

*Беляев Сергей Юрьевич*

магистрант

*Павлова Светлана Михайловна*

канд. пед. наук, доцент

Муромский институт (филиал)

ФГБОУ ВО «Владимирский государственный

университет им. А.Г. и Н.Г. Столетовых»

г. Муром, Владимирская область

## **РАСПОЗНАВАНИЕ ОТДЕЛЬНЫХ ПЛОСКИХ ОБЪЕКТОВ МЕТОДОМ БЛИЖАЙШЕГО СОСЕДА ПО БЕЗРАЗМЕРНЫМ ПРИЗНАКАМ**

*Аннотация:* в данной статье описаны основные алгоритмы, необходимые для распознавания плоских объектов. На основании этих алгоритмов реализовано приложение, в результате работы которого сделаны выводы о качестве распознавания плоских объектов.

*Ключевые слова:* безразмерные признаки, выпуклая оболочка, алгоритм ближайшего соседа, маркировка, алгоритм Брезенхема.

Целью данной работы является разработка приложения для распознавания объектов на основе безразмерных признаков их контуров и выпуклых оболочек методом ближайшего соседа. ПО должно быть простым в использовании и не требовать от пользователя дополнительных знаний в этой области.

Основой метода ближайшего соседа является формирование вектора признаков. Признаки представляют собой дробные числа, расположенные в диапазоне от 0 до 1. Коэффициенты считаются в два этапа: сначала первичные, потом вторичные. Набор необходимых коэффициентов был уже сформирован и был взят из научных статей [1; 2]. Задачей данной работы является формирование векторов признаков и их классификация методом ближайшего соседа.

Первым этапом процесса распознавания объектов является процесс получения и обработки изображения. На этом этапе последовательно происходят

следующие операции: получение изображения с web-камеры, перевод входного изображения в полутоновый формат, фильтрация, бинаризация, маркировка.

После того как процесс начальной обработки изображения завершен, ставится задача формирования вектора признаков полученного объекта. Этот этап включает в себя следующие операции: выделение выпуклой оболочки объекта, выделение контура объекта, расчет первичных признаков и расчет вторичных признаков.

Для расчета выпуклой оболочки был выбран алгоритм QuickHull.

QuickHull – один из алгоритмов выделения выпуклой оболочки объекта, основанный на идее быстрой сортировки Хоара. Этот алгоритм был выбран, так как отличается своим быстродействием.

После выполнения алгоритма программа возвращает набор наиболее отдаленных точек принадлежащих выпуклой оболочке. Остается только их соединить, после чего получится набор всех точек, принадлежащих к выпуклой оболочке. Чтобы это сделать используется алгоритм Брезенхема. Алгоритм Брезенхема – это алгоритм, определяющий, какие точки *двумерного растра нужно закрасить, чтобы получить близкое приближение прямой линии между двумя заданными точками.*

После того как выпуклая оболочка была построена, строится контур объекта. Контур вычисляется по методу направленного перебора.

Алгоритм основан на том, что на изображении отыскивается объект (первая встретившаяся точка объекта) и контур объекта отслеживается и векторизуется. Достоинством алгоритма является его простота, к недостаткам можно отнести последовательную реализацию.

Алгоритм начинает движение с белой области по направлению к черной. Как только он попадает на черный элемент, он поворачивает налево и переходит к следующему элементу. Если этот элемент белый, то алгоритм поворачивается направо, иначе – налево. Процедура повторяется до тех пор, пока алгоритм не вернется в исходную точку. Координаты точек перехода с черного на белое и с белого на черное и описывают границу объекта.

В результате после прохода алгоритма мы получим координаты контура объектов.

После того как выпуклая оболочка и контур сформированы, считаются первичные признаки (по отдельности). Первым делом считаются площадь и периметр. Далее считается длина контура (выпуклой оболочки).

$$L_{\text{кон.}} = [(N1 + N3 + N5 + N7) + \sqrt{2} * (N2 + N4 + N6 + N8)],$$

$N1, N3, N5, N7$  – линейные точки (следующая точка смещена либо по  $X$ , либо по  $Y$ );

$N2, N4, N6, N8$  – диагональные точки (следующая точка смещена и по  $X$  и по  $Y$ ).

Далее посредством сканирования изображения и сканирования масками  $3 \times 3$ , мы находим значения кривизны в каждой точке контура (выпуклой оболочке).

В итоге после сравнений получаем следующие коэффициенты:

$M1$  – количество точек со значением  $+90$ ;

$M2$  – количество точек со значением  $-90$ ;

$M3$  – количество точек со значением  $+135$ ;

$M4$  – количество точек со значением  $-135$ ;

$K$  – количество линейных точек со значением  $0$ ;

$T$  – количество диагональных точек со значением  $0$ ;

Остальные первичные признаки будут сформированы на основе уже существующих.

После того как все первичные признаки будут сформированы, останется лишь рассчитать вторичные признаки и сформировать из них вектор признаков. Все вторичные признаки представляют собой отношения меньшего к большему и находятся в диапазоне от  $0$  до  $1$ . После расчета вторичных признаков мы получаем два вектора: для контура вектор состоит из  $18$  значений. Примеры вторичных признаков:

$$K1 = P / S;$$

$$K2 = M1 / S;$$

$$K10 = M1 / (M1 + M2 + M3 + M4);$$

$$K15 = (K + T) / (P0 + S0);$$

$$K18 = ((M1 + M2 + M3 + M4) * T) / (P0 * S0);$$

Под распознаванием в данной работе понимается кластеризация объекта, то есть отнесение объекта к одному из существующих классов. Чтобы функция распознавание была доступна пользователю необходимо, чтобы в приложении уже хранилась база эталонов с которой будет сравниваться распознаваемый объект.

Как видно из блок-схемы (рис. 1) на вход поступает база эталонов и вектор распознаваемого объекта. Далее между этим вектором и всеми векторами в базе высчитываются расстояния. Среди этих расстояний ищется минимальный. После того как минимальное расстояние найдено, необходимо последовательно узнать к какому вектору из базы это расстояние принадлежит, и к какому классу принадлежит найденный вектор. Номер этого класса и будет результатом распознавания.

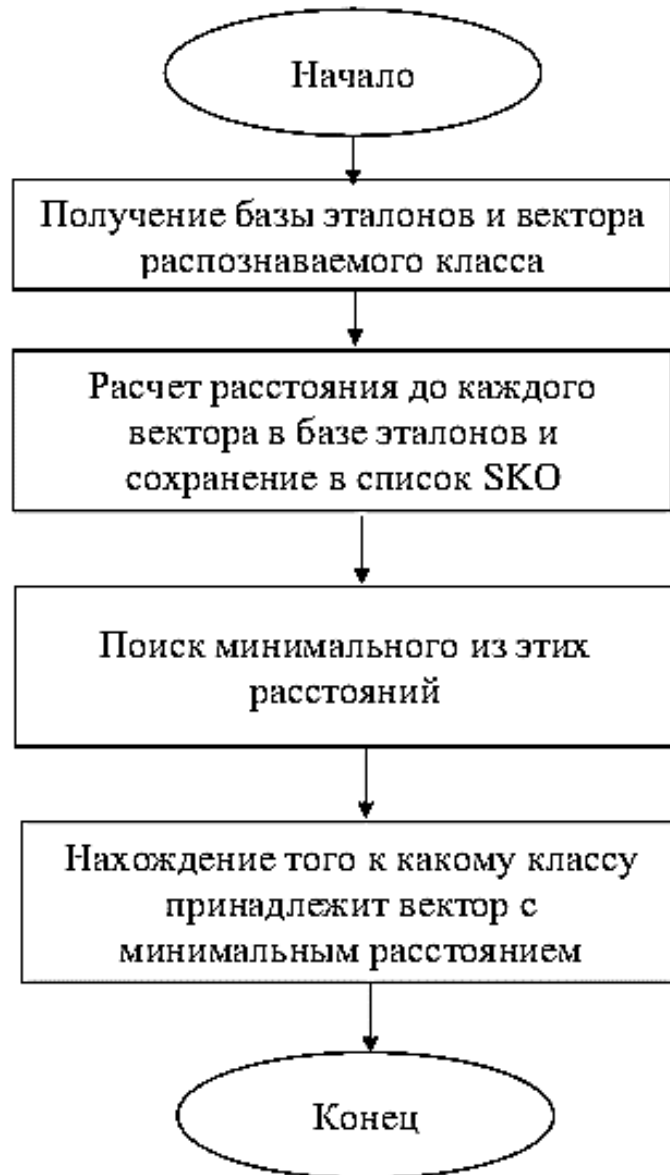


Рис. 1. Блок-схема алгоритма классификации

В ходе работы было принято решение, что для расчета расстояния между векторами будет использоваться метрика квадрата Евклидова расстояния (1). Другие метрики такие как Евклидово расстояние, Среднеквадратичное отклонение включают в себя большее количество действий, но на результат не влияют [3].

$$d(p, q) = \sum_{k=1}^n (p_k - q_k)^2 \quad (1)$$

На основе описанных алгоритмов было реализовано приложение на языке С#, классифицирующее объекты попадающие в поле зрения камеры (рис. 2).

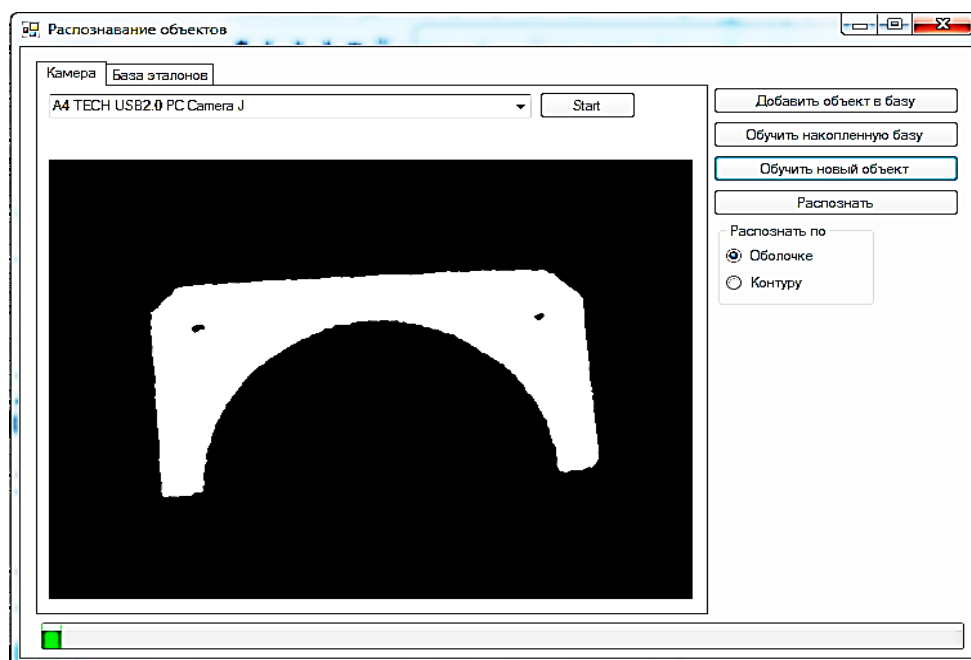


Рис. 2. Интерфейс приложения

В ходе проведения работы были получены результаты, такие как вероятность распознавания по выпуклой оболочке и контуру. Вероятность распознавания любого объекта по выпуклой оболочке составляет минимум 0,9. Вероятность распознавания по контуру имеет сильный разброс и находится в интервале от 0,1 до 0,8.

### ***Список литературы***

1. Садыков С.С. Формирование безразмерных коэффициентов формы замкнутого дискретного контура / С.С. Садыков // Алгоритмы, методы и системы обработки данных. – 2014. – №4 (29). – С. 91–98.
2. Садыков С.С. Оценка возможности распознавания отдельных тестовых плоских объектов на основе цепных кодов их контуров / С.С. Садыков, Я.Ю. Кульков. – 2016. – С. 101–102.
3. Прогрессивные технологии в машиностроении: Тематический сборник научных трудов / В.И. Гузеев. – Челябинск: Изд-во ЮУрГУ, 2006. – 239 с.