

## **РАЗРАБОТКА АЛГОРИТМОВ ОБРАБОТКИ ИЗОБРАЖЕНИЙ ИНТЕЛЛЕКТУАЛЬНЫМИ МОБИЛЬНЫМИ РОБОТАМИ НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ И НЕЙРОННЫХ СЕТЕЙ**

**Аунг Чжо Хтун<sup>1</sup>, Тант Зин Пьо<sup>2</sup>, Федоров А.Р.<sup>2</sup>, Федоров П.А.<sup>2</sup>**

<sup>1</sup>*Академия Оборона Республики Союза Мьянма, Пьен У Лвин, Мьянма*

<sup>2</sup>*Национальный исследовательский университет, Зеленоград, Москва, Россия (124498, Москва, Зеленоград, проезд 4806, д.5, МИЭТ), e-mail: kyawzawye85@gmail.com*

---

**В данной статье разработаны алгоритмы обработки изображений интеллектуальными мобильными роботами на основе нечеткой логики и нейронных сетей, обеспечивающие для выделения границ в изображении с помощью оператора Собеля. Сущность обработки изображения заключается в приведении исходного изображения сцены к виду, позволяющему решить задачу распознавания ее объектов. Рассмотрены основные проблемы, а также пути их решения при первичной подготовке изображения к распознаванию. Детально разобраны алгоритм предварительной обработки с помощью нечеткой логики и процесс бинаризации изображения. Построен алгоритм нечеткой обработки для выделения границ в изображении с помощью оператора Собеля.**

---

Ключевые слова: обработка изображения, нечеткая логика, интеллектуальная система, роботы, распознавание объектов.

## **DEVELOPMENT OF IMAGE PROCESSING ALGORITHMS FOR INTELLIGENT MOBILE ROBOTS BASED ON FUZZY LOGIC AND NEURAL NETWORKS**

**Aung Kyaw Tun<sup>1</sup>, Thant Zin Phyoe<sup>2</sup>, Fedorov A.R.<sup>2</sup>, Fedorov P.A.<sup>2</sup>**

<sup>1</sup>*Defence Academy of the Republic of the Union of Myanmar, Pyin Oo Lwin, Myanmar*

<sup>2</sup>*National Research University, Zelenograd, Moscow, Russia (124498, Moscow, Zelenograd, passage 4806, House.5, MIET), e-mail: kyawzawye85@gmail.com*

---

**In this paper, we develop algorithms for image processing intelligent mobile robots based on fuzzy logic and neural networks, providing for the allocation of boundaries in the image using the Sobel operator. The essence of the image processing is to bring the original image of the scene to the form, which allows to solve the problem of the recognition of its objects. The main problems and their solutions during the initial preparation of images for recognition. Detailed understanding of pre-processing algorithms using fuzzy logic and the process of image binarization. We construct an algorithm of fuzzy processing to extract the boundaries of the image using the Sobel operator.**

---

Keywords: image processing, fuzzy logic, intelligent system, robots, object recognition.

В настоящее время автоматическая обработка изображений является одним из важнейших направлений в области искусственного интеллекта и подразумевает разработку робототехнических комплексов, осуществляющих распознавание образов. Одним из наиболее эффективных инструментов для распознавания образов является системы, построенные на нечеткой логике и искусственных нейронных сетях. В системе технического зрения (СТЗ) необходимо нескольких методов и алгоритмов, решающие одну и ту же задачу различными способами, обеспечивая при этом необходимые показатели по быстродействию и достоверности идентификации [1-3].

Сущность гибридного алгоритма обработки изображений в СТЗ мобильного робототехнических комплексов (МРК) заключается в приведении исходного изображения сцены к виду, позволяющему решить задачу распознавания ее объектов.

## Алгоритм предварительной обработки изображения с помощью нечеткой системы в СТЗ

К обработке изображений нечеткая обработка представляет собой множество различных нечетких подходов, которыми являются понимание, представление, обработки изображений, сегменты и нечеткие множества. В процессе распознавания образов огромное значение имеет процесс предварительной нечеткой обработки изображений, так как именно от него зависит качество данных, далее поступающих на входы нейронной сети. В рамках решаемой задачи, разработанный алгоритм предварительной нечеткой обработки можно представить в виде следующей последовательности шагов (рис. 1): захват изображения с помощью веб-камеры; преобразование полученного цветного изображения в изображение в градациях серого цвета; нечеткая обработка изображений.

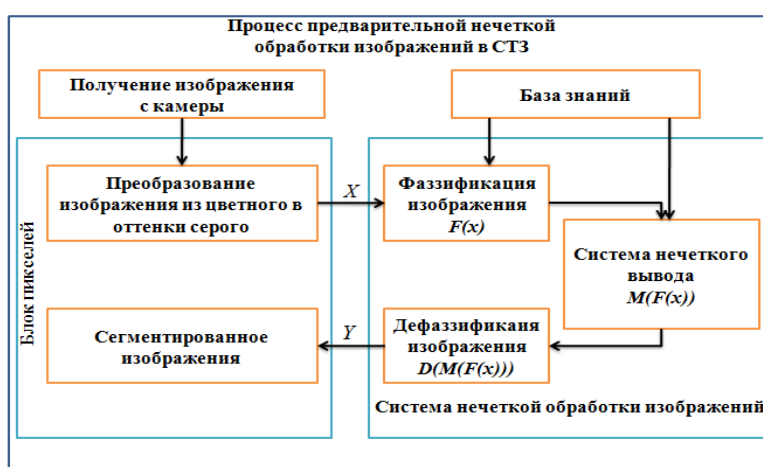


Рис. 1. Алгоритм предварительной нечеткой обработки изображения

Таким образом, первым шагом предварительной нечеткой обработки является преобразование изображение из цветного в оттенки серого. Преобразование цветов изображения в оттенки серого цвета осуществляется следующим образом. Вся цветовая палитра представляется в виде куба, вершины которого соответствуют различным цветам. Серая шкала располагается на диагонали куба, соединяющей черную и белую вершины.

Для преобразования изображения в оттенки серого для каждой точки изображения выделяются интенсивности красной, зеленой и синей составляющей цвета, а затем осуществляется преобразование цвета по следующей формуле:

$$Y = 0.255R + 0.255G + 0.255B, \quad (1)$$

где  $Y$  – новое значение цвета,  $R$  – интенсивность красной составляющей цвета,  $G$  – интенсивность зеленой составляющей цвета, а  $B$  – интенсивность синей составляющей цвета. Выход каждого алгоритма оттенки серого между 0 и 1. Для преобразования изображений в использующие только оттенки серого существует некоторые методов. В методе определения светлоты используется среднее значение между двумя наиболее и

наименее значимыми цветами:  $G_{lightness} = \frac{1}{2}(\max(R, G, B) + \min(R, G, B))$ . В методе среднего используется среднее значение всех трёх цветов:  $G_{Intensity} = \frac{1}{3}(R + G + B)$ . В методе определения яркости используется взвешенное среднее значение всех трех цветов, учитывающее человеческое восприятие. Так, поскольку человеческий глаз наиболее восприимчив к зеленому цвету, его вес считается наиболее важным:  $G_{Luminance} = (0.3R + 0.59G + 0.11B)$ . Метод определения яркости используется программное обеспечение для обработки изображений. Он реализован функцию «rgb2gray» в среде MATLAB и это часто используется для компьютерного зрения [4]. В процессе предварительной нечеткой обработки имеет процесс преобразования изображений из цветного (RGB) в оттенки серого с помощью метода определения яркости. Далее изображение преобразуется из оттенки серого в черно-белый [0 1] (рис. 2).



Рис. 2. процесс преобразования изображений из цветного в оттенки серого

### **Бинаризация изображения при предварительной обработке**

Целью предварительной нечеткой обработки изображения является формирование и последующее улучшение изображения, его бинаризация и кодирование (в частности, получение контурного представления). Бинаризация изображения представляет собой процесс преобразования изображения, состоящего из градации одного цвета (в нашем случае - серого), в бинарное изображение, т.е. изображение, в котором каждый пиксель может иметь только два цвета (в нашем случае это черный и белый цвета). В результате такого преобразования, цвет пикселя условно считают равным нулю или единице, при этом, пиксели с нулевым значением (в данном случае это пиксели белого цвета) называют задним планом, а пиксели со значением равным единице (черного цвета) называют передним планом. Но бинарное изображение, полученное в результате такого преобразования, искажается, по сравнению с оригиналом, что характеризуется появлением разрывов и размытостей на объектах, возникновением зашумлений изображения в однородных областях, а так же к потере целостности структуры объектов.

Потеря целостности объекта, а так же разрыв объекта, возникают в силу ряда причин, таких как, большая неравномерность освещения объекта или касание (или наложение объектов друг на друга). Особую сложность в обработке вызывает именно наложение (или касание – как частный случай наложения), т.к. с одной стороны, изображение нескольких объектов может быть интерпретировано как один объект, а с другой стороны, алгоритмы, проверяющие геометрическую целостность объекта, в местах наложения будут формировать разрывы, представляя эти области в виде заднего плана. Сложность обработки заключается в отсутствии теоретического решения задачи интерпретации наложения объектов, так как часть информации утеряна. В реализации алгоритмов на практике, в качестве верного решения принимают один из указанных вариантов – либо пересечение считается продолжением текущего объекта, либо область наложения будет считаться задним планом.

Пороговая обработка преобразовывает цветное или серое изображение в черно-белое изображение. Пороговые преобразования занимают центральное место в прикладных задачах сегментации изображений благодаря интуитивно понятным свойствам и простоте реализации. Для каждого пикселя в изображении, его уровень интенсивности исследован, если его значение – выше некоторого порогового уровня, это соответствует белому цвету. Если это – ниже порога набора, это установлено в черный. Пороговый уровень будет между 0 и 255.

В настоящее время существует большое количество методов бинаризации. Суть данного преобразования растровых изображений заключается в сравнительном анализе яркости текущего пикселя  $P(x, y)$  с неким пороговым значением  $P_T(x, y)$ : если яркость текущего пикселя превышает пороговое значение, т.е.  $P(x, y) > P_T(x, y)$ , то цвет пикселя на бинарном изображении будет белым, в противном случае цвет будет черным. Пороговой поверхностью  $P_T$  является матрица, размерность которой соответствует размерности исходного изображения [5].

В процессе бинаризации все методы делятся на две группы по принципу построения пороговой поверхности – это методы глобальной и локальной обработки бинаризации. В методах глобальной обработки бинаризации пороговая поверхность является плоскостью с постоянным значением пороговой яркости, т.е. значение порога рассчитывается исходя из анализа гистограммы всего изображения и является одинаковым для всех пикселей исходного изображения. Глобальная пороговая обработка имеет существенный недостаток - если исходное изображение имеет неоднородное освещение, области, которые освещены хуже, целиком классифицируются как передний план. В локальных же методах обработки бинаризации пороговое значение меняется для каждой точки исходя из некоторых признаков области, принадлежащей некоторой окрестности данной точки. Недостатком такого рода

преобразований является низкая скорость работы алгоритмов, связанная с пересчетом пороговых значений для каждой точки изображения.

В качестве метода для решения поставленной задачи воспользуемся методом Бернсена [4,5]. Метод базируется на идее сопоставления уровня яркости преобразуемого пикселя со значениями локальных средних, вычисляемых в его окружении. Пиксели изображения обрабатываются поочередно путем сравнения их интенсивности со средними значениями яркости в окнах с центрами в точках  $P_i (i = 0,1,\dots,7)$  (рис.3).

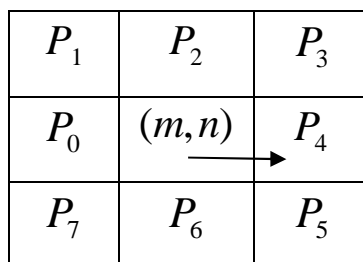


Рис. 3. Преобразование пикселя изображения

### **Алгоритм нечеткой обработки для выделения границ и сегментации изображений**

После преобразования изображения в черно-белый, получается градиентное изображение с помощью оператора Собеля и поступает на входы нечеткого обработки изображения (НОИ) (рис. 4).

Нечеткая обработка изображений состоит из трех основных этапов: фаззификация изображений  $F$ , система нечеткого вывода  $M$  на значениях принадлежности и дефаззификация изображений  $D$ . Главная нечеткая обработка изображений находится в среднем шаге (система нечеткого вывода  $M$ ). После передачи данных изображения из серого уровня в фаззификацию, система нечеткого вывода определяется значениям принадлежности. Фаззификация – кодирование данных изображений и дефаззификация – декодирование результатов, которые позволяют обрабатывать изображения с нечеткими методами.

Изображение  $X$  – размер  $M \times N$  с  $L$  серыми уровнями  $g = 0,1,2,\dots,L-1$  и может быть определено как массив нечеткое одноточечное множество (нечеткие множества можно поддерживать только с одной точкой), указывающее значение принадлежности каждого пикселя в отношении по заранее свойства изображения (например – яркость, гладкость и т.д.).

$$X = \bigcup_{m=1}^M \bigcup_{n=1}^N \frac{\mu_{mn}}{g_{mn}}, \quad \mu_{mn} \in [0,1], \quad (1)$$

где  $\mu_{mn}$  и  $g_{mn}$  – принадлежности пикселя  $mn$  в обозначении нечетких множеств. Определение значений принадлежности зависит от конкретных требований особенного применения и от соответствующей базы знаний.

Выход системы  $Y$  для входной системы  $X$  задается следующей формулу:

$$Y = D(M(F(X))). \quad (2)$$



Рис. 4. Алгоритм нечеткой обработки изображений для выделения границ

### Применение нейронных сетей для распознавания образов

Многослойным персептроном называют искусственную нейронную сеть, состоящую из нескольких входных узлов, образующих входной слой, одного или нескольких вычислительных слоев нейронов и одного выходного слоя (рис. 6). В таких сетях сигнал, подающийся на входной слой, передается последовательно в прямом направлении от слоя к слою. Данный тип ИНС успешно применяется для решения разнообразных задач, в частности для задачи распознавания образов [4].

Нейронная сеть обратного распространения ошибки состоит из нескольких слоев нейронов, причем каждый нейрон предыдущего слоя связан с каждым нейроном последующего слоя. В таких сетях, после определения количества слоев и количества элементов каждого слоя, требуется вычислить значения весов и порогов сети таким образом, что бы минимизировать ошибку прогноза. Данная задача решается при помощи разнообразных алгоритмов обучения. Суть этих алгоритмов заключается в подгонке сети к обучающим данным. Ошибка реализованной сети будет определяться путем прогона всех входных данных и сравнением реально полученных значений на выходе сети с целевыми значениями. Затем полученные разности суммируются в общую, так называемую, функцию ошибок, которая характеризует общую ошибку сети. Но чаще, в качестве функции ошибок, берется сумма квадратов ошибок.

Одним из наиболее распространенных алгоритмов обучения многослойных нейронных сетей является алгоритм обратного распространения ошибки. В этом алгоритме вычисляется вектор градиента поверхности ошибок. Затем продвигаемся на некоторую величину в направлении вектора (он будет указывать нам направление наискорейшего спуска), где значение ошибки будет уже меньше. Такое последовательное продвижение постепенно приведет к минимизации ошибки. Здесь возникает трудность с определением величины, на которую следует продвигаться. Если величина шага будет относительно большой, это приведет к наискорейшему спуску, однако есть вероятность «перепрыгнуть» искомую точку или уйти в неверном направлении, если поверхность имеет достаточно сложную форму. Например, в случае, если поверхность представляет собой узкий овраг с крутыми склонами, алгоритм будет очень медленно продвигаться, прыгая с одного склона на другой. Если же величина шага будет маленькой, это приведет к нахождению наиболее оптимального направления, однако может значительно увеличить число итераций. Для достижения наиболее оптимального результата величина шага берется пропорционально крутизне склона с некоторой константой- скоростью обучения. Выбор этой константы осуществляется экспериментальным путем и зависит от условий конкретной задачи.

Введем следующие обозначения. Матрицу весовых коэффициентов от входов к скрытому слою обозначим  $W$ , а матрицу весов, соединяющих скрытый и выходной слой –  $V$ . Для индексов примем следующие обозначения: входы будем нумеровать только индексом  $i$ , элементы скрытого слоя – индексом  $j$ , а выходы – индексом  $k$ . Число входов сети равно  $n$ , число нейронов в скрытом слое –  $m$ , число нейронов в выходном слое –  $p$ . Пусть сеть обучается на выборке  $(X^t, D^t)$ ,  $t = \overline{1, T}$ . Тогда алгоритм обучения многослойного персептрона будет выглядеть следующим образом [5]:

*Шаг 1.* Инициализация сети. Весовым коэффициентам присваиваются малые случайные значения, например, из диапазона  $(-0.3, 0.3)$ ; задаются  $\varepsilon$  – параметр точности обучения,  $\alpha$  – параметр скорости обучения (как правило,  $\approx 0.1$  и может еще уменьшаться в процессе обучения),  $N_{\max}$  – максимально допустимое число итераций.

*Шаг 2.* Вычисление текущего выходного сигнала. На вход сети подается один из образов обучающей выборки, и определяются значения выходов всех нейронов нейросети.

*Шаг 3.* Настройка синоптических весов. Рассчитать изменение весов для выходного слоя нейронной сети по формулам:

$$V_{jk}^{N+1} = V_{jk}^N - \alpha \frac{\partial E}{\partial V_{jk}}, \text{ где } \frac{\partial E}{\partial V_{jk}} = \delta_k y_j^c, \delta_k = (y_k - d_k) y_k (1 - y_k). \text{ Рассчитать изменение весов для}$$

$$\text{скрытого слоя по формулам: } W_{ij}^{N+1} = W_{ij}^N - \alpha \frac{\partial E}{\partial W_{ij}}, \text{ где } \frac{\partial E}{\partial W_{ij}} = \left( \sum_{k=1}^p \delta_k V_{jk}^{N+1} \right) y_j^c (1 - y_j^c) X_i$$

*Шаг 4.* Шаги 2-3 повторяются для всех обучающих векторов. Обучение завершается по достижении для каждого из обучающих образов значения функции ошибки, не превосходящего  $\epsilon$  или после максимально допустимого числа итераций.

На шаге 2 векторы из обучающей последовательности лучше предъявлять на вход в случайном порядке.

Количество входов и выходов сети, как правило, диктуется условиями задачи, а размер скрытого слоя находят экспериментально. Обычно число нейронов в нем составляет 30-50% от числа входов. Слишком большое количество нейронов скрытого слоя приводит к тому, что сеть теряет способность к обобщению (она просто досконально запоминает элементы обучающей выборки и не реагирует на схожие образцы, что неприемлемо для задач распознавания). Если число нейронов в скрытом слое слишком мало, сеть оказывается просто не в состоянии обучиться.

### **Заключение**

Рассмотрены основные проблемы, а также пути их решения при первичной подготовке изображения к распознаванию. Детально разобраны алгоритм предварительной обработки с помощью нечеткой логики и процесс бинаризации изображения. Построен алгоритм нечеткой обработки для выделения границ в изображении с помощью оператора Собеля.

### **Список литературы**

1. Веснин Е.Н., Вето А.В., Царев В.А. К вопросу о разработке и применении адаптивных оптоэлектронных систем технического зрения// Автоматизация в промышленности, 2009.- № 11.- С. 48-52.
2. Гришин В.А. Системы технического зрения в решении задач управления беспилотными летательными аппаратами // Датчики и системы, №2, 2009.- С. 46-52.
3. Клевалин В.А., Поливанов А.Ю. Цифровые методы распознавания в системах технического зрения промышленных роботов// Мехатроника, автоматизация, управление, 2008, № 5.- С. 56-56.
4. Михайлов С.В., Романов В.В., Заикин Д.А. Система технического зрения для диагностики процесса резания материалов// Вестник компьютерных и информационных технологий, 2007, № 3.- С. 12-19.
5. Семин М.С. Обзор решения прикладных задач с помощью систем технического зрения// <http://www.videoscan.ru/page/718#13>.



**Рецензенты:**

Гагарина Л.Г., д.т.н., профессор, заведующий кафедрой «Информатика и программное обеспечение вычислительных систем» Национального исследовательского университета «МИЭТ», г.Москва.

Портнов Е.М., д.т.н., профессор кафедры «Информатика и программное обеспечение вычислительных систем», начальник научно-исследовательской лаборатории «Управляющие информационные системы» Национального исследовательского университета «МИЭТ», г.Москва.