

УДК 004.932

DOI: 10.15593/2224-9397/2020.3.09

**В.Р. Зайникова<sup>1</sup>, С.А. Зыкин<sup>1</sup>, Р.А. Файзрахманов<sup>2</sup>**

<sup>1</sup>Лысьвенский филиал Пермский национальный исследовательский политехнический университет, Лысьва, Россия

<sup>2</sup>Пермский национальный исследовательский политехнический университет, Пермь, Россия

## **АЛГОРИТМ НАХОЖДЕНИЯ КОНТУРА ДЕЛОВОГО ОСТАТКА НЕСТАНДАРТНОЙ ФОРМЫ ПО ЦИФРОВОЙ ФОТОГРАФИИ СРЕДСТВАМИ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON НА ОСНОВЕ БИБЛИОТЕКИ КОМПЬЮТЕРНОГО ЗРЕНИЯ OPENCV**

В современном мире в связи со стремительным развитием технологий и ростом их объемов возникает необходимость в своевременном извлечении информации, содержащейся на цифровых изображениях. Обнаружение объектов на фотографиях привлекает большой интерес благодаря широкому использованию их в повседневной жизни. Технология была обусловлена увеличением вычислительной мощности, доступной в программном и аппаратном обеспечении. В научной сфере достаточно актуальна задача нахождения и распознавания объектов на изображении, однако она сконцентрирована на распознавании их контуров без приведения контура к реальным размерам и не решает задачу в полном объеме. **Цель исследования:** разработать алгоритм нахождения контура делового остатка нестандартной формы по цифровой фотографии. **Методы:** исправив перспективу цифровой фотографии, для получения вида «сверху-вниз» найден контур листа А4 для приведения изображения к реальным размерам. После получения изображения с реальными размерами найден контур делового остатка. **Результаты:** функции исправления перспективы позволяют получить изображение вида «сверху-вниз» для получения точных контуров. Используя метод сегментации Саппу, были найдены все контуры в изображении. Аппроксимируя контуры изображения, найден контур листа А4 с его координатами. Вычислив евклидово расстояние между координатами и сравнив с известными размерами листа А4, изображение преобразовано к точному размеру, где 1 пиксель в изображении равен 1 мм. Выполнив ряд морфологических операций, получен единый замкнутый контур с реальными размерами, который содержит все координаты делового остатка. Проанализирована работоспособность алгоритма и выявлены основные плюсы алгоритма. **Практическая значимость:** разработанный алгоритм позволяет сократить время, которое тратится при ручных замерах, а контур, полученный в результате, получает реальные данные делового остатка.

**Ключевые слова:** Python, деловой остаток, OpenCV, NumPy, Image, обработка изображений, компьютерное зрение, исправление перспективы, распознавание образов, контур, сегментация.

**V.R. Zaynikova<sup>1</sup>, S.A. Zykin<sup>1</sup>, R.A. Fayzrahmanov<sup>2</sup>**

Lysva branch Perm National Research Polytechnic University, Lysva,  
Russian Federation

Perm National Research Polytechnic University, Perm, Russian Federation

## **ALGORITHM FOR FINDING THE CONTOUR OF A BUSINESS REMAINDER OF A NON-STANDARD FORM USING DIGITAL PHOTOGRAPHY USING THE PYTHON PROGRAMMING LANGUAGE BASED ON THE OPENCV COMPUTER VISION LIBRARY**

In the modern world, due to the rapid development of technologies and the growth of their volumes, there is a need for timely search for information contained in digital images. The discovery of objects in photos is of great interest due to their wider use in everyday life. This technology was driven by an increase in computing power available in software and hardware. In the scientific field, the problem of finding and recognizing objects in an image is quite relevant, but they are focused on recognizing their contours, without reducing the contour to real dimensions, and do not solve the problem in full.

**Purpose:** develop an algorithm for finding the contour of the business balance, non-standard form, from a digital photo. **Methods:** after correcting the perspective of the digital photo, to get a top - down view, the contour of the A4 sheet was found to bring the image to real dimensions. After getting the image with the actual dimensions, the contour of the business remainder is found. **Results:** the perspective correction functions allow you to get a top-down view of the image to get accurate contours. Using the Canny segmentation method, all the contours in the image were found. Approximating the image contours, the contour of the A4 sheet with its coordinates is found. After calculating the Euclidean distance between the coordinates and comparing it with the known dimensions of the A4 sheet, the image is converted to the exact size, where 1 pixel in the image is equal to 1 mm. After performing a number of morphological operations, a single closed contour with real dimensions is obtained, which contains all the coordinates of the business remainder. The efficiency of the algorithm is analyzed and the main advantages of the algorithm are revealed. **Practical relevance:** the developed algorithm allows you to reduce the time spent on manual measurements, and the resulting contour receives real data on the business balance.

**Keywords:** python, business balance, OpenCV, NumPy, Image, image processing, computer vision, perspective correction, pattern recognition, contour, segmentation.

**Введение.** В современном мире в связи со стремительным развитием технологий и ростом их объемов возникает необходимость в своевременном извлечении информации, содержащейся на цифровых изображениях, с целью использования этой информации в различных областях человеческой деятельности.

Одной из этих областей является производство, в котором после использования листового материала остаются деловые остатки.

Деловыми остатками (ДО) называют остатки материала от раскроя, которые складывают и в дальнейшем используют для производства деталей. Их рассматривают как листы меньшего размера.

Вопрос использования деловых остатков актуален в настоящее время, поскольку это решает ряд задач производства по хранению, утилизации и повторному использованию остатков. Для повторного использования ДО их необходимо измерить. Замеры вручную могут занять много времени, а сфотографировав остатки и взяв всю необходимую информацию о размерах из цифровой фотографии, можно значительно сократить время работы на производстве.

В научной и производственной сфере достаточно актуальна задача нахождения и распознавания объектов на изображении. В основном рассмотрены методы нахождения текстовой информации [1] и биометрических персональных данных [2–4]. Авторы многократно рассматривают алгоритмы распознавания лиц [2, 4] и нахождения контурной границы [5–6].

**Цель:** разработать алгоритм нахождения контура ДО нестандартной формы по цифровой фотографии. В отличие от вышеперечисленных методов конечным результатом алгоритма является контур с реальными размерами ДО.

Для реализации поставленной цели необходимо выполнить следующие задачи:

- 1) проанализировать методы существующих решений;
- 2) разработать модель нахождения изображения с реальными размерами;
- 3) реализовать нахождение контура делового остатка;
- 4) провести эксперименты времени работы алгоритма.

Данный алгоритм разработан средствами языка программирования Python на основе библиотек OpenCV [7], NumPy и Image.

**1. Анализ существующих решений нахождения деловых остатков.** Используя фотографию ДО нестандартной формы, необходимо получить контур, содержащий в себе  $n$  точек границ контура. ДО на фотографии может находиться под любым углом (рис. 1).

Для приведения изображения к реальным размерам на ДО положен лист формата А4.

Главное условие – ДО должен поместиться на фотографии. Соблюдение этого условия влияет на точность измерения. Располагать лист А4 необходимо вертикально на поверхности ДО (см. рис. 1).

Фон фотографии не должен совпадать с цветом ДО: если цвет остатка темный, то фон должен быть светлым, и, наоборот, границы листа А4 должны хорошо выделяться на изображении.

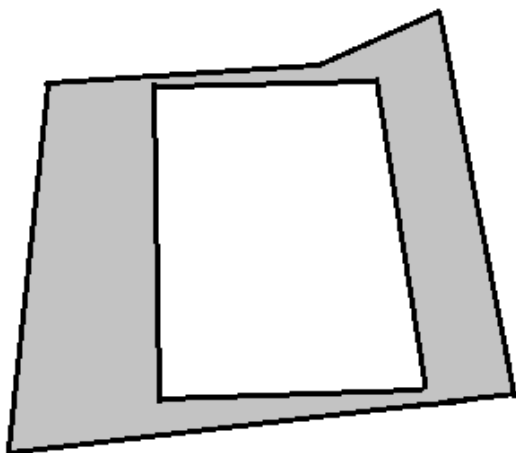


Рис. 1. Пример расположения ДО и листа формата А4

Подобная задача была рассмотрена в работе [8]. В ней рассмотрен алгоритм контурной сегментации и распознавания образцов объектов. В данном исследовании над изображением проведена предварительная обработка, далее выделены контуры для формирования словаря признаков, после чего шло сопоставление образцов с эталонами из базы данных. Данный метод лишь частично подходит для реализации алгоритма, так как в нем рассмотрено только нахождение контура без его приведения к реальным размерам.

Задача обнаружения объекта включает в себя распознавание объекта, присутствующего в изображении, а также локализацию, чтобы вычислить ограничивающую рамку вокруг него. Эта проблема в области компьютерного зрения изучена в работах [9, 10, 11, 12]. В этих методах обычно используется информация о цветной камере. Во многих средах методы, основанные на сверточных нейронных сетях (CNNs), демонстрируют очень высокую точность обнаружения объектов. Тем не менее эти методы уязвимы к ошибкам из-за качества цветных изображений, освещения и отсутствия цветового дифференцирования для обнаружения объекта.

В статье [13] для нахождения объектов и их размеров на изображении собираются мультимодальные данные с помощью пользовательского приложения Android. Также выполняется предварительная обработка данных для выполнения пространственно-временного выравнивания информации датчика о камере. Пространственное выравнивание требуется из-за разницы в положении камеры на мобильном

устройстве. Временное выравнивание требуется из-за временного промежутка между сбором данных камеры. Таким образом, учитывается разница в положении устройства в разных случаях, когда собирается информация. После этого выравнивания получают информацию о глубине для некоторых пикселей цветных изображений камеры. Для получения информации о глубине для всех пикселей выполняется билинейная интерполяция и используется  $k-d$ -дерево для быстрого поиска ближайшего пикселя с информацией о глубине. Реализуется несколько методов расчета размеров интересующего объекта в метрах и показывается работоспособность в различных средах как внутри помещения, так и снаружи.

Основными проблемами данного метода являются уязвимость к ошибкам из-за качества цветных изображений камеры, освещение и отсутствие цветового дифференцирования для обнаружения объекта. Также он имеет большую погрешность в расчетах. Так, задокументировано, что датчики, используемые в [13], демонстрируют погрешности, равные 4 см. Даже при более высокой точности, используя лидарные датчики, погрешность равна 2 см.

Таким образом, сформирована задача нахождения контура ДО. Рассмотрены существующие модели решения задачи. Выявлены их основные недостатки.

**2. Модель нахождения контура делового остатка.** С учетом рассмотренных выше моделей сформирован порядок действий для реализации алгоритма (рис. 2).

Исправление перспективы изображения необходимо для получения вида «сверху вниз» (рис. 3) из цифровой фотографии. Для исправления перспективы использована библиотека алгоритмов обработки изображений, компьютерного зрения, а также численных алгоритмов общего назначения OpenCV и библиотека NumPy. NumPy – это одна из основных Python-библиотек с поддержкой массивов. Изображение представляет собой стандартный массив NumPy, содержащий пиксели точек данных. Таким образом, при выполнении основных NumPy-операций можно изменять пиксельные значения изображения.

Для исправления перспективы ДО вписан в прямоугольник  $ABCD$  (см. рис. 3), и при помощи его вершин вычислена матрица преобразования перспективы. После нахождения матрицы применяется преобразование перспективы к изображению.

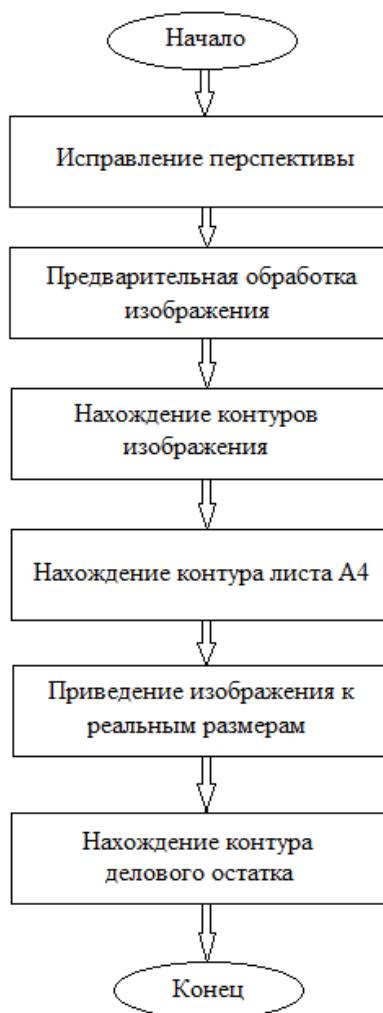


Рис. 2. Алгоритм нахождения контура ДО

Используя функцию `cv2.getPerspectiveTransform (src, dst)`, найдена матрица преобразования перспективы. Эта функция требует двух аргументов, списка координат вершин описывающего четырехугольника `src`, и `dst`, который является списком преобразованных точек, т.е. `cv2.getPerspectiveTransform` возвращает матрицу  $M$ , которая вычисляет  $3 \times 3$  матрицу аффинного преобразования так, что:

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \text{map}_{matrix} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}, \quad (1)$$

где  $\text{dst}(i) = (x'_i, y'_i)$ ,  $\text{src}(i) = (x_i, y_i)$ ,  $j = 0, 1, 2, 3$ .

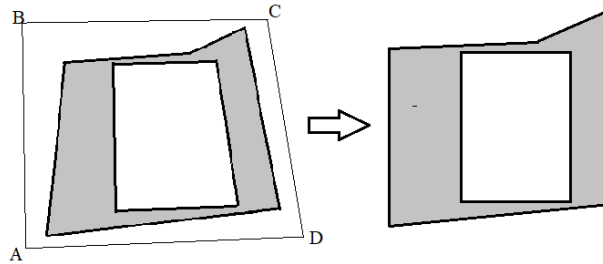


Рис. 3. Преобразование перспективы

Матрица применяется в функции `cv2.warpPerspective (src, M, dsize) → dst`, результатом которой является желаемое изображение вида «сверху вниз». Эта функция требует три аргумента, входное изображение `src`, матрицу преобразования перспективы `M` и размеры выходного изображения `dsize`. Функция `warpPerspective` преобразует исходное изображение с помощью матрицы:

$$\text{dst}(x, y) = \text{src} \left( \frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}}, \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}} \right). \quad (2)$$

Получив изображение с исправленной перспективой, необходимо найти контур листа А4. Перед нахождением контуров необходимо провести предварительную обработку изображения. Задачей предварительной обработки является эффективное подавление шума при сохранении важных для последующего распознавания элементов изображения [14]. Использование возможностей обработки изображений значительно расширилось за последние годы [15], охватывая такие области, как криминалистика и медицина [16].

Для подавления шума использован фильтр Гаусса [17] (рис. 4).

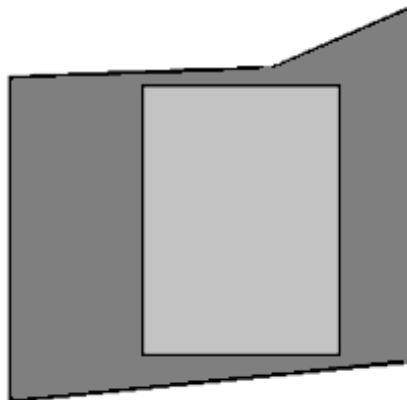


Рис. 4. Отфильтрованное изображение

Для этого применим функцию `cv2.GaussianBlur(src, ksize, sigmaX) → dst`. Первый параметр `src` – это массив фотографии, далее `ksize` – размер ядра фильтра, чем больше значение, тем больше размытие (значения должны быть нечетными). Далее идет `sigmaX`, стандартное отклонение по оси `X`. В нашем случае размер ядра размытия (5, 5).

Различное множество алгоритмов определения объектов рассматривается в работе [18]:

- 1) примитивы Хаара;
- 2) Adaptive Boosting;
- 3) механизм AdaBoost;
- 4) метод Виолы–Джонса;
- 5) алгоритм Хафа;
- 6) алгоритм сравнения с шаблоном;
- 7) сегментация изображения.

Выделение контуров является одним из методов сегментации изображений [14, 19–21]. Под сегментацией принято понимать разбиение изображения на группы пикселей по близким значениям некоторого показателя [22]. Большое распространение получили алгоритмы, основанные на выделении контурных границ с помощью вычисления первой производной (операторы Собеля и др. [23, 24]) или второй производной (оператор Лапласа [23]). В этом случае осуществляется поиск границ, которые в дальнейшем объединяются в контуры. В исследовании, приведённом в [25], показано, что методы пограничной обработки уступают по качеству выделения контуров методу, предложенному Canny [17].

Детектор границ Canny включает следующую последовательность действий. После фильтрации помех на изображении гауссовым фильтром с заданным параметром сглаживания  $\sigma$  в каждой точке изображения вычисляется градиент, который характеризуется значением модуля:

$$g(x, y) = [G_x^2 + G_y^2]^{\frac{1}{2}} \quad (3)$$

и направлением:

$$\alpha(x, y) = \arctg \left[ \frac{G_x}{G_y} \right]. \quad (4)$$

После получения величины градиента и направления производится полный анализ изображения, чтобы удалить любые нежелательные пиксели, которые не могут составлять ребро. Результатом является двоичное изображение с «тонкими краями» (рис. 5).



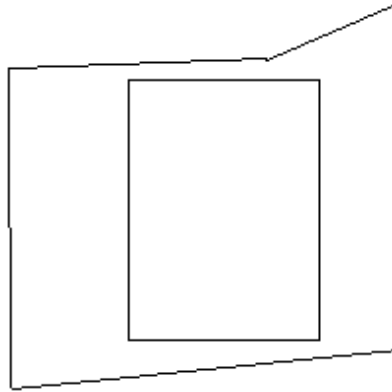


Рис. 5. Детектор границ Canny

Чтобы применить детектор границ Canny, необходимо использовать функцию `cv2.Canny(image, threshold1, threshold2) → edges`. Функция имеет 3 аргумента, входное изображение `image`, одноканальное изображение для обработки (градации серого), `threshold1` – порог минимума, `threshold2` – порог максимума.

Для перевода изображения в одноканальное используется функция `cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)`, она принимает два аргумента и отдает итоговое изображение. Первый аргумент – исходное изображение, второй – направление трансформации цветового пространства.

Получив двоичное изображение, найдены координаты всех контуров в изображении функцией `cv2.findContours (image, mode, method) → contours, hierarchy`. Эта функция требует трех аргументов, `image`-изображения, полученного детектором границ Канни, `Mode`-режима группировки контуров. Использован режим `CV_RETR_LIST` для извлечения контуров без установления каких-либо иерархических связей. `Method` – метод упаковки контуров. Использован метод `CV_CHAIN_APPROX_SIMPLE`, который склеивает все горизонтальные, вертикальные и диагональные контуры и оставляет только их конечные точки. Выходными данными функции являются `contours` и `hierarchy`. В `contours` каждый контур хранится в виде вектора точек, а `hierarchy` – необязательный выходной вектор, содержащий информацию о топологии изображения.

Для проверки, является ли контур листом А4 или нет, выполнен цикл по каждому контуру. Для каждого из контуров вычислен

периметр, используя функцию `cv2.arcLength`, далее аппроксимируем (сглаживаем) контур, используя функцию `cv2.approxPolyDP`.

Причина, по которой используется аппроксимация контура, заключается в том, что он может не быть идеальным прямоугольником. Из-за зашумления и теней на фото вероятность того, что у листа будет ровно 4 вершины, невелика. Аппроксимируя контур, эта проблема решена.

В нашем случае функции выглядят так:

```
peri = cv2.arcLength(contours, True)
```

```
approx = cv2.approxPolyDP(contours, 0.02 * peri, True)
```

Функция `cv2.approxPolyDP` принимает аргументы контур `contours` и `0.02 * peri` параметр, задающий точность аппроксимации. Результат выполнения `approx`, вектор координат контура. Если количество координат равно четырем, то контур является четырехугольником (рис. 6).

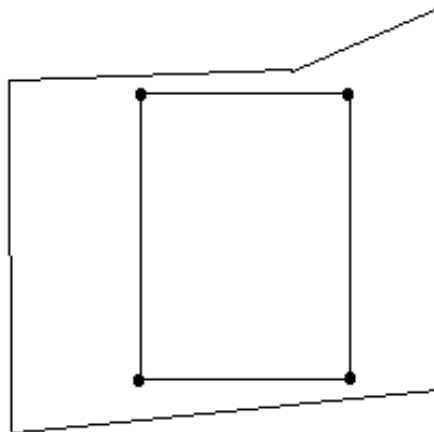


Рис. 6. Координаты листа А4

После нахождения четырехугольника по его координатам найдены длины его сторон, вычислив евклидово расстояние между точками  $i$  и  $j$ :

$$d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (5)$$

Рассчитав ширину и длину листа А4 на изображении, найдем, во сколько раз необходимо уменьшить или увеличить ширину и длину изображения относительно реальных размеров листа А4, чтобы пиксельное значение ширины и длины изображения в миллиметрах соот-

ветствовало реальным размерам. Вычислив новые значения ширины и длины изображения, с помощью библиотеки Image изменим размеры изображения на новые, используя функцию Image.resize(size). Эта функция требует один аргумент size, запрошенный размер в пикселях, как 2-кортеж (ширина, высота). Результатом выполнения функции является изображение заданного размера.

Для проверки, корректно ли был найден лист А4 на изображении, повторно находим его координаты на новом изображении с реальными размерами, используя описанный выше порядок действий.

Далее находим площадь контура листа А4, используя функцию cv2.contourArea(contour). Она требует один аргумент contour, это входной вектор вершин контура, в нашем случае – вершин листа А4. Если лист найден корректно, то площадь удовлетворяет условиям:

$$\begin{cases} S \geq 62338, \\ S \leq 62402. \end{cases} \quad (6)$$

Площадь листа А4 равна 62370 мм<sup>2</sup>, согласно условию площадь может иметь отклонение на 32 мм<sup>2</sup>,

$$p = \frac{100 \cdot 32}{62370} = 0,05 \%, \quad (7)$$

таким образом, погрешность измерения не будет превышать ±0,05 %. Данная погрешность по ГОСТу [26] соответствует измерению штангенциркулем.

Для нахождения контура ДО используем функцию cv2.Canny, чтобы получить 8-битное одноканальное изображение.

Однако на изображении между вертикальными и горизонтальными полосками контура может оказаться пустое пространство. Чтобы его закрыть, произведем ряд простых морфологических операций [27].

Начнем с создания прямоугольника между полосками с помощью функции cv2.getStructuringElement(shape, ksize) → kernel. Функция требует двух аргументов – shape и ksize. Shape – форма элемента, в нашем случае четырехугольник MORPH\_RECT. Размер структурирующего элемента – ksize. Функция строит и возвращает структурирующий элемент kernel, который передан в функцию morphologyEx().

Функция cv2.morphologyEx(src, op, kernel) → dst¶¶ производит морфологическую операцию, применив ядро к двоичному изображению, замазывая пространство между полосками. Функции необходимо три

аргумента: `src`, `op` и `kernel`. `Src` – двоичное изображение, `op` – тип морфологической операции, в нашем случае `MORPH_CLOSE`, для закрытия пустого пространства, `kernel` – структурирующий элемент:

$$\text{dst}(x, y) = \text{close}(\text{str}, \text{element}) = \text{erode}(\text{dilate}(\text{str}, \text{element})). \quad (8)$$

После морфологических операций над изображением находим контур делового остатка функцией `cv2.findContours()`, используя режим группировки **RETR\_EXTERNAL** для извлечения внешних контуров и метод `CV_CHAIN_APPROX_SIMPLE`.

Чтобы получить контур делового остатка, отсортируем найденные контуры с помощью функции `sorted()`, которая возвращает новый отсортированный список. С помощью параметра `key` можно указывать, как именно выполнять сортировку. Флаг `reverse` позволяет управлять порядком сортировки. По умолчанию сортировка будет по возрастанию элементов. В нашем случае параметр `key` равен `cv2.contourArea`, выполняя сортировку по площади контура.

Отсортировав контуры, сохраняем только самый большой из них, который содержит контур ДО (рис. 7).

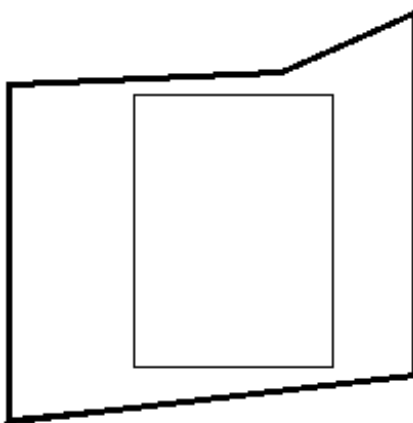


Рис. 7. Контур ДО

Таким образом, применяя функции `cv2.getPerspectiveTransform`, `cv2.warpPerspective`, получено изображение с исправленной перспективой. При помощи листа А4 получили изображение, которое соответствует реальным размерам, где 1 пиксель соответствует 1 мм. Обработав изображение детектором границ `Canny` и морфологической операцией `morphologyEx()`, найдены контуры изображения. Отсортировав их и

сохранив только самый большой, нашли контур ДО, который так же содержит в себе его координаты.

**Результаты проведения эксперимента, применения модели нахождения ДО.** В разработанном алгоритме использован метод обработки изображения для получения контура из работы [8]. Но он был улучшен за счет исправления перспективы, выделяет более точный контур объекта. В отличие от метода, рассмотренного в статье [12], точность нахождения ДО значительно выше, так как погрешность нахождения размеров разработанной модели равна 0,05 %, а в методе превышает 1 %. Главным недостатком является то, что в методе не выделяется контур, объекты вписывают в прямоугольник и выводят его размер, в результате получая только длину и ширину ДО. Результатом разработанного алгоритма является контур ДО с его координатами.

Кроме того, обнаружение объектов на основе только цветных изображений может обеспечить относительный размер в пикселях, который не имеет никакого отношения к реальным размерам изображения. Игрушечный автомобиль рядом с камерой может казаться таким же большим, как настоящий автомобиль, который находится далеко от камеры.

Трудоёмкость алгоритма и метода [8] различается в том, что в разработанной модели объект на фотографии может быть изображен под любым углом, а в методе [8] для выделения более точного контура нужно выполнить ряд действий для получения изображения объекта вида «сверху вниз». Скорость работы алгоритма не критично зависит от размера изображения (таблица).

Время работы программы

Размер изображения	Время, с
434×422	0,098
612×582	0,106
1024×992	0,113
2000×1900	0,125
3456×4608	0,146

Таким образом, проанализирована работоспособность алгоритма. Выявлены основные плюсы алгоритма в отличие от рассмотренных методов.

**Выводы.** В работе рассмотрен алгоритм для нахождения контура ДО нестандартной формы по цифровой фотографии. В отличие от алгоритмов распознавания объектов, которые находят граничные точки

контура, разработанный алгоритм имеет улучшенные функции для нахождения реальных размеров ДО.

Функция исправления перспективы позволяет получить изображение вида «сверху вниз». Контуры, полученные с этого преобразования, точнее передают форму ДО, повышая таким образом точность для дальнейшего использования листового материала.

Используя метод сегментации, который играет важную роль в системах компьютерного зрения для решения задач, связанных с распознаванием и выделением объектов, были найдены все контуры в изображении. Используя аппроксимацию для нахождения листа А4, найдены его координаты. Вычислив евклидово расстояние между координатами и сравнив с известными размерами листа А4, преобразовали изображение к точному размеру, где 1 пиксель в изображении равен 1 мм.

Выполнив ряд морфологических операций, получен единый замкнутый контур с точными размерами, который содержит все координаты ДО.

Полученные данные возможно внести в базу данных для дальнейшего использования. Разработанный алгоритм позволяет сократить время, которое тратится при ручных замерах, а контур, полученный в результате, в отличие от стандартных алгоритмов нахождения объектов передает точные данные ДО.

Перспективой для дальнейших исследований является доработка алгоритма, чтобы реализовать возможность выбирать шаблон для приведения изображения к реальным размерам, используя не только лист А4. Эта функция позволит работать с ДО любых размеров.

### **Библиографический список**

1. Нанавова Т.А. Алгоритм извлечения текста из видео с использованием библиотеки компьютерного зрения OPENCV // Ростовский научный журнал. – 2016. – № 7. – С. 21–40.
2. Михалин Д.А., Белов Ю.С. Применение библиотеки OPENCV для задачи распознавания лиц // Актуальные проблемы авиации и космонавтики. – 2016. – № 12. – С. 558–559.
3. Назарова Т.Ю., Лавров Д.Н. Компьютерное моделирование идентификации личности по радужной оболочке глаза на основе OPENCV // Математические структуры и моделирование. – 2014. – № 1(29). – С. 43–64.

4. Тихонова Т.С., Белов Ю.С. Основные подходы к отслеживанию и распознаванию лица // Наука, техника и образование. – 2016. – № 2(6). – С. 111–115.
5. Lienhart R., Kuranov A., Pisarevsky V. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection: MRL Technical Report, 2002.
6. Чабан Л.Н. Методы и алгоритмы распознавания образов в автоматизированном дешифрировании данных дистанционного зондирования: учеб. пособие. – М.: Изд-во МИИГАиК, 2016. – 94 с.
7. Bradsky G., Kaehler A. Learning OpenCV. – O'Reilly, 2008. – P. 571.
8. Нгуен К.М., Колочкин В.Я. Алгоритмы контурной сегментации и распознавания образов объектов систем технического зрения // наука и образование: научное издание МГТУ им. Н.Э. Баумана. – 2013. – № 4. – С. 187–200.
9. You only look once: Unified, real-time object detection / J. Redmon, S.K. Divvala, R.B. Girshick, A. Farhadi // CoRR. – 2015. – abs/1506.02640.
10. Rich feature hierarchies for accurate object detection and semantic segmentation / R.B. Girshick, J. Donahue, T. Darrell, J. Malik // CoRR. – 2013. – abs/1311.2524.
11. Girshick R.B. Fast R-CNN // CoRR. – 2015. – abs/1504.08083.
12. Faster R-CNN: towards real-time object detection with region proposal networks / S. Ren, K. He, R.B. Girshick, J. Sun // CoRR. – 2015. – abs/1506.01497.
13. Estimating the Object Size from Static 2D Image / O. Kainz, M.W. Horecny, F. Jakab, D. Cymbalak // CoRR. – 2015. – P. 15–17.
14. Яне Б. Цифровая обработка изображений: пер. с англ. – М.: Техносфера, 2007. – 584 с.
15. Chang P.L., Teng W.G. Exploiting the self-organizing map for medical image segmentation in Computer-Based Medical Systems // Twentieth IEEE International Symposium on. – 2007. – P. 281288.
16. Zhou Z.J., Wu H. Digital Image Processing. Part 1. Ventus Publishing ApS. – 2010.
17. Canny J.E. A computational approach to edge detection // IEEE Trans Pattern Analysis and Machine Intelligence. – 1986. – No. 8. – P. 679–698.

18. Алгоритмы распознавания объектов / А. А. Цветков, Д. К. Шорох, М. Г. Зубарева, С. В. Юрсков, А. В., Шуклин А. Л. Хамуш, И. Б. Ануфриев // Технические науки: проблемы и перспективы. – 2016. – С. 20–28.

19. Шитова О.В., Пухляк А.Н., Дроб Е.М. Анализ методов сегментации текстурных областей изображений в системах обработки изображений // Научные ведомости Белгород. гос. ун-та. Сер. Экономика. Информатика. – 2014. – № 8(179). – С. 182–188.

20. Родзин С.И., Родзина О.Н., Эль-Хатиб С.А. Гибридный муравьиный алгоритм сегментации медицинских изображений // Вестник Чувашского ун-та. – 2017. – № 3. – С. 262–272.

21. Комков В.С., Дюдин М.В. Метод сегментации растровых изображений на основе выбора оптимальных направлений обработки пикселей изображения // Российская наука и образование сегодня: проблемы и перспективы. – 2015. – № 2(5). – С. 151–154.

22. Шапиро Л., Стокман Дж. Компьютерное зрение. – М.: Бином, 2006. – 752 с.

23. Гонзалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2006. – 1072 с.

24. Rosenfeld A., Как А.С. Digital Picture Processing. – New York: Academic press, 1976.

25. Стругайло В.В. Обзор методов фильтрации и сегментации цифровых изображений // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. – 2012. – № 5. – С. 17.

26. ГОСТ 166-89. Межгосударственный стандарт. Штангенциркули. Технические условия // Доступ из справ.-правовой системы КонсультантПлюс.

27. Власов А.А. Метод контурной сегментации канни с использованием морфологических операций // Решетневские чтения. – 2010. – № 2. – С. 479.

### **References**

1. Nanavova T.A. Algorithm izvlecheniia teksta iz video s ispol'zovaniem biblioteki komp'iuternogo zreniia OPENCV [Algorithm for extracting text from a video using a library computer vision opencv]. *Rostovskii nauchnyi zhurnal*, 2016, no. 7, pp. 21-40.

2. Mikhalin D.A., Belov Iu.S. Primenenie biblioteki OPENCV dlia zadachi raspoznavaniia lits [Using the OPENCV library for face recog-



dition tasks]. *Aktual'nye problemy aviatsii i kosmonavtiki*, 2016, no. 12, pp. 558-559.

3. Nazarova T.Iu., Lavrov D.N. Komp'yuternoe modelirovanie identifikatsii lichnosti po raduzhnoi obolochke glaza na osnove OPENCV [Computer modeling of identification by the iris of the eye based on opencv]. *Matematicheskie struktury i modelirovanie*, 2014, no. 1(29), pp. 43-64.

4. Tikhonova T.S., Belov Iu.S. Osnovnye podkhody k otslezhivaniuu i raspoznavaniuu litsa [Basic approaches to face tracking and recognition]. *Nauka, tekhnika i obrazovanie*, 2016, no. 2(6), pp. 111-115.

5. Lienhart R., Kuranov A., Pisarevsky V. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection: MRL Technical Report, 2002.

6. Chaban L.N. Metody i algoritmy raspoznavaniia obrazov v avtomatizirovannom deshifrirovanii dannykh distantsionnogo zondirovaniia [Methods and algorithms for pattern recognition in automated decryption of remote sensing data]. Moscow: Moskovskii gosudarstvennyi universitet geodezii i kartografii, 2016, 94 p.

7. Bradsky G., Kaehler A. Learning OpenCV. O'Reilly, 2008, 571 p.

8. Nguen K.M., Koliuchkin V.Ia. Algoritmy konturnoi segmentatsii i raspoznavaniia obrazov ob"ektov sistem tekhnicheskogo zreniia [Algorithms for contour segmentation and image recognition of objects in vision systems]. *Nauka i obrazovanie: nauchnoe izdanie Moskovskogo gosudarstvennogo tekhnicheskogo universiteta imeni N.E. Baumana*, 2013, no. 4, pp. 187-200.

9. Redmon J., Divvala S.K., Girshick R.B., Farhadi A. You only look once: Unified, real-time object detection. *CoRR*, 2015, abs/1506.02640.

10. Girshick R.B., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, 2013, abs/1311.2524.

11. Girshick R.B. Fast R-CNN. *CoRR*, 2015, abs/1504.08083.

12. Ren S., He K., Girshick R.B., Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, 2015, abs/1506.01497.

13. Kainz O., Horecny M.W., Jakab F., Cymbalak D. Estimating the Object Size from Static 2D Image. *CoRR*, 2015, pp. 15-17.

14. Iane B. Tsifrovaia obrabotka izobrazhenii [Digital image processing]. Moscow: Tekhnosfera, 2007, 584 p.

15. Chang P.L., Teng W.G. Exploiting the self-organizing map for medical image segmentation in Computer-Based Medical Systems. *Twentieth IEEE International Symposium on*, 2007, pp. 281288.

16. Zhou Z.J., Wu H. Digital Image Processing: Part 1. Ventus Publishing ApS, 2010.

17. Canny J.E. A computational approach to edge detection. *IEEE Trans Pattern Analysis and Machine Intelligence*, 1986, no. 8, pp. 679-698.

18. Tsvetkov A. A., Shorokh D. K., Zubareva M. G., Iurskov S. V., Shuklin A. V., Khamush A. L., Anufriev I. B. Algoritmy raspoznavaniia ob"ektov [Object recognition algorithms]. *Tekhnicheskie nauki: problemy i perspektivy*, 2016, pp. 20-28.

19. Shitova O.V., Pukhliak A.N., Drob E.M. Analiz metodov segmentatsii teksturnykh oblastei izobrazhenii v sistemakh obrabotki izobrazhenii [Analysis of methods of segmentation texture regions in the systems of image processing]. *Nauchnye vedomosti Belgorodskogo gosudarstvennogo universiteta. Ekonomika. Informatika*, 2014, no. 8(179), pp. 182-188.

20. Rodzin S.I., Rodzina O.N., El'-Khatib S.A. Gibridnyi murav'inyi algoritm segmentatsii meditsinskikh izobrazhenii [Hybrid segmentation ant algorithms of medical images]. *Vestnik Chuvashskogo universiteta*, 2017, no. 3, pp. 262-272.

21. Komkov V.S., Diudin M.V. Metod segmentatsii rastroykh izobrazhenii na osnove vybora optimal'nykh napravlenii obrabotki pikselei izobrazheniia [Method of segmentation of bitmapped images on basis of choice of optimal directions of treatment of pixels of image]. *Rossiiskaia nauka i obrazovanie segodnia: problemy i perspektivy*, 2015, no. 2(5), pp. 151-154.

22. Shapiro L., Stokman Dzh. Komp'iuternoe zrenie [Computer vision]. Moscow: Binom, 2006, 752 p.

23. Gonzales R., Vuds R. Tsifrovaia obrabotka izobrazhenii [Digital imaging]. Moscow: Tekhnosfera, 2006, 1072 p.

24. Rosenfeld A., Kak A.C. Digital Picture Processing. New York: Academic press, 1976.

25. Strugailo V.V. Obzor metodov fil'tratsii i segmentatsii tsifrovyykh izobrazhenii [Overview of digital image filtering and segmentation methods]. *Nauka i obrazovanie: nauchnoe izdanie Moskovskogo gosudarstvennogo tekhnicheskogo universiteta imeni N.E. Baumana*, 2012, no. 5, 17 p.

26. GOST 166-89. Mezhgosudarstvennyi standart. Shtangentsirkuli. Tekhnicheskie usloviia [GOST 166-89. Interstate standard. Calipers. Technical conditions]. Dostup iz spravочно-pravovoi sistemy Konsul'tantPlius.

27. Vlasov A.A. Metod konturnoi segmentatsii kanni s ispol'zovaniem morfologicheskikh operatsii [Canny contour segmentation method using morphological operations]. *Reshetnevskie chteniia*, 2010, no. 2, 479 p.

### **Сведения об авторах**

**Зайникова Виктория Рустемовна** (Лысьва, Россия) – студентка Лысьвенского филиала Пермского национального исследовательского политехнического университета (618905, Лысьва, ул. Ленина, 2, e-mail: zaunikowa@yandex.ru).

**Зыкин Сергей Анатольевич** (Лысьва, Россия) – старший преподаватель кафедры ОНД Лысьвенского филиала Пермского национального исследовательского политехнического университета (618905, Лысьва, ул. Ленина, 2, e-mail: zykin\_serгей@mail.ru).

**Файзрахманов Рустам Абубакирович** (Пермь, Россия) – доктор экономических наук, профессор, заведующий кафедрой «Информационные технологии и автоматизированные системы» Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: fayzrakhmanov@gmail.com).

### **About the authors**

**Zaynikova Victoria Rustemovna** (Lysva, Russian Federation) is a Student Lysvensky branch Perm National Research Polytechnic University (618905, Lysva, Lenin, 2, e-mail: zaunikowa@yandex.ru).

**Zykin Sergey Anatolyevich** (Lysva, Russian Federation) is a Senior Lecturer, Department of OND, Lysva branch of Perm National Research Polytechnic University (618905, Lysva, Lenina, 2, e-mail: zykin\_serгей@mail.ru).

**Fayzrakhmanov Rustam Abubakirovich** (Perm, Russian Federation) is a Doctor of Economics, Professor, Head of the Department of Information Technologies and Automated Systems Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: fayzrakhmanov@gmail.com).

Получено 17.08.2020