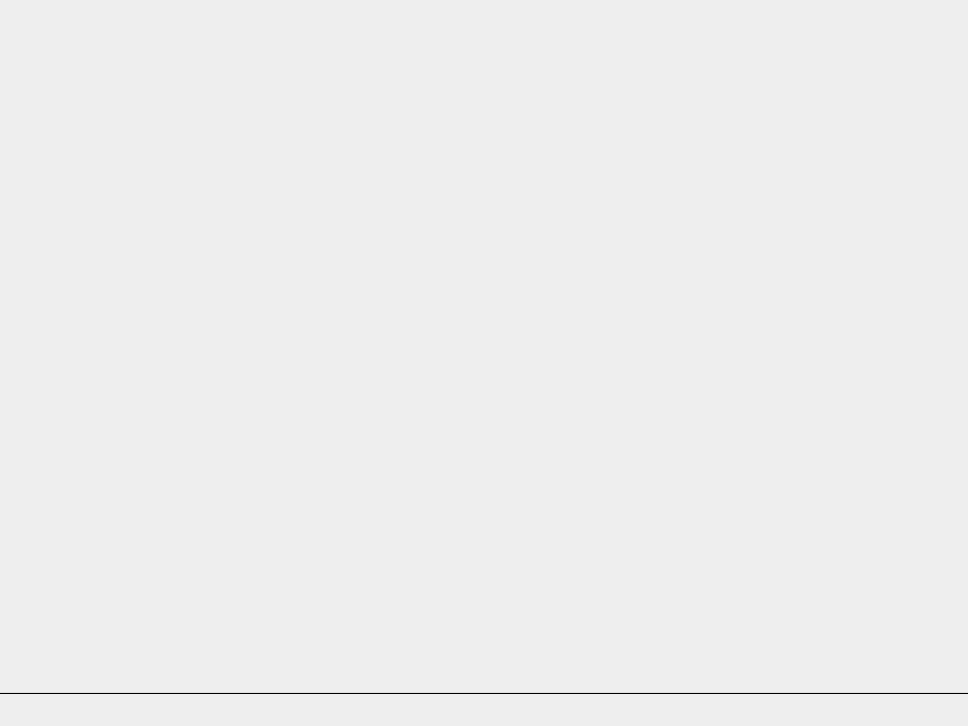
Фотографии Прокудина-Горского

Влад Шахуро



1. Загружаем скан фотопленки из файла
 2. Преобразуем значения изображения в вещественные
 3. Делим его на три равные части по высоте, обрезаем края (например, по 5%)
 4. Сдвигаем синий канал, накладываем на зеленый, считаем похожесть
 5. Сдвигаем красный канал, накладываем на зеленый, считаем похожесть
 6. Совмещаем три канала с наилучшими сдвигами
-







```
b = numpy.roll(b, row_shift, axis=0)  
b = numpy.roll(b, col_shift, axis=1)
```

Похожесть изображений b и g :

```
correlation = (b * g).sum()
```

Чем больше корреляция, тем более похожи два изображения

Как оценить качество сопоставления?

1. Визуально
2. Автоматически по разметке



Прототип функции сопоставления:

```
def align(img, (row_g, col_g)):  
    # считаем сдвиги каналов  
  
    # сдвигаем точку на зеленом канале  
    # на другие каналы  
  
    return (row_b, col_b), (row_r, col_r)
```

Прототип функции сопоставления:

```
def align(img, (row_g, col_g)):  
    # считаем сдвиги каналов  
  
    # сдвигаем точку на зеленом канале  
    # на другие каналы  
  
    return (row_b, col_b), (row_r, col_r)
```

Оценка качества сопоставления:

$$\text{diff} = |\text{row}_b - \text{true_row}_b| + |\text{col}_b - \text{true_col}_b| + \\ + |\text{row}_r - \text{true_row}_r| + |\text{col}_r - \text{true_col}_r|$$
