

• ВЕСЁЛАЯ ИГРА •

Дуббл – это захватывающая игра
на внимательность для детей и взрослых.

Внутри вы найдёте карточки с уникальным набором картинок!
Ни одна карта не повторяется, но удивительным образом
на каждой паре карт есть одна общая для них картинка.

Убедитесь в этом сами – найдите все
совпадения на скорость!



Уважаемые коллеги. Есть в моем доме игра “дубль”, бывшая в руках детей, и потрепанная жизнью изрядно. В моей голове давно работала мысль: как применить классические алгоритмы компьютерного зрения

для этой игры.

О том что было сделано за некоторое количество дней, и как это работает, я хочу поведать habr сообществу. В основе лежит FFT (fast fourier transform), а именно "An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration" описанной в статье авторов: B. Srinivasa Reddy and B. N. Chatterji. Другие подходы к решению, конечно, имеют право на существование, но данный метод эффективен, и чем сожнее объекты - тем лучше матчит, а ясность и красота математической идеи не оставит равнодушным того кто уделил достаточно внимания материалу упомянутой выше статьи. Для реализации проекта я использовал C++ и OpenCV library.

Отснятые на телефон карточки находятся в папке в JPEG формате. Теперь надо вырезать каждый рисунок. Белый картон (а он условно белый, т.к. есть следы использования карточек, и баланс бело "пляшет" на снимках) - это фон. Чтобы получить бинарную маску (фон - не-фон) создан байесовский классификатор: цвет RGB -> класс. Есть 2 класса, каждый представлен смесью гауссиан.

Теперь используем cv::findContours() с флагом cv::RETR_EXTERNAL чтобы найти внешний контур карточки, а затем та же функция чтобы получить внешние контуры каждого рисунка.

Разбиваем одну карточку и другую

```
std::vector<cv::Mat3b> pict_sampl0, pict_sampl1;  
  
split_jpg(fn0, img0, pict_sampl0, v0);  
split_jpg(fn1, img1, pict_sampl1, v1);
```

Как теперь найти рисунок, который присутствует в обоих списках ?

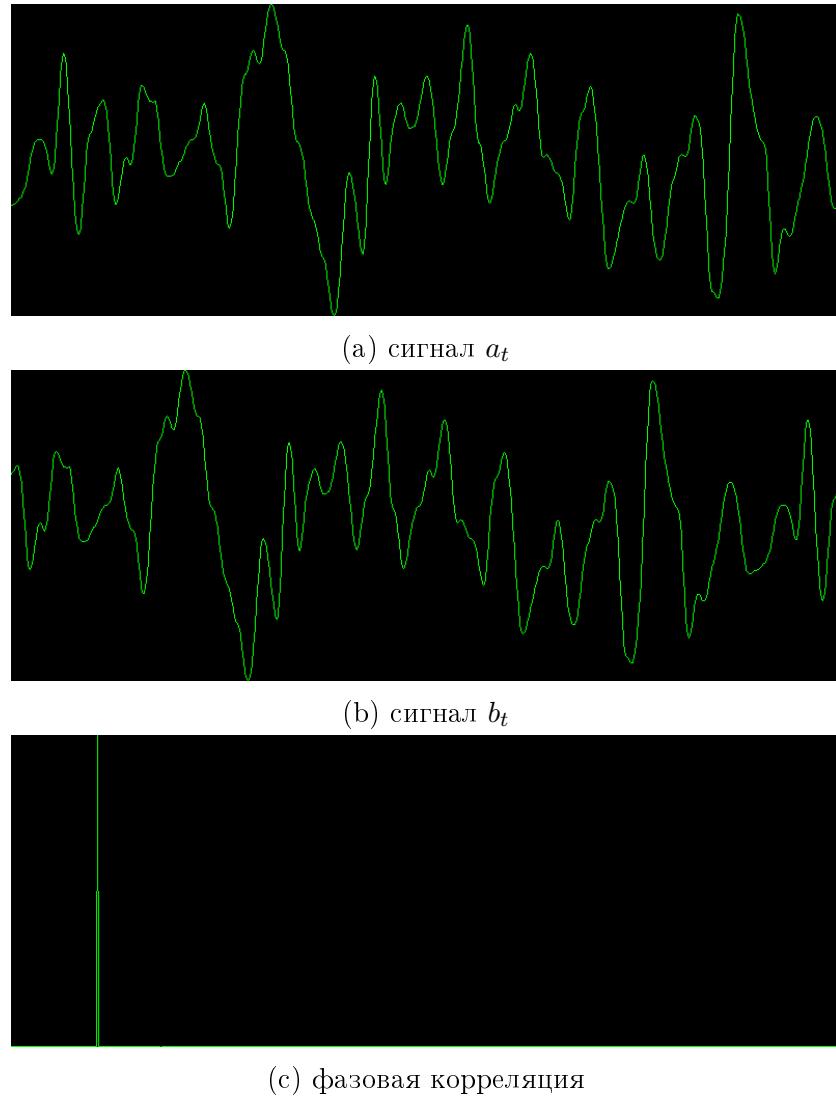
Начнем с более простой задачи и метода "Phase correlation".

Допустим, есть 1D сигнал (циклический с периодом N) a_t и другой сигнал b_t сдвинутый на Δt

$$a_t \stackrel{\text{def}}{=} b_{(t-\Delta t) \bmod N}$$

вычислим спектр A_k , B_k

$$A_k = \sum_{t=0}^{N-1} a_t \cdot e^{-i2\pi \frac{k}{N} t}$$
$$B_k = \sum_{t=0}^{N-1} b_t \cdot e^{-i2\pi \frac{k}{N} t}$$



ПОДСТАВИМ a_t

$$A_k = \sum_{t=0}^{N-1} b_{(t-\Delta t) \bmod N} \cdot e^{-i2\pi \frac{k}{N} t} = \sum_{t=0}^{N-1} b_t \cdot e^{-i2\pi \frac{k}{N} (t+\Delta t)} = e^{-i2\pi \frac{k}{N} \Delta t} \cdot B_k$$

Домножим обе части равенства на B_k^* - сопряженное к B_k , а затем поделим на вещественное число $B_k B_k^* = |B_k B_k^*|$

$$\frac{A_k B_k^*}{|B_k B_k^*|} = e^{-i2\pi \frac{k}{N} \Delta t}$$

а теперь, внимание (!) трюк. подставим B_k в знаменателе. Модуль произведения равен произведению модулей, а модуль мнимой экспоненты равен единице.

$$|B_k B_k^*| = |e^{i2\pi \frac{k}{N} \Delta t} \cdot A_k B_k^*| = |e^{i2\pi \frac{k}{N} \Delta t}| \cdot |A_k B_k^*| = |A_k B_k^*|$$

итак, применяя формулу к компонентам спектра, получим новый спектр:

$$e^{-i2\pi \frac{k}{N} \Delta t} = \frac{A_k B_k^*}{|A_k B_k^*|}$$

это ничто иное, как спектр единичного импульса.

$$e^{-i2\pi \frac{k}{N} \Delta t} = \sum_{t=0}^{N-1} c_t \cdot e^{-i2\pi \frac{k}{N} t}$$

$$c_t = \begin{cases} 1 & \text{если } t = \Delta t, \\ 0 & \text{иначе.} \end{cases}$$

Аналогично, метод фазовой корреляции применяется для 2D данных. Выполняем быстрое Фурье преобразование (FFT) для двух изображений, вычисляем cross-power спектр, выполняем обратное быстрое Фурье преобразование, ищем максимум. Алгоритмическая сложность вычисления кросс-корреляции “по определению” - $O(N^2)$, данный же алгоритм имеет сложность $O(N \cdot \ln(N))$, где N - число элементов (в данном случае пикселей). Итак, сдвиг изображения приводит к сдвигу фазы, т.е. каждая компонента спектра домножается на мнимую экспоненту от k . В то же время сдвиг не изменяет магнитуду.

Рассмотрим теперь поворот (rotation) и изменение масштаба (scale). Если повернуть изображение, то спектр (непрерывный) повернется на тот же угол. Доказательство приводить не будем, но сделаем иллюстрацию (см. рисунок). Для пары изображений возьмем магнитуду спектра. Тогда сдвиг не играет роли, а поворот рисунка относительно другого мы можем определить: Для этого перейдем к полярным координатам и вычислим смещение по Y (которое будет соответствовать углу) методом фазовой корреляции. Если же изображение увеличить в scale раз то спектр сожмется в scale раз. Поэтому, если представить магнитуду спектра в системе координат logpolar (логарифм от радиуса), то rotation и scale можно определить фазовой корреляцией.

Так как в нашем случае не пара изображений, а два списка - каждый снимок предварительно подготавливается: вычисляется FFT, магнитуда

```

void phase_corr_eq(complex_t& a, complex_t& b, complex_t& dst)
{
    float v1 = a.Re * b.Re + a.Im * b.Im;
    float v2 = a.Im * b.Re - a.Re * b.Im;
    float mag = hypotf(v1, v2);

    if (mag == 0)
    {
        dst.Re = 0;
        dst.Im = 0;
    }
    else
    {
        dst.Re = v1 / mag;
        dst.Im = v2 / mag;
    }
}

void perf_phase_corr_eq(cv::Mat1f& FFT1, cv::Mat1f& FFT2, cv::Mat1f& C)
{
    // CCS packed format (complex-conjugate-symmetrical)
    // ...
}

cv::Mat<float> s0, s1, s0_FFT, s1_FFT, cross_power_spec, cross_power_spec_inv;
//...
dft(s0, s0_FFT, DFT_REAL_OUTPUT);
dft(s1, s1_FFT, DFT_REAL_OUTPUT);
perf_phase_corr_eq(s0_FFT, s1_FFT, cross_power_spec);
idft(cross_power_spec, cross_power_spec_inv, DFT_REAL_OUTPUT);

```

Рис. 2: реализация: FFT, cross-power spectrum, inverse FFT (idft() - OpenCV)

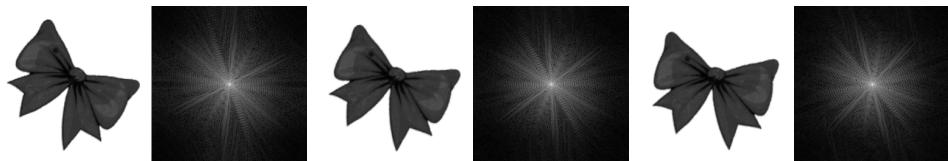


Рис. 3: поворот изображения и магнитуда спектра

спектра, и переводится в систему координат logpolar. Затем выполняем FFT для logpolar представления, полученного на предыдущем шаге. Вычисляем cross-power spectrum для каждой пары (первое изображение из первого списка, второе - из второго). Выполняем обратное FFT. Ищем максимум. Та пара, которая имеет максимальное значение (response) и будет вероятным претендентом. По позиции этого максимума определяется scale и rotaion. Затем выполняется аффинная трансформация второго изображения (rotaion, scale) и методом фазовой корреляции ищется

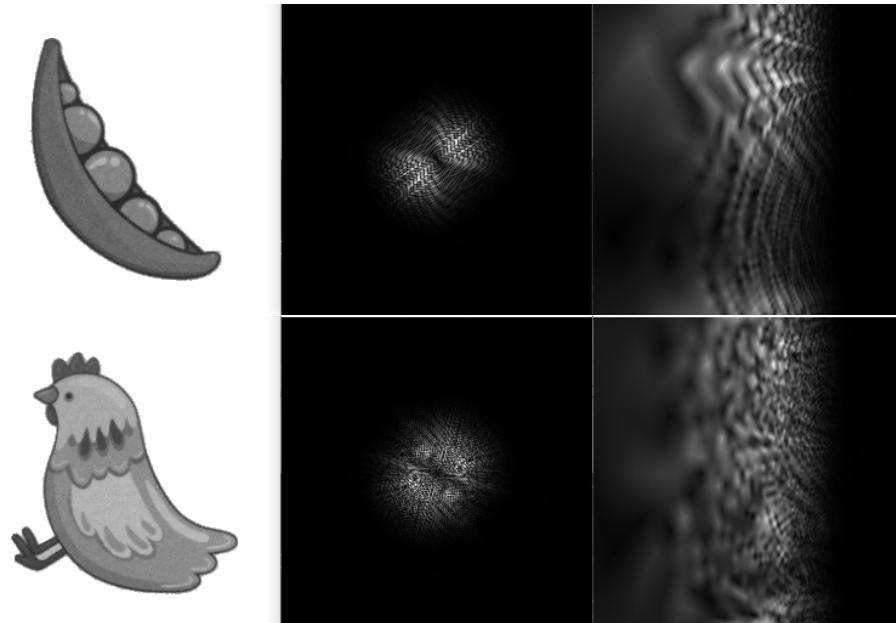


Рис. 4: изображение, мангитуда, logpolar представление



Рис. 5: предварительная фильтрация: high pass filtering, gaussian filtering, apply two filters

смещение. Следует заметить, что модуль спектра - симметричен. Поэтому, есть неопределенность угла поворота с кратностью 180° . Необходимо проверить изображение повернутое на $(\alpha + 180^\circ)$ и выбрать вариант у которого больше response.

[демо на youtube](#)
[проект на github](#)