

РАЗРАБОТКА АЛГОРИТМА СЕГМЕНТАЦИИ РУКОПИСНОГО ТЕКСТА С ИСПОЛЬЗОВАНИЕМ МАСКИ

В работе освещается проблема выделения рукописного текста на изображении для последующего распознавания. Приводится описание широко распространенных методов сегментации, позволяющих решить данную проблему: метод цветового анализа изображений, бинарная сегментация изображений методом фиксации уровня, а также метод выделения и анализа контуров. Предложен алгоритм сегментации текста, основанный на использовании маски исходного изображения, позволяющий существенно упростить сегментацию текста на изображениях с немонокромным фоном. Также производится сравнительный анализ методов.

Ключевые слова: сегментация рукописного текста, использование маски, удаление фона изображения.

Введение. В задачах оптического распознавания объектов на изображении основой успешного распознавания является качество обрабатываемого изображения. В связи с этим вероятность успеха напрямую зависит от алгоритмов предобработки изображения, позволяющих решать такие задачи, как повышение контрастности, восстановление поврежденных изображений, подавление шумов, выделение фона и т.д.

В настоящий момент существует несколько алгоритмов, позволяющих в той или иной степени решать поставленную задачу.

Обзор существующих аналогов.

1. Цветовой анализ изображений.

Данный метод является наиболее простым и может быть применим как к цветным изображениям, так и к изображениям в градациях серого, в которых объект сегментации имеет высокую контрастность по отношению к фону изображения.

Основа метода заключается в представлении изображения в виде некоторой цветовой модели, например, RGB, в которой исходное изображение разбивается на 3 канала: красный, зеленый и синий, соответственно, или HSV — тон, насыщенность, значение и т.д.

Рассмотрим структуру дискретного цветового изображения в RGB представлении.

Цветовая модель изображения имеет следующий вид:

$$\begin{aligned} M[f(m, n)] &= \{p_0[x_0(m), y_0(n), c_0(r(l), g(l), b(l))]\} \\ p_1[x_1(m), y_1(n), c_1(r(l), g(l), b(l))] & \\ \dots & \\ p_k[x_k(m), y_k(n), c_k(r(l), g(l), b(l))] & \} \end{aligned} \quad (1)$$

где M — массив пикселей; $[f(m, n)]$ — функция расположения пикселей на изображении; p — пиксель, состоящий из трех цветов (красный, зеленый, синий); x и y — индексы строки и столбца положения пикселя в матрице изображения; c — код цвета; r — красный цвет; g — зеленый цвет; b — синий цвет; l — уровень градации цвета; k — количество пикселей в изображении.

Для решения задачи выделения объектов на цветных дискретных изображениях методом цветового анализа предлагаются следующие алгоритмы:

- определение наличия чистых цветов по шкале RGB;
- разделение исходного изображения на два (серое и чисто цветное);
- снижение числа градаций цвета путем сдвига градаций к нижнему уровню, среднему уровню, верхнему уровню или определенному уровню цвета без потерь пикселей данного цвета;
- поиск областей скопления пикселей выбранного цвета на изображении путем использования спиральной развертки;
- выявление контура цветного объекта в выбранной области.

Рассмотрим работу метода цветового анализа изображения. Алгоритм работы представлен на рис. 1.

В используемом алгоритме можно выделить 6 основных пунктов. После загрузки исходного изображения (пункт 1) необходимо определить параметры фрагмента сегментации (пункт 2). В данном случае фрагментом сегментации будет являться зеленый кубик.

Определим соотношения RGB-составляющих в выбранном фрагменте. Для удобства визуального представления воспользуемся возможностями встроенной функции «imtool» [1], входящей в со-

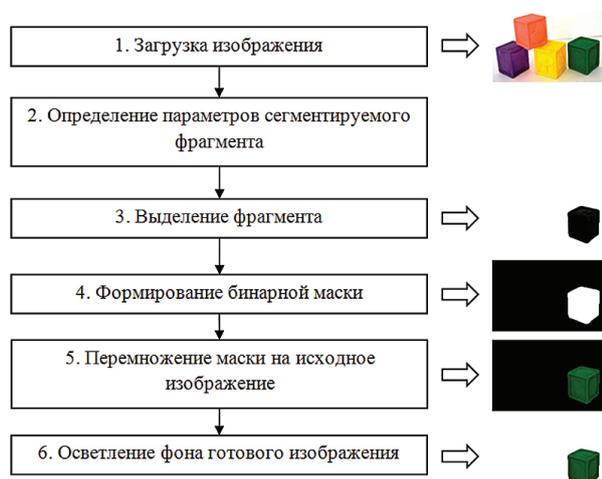


Рис. 1. Блок-схема алгоритма цветового анализа изображений

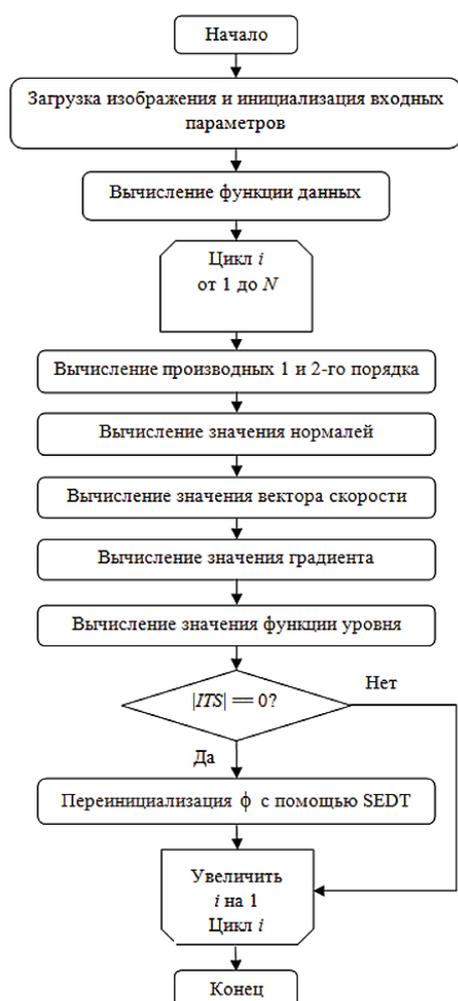


Рис. 2. Блок-схема бинарной сегментации изображений методом фиксации уровня

став пакета Image Processing Toolbox программного обеспечения Matlab, и оценим яркость пикселей по 3-м каналам.

После определения параметров фрагмента сегментации осуществляется выделение фрагмента (пункт 3) путем изменения соотношения основных цветов. После чего формируется маска изображения (пункт 4).

Пункты 5 и 6 являются необязательными и используются в случае необходимости получения фрагмента сегментации в исходном виде. В таком случае полученную маску необходимо поэлементно умножить на исходное изображение и в случае необходимости изменить цвет фона.

Главными достоинствами данного метода являются простота и наглядность.

Основными недостатками являются: необходимость хорошей различимости фрагмента сегментации изображения по отношению к фону и побочным объектам, необходимость ручной настройки параметров для получения качественного результата, а также невозможность применения метода к бинарным изображениям.

2. *Бинарная сегментация изображений методом фиксации уровня.*

Данный метод был введен американскими математиками Стэнли Осером и Джеймсом Сетьяном в 80-е годы прошлого века. Он стал популярным во многих дисциплинах, таких как компьютерная графика, обработка изображений, вычислительная геометрия, оптимизация, вычислительная гидродинамика и вычислительная биофизика.

Метод фиксации уровня неявно преобразует контур (в двух измерениях) или поверхность (в трех измерениях), используя функцию более высокой размерности, называемую функцией фиксации уровня $\phi(x, t)$.

Преобразованный контур или поверхность можно получить как кривую $\Gamma(x, t) = \{\phi(x, t) = 0\}$ [2].

Блок-схема алгоритма представлена на рис. 2.

Перед началом работы алгоритма производится загрузка исходного изображения I , а также проводится инициализация основных переменных, таких как исходная маска m ; порог T ; ошибка ε ; число итераций N ; число итераций для пересчета карты расстояний, ITS .

Далее следует вычисление функции данных по формуле:

$$D(I) = \varepsilon - |I - T| \quad (2)$$

Функция $D(I)$, по сути, управляет всей сегментацией, сжимая или расширяя модель в зависимости от попадания пикселя в диапазон $(T - \varepsilon, T + \varepsilon)$ [2].

Производные вычисляются вычитанием сдвинутых матриц функции ϕ . Для двух измерений требуется только $D_x^+, D_x^-, D_y^+, D_y^-$ [2]:

$$\begin{aligned} D_x^+ &= \frac{\phi_{i+1,j+1} - \phi_{i-1,j+1}}{2} & D_x^- &= \frac{\phi_{i+1,j-1} - \phi_{i-1,j-1}}{2} \\ D_y^+ &= \frac{\phi_{i+1,j+1} - \phi_{i+1,j-1}}{2} & D_y^- &= \frac{\phi_{i-1,j+1} - \phi_{i-1,j-1}}{2} \end{aligned} \quad (3)$$

Вычисление нормалей n^+ и n^- производится по формуле:

$$n^+ = \begin{bmatrix} \frac{D_x^+}{(D_x^+)^2 + \left(\frac{D_y^+ + D_y^-}{2}\right)^2} \\ \frac{D_y^+}{(D_y^+)^2 + \left(\frac{D_x^+ + D_x^-}{2}\right)^2} \end{bmatrix} \quad n^- = \begin{bmatrix} \frac{D_x^-}{(D_x^-)^2 + \left(\frac{D_y^- + D_y^+}{2}\right)^2} \\ \frac{D_y^-}{(D_y^-)^2 + \left(\frac{D_x^- + D_x^+}{2}\right)^2} \end{bmatrix} \quad (4)$$

Изменение вида вектора F с точностью до параметров позволяет направлять и изменять линию уровня различным образом [2].

$$F = \alpha D(I) + (1 - \alpha) \nabla \cdot \frac{\nabla}{|\nabla \phi|}, \quad (5)$$

где α — весовой параметр, задаваемый пользователем; $\nabla \cdot \frac{\nabla}{|\nabla \phi|}$ — средний член кривизны функции ϕ , который будет сохранять эту функцию гладкой.

Вычисление градиента $\nabla \phi$ определяется в зависимости от того, является ли $F_{i,j,k} > 0$ или $F_{i,j,k} < 0$ [2]:

$$\nabla \phi = \begin{cases} \|\nabla \phi_{\max}\|_2 & \text{если } F_{i,j,k} > 0 \\ \|\nabla \phi_{\min}\|_2 & \text{если } F_{i,j,k} < 0 \end{cases}. \quad (6)$$

$\nabla \phi_{\max}$ и $\nabla \phi_{\min}$ рассчитываются по формулам [2]:

$$\nabla \phi_{\max} = \begin{bmatrix} \sqrt{\max(D_x^+, 0)^2 + \max(-D_x^+, 0)^2} \\ \sqrt{\max(D_y^+, 0)^2 + \max(-D_y^+, 0)^2} \end{bmatrix}, \quad (7)$$

$$\nabla \phi_{\min} = \begin{bmatrix} \sqrt{\min(D_x^+, 0)^2 + \min(-D_x^+, 0)^2} \\ \sqrt{\min(D_y^+, 0)^2 + \min(-D_y^+, 0)^2} \end{bmatrix}.$$

Производим пересчет функции уровня [2] (рис. 3):

$$\phi(t + \Delta t) = \phi(t) + \Delta t \cdot F \cdot |\nabla \phi|. \quad (8)$$

Преимущество использования этого метода состоит в том, что топологические изменения, такие как слияние и разделение контура или поверхности, решаются неявно.

Недостатками являются значительные временные затраты, а также невозможность обработки бинарных изображений.

3. Выделение и анализ контуров.

Данный метод в основном применяется в случае, если объект сегментации существенным образом не выделяется на фоне по цвету или имеет сложную раскраску, т.е. в случаях, когда применение методов цветового анализа и бинарной сегментации не даст хороших результатов. В этом случае можно попробовать применить метод выделения и анализа контуров.

Особенность данного метода заключается в выделении границы объекта сегментации на изображении. Граница — это место с резким изменением градиента яркости. Поиск границы может быть осуществлен с помощью методов Canny, Sobel, Prewitt, или Roberts [3]. Среди представленных детекторов наибольшую популярность получил метод выделения границ Сэнну.

В 1986 г. Джон Ф. Канни разработал алгоритм обнаружения краев для изображений, искажённых

шумами. Целью Канни было разработать оптимальный алгоритм обнаружения края, который уменьшал бы вероятность обнаружения ложных краев и выдавал острые края [4].

Детектор границ Сэнну является оптимальным методом, т.к. имеет ряд преимуществ:

- обеспечивает хорошее обнаружение края;
- чёткий отклик;
- обеспечивает хорошую локализацию (рис. 4).

Алгоритм детектора границ Сэнну.

1. Ослабление шумов с помощью размытия.

Перед началом непосредственной обработки изображения необходимо избавиться от шума. Шум на изображении устраняется путем применения операции свертки исходного изображения $Image(i, j)$ с фильтром Гаусса G , применение которого приводит к сильному размытию изображения. Результат сглаживания представляется формулой (5) и описывается выражением:

$$F(i, j) = G \cdot Image(i, j). \quad (9)$$

2. Вычисление градиента.

Суть данного этапа заключается в определении мест, в которых изменение уровня интенсивности оттенков серого максимально. Данные места можно обнаружить при помощи градиента изображения, который вычисляется для i и j направлений.

Вычисление градиентов в направлениях i и j определяется формулами (7) и представляется в виде [5]:

$$G_i = D_i \cdot F(i, j) \quad \text{и} \quad G_j = D_j \cdot F(i, j), \quad (10)$$

где G_i и G_j — градиенты в направлении i и j соответственно;

$$D_i = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{и} \quad D_j = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \text{— операторы Собеля в направлениях } i \text{ и } j \text{ соответственно.}$$

3. Подавление не максимумов.

Границы изображения будут проходить в точках, в которых градиент находится на максимальном уровне. Таким образом, все точки не в максимуме должны быть подавлены. Подавление не максимумов выполняется для сохранения всех локальных максимумов в градиентном изображении, что приводит к получению тонких краев [6].

4. Пороговая фильтрация.

Пороговая фильтрация применяется для подавления шумов, созданных в результате подавления не максимумов. С целью улучшения качества фильтрации применяют два порога.

Выходной сигнал после подавления не максимумов по-прежнему содержит локальные максимумы,

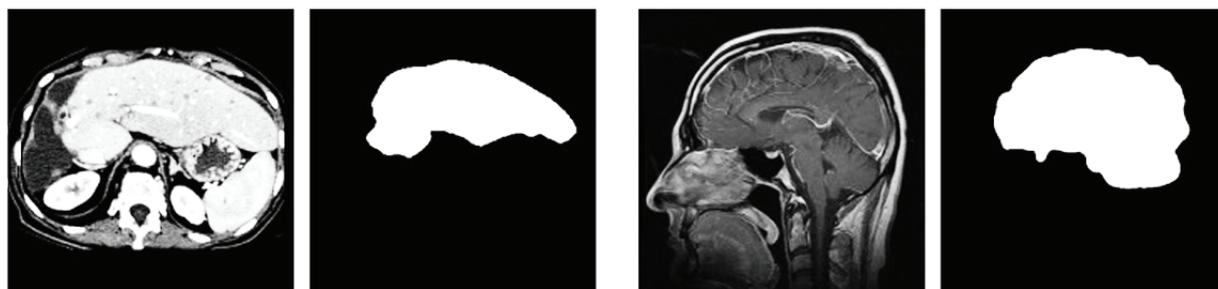


Рис. 3. Пример использования бинарной сегментации изображений методом фиксации уровня

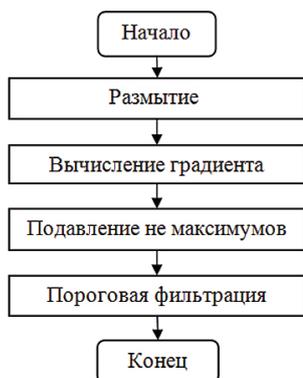


Рис. 4. Блок-схема алгоритма, используемого в детекторе Canny

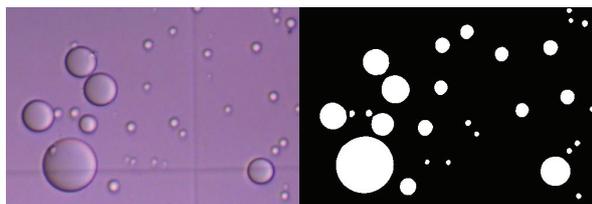


Рис. 5. Пример использования метода выделения контуров

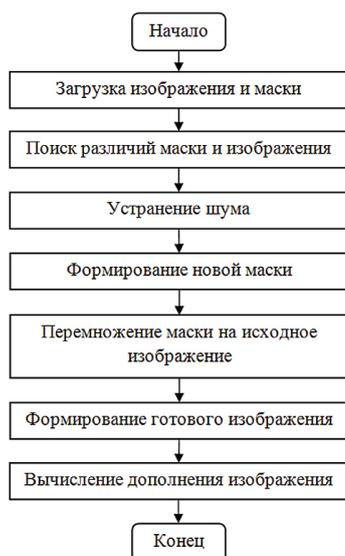
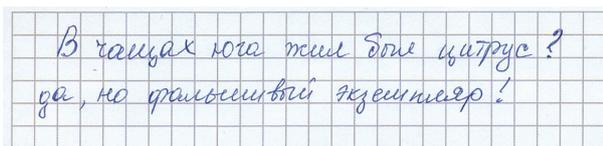
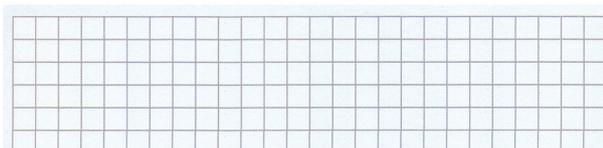


Рис. 6. Блок-схема алгоритма



а)



б)

Рис. 7. а — исходное изображение; б — маска

созданные шумом. Для улучшения качества фильтрации используют не один, а два порога t_H и t_L [7].

Если $G > t_L$ пиксель не является границей.

Если $G > t_H$ пиксель является границей.

Если $t_L > G > t_H$ и любой из соседних пикселей в зоне 3×3 имеет градиент выше, чем t_H , то пиксель является границей.

Также данный метод позволяет осуществить проверку получившихся линий-границ на соответствие различным геометрическим контурам. Данную проверку можно осуществить, используя метод Хафа (Hough Transform). Подробнее о методе анализа контуров и методе Хафа описано в [3, 8] (рис. 5).

Достоинства: возможность применения в изображениях, когда объект сегментации существенным образом не выделяется на фоне по цвету или имеет сложную раскраску, высокая адаптивность с помощью регулировки параметров и низкая чувствительность к шуму.

Недостатком является большая вычислительная сложность, а также сложность применения в случае, когда фон не является монотонным.

Предлагаемый алгоритм сегментации. Предлагаемый способ решения данной проблемы заключается в использовании маски исходного изображения. Блок-схема предлагаемого алгоритма представлена на рис. 6.

Загрузка изображения и маски.

Реализуем данный алгоритм сегментации в системе Matlab. В качестве исходного изображения будет выступать отсканированный фрагмент тетрадного листа с рукописным текстом, представленный на рис. 7а. Под маской в данном случае понимается исходное изображение без рукописного текста (рис. 7б).

Поиск различий маски и изображения.

На данном этапе необходимо осуществить поиск различий исходного изображения и маски путем алгебраического вычитания:

$$Diff(i, j) = Image(i, j) - Mask(i, j). \quad (11)$$

Устранение шума.

Так как маска не идеально повторяет фон исходного изображения, в связи с этим на полученном изображении наблюдается большое количество шумов и линий, не предназначенных для сегментации. Для фильтрации полученного изображения воспользуемся усредняющим (медиальным) фильтром, который выполняет среднюю фильтрацию изображения в двух измерениях, где каждый выходной пиксель содержит среднее значение в 3×3 окружении соответствующего пикселя во входном изображении [1].

Для устранения оставшихся шумов воспользуемся операцией эрозии. Операция максимальной эрозии изображения заключается в преобразовании локального максимума евклидова расстояния на изображении [1].

Формирование новой маски изображения.

Полученное изображение имеет разрывы в местах прохождения линий клеток. Одним из вариантов решения проблемы является использование функций интерполяции фона, как описано в статье [9].

В данной статье рассматривается наиболее простой метод решения данной проблемы, заключающийся в использовании получившегося изображения в качестве новой маски.

Для формирования новой маски воспользуемся операцией свертки полученного изображения с матричным фильтром наращивания размерности 5×5:

$$F(i, j) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot Diff(i, j).$$

Формирование готового изображения.

Далее, производится поэлементное умножение элементов, полученной маски, на исходное изображение. В связи с тем, что сформированная маска несколько шире необходимого нам фрагмента, то в результате умножения на полученном изображении будут наблюдаться части элементов фона исходного изображения.

Данный эффект можно устранить, профильтровав полученное изображение медиальным фильтром.

В заключение необходимо осуществить вычисление дополнения изображения для восстановления цветов (фон — белый, текст — черный) (рис. 8).

Сравнение результатов работы алгоритмов сегментации.

1. Цветовой анализ изображений.

Разложим исходное изображение (рис. 7а) на RGB-составляющие. На основании произведенного разложения видно, что написанный текст имеет ярко выраженную синюю составляющую. Произведем цветовую селекцию с помощью приложения Matlab «Color Thresholder» [10] (рис. 9).

С помощью бегунков подбираем параметры так, чтобы остался только рукописный текст. После устранения шумов медиальным фильтром получаем готовое изображение (рис. 10).

2. Бинарная сегментация изображений методом фиксации уровня.

Применим бинарную сегментацию изображений методом фиксации уровня. В данном случае можно ограничиться применением в качестве порога плоскость с заданным уровнем, поскольку сегментируемый фрагмент хорошо различается на общем фоне изображения, как показано на рис. 11.

С целью улучшения качества фрагмента сегментации, уровень выставяемого порога выбирается таким образом, чтобы обеспечить максимальную сохранность сегментируемого фрагмента. В связи с этим на получившемся изображении наблюдаются шумы. Для устранения шумов воспользуемся медиальным фильтром (рис. 12).

3. Сравнение результатов работы алгоритмов сегментации.

В качестве эталона для последующего анализа воспользуемся алгоритмом обработки изображений, используемым в методе цветовой селекции, поскольку данный метод показал наилучшее качество сегментируемого фрагмента.

Произведем сравнительный анализ работы двух алгоритмов. Результат подсчёта идентичных пикселей, т.е. пикселей, имеющих идентичное значение и находящихся на идентичных позициях, показал, что сравниваемые изображения совпадают на 99,02 %. Таким образом, предложенный метод показывает результат, сопоставимый с уже зарекомендованными методами сегментации.

*В тащах юга тил боте цитрус?
да, но фримышвэй ядешмиер!*

Рис. 8. Конечный результат преобразований

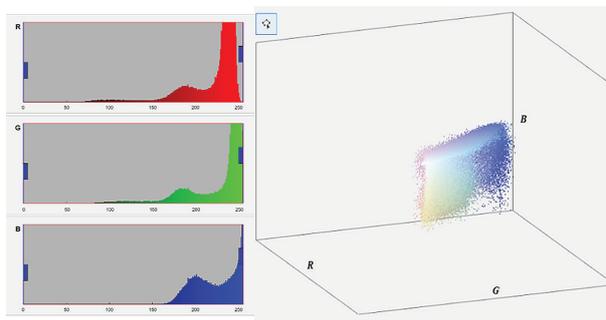


Рис. 9. Гистограммы и диаграмма рассеяния исходного изображения

*В тащах юга тил боте цитрус?
да, но фримышвэй ядешмиер!*

Рис. 10. Готовое изображение

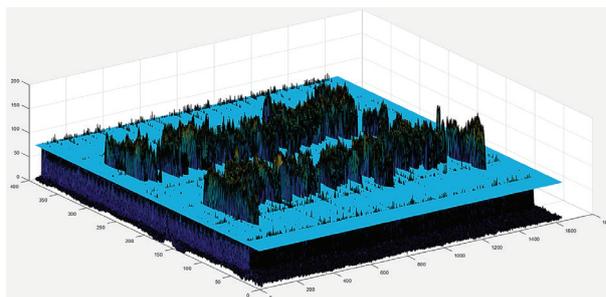


Рис. 11. Результат сегментации в 3D-представлении

*В тащах юга тил боте цитрус?
да, но фримышвэй ядешмиер!*

Рис. 12. Результат применения метода бинарной сегментации изображений с фиксацией уровня

Заключение. Предложен алгоритм сегментации текста, основанный на использовании маски исходного изображения. Данный метод позволяет существенно упростить процесс сегментации текста на изображениях с немонокромным фоном. Предложенный метод позволяет работать с RGB изображениями; с изображениями в оттенках серого, имеющими низкую контрастность; а также с бинарными изображениями. Кроме того, процесс обработки изображений может осуществляться в автоматическом режиме.

На основании проведенного сравнительного анализа можно сделать вывод, что изображения, полученные в результате работы алгоритмов, совпадают

более чем на 99 %. Это говорит о том, что предлагаемый алгоритм показывает результат, сопоставимый с уже хорошо зарекомендовавшими себя методами сегментации.

Из недостатков данного метода можно выделить необходимость наличия маски исходного изображения, большое количество операций для получения нужного фрагмента изображения, а также чувствительность к повреждению сегментируемого фрагмента.

Библиографический список

1. Список функций Image Processing Toolbox. URL: <https://hub.exponenta.ru/post/spisok-funktsiy-image-processing-toolbox152#bwm0-grh> (дата обращения: 20.03.2020).
2. Бинарная сегментация изображений методом фиксации уровня (Level set method). URL: <https://habr.com/ru/post/332692/> (дата обращения: 20.03.2020).
3. Горитов А. Н., Яковленко С. И. Выделение на изображении низкого разрешения параметрически задаваемых объектов // Доклады Томского государственного университета систем управления и радиоэлектроники. 2017. № 2. С. 88–90. DOI: 10.21293/1818-0442-2017-20-2-88-90.
4. Sanny J. A Computational Approach to Edge Detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986. No. 6. P. 679–698. DOI: 10.1109/TPAMI.1986.4767851.
5. Rashmi, Kumar M., Saxena R. Algorithm and technique on various edge detection: A survey // Signal & Image Processing: An International Journal (SIPIJ). 2013. No. 3. P. 65–75. DOI: 10.5121/sipij.2013.4306.
6. Saini S., Kasliwal B., Bhatia S. Comparative Study of Image Edge Detection Algorithms. 2013. URL: <https://arxiv.org/abs/1311.4963> (дата обращения: 20.03.2020).
7. Prasad R., Sresh S. A Review on Edge Detection Algorithms // International Journal of Mechanical Engineering and Computer Applications. 2016. No. 1. P. 7–11.

8. Гайдуков Н. П., Савкова Е. О. Применение преобразования Хафа для распознавания текста // Информатика и компьютерные технологии. 2012. С. 136–140. URL: <http://ea.donntu.org/handle/123456789/15585> (дата обращения: 20.03.2020).

9. Firas A. Jassim. Image Inpainting by Kriging Interpolation Technique // World of Computer Science and Information Technology Journal (WCSIT). 2013. Vol. 3, no. 5. P. 91–96.

10. Создание маски изображения, используя Color Treshold. URL: <https://docs.exponenta.ru/R2019b/images/image-segmentation-using-the-color-thesholder-app.html> (дата обращения: 20.03.2020).

КОБЕНКО Вадим Юрьевич, доктор технических наук, доцент (Россия), профессор кафедры «Электроника».

SPIN-код: 9011-1591

AuthorID (SCOPUS): 55674672100

ResearcherID: D-7342-2013

ФРОЛОВ Святослав Олегович, магистрант гр. ЭНМ-191 факультета элитного образования и магистратуры.

ТАЛАЛАЕВ Владимир Ильич, магистрант гр. ЭНМ-191 факультета элитного образования и магистратуры.

Адрес для переписки: svyatoslav.frolov.97@mail.ru

Для цитирования

Кобенко В. Ю., Фролов С. О., Талалаев В. И. Разработка алгоритма сегментации рукописного текста с использованием маски // Омский научный вестник. 2020. № 4 (172). С. 47–52. DOI: 10.25206/1813-8225-2020-172-47-52.

Статья поступила в редакцию 27.05.2020 г.

© В. Ю. Кобенко, С. О. Фролов, В. И. Талалаев