

## Многозадачная система распознавания образов реального времени

А.Ю. Кручинин (ГОУ ВПО "Оренбургский государственный университет")

Представлена многозадачная система распознавания образов, работающая в РВ и позволяющая добиться максимального значения достоверности при решении нескольких периодических задач распознавания образов.

Ключевые слова: многозадачная система, распознавание образов, РВ, достоверность, адаптивная система, эффективность распознавания образов.

Современные системы распознавания образов (СРО) РВ в большинстве случаев предназначены для решения одной конкретной задачи и реализуются в виде отдельного модуля. Решение на той же вычислительной системе одновременно других задач распознавания образов либо той же задачи для другого канала данных (например, видеопотока) осуществляется введением новой независимой системы, работающей параллельно, либо путем обращения к тому же программному модулю, что не всегда позволяет оптимально распределить ресурсы вычислительной системы. Тенденция к увеличению числа решаемых задач распознавания на одной вычислительной системе приводит к необходимости разработки новых подходов к организации структуры СРО РВ. При этом система должна быть адаптивной, надежной и эффективной.

Чтобы построить адаптивную СРО РВ можно воспользоваться подходом, описанным в работе [1], где показано, как в зависимости от поступающих данных система подстраивается под них. В работе [2] рассмотрены особенности разработки ПО для распознавания образов и предложена общая структурная схема СРО РВ, которую можно использовать и при проектировании многозадачной СРО РВ. На рис. 1 представлена модель многозадачной СРО РВ, построенная на основе описанных в работах [1, 2] принципов.

Информация для распознавания образа  $X$  поступает с аппаратного обеспечения или программных библиотек, не входящих в состав СРО, на подсистемный сборщик данных, назначение которой состоит в формировании очереди заданий  $\lambda$ . По мере посту-

пления заданий планировщик распределяет каждую задачу определенному агенту распознавания. Агент распознавания – это отдельный модуль (обычно отдельный процесс). В каждый момент времени  $t$  планировщику доступно только одно задание из последовательной входной очереди:

$$\lambda(t) = x_c, x_c = (I_c, P_c, T_c, Z_c),$$

где  $x_c$  – текущее задание,  $I_c$  – данные для распознавания,  $P_c$  – приоритет текущего задания,  $T_c$  – ограничение по времени выполнения,  $Z_c$  – тип задания (для выбора класса агента).

В частных случаях распределение приоритетов может быть различным, однако в общем можно выделить три класса приоритетов:

- 0 – приоритет РВ;
- 1 – приоритет разделения времени (отсутствует жесткое ограничение времени);
- 2 – низкий приоритет для задач не связанных с распознаванием образов.

Планировщик задач СРО РВ поддерживает список заданий и список агентов (свободных и работающих). Свободный агент блокирован на механизме синхронизации ОС и может быть разблокирован планировщиком по мере необходимости, что происходит после отправки требуемой для работы информации  $\lambda'$ :

$$\lambda'(t) = x'_c, x'_c = (I'_c, P'_c),$$

где  $I'_c$  – указатель на данные (либо, если агент удаленный, сами данные),  $P'_c$  – процедура установки приоритета для агента.

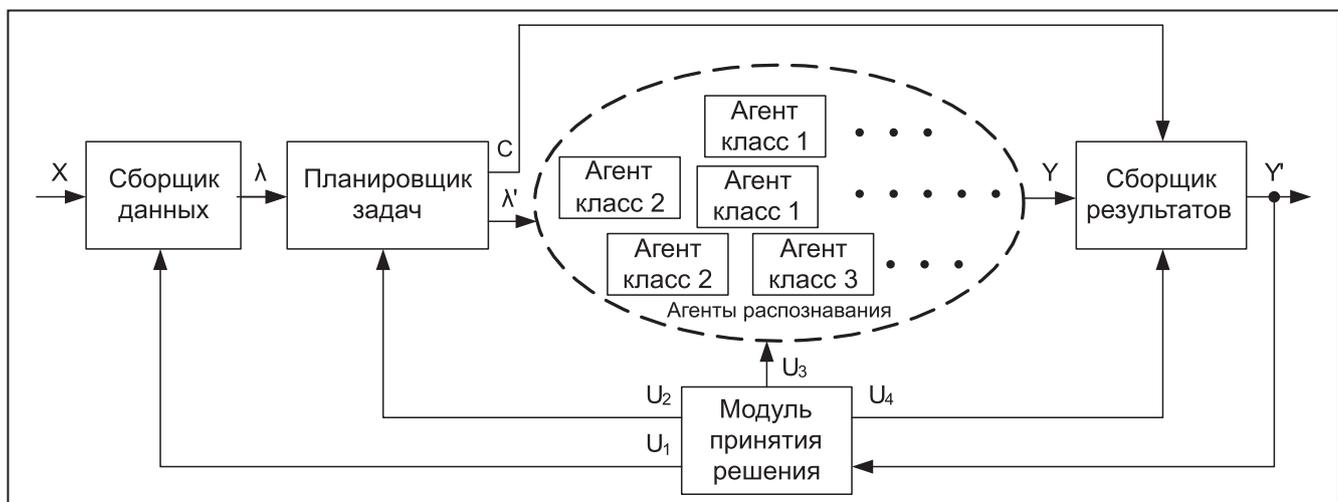


Рис. 1. Модель многозадачной СРО РВ

Алгоритм работы планировщика задач:

- поиск агента нужного типа для решения текущего задания;
- завершение работы с данными  $I_c$  и освобождение критической области для разрешения доступа агенту распознавания;
- установка приоритета для агента (процесса) распознавания;
- установка таймера для задачи РВ;
- разблокирование агента.

При срабатывании таймера планировщик должен проверить выполнение задачи. Если задача еще не выполнена, то необходимо принудительно завершить работу агента распознавания. Информация ( $C$ ) о выполненных и невыполненных задачах, свободных и занятых агентах передается сборщику результатов и может затем использоваться модулем принятия решения. По сути, назначение планировщика задач сходно с функциями планировщика доступа в системах пакетной обработки [3].

Агенты распознавания являются независимыми друг от друга и после выполнения задания формируют результат распознавания  $Y$ . Сборщик результатов формирует выходную последовательную очередь  $Y' = \{Y, C\}$ , которая также поступает на модуль принятия решения.

Модуль принятия решения осуществляет управление многозадачной системой автоматически или при помощи пользователя системы. Основные функции управления:

- $U_1$  – настройки режима сбора данных (например, частота кадров видеокамеры);
- $U_2$  – настройки режимов работы планировщика (обновление списка агентов, настройка алгоритма планирования);
- $U_3$  – создание/удаление агентов;
- $U_4$  – настройка режимов сохранения результатов.

Данные функции управления позволяют построить адаптивную СРО РВ путем влияния на собираемую информацию и выбор оптимального алгоритма распознавания.

Описанная модель может быть реализована как в виде отдельных процессов, так и отдельных потоков. При реализации потоков нет одной части проблемы межпроцессного взаимодействия – передачи информации от одного процесса другому. Однако использование потоков обладает одним недостатком – при возникновении ошибки в одном программном потоке, вероятно, что и остальные потоки прекратят свою работу, то есть у такого решения надежность ниже. А при разделении системы на процессы необходимо обеспечить быстрый обмен и доступ к общим данным.

Известно, что для реализации системы РВ необходимо выполнение следующего условия [3]:

$$\sum_{i=1}^m \frac{\tau_i}{\Delta t_i} \leq 1, \quad (1)$$

где  $i$  – номер периодического события,  $m$  – число периодических событий,  $\Delta t_i$  – период события  $i$ ,  $\tau_i$  –

длительность обработки события  $i$ . В дальнейшем отношение  $\Delta i / \Delta t_i$  будем обозначать, как  $R_i$ . Система РВ тогда будет поддаваться планированию (возможна для реализации), если будет в состоянии своевременно обработать все события, что показано в выражении (1).

Если предположить, что ресурсов оперативной памяти достаточно для решения поставленных задач, то для реализации СРО РВ необходимо, чтобы все программные компоненты системы своевременно получали ресурсы центрального процессора (для упрощения исключим возможность использования видеокарт для реализации алгоритмов распознавания). Невозможно 100% ресурсов процессора использовать только для процессов СРО, например, есть необходимость тратить время на переключения между процессами, реализацию функций ОС и решение некоторых неперiodических задач (запросы пользователя о состоянии системы). Поэтому для СРО должно быть предоставлено  $R_p$  ресурсов процессора. При этом значение  $R_n$ , равное  $(100\% - R_p)$ , представляет собой необходимый процент ресурсов для обработки неперiodических событий и служебных функций ОС.

Поскольку в системах РВ всегда необходимо исходить из максимально возможного времени выполнения процедуры, то  $R_p$  соответствует максимально возможному количеству ресурсов ( $R_{max}$ ) центрального процессора, потребляемых процессами. При этом среднее значение потребляемых ресурсов ( $R_{cp}$ ) может быть значительно меньше.

Конечно, при реализации СРО РВ желательно, чтобы вычислительная машина обладала максимальной производительностью, которая бы с запасом позволяла решать все поставленные задачи с высокой достоверностью, однако в большинстве случаев ресурсы ограничены, поэтому при проектировании и эксплуатации СРО РВ необходимо исходить из компромисса между уровнем достоверности и быстродействия системы [1, 2]. При этом ресурсы системы не должны простаивать, если есть возможность их использования. Исходя из этого, целевая функция эффективности функционирования многозадачной СРО РВ с заданными вычислительными ресурсами принимает следующий вид:

$$E = \max D_{общ}; \text{ при } D_i \geq D_{i-3}, \sum_{i=1}^m R_i \leq R_{max}, \quad (2)$$

где  $D_{общ}$  – общая достоверность результатов распознавания,  $i$  – периодическое событие ( $i = \{1, 2, \dots, m\}$ ),  $D_i$  – фактическое значение достоверности результата распознавания  $i$ -го периодического события,  $D_{i-3}$  – заданное значение достоверности результата распознавания  $i$ -го периодического события,  $R_i$  – требуемые ресурсы центрального процессора для  $i$ -го события в единицу времени.

Необходимо стремиться к максимально возможному использованию ресурсов  $R_{max}$  при сохранении требуемых значений достоверности для каждого периодического события. Однако здесь возникает вопрос, как

Таблица . Матрица средних значений достоверности распознавания ( $D$ ) и требуемых в среднем ресурсов ( $R$ )

Алгоритмы распознавания	Режимы формирования и предварительной обработки данных			
	1	2	...	$N$
1	$D_{11}, R_{11}$	$D_{12}, R_{12}$	...	$D_{1N}, R_{1N}$
2	$D_{21}, R_{21}$	$D_{22}, R_{22}$	...	$D_{2N}, R_{2N}$
...	...	...	...	...
$M$	$D_{M1}, R_{M1}$	$D_{M2}, R_{M2}$	...	$D_{MN}, R_{MN}$

и в каких задачах улучшать достоверность распознавания образов? Какой задаче выделить больше ресурсов центрального процессора, чем необходимо для поддержания заданного уровня  $D_{i,3}$ ? Решением данного вопроса должен заниматься модуль принятия решения.

Для управления достоверностью распознавания образов возможно два пути:

- изменение объема и качества поступающих данных;
- изменение алгоритма распознавания или его настроек.

Для каждой задачи можно построить матрицу средних значений достоверности в зависимости от поступающих данных и алгоритма распознавания. Пусть система поддерживает  $N$  режимов предварительной обработки данных, которые упорядочены последовательно (увеличение достоверности), а также  $M$  алгоритмов и/или конфигураций алгоритма, упорядоченных аналогичным образом (таблица). Пусть наиболее быстрые режимы не подходят для выполнения поставленной задачи по причине низкой достоверности, что помечено в таблице серым. Тогда доступными режимами являются все остальные, поскольку они не ограничены  $D_{i,3}$ , а максимально возможное использование ресурсов ограничивается в выражении (2) общей суммой.

Для каждого типа задач можно построить свою матрицу режимов, и тогда выбор оптимального режима работы многозадачной системы согласно критерию (2) сводится к перебору всех возможных конфигураций и выбору наиболее подходящей. Для этого необходимо для каждой конфигурации получить общее значение достоверности и величину используемых ресурсов. Требуемые ресурсы вычисляются по выражению (1), а достоверность определяется следующим образом:

$$D_{\text{общ}} = \frac{\left( \frac{D_1}{\Delta t_1} + \frac{D_2}{\Delta t_2} + \dots + \frac{D_m}{\Delta t_m} \right)}{\left( \frac{1}{\Delta t_1} + \frac{1}{\Delta t_2} + \dots + \frac{1}{\Delta t_m} \right)}. \quad (3)$$

В данном выражении учитывается, сколько раз в единицу времени вызывается каждая задача, что определяет вклад в общую достоверность распознавания образов.

В работе [1] описан подход к управлению процессом распознавания на основе анализа сложности распознавания образов, а в работе [2] показана эффективность такого управления при зависимости образов в поступающей на распознавание последовательно-

сти друг от друга, например, видеопоток с видеокamerы. Однако в таком случае усложняется вычисление общей достоверности, поскольку выражение (3) преобразуется в функционал:

$$D_{\text{общ}} = \frac{\left( \frac{D_1(x_1(t))}{\Delta t_1} + \frac{D_2(x_2(t))}{\Delta t_2} + \dots + \frac{D_m(x_m(t))}{\Delta t_m} \right)}{\left( \frac{1}{\Delta t_1} + \frac{1}{\Delta t_2} + \dots + \frac{1}{\Delta t_m} \right)}.$$

где  $x_i(t)$  – данные для распознавания.

При таком подходе к управлению может возникнуть ситуация, когда достоверность распознавания будет увеличена до максимума в одной или нескольких периодических задач распознавания, а в остальных останется равной заданному уровню. Поэтому при необходимости можно ввести дополнительные критерии, однако задача оптимизации в таком случае становится многокритериальной, и оптимальное решение следует искать на основе компромиссов Парето.

Чтобы показать преимущество многозадачной СРО РВ перед системой независимого распознавания рассмотрим простейший пример. Пусть имеются две схожие периодические задачи распознавания, которые поступают с одинаковым периодом времени  $\Delta t_1 = \Delta t_2$ . Например, два видеопотока с видеокamer. Пусть сложность распознавания второго видеопотока значительно больше, чем первого ( $S1_1 \ll S1_2$ ). Например, на первой камере идеальные условия для наблюдения за образами, а вторая камера находится в условиях повышенного шума, плохого освещения и качество изображения плохое. То есть изображение с первой камеры значительно проще распознать. Если используется независимое распознавание, то тогда в общем случае возможны два варианта:

- СРО РВ настроена так, что самые сложные образы ( $S1_2$ ) распознавались с уровнем достоверности не меньшим заданного ( $D_3$ ), хотя для первой периодической задачи заданный уровень можно получить с более низкими затратами по производительности:  $R_1 = R_2, D_1 > D_2$ ;
- ресурсы процесса делятся изначально, например, в пропорциях 30 на 70% между двумя задачами, в этом случае процент загрузки зависит от распределения пользователем, а достоверность от количества ресурсов и сложности задачи.

Второй вариант предусматривает многозадачную систему, где роль планировщика отводится пользователю, что является неоптимальным и неправильным. Например, во время работы системы качество изображения на одной из камер резко изменилось, но системы не могут перенастроиться без вмешательства пользователя, а это значит возможное снижение достоверности распознавания или неоптимальное использование ресурсов.

В отличие от этих вариантов многозадачная СРО РВ предполагает максимальное использование ресурсов ( $R_{\text{max}}$ ), а значение общей достоверности максимальным:

$$D_1 + D_2 = \text{max}.$$

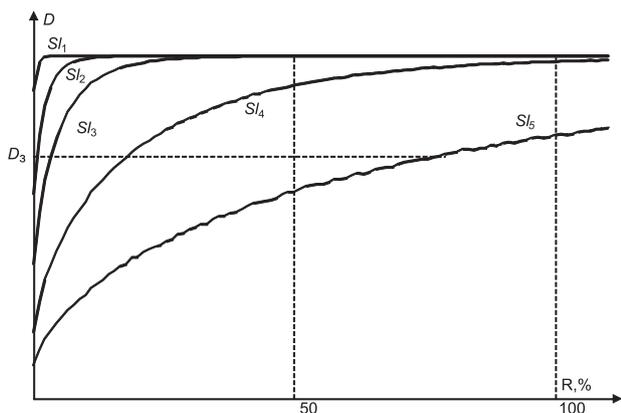


Рис. 2. Зависимость достоверности распознавания образов (D) от затраченных ресурсов (R) при различном значении сложности

При одинаковой величине выделяемых на распознавание ресурсов ( $R_{max}$ ) общая достоверность многозадачной СРО РВ всегда будет больше или равна общей достоверности аналогичных независимых систем.

Концептуально зависимость достоверности распознавания образов от процента используемых ресурсов представлена на рисунке 2. Как видно из рисунка, если использовать один модуль с одинаковыми настройками распознавания, то может возникнуть ситуация, когда при одной сложности распознавания ( $Sl_1$ ) напрасно расходуются 50% ресурсов процессора, в то время как для  $Sl_5$  ресурсов будет недостаточно, и значение достоверности будет меньше  $D_3$ . Если же использовать многозадачную систему, то можно на распознавание со сложностью  $Sl_1$  выделить меньше ресурсов, что позволит выполнить условие (2).

Программные системы РВ реализуются на ОС РВ, например, QNX Neutrino, Windows CE, или на обыч-

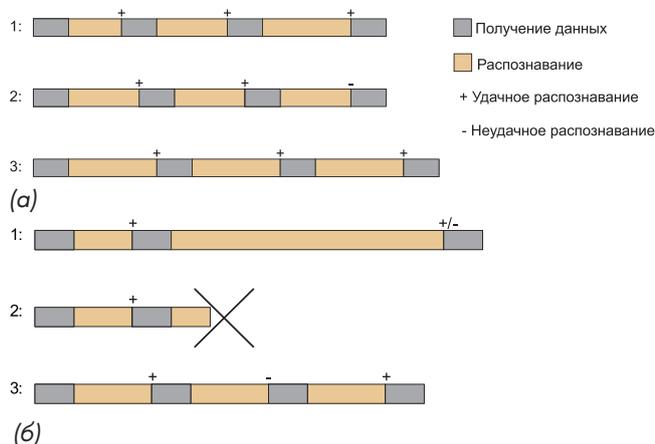


Рис. 3. Временные диаграммы процесса распознавания образов в РВ: а) обычная система (1); СРО РВ с малым (2) и большим (3) интервалами распознавания; б) обычная система с ошибками задержки времени распознавания (1) и зависанием (2), СРО РВ с теми же ошибками (3)

**Кручинин Александр Юрьевич** — канд. техн. наук, старший преподаватель ГОУ ВПО "Оренбургский государственный университет".

Контактный телефон (3532) 27-09-12.

E-mail: [kruchinin-al@mail.ru](mailto:kruchinin-al@mail.ru), <http://vidikon.com>

ных ОС (Windows, Linux) с расширениями РВ. Особенностью подобных ОС является поддержка вытеснения потоков, что позволяет полностью вытеснить потоки с меньшим приоритетом при поступлении более важных потоков. В качестве примера построения структуры была разработана программная имитационная модель (Pattern recognition system – Real time), работающая на обычной версии ОС Windows XP. Исходный код модели реализован на языке С и с версией 1.0 выложен под лицензией BSD на сайте <http://vidikon.com>. Данная модель реализует принципы, описанные в работе [2] и с некоторыми доработками может быть использована для построения многозадачной СРО РВ.

Главным достоинством программной модели СРО РВ является надежность функционирования, выраженная в устойчивости к ошибкам и исполнении заданий точно в срок. Недостаток – более низкая производительность. Достоинства и недостатки проиллюстрированы на рис. 3.

На рис. 3 а показано, что при стабильном успешном распознавании (диаграмма 1) обычная система работает лучше СРО РВ с точки зрения достоверности (диаграмма 2) или производительности при увеличении интервала распознавания (диаграмма 3). В данном случае преимущество СРО РВ в том, что она предсказуема и может работать синхронно с АСУ. На рис. 3 б показано, что при возникновении ошибок СРО РВ ведет себя намного лучше: при значительном превышении времени распознавания (диаграмма 1) результат распознавания уже не важен; при неисправимой ошибке или зависании (диаграмма 2) обычная система нуждается во внешнем вмешательстве для продолжения работы. В обоих случаях СРО РВ ведет себя одинаково: процесс распознавания принудительно завершается, переключаясь на другой процесс без потери времени (диаграмма 3).

В обмен на некоторые затраты процессорного времени и усилия в синхронизации процессов и потоков система приобретает предсказуемость, фиксированное время распознавания и безболезненную обработку неисправимых ошибок распознавания. А при добавлении многозадачности СРО РВ позволяет оптимально управлять ресурсами вычислительной системы.

**Список литературы**

1. Кручинин А.Ю. Управление процессом распознавания образов в реальном времени//Автоматизация и современные технологии. 2010. № 3.
2. Кручинин А.Ю. Особенности разработки программных систем распознавания образов реального времени//Автоматизация в промышленности. 2011. № 7.
3. Таненбаум Э. Современные операционные системы. 3-е изд. СПб.: Питер. 2010.