

Image processing

Lab 3

Бинаризация, сегментация, анализ формы объектов.

1. Реализовать алгоритм сегментации (на выбор).
2. Протестировать работу алгоритма. Тестовую базу найти самостоятельно или самостоятельно сформировать.

Results

Otsu's method

Метод [\[править | править код \]](#)

Метод Оцу ищет порог, уменьшающий **дисперсию** внутри класса, которая определяется как взвешенная сумма дисперсий двух классов:

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t),$$

где веса ω_i — это вероятности двух классов, разделенных порогом t ,
 σ_i^2 — дисперсия этих классов.

Оцу показал, что минимизация дисперсии *внутри* класса равносильна максимизации дисперсии *между* классами:^[1]

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2$$

которая выражается в терминах вероятности ω_i и **среднего арифметического** класса μ_i , которое, в свою очередь, может обновляться **итеративно**. Эта идея привела к эффективному алгоритму.

Алгоритм [\[править | править код \]](#)

Пусть дано монохромное изображение $G(i, j)$, $i = \overline{1, Height}$, $j = \overline{1, Width}$. Счетчик повторений $k = 0$.

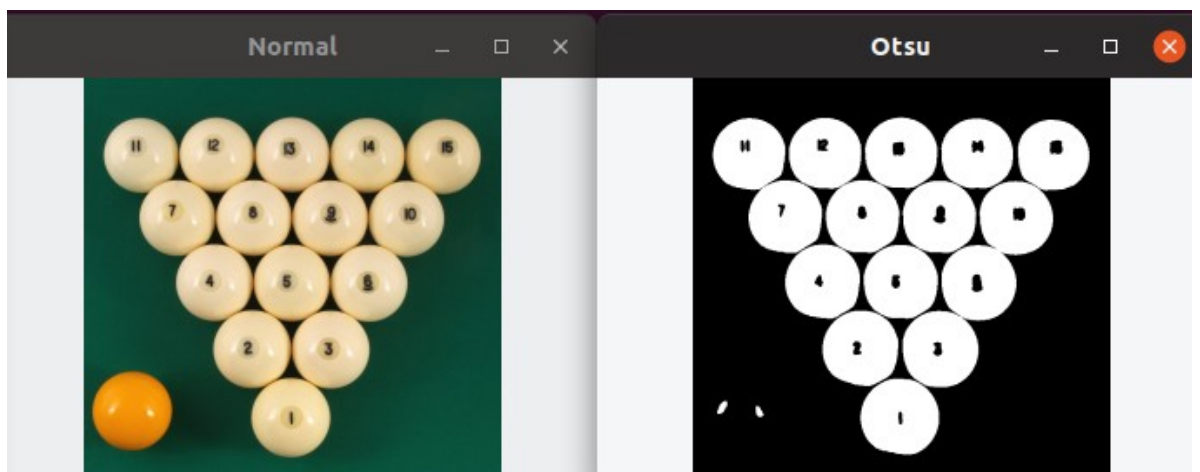
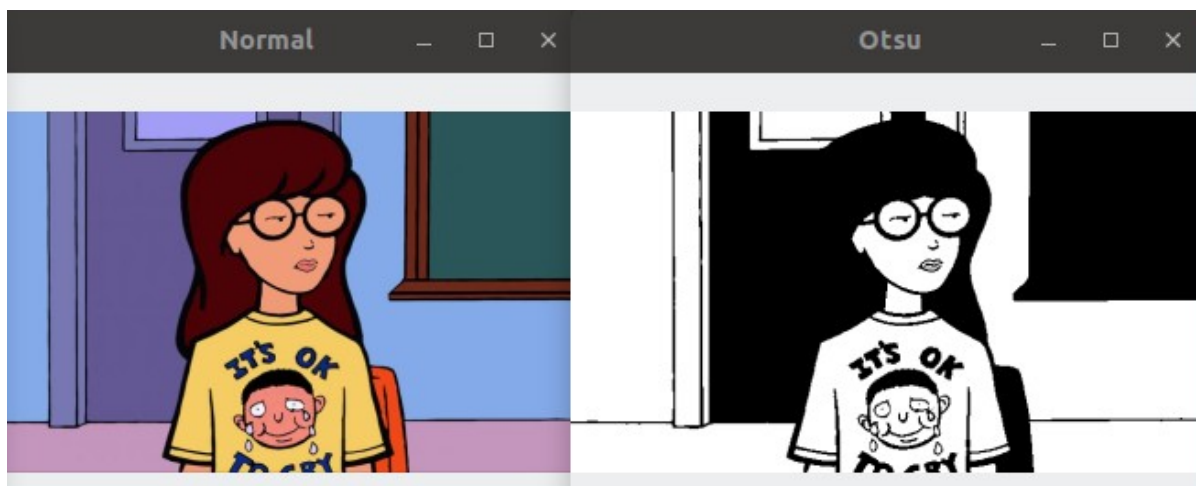
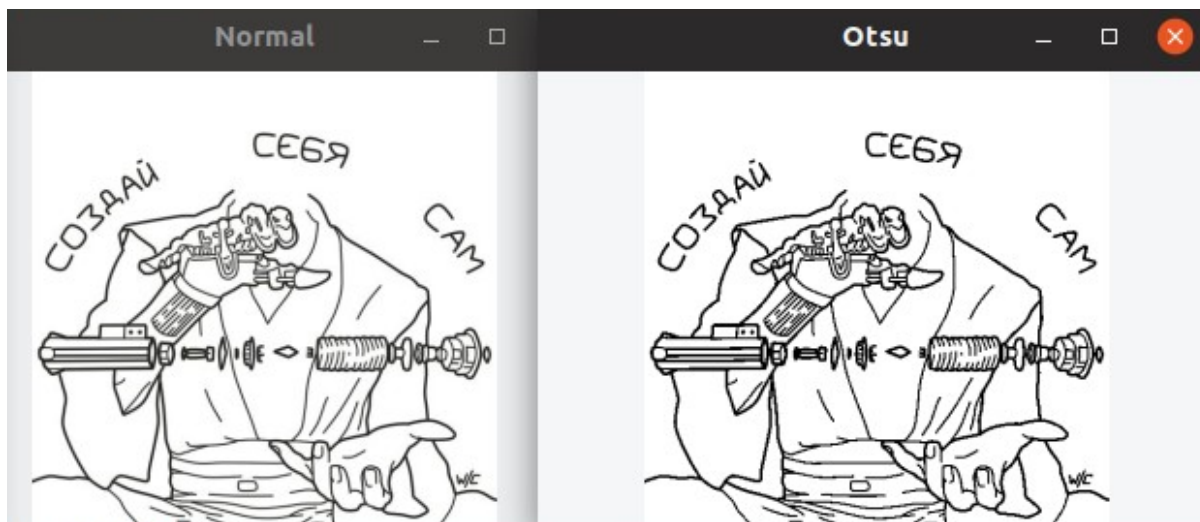
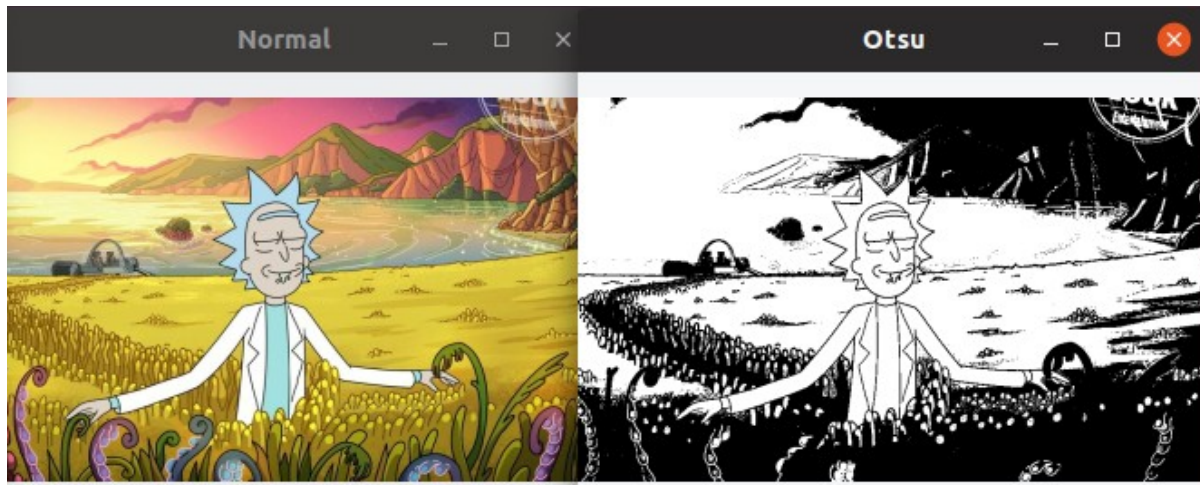
1. Вычислить **гистограмму** $p(l)$ изображения и **частоты** $N(l)$ для каждого уровня интенсивности изображения G .
2. Вычислить начальные значения для $\omega_1(0)$, $\omega_2(0)$ и $\mu_1(0)$, $\mu_2(0)$.
3. Для каждого значения $t = \overline{1, max(G)}$ — полутона — горизонтальная ось гистограммы:
 1. Обновляем ω_1, ω_2 и μ_1, μ_2
 2. Вычисляем $\sigma_b^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2$.
 3. Если $\sigma_b^2(t)$ больше, чем имеющееся, то запоминаем σ_b^2 и значение порога t .
4. Искомый порог соответствует максимуму $\sigma_b^2(t)$.

$$N_T = \sum_{i=0}^{max(G)} p(i),$$

$$\omega_1(t) = \frac{\sum_{i=0}^{t-1} p(i)}{N_T} = \sum_{i=0}^{t-1} N(i), \quad \omega_2(t) = 1 - \omega_1(t),$$

$$\mu_T = \frac{\sum_{i=0}^{max(G)} i \cdot p(i)}{N_T} = \sum_{i=0}^{max(G)} i \cdot N(i),$$

$$\mu_1(t) = \frac{\sum_{i=0}^{t-1} i \cdot p(i)}{N_T \cdot \omega_1(t)} = \frac{\sum_{i=0}^{t-1} i \cdot N(i)}{\omega_1(t)}, \quad \mu_2(t) = \frac{\mu_T - \mu_1(t) \cdot \omega_1(t)}{\omega_2(t)}.$$



K-Means method

Алгоритм

данных

Итак, если мера близости до центроида определена, то разбиение объектов на кластеры сводится к определению центроидов этих кластеров. Число кластеров k задается исследователем заранее.

Рассмотрим первоначальный набор k средних (центроидов) μ_1, \dots, μ_k в кластерах S_1, S_2, \dots, S_k . На первом этапе центроиды кластеров выбираются случайно или по определенному правилу (например, выбрать центроиды, максимизирующие начальные расстояния между кластерами).

Относим наблюдения к тем кластерам, чье среднее (центроид) к ним ближе всего. Каждое наблюдение принадлежит только к одному кластеру, даже если его можно отнести к двум и более кластерам.

Затем центроид каждого i -го кластера перевычисляется по следующему правилу:

$$\mu_i = \frac{1}{s_i} \sum_{x^{(j)} \in S_i} x^{(j)}$$

Таким образом, алгоритм k -средних заключается в перевычислении на каждом шаге центроида для каждого кластера, полученного на предыдущем шаге.

Алгоритм останавливается, когда значения μ_i не меняются: $\mu_i^{\text{mar } t} = \mu_i^{\text{mar } t+1}$

Важно: Неправильный выбор первоначального числа кластеров k может привести к некорректным результатам. Именно поэтому при использовании метода k -средних важно сначала провести проверку подходящего числа кластеров для данного набора данных.

Итак, еще раз подчеркнем некоторые особенности метода k -средних:

1. В качестве метрики используется Евклидово расстояние
2. Число кластеров заранее не известно и выбирается исследователем заранее
3. Качество кластеризации зависит от первоначального разбиения

