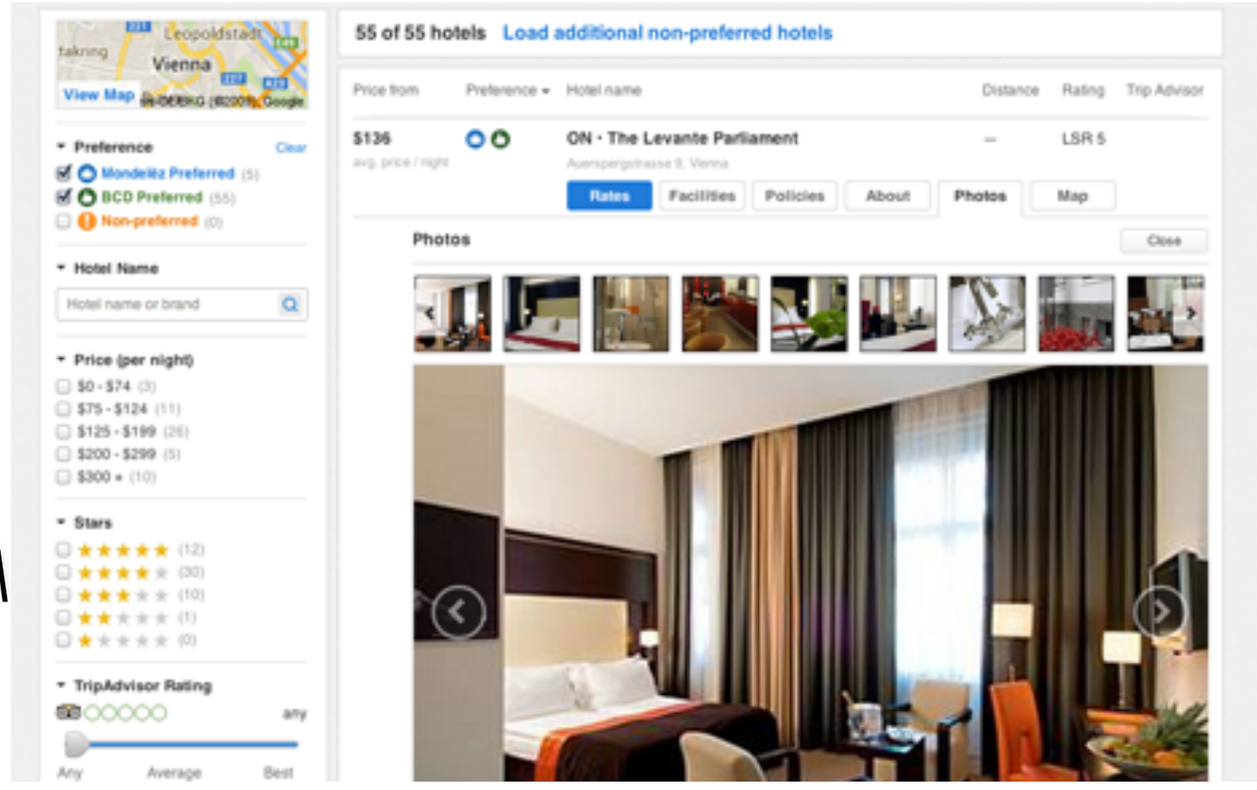
HOW TO DETECT SIMILAR IMAGES WITH PYTHON

Viktor Drachov

ABOUT ME

- GETGOING.COM
  - B2B, CORPORATE TRAVEL
  - КРУПНЕЙШИЙ ЗАКАЗЧИК - BCD TRAVEL
  - 4 ПРОВАЙДЕРА
  - >100К ОТЕЛЕЙ В КАЖДОМ
  - ~7М ИЗОБРАЖЕНИЙ
  - НЕ ХОТИМ ПОКАЗЫВАТЬ ОДНО И ТО ЖЕ



DISCLAIMER

Everybody lies

- Расскажу
  - Общие подходы
  - Как это делали мы
  - Как это не стоит делать
- Не расскажу
  - Распознавание объектов
  - Поиск “похожих” картинок
  - Silver Bullet

Подходы к сравнению изображений

- Каждый с каждым
  - Долго, неудобно, $O(N^2)$
- Хэширование. Perceptual hashing.
  - Не обязательно сравнивать N с N
  - Занимает меньше места
  - Можно искать (KD-Tree, MVP-Tree, etc). O(N) в **худшем** случае.
    - Или даже в PostgreSQL
  - По сути мы строим features для классификатора

OPEN SOURCE

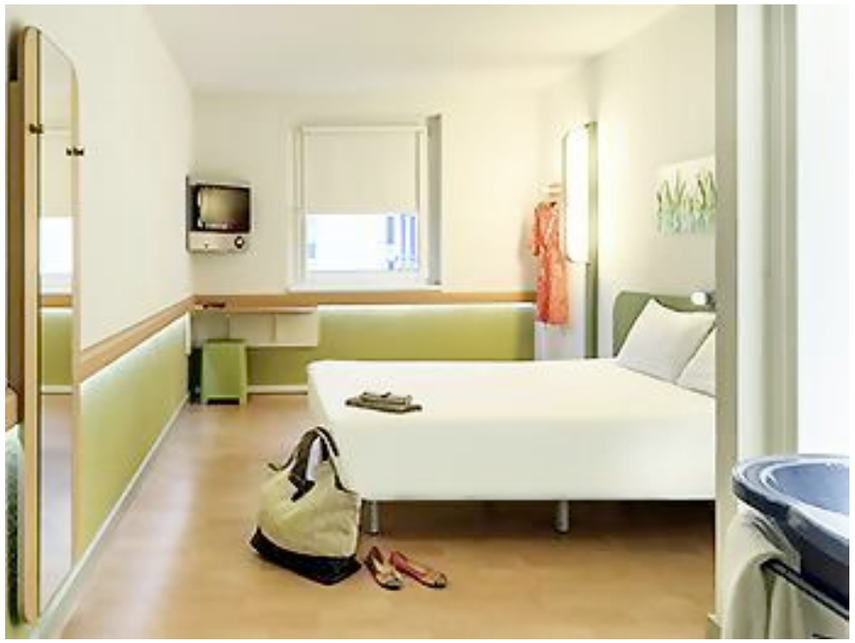
- ImageHash
- pHash - http://phash.org
  - http://pypi.python.org/pypi/phash/
  - https://github.com/polachok/py-phash
- imgSeek
  - http://www.imgseek.net/isk-daemon
- Lucene
  - http://www.lire-project.net/

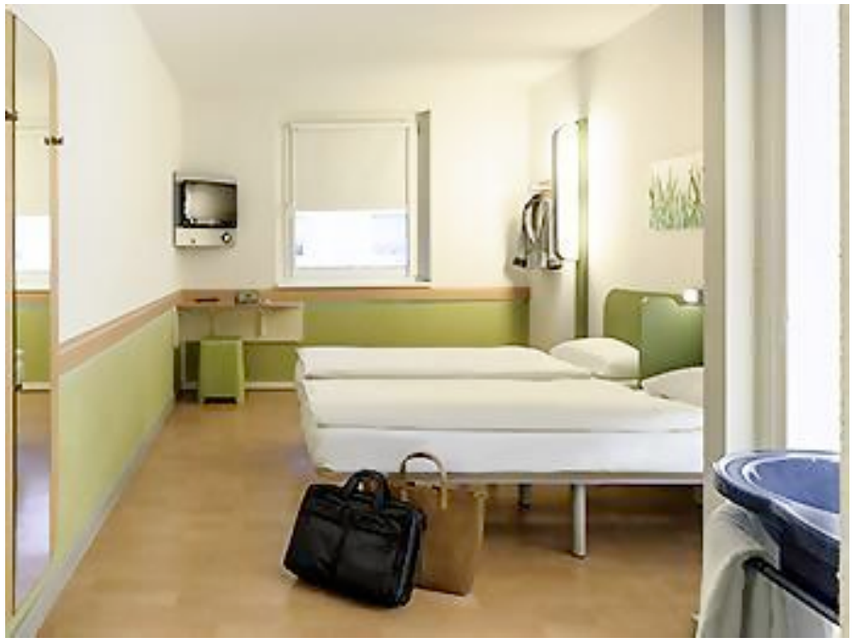
КАК НАМ ПОРТЯТ ЖИЗНЬ

- Изменение одного и того же оригинала
  - Ресайз с изменением aspect ratio
  - Цветокоррекция, баланс белого, контраст
  - Кроп
  - Поворот - нет, не слышал
- Разные фото одного и того же
  - БОЛЬ!

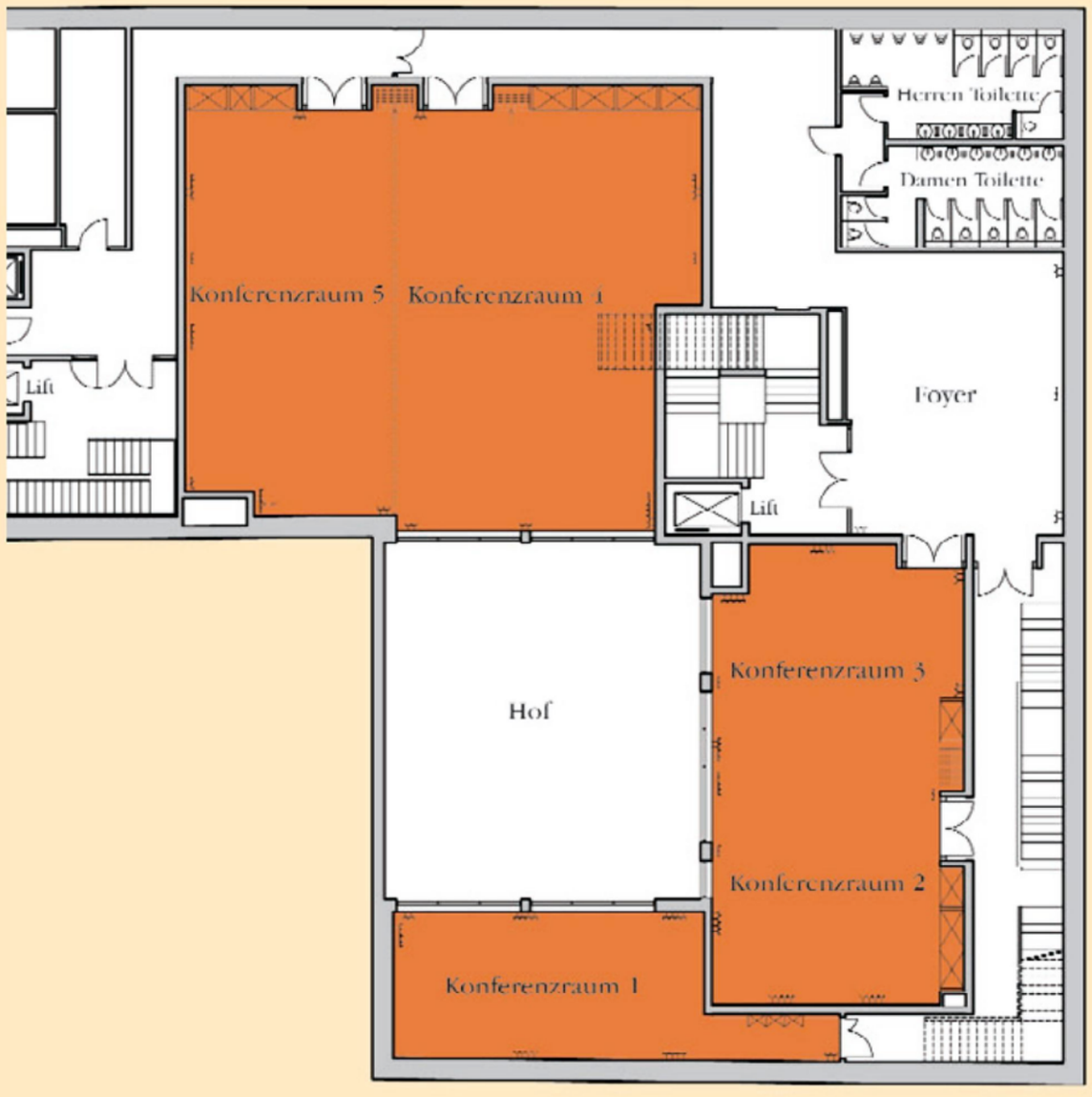
ДАННЫЕ - ВСЕ!

- Что такое похожие картинки?
- Много примеров, хороших и разных
- Могут быть неожиданные сюрпризы
- Вы узнаете что будет работать, а что нет





1. Untergeschoss



2. Untergeschoss

TEST IMAGES

KODAK TRUE COLOR TEST IMAGES

LOW QUALITY

В 4 раза меньше

JPEG quality = 10

CONTRAST & BRIGHTNESS

CROPS

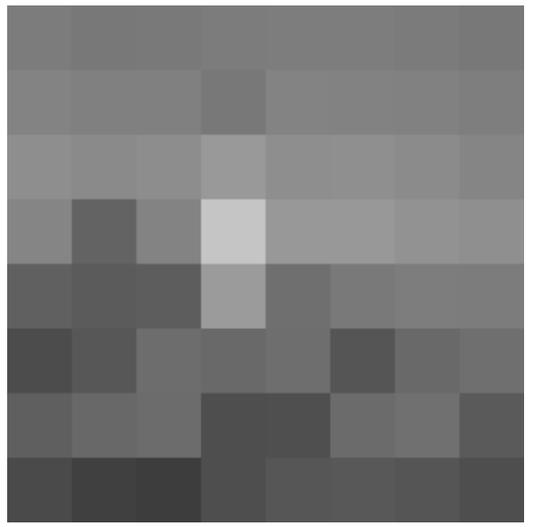
AFFINE TRANSFORMS

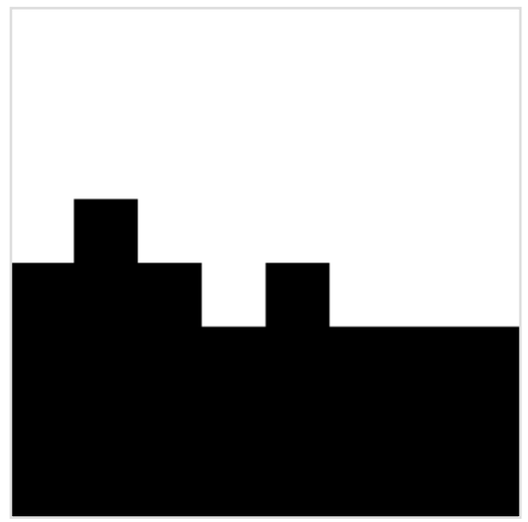
Препроцессинг

- Ресайзим до удобоваримого размера
  - Crop vs искажение
  - Hint: большие картинки можно ресайзить в два этапа
- Нужен ли цвет?
  - Если нужен, то HSV, HSL, YCbCr
- Blur?

AVERAGE HASHING

```
image = image.convert('L').resize((8, 8))
pixels = numpy.array(image)
avg = pixels.mean()
diff = pixels > avg
```

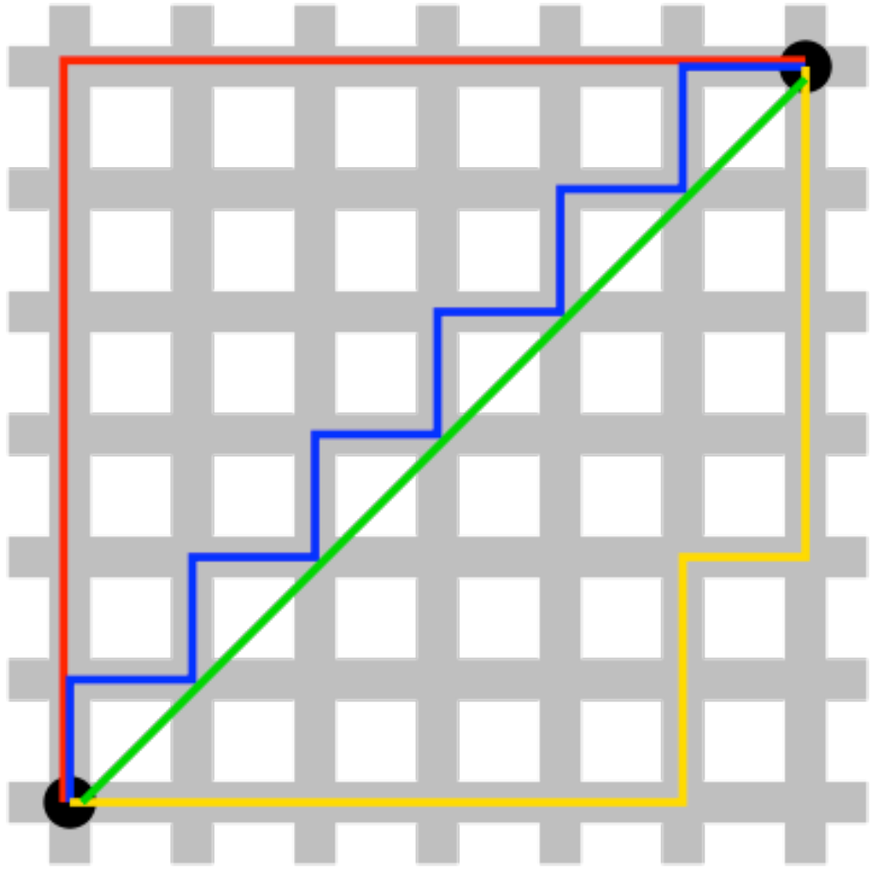




```
array([[ True,  True,  True,  True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True,  True,  True,  True],
       [ True, False,  True,  True,  True,  True,  True,  True],
       [False, False, False,  True, False,  True,  True,  True],
       [False, False, False, False, False, False, False, False],
       [False, False, False, False, False, False, False, False],
       [False, False, False, False, False, False, False, False]], dtype=bool)
```

А КАК СРАВНИВАТЬ ХЭШИ?

- Совпал или нет
- Расстояние Хэмминга
  - 00101101 XOR 10011101 == 10110000
- Расстояние Манхеттена
- Евклидово расстояние
- Косинусное расстояние
  - 1 - cos(Θ) между векторами

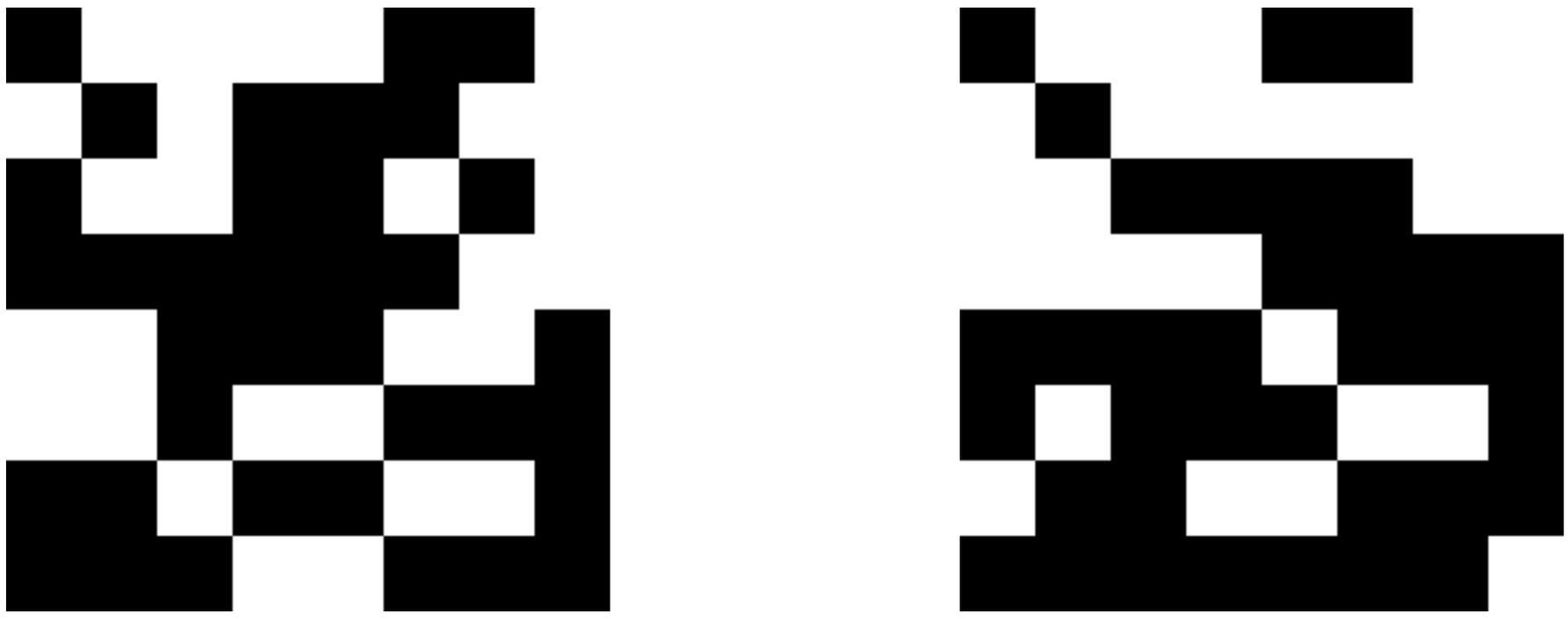


AVERAGE HASH RESULTS

	4x less	Q=10	Bright	Contrast	Crop small	Crop left	Rotate	Affine	Lenna
64b	1.0	1.0	1.0	1.0	0.94	0.89	0.92	0.97	0.56
256b	1.0	0.99	0.98	1.0	0.89	0.83	0.87	0.81	0.58
1024b	1.0	1.0	0.98	0.99	0.85	0.81	0.82	0.81	0.59

DIFFERENCE HASHING

```
image = image.convert('L').resize((8, 9))
pixels = numpy.array(image)
diff = pixels[1:,:] > pixels[:-1,:]
```



DIFFERENCE HASH RESULTS

	4x less	Q=10	Bright	Contrast	Crop small	Crop left	Rotate	Affine	Lenna
Vertical	1.0	0.96	0.97	1.0	0.89	0.72	0.83	0.81	0.52
Horizontal	1.0	0.95	0.9	0.95	0.81	0.58	0.8	0.70	0.63

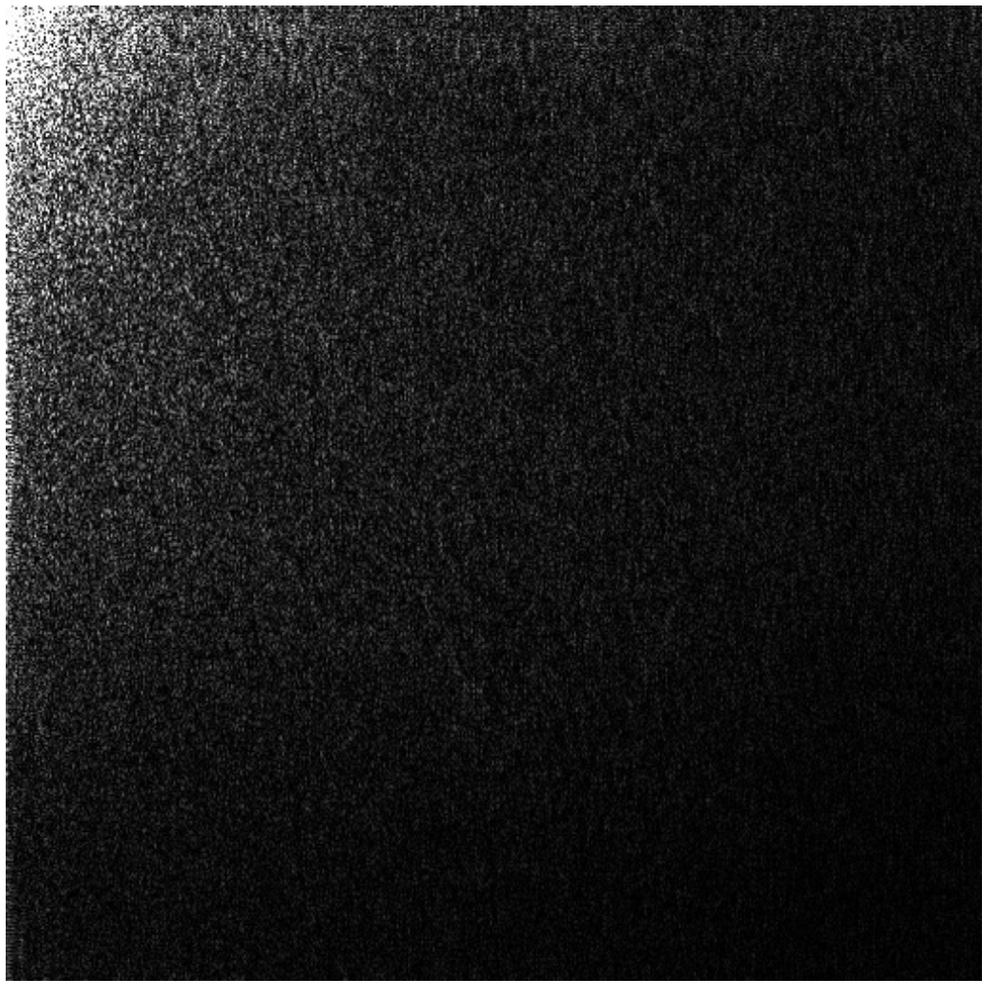
DCT (DISCRETE COSINE TRANSFORM)

- Частный случай DFT (Discrete Fourier Transform)
- Берем вещественную составляющую, отбрасываем комплексную
- Меньше вычислений, результат практически тот же

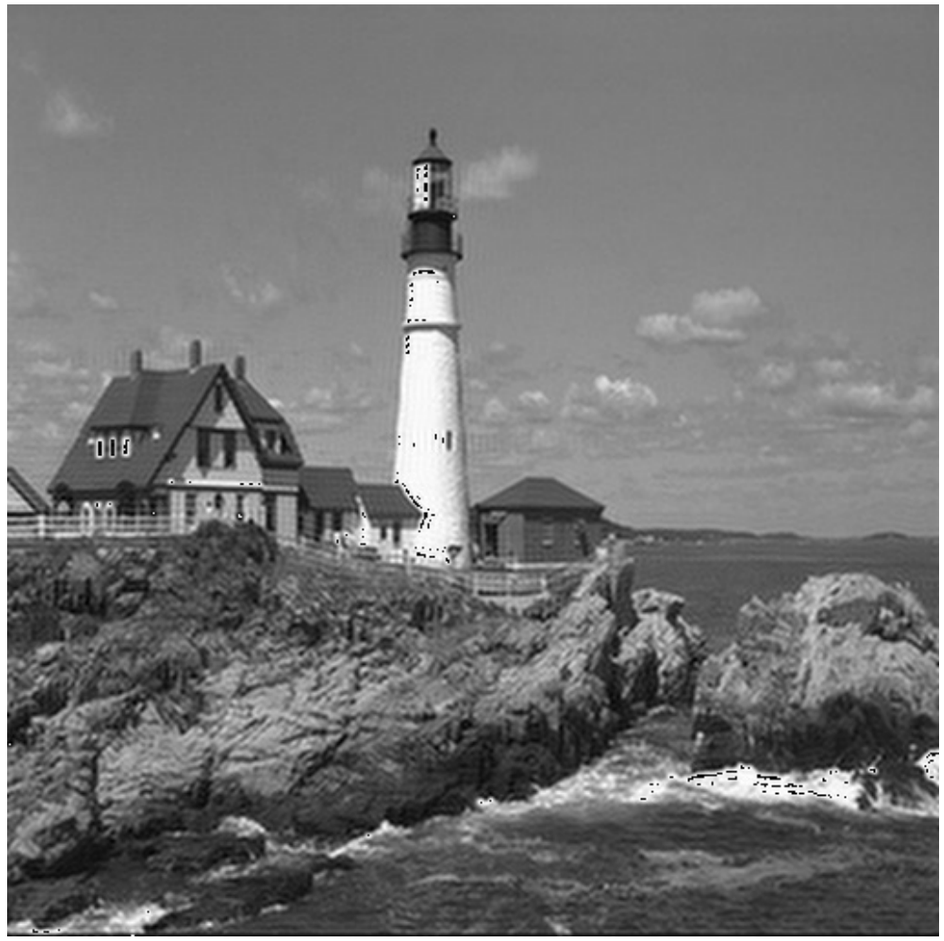
$$
\begin{aligned}
X_{k_1,k_2} &= \sum_{n_1=0}^{N_1-1}\left(\sum_{n_2=0}^{N_2-1} x_{n_1,n_2}\cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right]\right)\cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right] \\
&= \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x_{n_1,n_2}\cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right]\cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right].
\end{aligned}
$$

СТРОИМ DCT ХЭШ

```
pixels = np.array(image, dtype=np.float)
pixels_dct = dct(dct(pixels, axis=1), axis=0)
low_freq = pixels_dct[:hash_size, :hash_size]
median = np.median(low_freq)
diff = low_freq > median
```



512x512 DCT



256x256 DCT

128x128 DCT

64x64 DCT

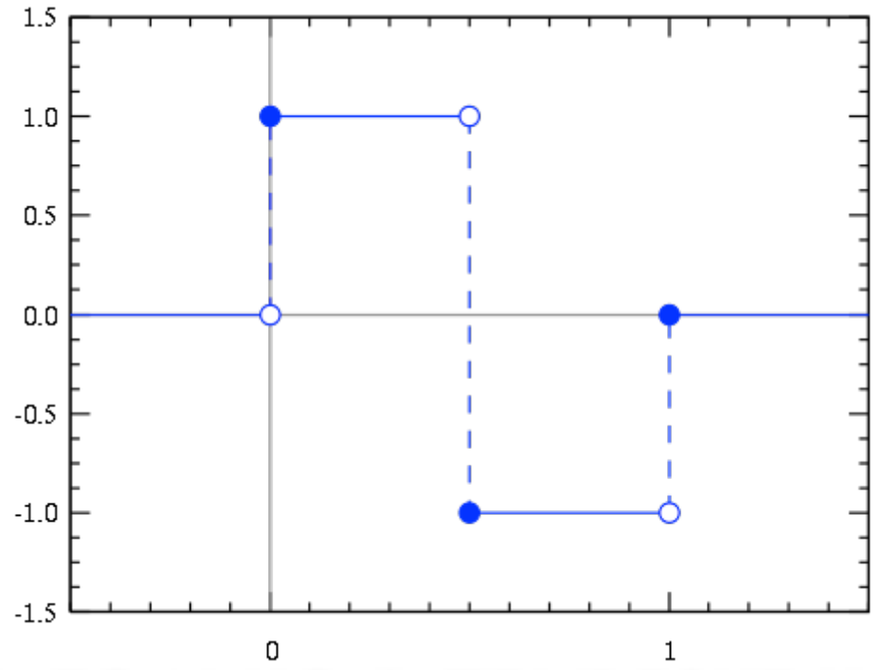
32x32 DCT

16x16 DCT

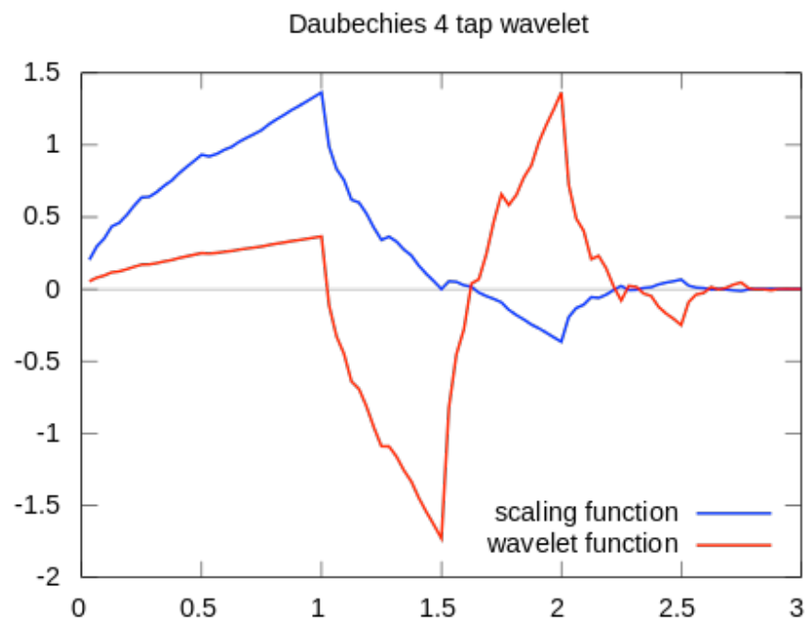
DCT HASH RESULTS

	4x less	Q=10	Bright	Contrast	Crop small	Crop left	Rotate	Affine	Lenna
Our	1.0	0.97	0.84	0.94	0.84	0.63	0.78	0.75	0.59
pHash	0.88	0.97	0.88	0.94	0.84	0.47	0.75	0.67	0.56
Our 256b	0.99	0.97	0.89	0.96	0.77	0.48	0.64	0.61	0.53
Our 1024b	0.998	0.97	0.9	0.95	0.58	0.49	0.56	0.52	0.5

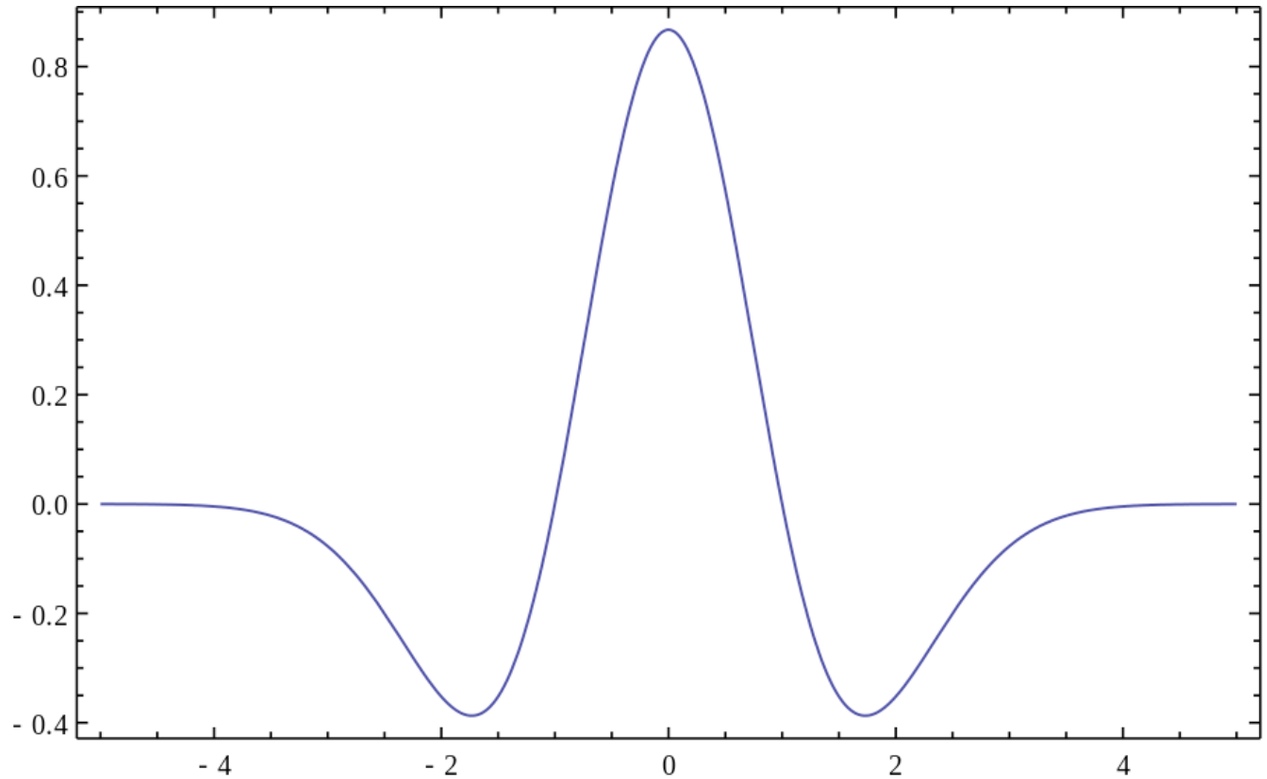
WAVELETS



- Они есть разные
- Описывают те же волны, только в привязке к положению
- JPEG2000
- Используются в imgSeek
- Есть реализация в pHash



MEXICAN HAT



4x less	Q=10	Bright	Contrast	Crop small	Crop left	Rotate	Affine	Lenna
0.77	0.82	0.88	0.9	0.51	0.53	0.5	0.5	0.5

STATISTICAL MOMENTS

- Статистический момент

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y)$$

  - Centroid: { x, y } = {M_{10}/M_{00}, M_{01}/M_{00} }
- Central moment

$$\mu_{pq} = \sum_m^p \sum_n^q \binom{p}{m} \binom{q}{n} (-\bar{x})^{(p-m)} (-\bar{y})^{(q-n)} M_{mn}$$

  - Translation invariant
  - Orientation
- Scale invariant moment

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\left(1+\frac{i+j}{2}\right)}}$$

HU INVARIANT MOMENTS

- Их 7 (8)
- Инвариантны к
  - Трансляции
  - Масштабированию
  - Повороту
- SKIMAGE.MEASURE.MOMENTS_HU
- https://en.wikipedia.org/wiki/Image_moment

Euclidian distance		Equalized
4x less	0.039	0.088
Q=10	0.0079	0.054
Bright	1.71	0.27
Contrast	0.52	0.0077
Crop	0.37	0.41
Crop	0.12	0.048
Rotate	0.24	0.044
Affine	0.22	0.092
Lenna	1.18	1.43

RADIAL VARIANCE HASH

- Дисперсия интенсивности вдоль осей
- Оси под углом, проходят через центр
- Есть реализация в pHash

4x less	Q=10	Bright	Contrast	Crop small	Crop left	Rotate	Affine	Lenna
0.99	1.0	0.77	0.98	0.98	0.34	0.94	0.73	0.46

ИНТЕРЕСНЫЕ ТОЧКИ

- Края (Edges)
- Углы (Corners)
- Blobs
- Ridges (Контур?)
- Регионы

- Точки - хорошо
- Дескрипторы - лучше
- OpenCV - features2d
- scikit image - sckimage.feature

Итоги

- Проблема сложная
- Решения есть
  - Простых нет
  - Комбинируйте разные признаки
  - Из 7M картинок мы убрали 600K дубликатов
    - 20 c3.large EC2 машин за сутки

КОНТАКТЫ

- http://github.com/ventura/talks
- Viktor.Drachov@gmail.com
- Внимание, голосование
  - https://github.com/ventura/image_compare