

10. Muna, M.J. Design Network Intrusion Detection System using hybrid Fuzzy-Neural Network / M.J. Muna, M. Mehrotra // International Journal of Computer Science and Security. – 2010. – Vol. 4 (3). – P. 258–294.
11. Giacinto, G. Fusion of multiple classifiers for intrusion detection in computer networks / G. Giacinto, F. Roli, L. Didaci // Pattern Recognition Letters. – 2003. – № 24. – P. 1795–1803.
12. Lee, W. Mining in a data-flow environment: Experience in network intrusion detection / W. Lee, S.J. Stolfo, K.W. Mok // The Fifth International Conference on Knowledge Discovery and Data Mining (KDD-99): proceedings, San Diego, CA, 15–18 August, 1999 / Eds.: S. Chaudhuri, D. Madigan. – ACM, 1999. – P. 114–124.
13. Lee, W. Adaptive intrusion detection: A data mining approach / W. Lee, S. J. Stolfo, K. W. Mok // Artificial Intelligence Review. – 2000. – Vol. 14. – №6. – P. 533–567.
14. Головкин, В.А. Нейронные сети: обучение, организация и применение / В. А. Головкин. – М.: ИПРЖР, 2001. – 256 с.
15. Ежов, А.А. Нейрокомпьютеринг и его приложение в экономике и бизнесе / А. А. Ежов, С.А. Шумский. – М.: МИФИ, 1998. – 222 с.
16. Kramer, M.A. Nonlinear principal component analysis using auto-associative neural networks / M.A. Kramer // J. Am. Institute Chem. Eng. (AIChE). – 1991. – Vol. 37. – №2. – P.233–243.

Материал поступил в редакцию 22.12.12

**KACHURKA P.A., GOLOVKO V.A. Ensemble of neuronetwork detectors in systems of detection of attacks**

Approach to detection of network attacks with use of recirculation neural networks as detectors of anomalies and detectors of abuses is presented. Sharing of these detectors within system of detection of attacks is proved. The experimental results confirming prospects of approach are given.

УДК 004.896

**Дёмин В.В., Кабыш А.С., Дунец И.П., Дунец А.П., Головкин В.А.**

**ИСПОЛЬЗОВАНИЕ RAM-BASED СЕТЕЙ ДЛЯ ДЕТЕКТИРОВАНИЯ ГРАФИЧЕСКОЙ МЕТКИ**

**Введение.** В данной работе рассмотрена задача распознавания графической метки ведущего робота в системе «ведущий-ведомый роботы». Для решения задачи был разработан метод детектирования на основе RAM-based сетей, позволяющий по расположению метки узнать положение и дальность ведущего робота по графическому паттерну. Обученная RAM-based сеть хранит характеристические особенности паттерна в разных секторах относительно ведомого робота. Сработавший дискриминатор сети будет указывать на сектор, в котором находится ведущий робот. В подробностях описана процедура обучения и настройки RAM-based сети. Преимуществом данного подхода является снижение вычислительных ресурсов, что позволяет применять данное решение на платформах с низкой производительностью.

**Использование камер в коммуникации стайных роботов.** В задаче следования за лидерами от роботов требуется сформировать паттерн формации, при котором каждый предыдущий робот следует за последующим, а ведущий – лидер, либо управляется оператором, либо следует по заранее заданному пути.

Существуют различные подходы к распознаванию роботами друг друга: инфракрасные датчики, световые паттерны, RFID-метки, компьютерное зрение и т.д. В данной работе распознавание ведущего основано на детектировании визуальной графической метки, закрепленной на корпусе робота при помощи RAM-based нейронных сетей.

Роботы, оснащенные видеосистемой, могут получить необходимый минимум информации о находящихся поблизости роботах группы, если роботов достаточно легко детектировать по их особенностям, паттернам или световым меткам. Цель работы состоит в разработке методики распознавания ведущего робота, не требующей достаточных вычислительных мощностей и устойчивой к помехам во внешней среде.

Отсутствие камеры в целом ограничивает область применения swarm роботов. Большинство камер на роботах swarm-масштаба не являются пригодными для глобального восприятия окружающей среды, а используются только для локальных областей, с максимальным радиусом до 1 метра.

В работе [1] описано стайное поведение группы роботов, окра-

шенных в красный цвет. Эта их характеристика (красный окрас) была использована при детектировании роботов друг другом с помощью камеры. Если область не менее чем 25 на 25 пикселей по границам имела красный цвет, то область помечалась как вероятный сосед. После фазы распознавания принимались решения о факте наличия робота и дистанции до него.

В работе [2] роботы «s-bot» имели светодиодное кольцо, по периметру робота формирующее триангулярный паттерн из разных цветов, где красный цвет указывает направление движения робота (рис. 1). На роботе установлена всенаправленная камера с эффективной областью обзора в 60 см. Алгоритм обработки изображения на роботе определяет по «цвету» направление движения соседей в области видимости камеры. Для этого используется вероятностный алгоритм определения направления движения соседей по изображению, при заранее заданных оценках распределения. Роботы в данном исследовании решали задачу кооперативного транспорта путем создания формации и определяли направление движения путем переговоров.



Рис. 1. Триангуляционный паттерн, формируемый светодиодами робота

**Описание используемых роботов**

**Ведущий или лидер (leader)** – автономный робот, основной задачей которого является движение к цели по некоторому маршруту и, возможно, без столкновений с препятствиями. За ведущим роботом следуют **ведомые (followers)** роботы, оборудованные видеокamerой для распознавания ведущего робота. В задаче следования за лидером от ведомых роботов требуется не потерять ведущего –

**Дёмин В.В.**, магистрант кафедры интеллектуальных информационных технологий Брестского государственного технического университета.

**Кабыш А.С.**, ст. преподаватель кафедры интеллектуальных информационных технологий Брестского государственного технического университета.

**Дунец И.П.**, аспирант кафедры интеллектуальных информационных технологий Брестского государственного технического университета.

**Дунец А.П.**, доцент кафедры интеллектуальных информационных технологий Брестского государственного технического университета.

Беларусь, БрГУ, 224017, г. Брест, ул. Московская, 267.

Физика, математика, информатика

лидера, который либо управляется оператором, либо следует по заранее заданному пути.

В качестве мобильной платформы используется DFRobotTurtle 2WDMobileplatform. Ведущий робот оснащен паттерном на задней части (рис. 2). Размер паттерна 28 на 7 см. Паттерн состоит из трех горизонтальных полос: черной, белой, черной. Высота черных полос 2,5 см, белой – 2 см.

Горизонтальное расположение линий паттерна обусловлены требованием горизонтальной схожести паттерна, т.к. детектирование производится на основе положения робота, а не его ориентации. Порядок черная-белая-черная линий выбран для детектирования промежуточных положений робота.

Ведущий и ведомый робот в связке во время проведения экспериментов изображены на рисунке 2.

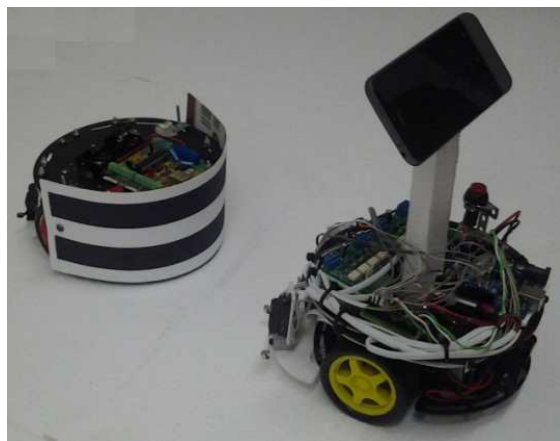


Рис. 2. Ведущий и ведомый роботы

**Система детектирования ведущего робота.** Система детектирования ведущего робота основана на использовании RAM-based сетей с предварительными фазами подготовки и обработки изображения. Вся последовательность выполняемых этапов системы детектирования изображена на рис. 3.

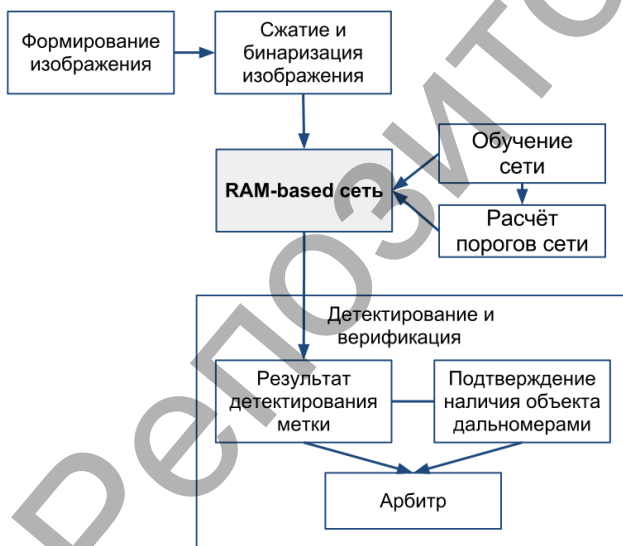


Рис. 3. Компоненты системы детектирования графической метки

Для распознавания области, в которой находится ведущий робот, изображение подвергается обработке с целью выделения графической метки. После предварительных фаз, изображение делится на 6 секторов, в которых выполняется поиск графической метки RAM-based сетью. На основе детектирования арбитр принимает решение о наличии или отсутствии графической метки в заданном секторе. Опционально, может использоваться верификация при

помощи инфракрасных дальномеров. На основе результатов распознавания робот принимает решение о дальнейших действиях, согласно алгоритму поведения.

**Подготовка изображения.** Расположение камеры ведомого позволяет получить область обзора камеры для детектирования от 15 до 70 см. Робот получает цветное изображение с камеры в виде массива размером 320 на 240 пикселей, где каждый пиксель представлен тремя байтами в цветовом пространстве RGB. Полученное изображение конвертируется в цветовое пространство Grayscale, содержащее изображение в градациях серого.

**Предобработка изображения.** Получаемое от камеры изображение содержит большое количество малоинформативных данных и нуждается в предварительной обработке. Обработка изображения состоит из следующих этапов:

1. Обрезка малоинформативных областей.
2. Сжатие изображения.

Обрезка малоинформативных областей изображения заключается в отсечении 40% верхней части изображения и 10% нижней, которые подобраны экспериментально (рис. 3). Отсечение верхней части изображения используется для удаления областей, содержащих графические метки роботов, находящихся на большом расстоянии от камеры. Эти метки имеют малый размер на изображении, что ведет к большому количеству ошибок при распознавании. Отсечение нижней части производится для уменьшения размера обрабатываемого изображения. Оставшаяся часть изображения делится на 6 секторов, размер которых пропорционален графической метке робота.

На рисунке 4-а показана эффективная рабочая область изображения. Рисунок 4-б показывает проекцию виртуальных секторов робота на реальные области.

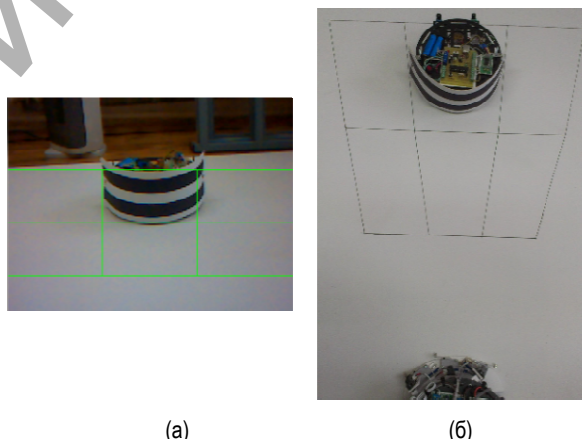


Рис. 4. Рабочая область изображения и пространства перед роботом

Сжатие изображения производится при помощи алгоритма линейной интерполяции. Из исходного изображения размером 320 на 120 пикселей формируется изображение размером 80 на 40 пикселей.

**Бинаризация изображения.** Для выделения элементов на изображении применяется контрастная бинаризация. Преобразование производится согласно формуле 1:

$$dst = \{x, y\} = \begin{cases} 0, & src(x, y) > T \\ 1, & src(x, y) \leq T \end{cases}, \quad (1)$$

где  $src(x, y)$  – пиксель исходного изображения в градации серого,  $dst\{x, y\}$  – пиксель результирующего изображения,  $T$  – пороговое значение подобрано эмпирически:  $src(x, y) \in [0..255]$ ,  $T = 30$ . Результат сжатия и бинаризации изображения показан на рисунке 5.



Рис. 5. Сжатие и бинаризация изображения

**RAM-based сети.** Низкие вычислительные мощности микроконтроллеров, используемых в мобильной робототехнике, накладывают ограничения на возможности обработки видеоданных в реальном режиме. В данной работе представлена архитектура RAM-based нейронной сети, которая является видом ассоциативной памяти с произвольным доступом. Сеть имеет ряд применений для распознавания образов; например для распознавания лиц и букв. В работе [7] реализована система объезда препятствий на базе микроконтроллера 8051 с 512 байтами памяти. Робот обрабатывал данные восьми инфракрасных датчиков и успешно решал поставленную задачу избегания препятствий в режиме реального времени. Отличительной особенностью сети является простота обучения, тестирования, а так же низкие требования к памяти и вычислительным мощностям.

В основе работы сети лежит теория  $n$  – кортежей. Кортеж – это блок изображения определенного размера. Биноризованное изображение шириной  $lw$  и высотой  $lh$  пикселей разбивается на кортежи, которые подаются на вход сети. Введем обозначение кортежа как  $T$ , а размер кортежа  $Ts$ . Тогда количество кортежей  $Tc$  изображения будет равно:

$$Tc = \frac{lw \cdot lh}{Ts} \quad (2)$$

Изображение, хранимое в памяти сети, называется *дискриминатором*. Одна RAM-based нейронная сеть может хранить  $K$  дискриминаторов. Один дискриминатор состоит из  $Tc$  блоков памяти произвольного доступа (RAM), каждый размером  $Ts$ . В каждом из блоков хранится образ для сравнения с кортежем, получаемым из изображения, подаваемого на вход сети. Если поступающий образ совпадает с образом RAM – на выходе выдается 1, иначе 0.

Количество совпадений RAM с образами на  $i$ -м дискриминаторе рассчитывается по формуле (3):

$$D_i = \sum_{j=1}^{Tc} (T_j \text{ and } RAM_j^i), \quad (3)$$

где  $T_j$  –  $j$ -й кортеж подаваемого на сеть изображения,  $RAM_j^i$  –  $j$ -й блок памяти дискриминатора.

Значение на выходе  $D_i$  сети является показателем соответствия входного изображения обученному образцу. В частном случае,  $D_i$  – это количество совпавших кортежей образа и дискриминатора. Сеть не имеет весовых коэффициентов ассоциированных со связями.

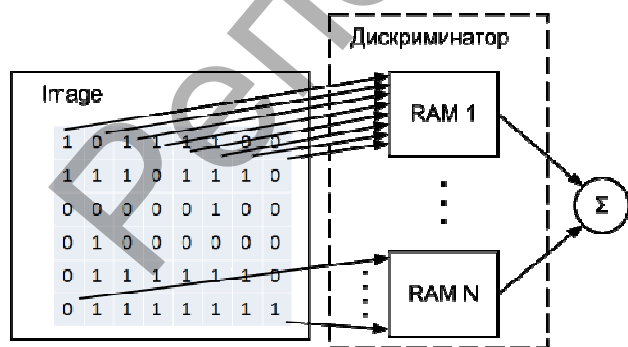


Рис. 6. Дискриминатор RAM-based сети, где  $N = Ts$

Обучение сети производится методом прямого распространения сигнала. Начальные значения образов дискриминаторов равны ну-

лю. Изображение разбивается на  $Tc$  кортежей, размером  $Ts$  каждый. Кортежи поступают на входной слой дискриминатора и сохраняются в постоянную память.

Для каждого хранимого образа сети, как правило, требуется меньше памяти (в зависимости от выбранного размера кортежа), чем размер используемого для распознавания. Размер памяти  $Z$ , необходимый для одной ячейки памяти RAM:

$$Z = 2^{Ts} \quad (4)$$

Размер памяти  $M$ , необходимый для хранения изображения сети:

$$M = Tc \cdot Z \quad (5)$$

**Архитектура RAM-based сети для детектирования паттерна.**

Архитектура сети применяемой для классификации изображения с целью определения сектора, в котором находится ведущий робот, приведена на рисунке 7. Сеть имеет  $d = 6$  дискриминаторов, каждому из которых соответствует свой сектор, в котором может находиться ведущий робот.

Распознавание метки заключается в определении класса дискриминатора соответствующего текущему изображению. Подаваемое на вход сети изображение подвергается обработке, разбивается на кортежи, сворачивается в дискриминатор и последовательно сопоставляется с каждым дискриминатором сети. Предварительно сеть обучается, настраивая каждый дискриминатор для определения положения робота в отдельном секторе изображения. Образ последовательно сравнивается с каждым из дискриминаторов, для каждого из которых рассчитывается количество срабатываний совпадений  $D_i$  по (6).

$$D_i = D_i^{\text{чёрные}} + D_i^{\text{остальные}}, \quad (6)$$

где  $D_i^{\text{чёрные}}$  и  $D_i^{\text{остальные}}$  – это количество совпадений по чёрному и остальным кортежам соответственно, рассчитываемые по формуле (3) с учётом цвета кортежа.

Выходы дискриминаторов  $D_i$  подаются на вход классификатора, который оценивает значение  $D_i$  относительно функции порога дискриминатора  $T_i(D_i)$ . Все дискриминаторы, количество срабатываний которых попадает в диапазон порога, становятся кандидатами в победители.

Итоговый дискриминатор-победитель определяется по максимуму количества срабатываний. Выходом классификатора является номер дискриминатора-победителя, которому соответствует сектор, в котором находится ведущий робот.

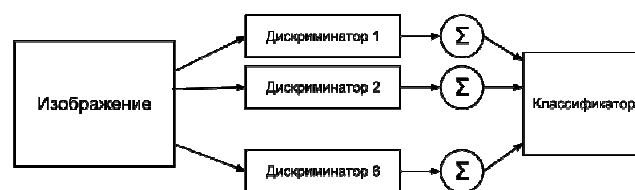


Рис. 7. Архитектура RAM-сети для определения сектора

Описанный подход распознавания ведущего робота на основе RAM-based нейронной сети позволяет определить ориентировочно расстояние (близко или далеко) и направление (лево-центр-право). Расстояние близко означает, что ведущий может располагаться на расстоянии от 25 до 45 см, далеко – от 45 до 70 см.

**Обучение RAM-based сети.** Обучение сети заключается в создании дискриминаторов для каждого из секторов. Размер сектора равен усредненному размеру паттерна ведущего робота. Количество секторов, где может находиться робот, определяется относительными размерами паттерна и рабочей областью изображения. В данной работе один дискриминатор содержит один образ паттерна. Обучающие образы подаются на вход сети, которая запоминает их в соответствующих дискриминаторах. Изображения, используемые в качестве основы для обучающих образов, приведены на рисунке 8. На рисунке 9 показаны обучающие образы после преобразования.

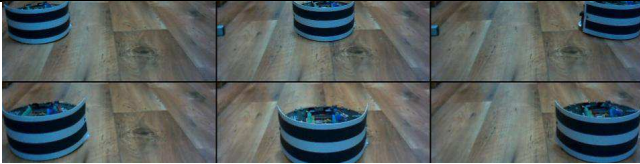


Рис. 8. Паттерны для обучения сети



Рис. 9. Паттерны после предобработки

Обучающие образы отражают эталонное положение ведущего робота относительно ведомого. Следовательно, если это же изображение будет подано на вход сети, количество совпадений на  $i$ -м дискриминаторе будет максимальным. Поэтому во время обучения на дискриминаторе запоминается  $D_i^{черные_{max}}$  – количество совпадений по чёрному цвету, и  $D_i^{остальные_{max}}$  – количество срабатываний по остальным. Конкретные значения этих величин зависят от размера кортежа  $Ts$ .

**Расчет порога Ram-based сети.** Предыдущие работы авторов [8] основывают выбор дискриминатора-победителя по количеству совпадений образов RAM. Простое определение сектора положения лидера на основе дискриминатора победителя в данной работе не представляется возможным из-за неравнозначности размера графической метки робота в дальних и ближних секторах. Ближние сектора содержат паттерн большего размера, и количество срабатываний в ближних секторах будет выше по фону. Следовательно, требуются разные пороги для разных секторов.

Для устранения сбоев введены пороги для каждого из дискриминаторов сети. Значение порогов получены исходя из средневзвешенной суммы результатов срабатывания дискриминатора соответствующего класса (формула 6).

Допустим, на вход сети подается произвольное изображение с роботом в позиции, попадающей в один из секторов. Если изображение близко к эталонному, то количество срабатываний будет стремиться к соответствующим значениям  $D_i^{черные_{max}}$  и  $D_i^{остальные_{max}}$ .

Если положение робота отличается, то количество срабатываний будет соответственно уменьшаться. Суть задания пороговой функции в том, что бы определить, при каких значениях  $D_i^{остальные}$  и  $D_i^{черные}$  метка считается распознанной, если значение количества срабатываний соответствующего дискриминатора попало в допустимый порог.

Для определения порога дискриминатора использовалась выборка положений робота по секторам. Ведущий робот  $P=20$  раз ставился в каждый сектор. Пример обучающей выборки для расчета порогов приведен на рисунке 10.

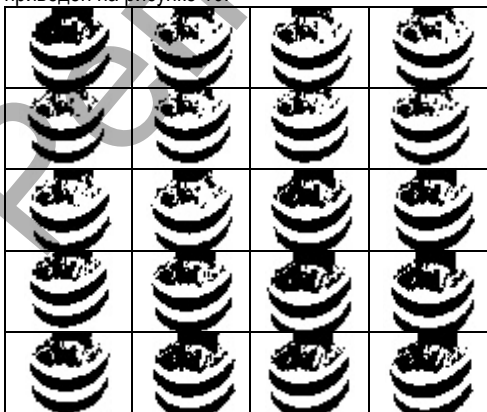


Рис. 10. Обучающая выборка для расчета порога для сектора «далеко-центр»

По каждому изображению для каждого из дискриминаторов высчитывалось срабатываний по полностью чёрным паттернам и по всем остальным. Нижний порог  $i$ -го дискриминатора для чёрных и белых совпадений определяется методом  $K$ -минимальных ближайших соседей.

$$T_i^{c_{min}} = \frac{1}{k} \sum_{t=1}^k D_i^c(t), \quad (7)$$

где  $T_i^c$  – это нижняя граница порога дискриминатора для цвета  $c \in$  (чёрный, остальные),  $k$  – количество нижних соседей, участвующих в расчете нижней границы порога.

Пороговая функция определяется следующим образом:

$$F_i(D_i) = \begin{cases} 1, & \text{если } T_i^{c_{min}} \leq D_i^c \leq T_i^{c_{max}}, \\ 0, & \text{иначе} \end{cases} \quad \text{где } c \in (\text{черные, остальные}). \quad (8)$$

После определения порогов сеть считается рабочей и готова к детектированию. В таблице 1 отображены внешние параметры сети, используемые в данной работе при настройке и обучении сети.

Таблица 1. Внешние параметры сети

№ п/п	Параметр	Обозначение	Значение
1	Размер изображений	ширина $lw$ высотой $lh$	Подбирается эмпирически
2	Количество обученных дискриминаторов в сети	$d$	6
3	Размер кортежа	$Ts$	Подбирается эмпирически
4	Размер дискриминатора	$Tc$	Рассчитывается по (2)
5	Порог бинаризации	$Tb$	Подбирается эмпирически
6	Нижняя граница дискриминатора $D_i^c$ по цвету $c$	$T_i^{c_{min}}$	Обучение, формула (7)
7	Верхняя граница дискриминатора $D_i^c$ по цвету $c$	$T_i^{c_{max}}$	
8	Количество значений в обучающей выборке, участвующие в расчете порога	$k$	Определяется эмпирически $0 < k < P_i$

**Детектирование.** Проведем  $M$  экспериментов детектирования с истинными и ложными целями. Целью экспериментов ставится определение качества детектирования при рассчитанных значениях порогов при заданном размере кортежа, а так же определение оптимального размера кортежа для детектирования метки.

Будем обозначать количество образов с ведущим роботом как  $P$  (от *positive*) и количество образов с ложными целями как  $N$  (от *negative*). Тестовая выборка состоит  $M=60$  образов, случайно подаваемых на вход сети. Количество образов с ведущим роботом  $P=30$ , где на каждом образе ведущий робот случайным образом расположен в некотором секторе. Ложные образы делятся на 3 группы по 10 штук ( $N=30$ ), где на каждом из образов в случайной позиции находится ложный предмет прямоугольной, квадратной или круглой формы.



Рис. 11. Оригинальные и бинаризованное примеры ложных образов разных категорий

Таблица 2. Таблица оценок срабатываний для обученного робота при размере кортежа

Выход сети	Эталонный выход сети	Описание	Размер кортежа				
			2	4	6	8	10
True	Positive	TP – Истинное срабатывание - паттерн детектирован, и он на самом деле есть там, где был определен.	19	21	21	29	30
True	Negative	TN - Паттерна нет, и сеть ничего не распознала.	15	17	13	2	0
False	Positive	FP – Ложное срабатывание – паттерн детектирован, но его на самом деле нет в секторе, т.н. ложные срабатывания, или <i>ошибки первого рода</i> .	15	13	17	28	30
False	Negative	FN – количество необнаруженных паттернов, паттерн находился в секторе, но сеть его не распознала, т.н. пропуск цели, или <i>ошибки второго рода</i>	11	9	9	1	0

Таблица 3. Характеристики и результаты детектирования

Описание	Формула расчета	Размер кортежа				
		2	4	6	8	10
ACC (accuracy) – точность классификации, отражающая качество детектирования	$ACC = \frac{TP + TN}{P + N}$	0.57	0.63	0.56	0.52	0.5
TPR (truepositive rate) – качество обнаружения паттерна, чувствительность, показывает, какой процент паттернов был успешно детектирован	$TPR = \frac{TP}{P}$	1	0.7	0.7	0.97	1
FPR(falsepositive rate) – уровень ложных срабатываний, вероятность ошибки первого рода или <i>уровень значимости</i> :	$\alpha = FPR = \frac{FP}{N}$	0.5	0.43	0.56	0.93	1
FNR (falsenegative rate) – показывает вероятность ошибки второго рода, или уровень ложного обнаружения паттерна	$\beta = 1 - TPR$ $\beta = \frac{FN}{P}$	0.37	0.3	0.3	0.33	0
TNR (truenegative rate) – показывает, в каком проценте экспериментов паттерн не был детектирован	$TNR = \frac{TN}{N}$	0.5	0.57	0.43	0.67	0

Целью разработки алгоритма детектирования паттерна робота является получение классификатора с наибольшей вероятностью обнаружения атак (TPR) при приемлемом уровне ложных срабатываний (FPR).

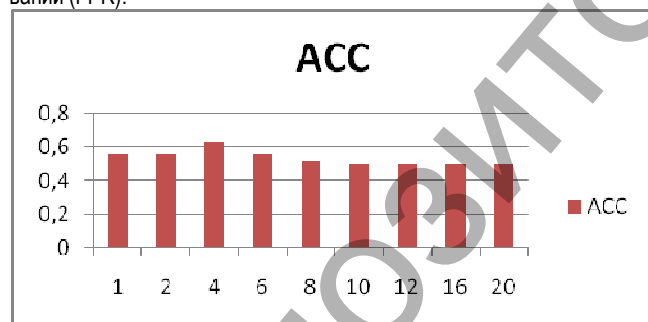


Рис. 12. Точность классификации для всех тестируемых размеров кортежей

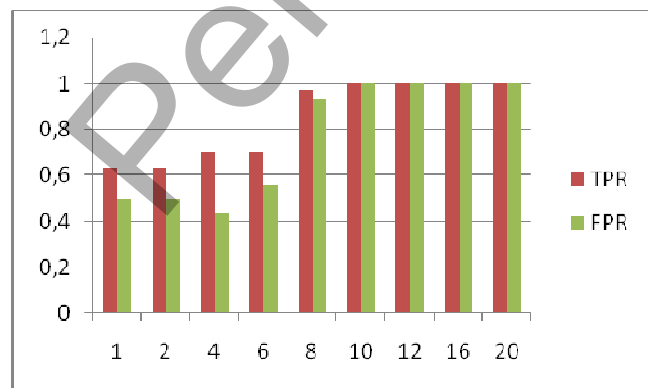


Рис. 13. Зависимость TPR/FNR от размера кортежа

Исходя из рассчитанных параметров, оптимальный размер кортежа для детектирования равен 4.

**Заключение.** В данной работе было показано применение RAM-based сетей для распознавания положения и дальности ведущего робота по графическому паттерну. Обученная RAM-based сеть хранит характеристические особенности паттерна в разных секторах относительно ведомого робота. Предобработанное изображение подается на вход RAM-based сети, которая по количеству совпадений в дискриминаторах определяет, в каком секторе находится паттерн. Порог срабатывания дискриминаторов сети индивидуальным образом характеризует необходимую степень схожести для распознавания паттерна в данном секторе.

Основным преимуществом применения RAM-based сетей для распознавания графических меток является снижение вычислительных ресурсов, что позволяет применять данное решение на платформах с низкой производительностью. Эффективность работы системы обусловлена предобработкой изображения, выбором размера кортежа. Основные отличия данного подхода от других, это малые вычислительные ресурсы, простота обучения сети и простота тестирования сети.

**СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ**

1. Caprari, G. The autonomous miniature robot alice: from prototypes to applications в In IEEE / RSJ International Conference on Intelligent Robots and Systems (IROS'00) / G. Caprari, K.O. Arras и R. Siegwart, 2000.
2. Haverinen, J. A miniature mobile robot with a color stereo camera system for swarm robotics research в IEEE International Conference on Robotics and Automation (ICRA2005) / J. Haverinen, M. Parpala и J. Rning. – Barcelona, 2005.
3. Yao, Q. A RAM-Based Neural Network for CollisionAvoidance in a Mobile Robot Neural Networks, Proc. of IEEE International Joint

- conference / Q. Yao, B. Beetner, D. Wunsch II и B. Osterloh. – P. 3157–3160. – Vol. 4, 2003.
4. Yao, Q. A RAM-Based Neural Network for Collision Avoidance in a Mobile Robotic в Neural Networks, Proc. of IEEE International Joint conference / Q. Yao, D. Beetner. – P. 3157– 3160. – Vol. 4, 2003.
5. Gupta, M. Collective Transport of Robots: emergent Flocking from Minimalist Multi-robot Leader-following в IEEE/RSJ International Conference on Intelligent Robots and Systems / M. Gupta, J. Das, M. Vieira, H. Heidarrsson, H. Vathsangam и G. Sukhatme, 2009.

Материал поступил в редакцию 21.12.12

**DZIOMIN V.V., KABYSH A.S., DUNETS I.P., DUNETS A.P., GOLOVKO V.V. Using RAM-based networks for visual mark detection**

In this paper we consider the problem of detection of the graphic mark of leading robot in a leader following task. The detection method based on RAM-based is purposed. Trained RAM-based network keeps the features of a visual mark in different sectors regarding the slave robot. The network winner-discriminator will indicate the sector in which the leading robot present. The advantage of this approach is reducing the computational resources, so it can be used on platforms with low productivity.

УДК 004.5;621.38

**Бутов А.А.**

**МЕТОД НАХОЖДЕНИЯ БУЛЕВОЙ ФОРМУЛЫ МНОГУГОЛЬНИКА В ДИЗЪЮНКТИВНОЙ НОРМАЛЬНОЙ ФОРМЕ БЕЗ ИСПОЛЬЗОВАНИЯ ДОПОЛНИТЕЛЬНЫХ ПРЕДИКАТНЫХ ПЕРЕМЕННЫХ**

**Введение.** В системах автоматизированного проектирования (САПР) топологии интегральных схем [1, 2] широко применяются способы описания геометрических объектов, основанные на использовании методов аналитической геометрии, векторной алгебры, теории матриц [1–5]. В последнее время появились альтернативные способы описания, основанные на использовании булевых формул: как в скобочной форме [6], так и в более привычной – дизъюнктивной нормальной форме [7, 8]. Простой и приемлемый на практике метод нахождения дизъюнктивной нормальной формы (ДНФ), описывающий прямоугольник, изложен в работе [8]. Однако недостаток этого метода заключается в том, что в найденную им формулу наряду с предикатными переменными, которые связаны с полуплоскостями, порождаемыми сторонами многоугольника, входят еще и дополнительные предикатные переменные, а также инверсии некоторых из них. В настоящей работе предлагается метод построения ДНФ многоугольника, свободный от этих недостатков и позволяющий в большинстве случаев получить более качественное решение, однако характеризующийся и более высокой вычислительной сложностью.

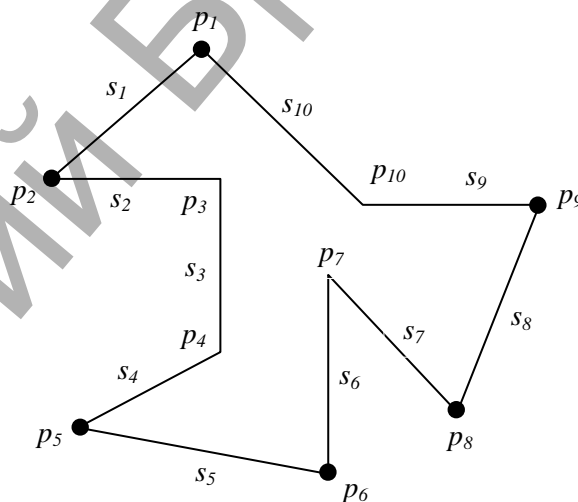
**1. Основные определения, постановка задачи**

Многоугольник, расположенный на плоскости, задается своей границей – замкнутой не пересекающейся ломаной линией, состоящей из отрезков прямых или *сторон* многоугольника. Эту границу можно определить последовательностью *угловых точек* или *вершин* многоугольника, получаемых при обходе его по границе справа:  $p_1, p_2, \dots, p_n$  (рис. 1, где  $n = 10$ ).

Так как каждая пара соседних угловых точек ограничивает соответствующую сторону многоугольника, то его границу можно задать также последовательностью сторон многоугольника:  $s_1, s_2, \dots, s_n$ , где  $s_1 = (p_1, p_2), s_2 = (p_2, p_3), \dots, s_n = (p_n, p_1)$ .

Вершина  $p_1$ , которая служит начальной точкой для последовательного обозначения отрезков, образующих границу многоугольника, называется *начальной*. В качестве начальной будем выбирать вершину, наиболее удаленную от координатной оси  $X$  (если таких вершин несколько, то среди них выбирается вершина, наиболее удаленная от координатной оси  $Y$ ).

Каждой стороне  $s_i$  многоугольника поставим в соответствие *ориентированную прямую*  $v_i$  содержащую точки  $p_i$  и  $p_{i+1}$ . Положим, что она ориентирована от  $p_i$  к  $p_{i+1}$ .



**Рис. 1.** Угловые точки и стороны многоугольника

Рассмотрим некоторую произвольную точку плоскости  $p$ , заданную парой декартовых координат  $(X, Y)$ . Будем считать, что точка  $p$  расположена *слева от прямой*  $v_i$ , если она принадлежит полуплоскости, расположенной слева от ориентированной прямой  $v_i$  или лежит на прямой  $v_i$ . Все возможные варианты левостороннего расположения точки  $p$  относительно ориентированной прямой  $v_i$  представлены на рисунке 2 (последние два варианта соответствуют случаю, когда прямая  $v_i$  параллельна координатной оси  $X$ ).

Как и в работе [6], будем в дальнейшем обозначать отрезки ломаной буквами  $a, b, c, \dots$ , границу многоугольника как  $abc\dots$ , а полуплоскости, расположенные слева от соответствующих ориентированных прямых – буквами  $A, B, C, \dots$  (считая, что каждая из этих полуплоскостей включает в себя еще и все точки порождающей ее ориентированной прямой). Введем также предикаты  $a, b, c, \dots$  для описания положения некоторой точки  $p$  на плоскости, полагая, что  $a(p) = 1$ , если и только если  $p \in A$ .

Основываясь на таких предикатных переменных, в работе [6] описан метод построения скобочной булевой формулы  $F$ , представляющей многоугольник и обладающей следующим свойством: если выполнить подстановку предикатных координат произвольной точки плоскости, то

**Бутов А.А.**, к.т.н., доцент кафедры экономической кибернетики Белорусского государственного университета информатики и радиоэлектроники. Беларусь, БГУИР, 220013, г. Минск, ул. П. Бровки, 6.