

## Интерпретатор языка SOQL для обработки растровых изображений

И. В. Бычков, Г. М. Ружников, Р. К. Фёдоров\*, Ю. В. Авраменко

Институт динамики систем и теории управления им. В.М. Матросова, Иркутск, Россия

\*Контактный e-mail: fedorov@icc.ru

Для повышения эффективности идентификации объектов на растровых изображениях необходимо использовать априорные знания, которые могут значительно меняться в зависимости от решаемой задачи. Для существующих методов идентификации характерно использование фиксированных наборов знаний. Это связано со сложностью формализации знаний и их применения в процессе идентификации. В данной статье приводится описание интерпретатора языка Spatial Object Query Language (SOQL) для обработки растровых изображений. Реализованный в интерпретаторе метод позволяет пользователю использовать различные знания в процессе идентификации.

*Ключевые слова:* интерпретатор, логические методы, язык Пролог, идентификация объектов на изображении, распознавание образов.

### Введение

В области компьютерного зрения активно развиваются различные методы анализа и обработки растровых изображений. Одной из важных задач является идентификация объектов определенной формы с заданными текстурными и спектральными свойствами. В существующих методах обычно осуществляется идентификация заранее заданного класса объектов, который пользователь не может значительно изменить. Ограниченность методов обусловлена сложностью изменения набора знаний, используемых при идентификации объектов. На разных изображениях в зависимости от решаемых задач большее или меньшее значение могут приобретать те или иные свойства объектов или их совокупность. В связи с этим актуальными проблемами являются формализация знаний об объекте и их применение в процедуре идентификации.

Так, в статье [1] предложено использовать деформируемые модели, что позволяет выбирать положение границ объектов не только на основе локальных характеристик, но и с учетом пространственного расположения фрагментов. Форма объекта определяется с помощью деформируемой модели и функции энергии. В работе [2] описаны упрощенная и дискретная версия применения деформируемых моделей и методы локальной оптимизации. Использование методов динамического программирования позволило применить более общий класс ограничений формы объектов [3].

Среди работ российских ученых следует отметить [4], где знания об объекте используются в виде древовидного представления примитивов. Одним из подходов к использованию знаний об объектах является их формализация с помощью некоторого языка.

Так, в работе [5] для распознавания объектов применен язык Пролог, в котором переборный механизм логического вывода решений для задачи распознавания оказался очень гибким и мощным. В этом случае информация с изображения представляется до выполнения логического вывода в виде фактов языка Пролог. Основная сложность данного метода заключается в переводе знаний, полученных с изображения, в факты. Так, на нечетких, размытых изображениях фактов, определяющих возможное положение границы объекта, может быть значительное количество. В статье [6] даны примеры пространственных отношений, которые обосновывают использование аппарата теории нечетких множеств. В работе [7] приводятся логический язык PDL, позволяющий описывать объекты с учетом нечетких отношений, и механизм вывода. Однако отсутствуют примеры практического применения данного языка, не разработаны средства борьбы со сложностью логического вывода.

Таким образом, различные подходы к использованию знаний об объектах при их распознавании и идентификации активно развиваются, но следует признать, что они недостаточно проработаны. Необходимо создавать и развивать формализмы для их представления, позволяющие учитывать структурные, текстурные, топологические свойства объектов при их распознавании, а также эффективно интерпретировать знания на изображении и идентифицировать объекты. Для того чтобы учитывать возможную сложность знаний (иерархичность описания свойств объектов, наличие пространственных отношений, необходимость дизъюнктивных ветвлений и т. д.), наиболее подходит формальный аппарат языка Пролог III [8].

В настоящей статье на основе деформируемых моделей и языка Пролог III предложен разработанный авторами язык Spatial Object Query Language (SOQL) [9], позволяющий пользователю представлять знания и требования к идентифицируемым объектам, и показана эффективность предложенного интерпретатора для идентификации объектов на растровых изображениях.

## 1. Язык Spatial Object Query Language

Язык SOQL базируется на языке Пролог, он дополнен такими конструкциями, как:

- 1) терм  $p(x, y)$ , описывающий положение точек на изображении (область определения  $x, y$  ограничена размерами изображения);
- 2) множество пространственных ограничений;
- 3) встроенные предикаты.

Пространственные отношения описываются при помощи предложений вида  $\langle \text{выражение} \rangle \langle \text{оператор сравнения} \rangle \langle \text{выражение} \rangle$ . Под выражением понимаются константа, функция, арифметическое выражение и т. д. На текущий момент реализованы следующие функции:

- $angle(p_1, p_2, p_3)$  — возвращает значение угла, образованного отрезками  $[p_1, p_2]$ ,  $[p_2, p_3]$ ;
- $dist(p_1, p_2)$  — возвращает расстояние между точками  $p_1, p_2$ .

Операторы сравнения:  $=, <, >, \leq, \geq$ .

Для получения информации из изображения разработаны следующие встроенные предикаты:

- $line(p_1, p_2)$  — поиск прямолинейных фрагментов границы объекта;
- $circle(p, r)$  — поиск объектов с границей в виде окружности с центром в точке  $p$  и радиусом  $r$ ;

- $texture([p_1, \dots, p_k], ModelFile)$  — определение соответствия спектральных и текстурных свойств изображения, ограниченного множеством точек  $[p_1, \dots, p_k]$ , заданной модели, расположенной в файле  $ModelFile$ .

Приведем пример правила для нахождения треугольника определенной формы:

$$triangle(A, B, C) : -line(A, B), line(B, C), line(C, A),$$

$$angle(A, B, C) = 135, dist(A, B) = 10, dist(B, C) = 20.$$

Каждый объект, удовлетворяющий логической модели, оценивается функцией энергии. Чтобы описать функцию энергии, введем  $InBP$  — множество встроенных предикатов. Описание объекта будем задавать набором правил вида  $P: -P_1, \dots, P_n, C_1, \dots, C_m$ , где  $P_i$  —  $i$ -й предикат в теле правила  $P$ ;  $C_j$  — пространственное  $j$ -ограничение в теле правила  $P$ ;  $i \in [1; n]$ ,  $j \in [1; m]$ . Значение функции энергии вычисляется по формуле

$$L(s_1, \dots, s_k) = \begin{cases} -1, & \text{если } n = 0; \\ -\mu_i(s_{i_1}, \dots, s_{i_a}), & \text{если } P_i \in InBP, n = 1; \\ L_i(s_{i_1}, \dots, s_{i_a}), & \text{если } P_i \notin InBP, n = 1; \\ \frac{1}{n} \sum_{i=1}^n L_i(s_{i_1}, \dots, s_{i_a}), & \text{если } n > 1. \end{cases}$$

Здесь  $n$  — количество предикатов в теле правила  $P$ ;  $\mu_i(s_{i_1}, \dots, s_{i_a})$  — значение функции принадлежности встроенного предиката  $P_i$ ;  $\mu_i \in [0; 1]$ ;  $s_i$  — переменная, соответствующая положению точки,  $s_i, s_{i_1}, \dots, s_{i_a} \in S$ ,  $S$  — множество точек;  $L_i$  — значение функции энергии  $P_i$  предиката в теле правила  $P$ .

Функция энергии вычисляется как среднее арифметическое значение функций энергии соответствующих дочерних предикатов в теле правила. Идентификация сводится к нахождению множества точек (определяющих положение объекта), удовлетворяющих логической модели и приводящих функцию энергии к локальному минимуму  $L_i(s_1, \dots, s_k) \rightarrow \min_{s_i \in S}$ . Объекты, для которых выполняется неравенство  $L_i(s_1, \dots, s_k) < \tilde{L}$ , где  $\tilde{L}$  — порог, являются решением.

Функции принадлежности  $\mu_i(s_{i_1}, \dots, s_{i_a})$  оценивают наличие графического примитива на определенной позиции растрового изображения или соответствие изображения заданной текстуре и спектральным свойствам. Оценка производится в интервале от нуля до единицы. Ноль означает полное отсутствие соответствующего свойства объекта на текущей позиции изображения, единица — явно выраженное присутствие.

При наличии множества локальных минимумов и возможности включения различных функций принадлежности с заранее неизвестными характеристиками требуется полный перебор возможных положений объекта в соответствии с заданной логической моделью.

## 2. Реализация интерпретатора языка SOQL

Работа интерпретатора основывается на классическом переборе с возвратами языка Пролог, дополненного механизмом отсечения неперспективных ветвей логического вывода путем проверки пространственных ограничений, отсечения по порогу, применения алгоритма  $A^*$  [10] и мултистарта. Структура интерпретатора приведена на рис. 1.

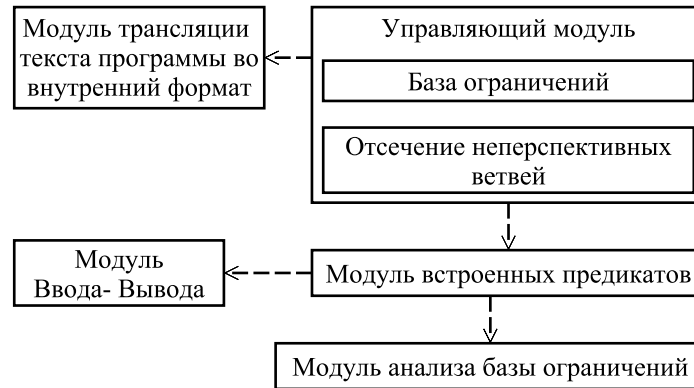


Рис. 1. Интерпретатор языка SOQL

### 2.1. Управляющий модуль

Управляющий модуль реализует переборный механизм с возвратами — поиск в глубину [11]. В процессе поиска в глубину производится накопление ограничений и связывание переменных ограничений со слотами, выделенными для хранения значений переменных правил. Для каждого найденного решения (ветки дерева перебора в глубину) вычисляется функция энергии. Сохраняются решения, которые имеют значения функции энергии меньше заданного порога. Идентификация объектов проводится в предположении, что границы объектов не пересекаются. Из значений меньше заданного порога, которым соответствуют пересекающиеся альтернативы положения объекта, выбирается наименьшее. Для отсекаания альтернативных решений с заведомо худшим значением переборный механизм модифицирован на основе алгоритма  $A^*$ .

Рассмотрим работу модуля на примере правила нахождения двух смежных ребер:

$$edges(A, B, C) : -line(A, B), line(B, C), angle(A, B, C) = 60.$$

Здесь предикат  $edges(A, B, C)$  — голова правила; предикаты  $line(A, B)$ ,  $line(B, C)$  — дочерние или тело правила;  $angle(A, B, C) = 60$  — пространственное ограничение на расположение ребер.

Для каждой унификации предиката создается фрейм, как показано на рис. 2. Фрейм, соответствующий вызову того или иного предиката, содержит значение функции принадлежности (слот  $e$ ), наилучшее значение функции энергии (слот  $record$ ) среди всех пересекающихся альтернатив и сумму функций всех дочерних предикатов (слот  $sum$ ). Эти слоты используются для определения перспективности продолжения поиска в глубину текущей ветви. Перед унификацией встроенного предиката на значения входящих в него переменных накладываются ограничения. При унификации встроенного предиката входящим в него переменным присваиваются значения, удовлетворяющие пространственным ограничениям. Затем вычисляется функция принадлежности, значение которой заносится в слот  $e$ . В родительском фрейме в слот  $sum$  добавляется значение слота  $e$ . Производится проверка на пересечение найденного примитива с другими альтернативами. Если такая альтернатива найдена, то рассматривается функция энергии найденной альтернативы. Если она лучше, то она помещается в слот  $record$  текущего фрейма. При откате обновляются слоты  $sum$  и  $e$  родительского фрейма.

Приведем алгоритм отсекаания неперспективных ветвей логического вывода путем применения алгоритма  $A^*$ . Смысл алгоритма заключается в следующем. Если в про-



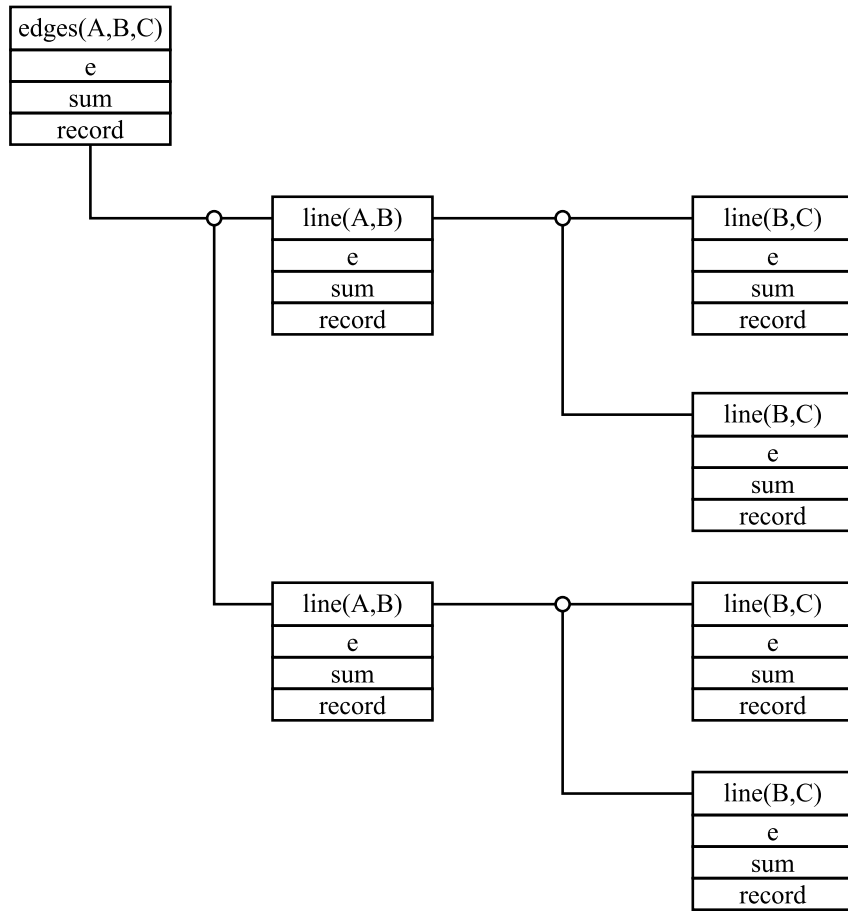


Рис. 2. Дерево логического вывода

цессе поиска в глубину для текущей ветки лучшее возможное решение хуже альтернативного, то продолжать эту ветку нет смысла. Допустим, что в процессе логического вывода в теле правила имеется последовательность  $\{P_1, \dots, P_{j-1}\}$  унифицированных и  $\{P_j, \dots, P_n\}$  неунифицированных предикатов,  $1 < j < n$ , где  $n$  — число предикатов в теле правила. Унифицированная последовательность частично определяет положение текущей альтернативы объекта. Наилучшее возможное значение функции энергии (оценка снизу) в этом случае вычисляется по формуле

$$H(s_1, \dots, s_k) = \frac{1}{n} \left[ \sum_{i=1}^{j-1} L_i(s_{i_1}, \dots, s_{i_b}) + \sum_{i=j}^n h_{L_i}(s_{i_1}, \dots, s_{i_b}) \right],$$

где  $h_{L_i}(s_{i_1}, \dots, s_{i_b})$  — оценка снизу функции энергии оставшихся дочерних предикатов  $\{P_j, \dots, P_n\}$ . Оценку снизу грубо можно построить следующим образом:  $h_{L_i}(s_{i_1}, \dots, s_{i_b}) = -1$ . Примем, что для всех оставшихся дочерних предикатов значение их функций принадлежности будет равно наилучшему значению. Необходимо произвести откат и выбрать следующую альтернативную унификацию для предиката  $P_{j-1}$ , если выполняется неравенство

$$H(s_1, \dots, s_k) > \tilde{L}.$$

Полученная таким способом оценка позволяет исключать ветки логического вывода, заведомо не содержащие лучшего решения, чем уже найденное.

## 2.2. Модуль встроенных предикатов

Модуль встроенных предикатов содержит их алгоритмы унификации, позволяющие получать информацию из изображения в виде набора унифицированных переменных с нечеткой оценкой (функцией принадлежности). Схема взаимодействия модулей для унификации встроенных предикатов приведена на рис. 3. Модуль встроенных предикатов использует следующие функции:

- сравнение гистограмм — определение фрагмента контура;
- формирование множества начальных точек — реализация детектора углов Харриса [12];
- формирование вектора признаков, учитывающих текстуру объектов с помощью оператора LBP (Local Binary Pattern) [13];
- реализация метода опорных векторов для проверки текстуры объекта на соответствие эталону с применением библиотеки SVM [14];
- определение множества точек ограниченного контуром объекта с использованием библиотеки GEOS [15] пространственных функций и операторов.

Приведем общее описание реализации встроенных предикатов. Для интерпретатора предикат — это некоторая функция, которая при каждом вызове соответствующим переменным присваивает конкретные значения и в слот  $e$  текущего фрейма заносит значение функции принадлежности, если оно ниже порогового. Функция должна сохранять контекст, чтобы для каждого вызова возвращать новый набор значений переменных.

Взаимодействие встроенных предикатов с интерпретатором происходит через единый интерфейс. Интерфейс определен как функциональный тип данных

$$\text{BuiltProc}(F, \text{LastReturnPoint}, \text{BuildVars}, \text{CurThreshold}, \text{CurEst}) : \text{Boolean}.$$

Здесь  $F$  — текущий фрейм,  $\text{LastReturnPoint}$  — ближайшая точка возврата,  $\text{BuildVars}$  — список переменных,  $\text{CurThreshold}$  — текущий порог,  $\text{CurEst}$  — текущая оценка. Применение функционального типа позволяет расширить набор встроенных предикатов без изменения основного кода программы.

## 2.3. Модуль анализа базы ограничений

При унификации встроенных предикатов производится перебор возможных значений переменных и проверка накопленных ограничений. Если домен имеет большую размер-

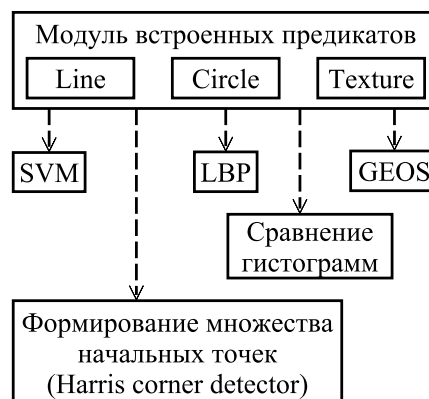
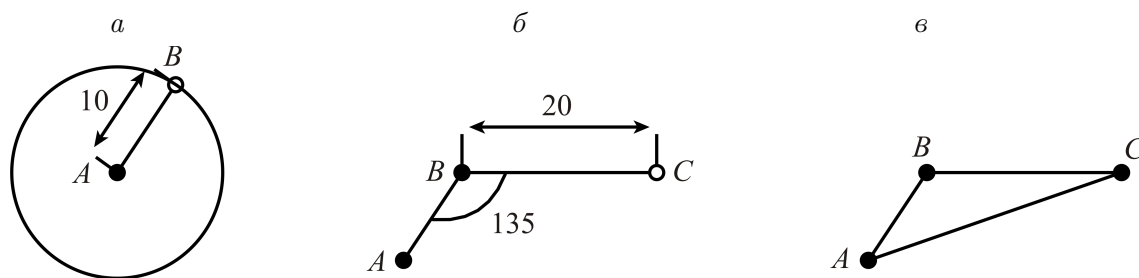


Рис. 3. Схема взаимодействия модулей для унификации встроенных предикатов

Рис. 4. Формирование множества перебора при унификации предиката *line*

ность переменных (все пиксели на изображении), то перебор может занимать значительное время. Уменьшение пространства перебора возможных значений переменных можно производить на основе анализа пространственных ограничений.

Рассмотрим работу модуля на примере правила для нахождения треугольника:

$$\begin{aligned} \text{triangle}(A, B, C) : & \neg \text{line}(A, B), \text{line}(B, C), \text{line}(C, A), \text{angle}(A, B, C) = 135, \\ & \text{dist}(A, B) = 10, \text{dist}(B, C) = 20. \end{aligned}$$

На рис. 4, *a* проиллюстрирована унификация первого в теле правила встроенного предиката *line*. При помощи детектора углов Харриса фиксируется некоторая точка изображения и сопоставляется с одной из переменных предиката (в данном примере с *A*). В результате рассмотрения ограничения  $\text{dist}(A, B) = 10$  перебор значений переменной *B* производится в точках окружности радиуса 10 с центром в точке *A*. На рис. 4, *б* показано, как при унифицированных переменных *A, B*, ограничениях  $\text{dist}(B, C) = 20$  и  $\text{angle}(A, B, C) = 135$  можно вычислить значение переменной *C*. Унификация последнего встроенного предиката приведена на рис. 4, *в*. Поскольку к моменту его унификации все переменные имеют конкретные значения, предикат только вычисляет функцию принадлежности.

## 2.4. Другие модули интерпретатора SOQL

Модуль трансляции текста во внутренний формат переводит текст программы во внутреннее представление. Для ограничений строятся специальные объекты, позволяющие проводить проверку ограничений на множестве найденных переменных. Модуль ввода-вывода производит чтение растровых файлов и запись результатов идентификации в векторные файлы.

## 3. Апробация интерпретатора языка SOQL

Результаты экспериментов приведены на рис. 5–8. В качестве примера приводятся описания формы объектов, размер входного изображения и время работы. На рис. 5 представлен результат идентификации зданий прямоугольной формы по правилу

$$\begin{aligned} \text{building}(A, B, C, D) : & \neg \text{line}(A, B), \text{line}(B, C), \text{line}(C, D), \text{line}(D, A), \text{dist}(A, B) > 28, \\ & \text{dist}(B, C) > 28, \text{dist}(C, D) > 228, \text{dist}(D, A) > 28, \text{dist}(A, B) < 42, \text{dist}(B, C) < 42, \\ & \text{dist}(C, D) < 42, \text{dist}(D, A) < 42, \text{angle}(A, B, C) = 90, \text{angle}(B, C, D) = 90, \\ & \text{angle}(C, D, A) = 90. \\ ? - & \text{building}(A, B, C, D). \end{aligned}$$



Рис. 5. Идентификация зданий (размер изображения  $600 \times 300$  пикселей, время работы 53 с)



Рис. 6. Идентификация здания нестандартной формы (размер изображения  $300 \times 280$  пикселей, время работы 1.05 мин)



Рис. 7. Идентификация автомобильных номеров (размер изображения  $430 \times 280$  пикселей, время работы 27 с)



Рис. 8. Идентификация предметов (размер изображения  $400 \times 370$  пикселей, время работы 1.14 мин)

На рис. 6 приведен результат идентификации здания нестандартной формы по правилу

$$\begin{aligned} & \text{pentagon}(A, B, C, D, E) : -\text{line}(A, B), \text{line}(B, C), \text{line}(C, D), \text{line}(D, E), \text{line}(E, A), \\ & \text{dist}(A, B) = 335, \text{dist}(B, C) = 335, \text{dist}(C, D) = 335, \text{dist}(D, E) = 335, \\ & \text{dist}(E, A) = 335, \text{angle}(A, B, C) = 108, \text{angle}(B, C, D) = 108, \text{angle}(C, D, E) = 108, \\ & \text{angle}(D, E, A) = 108, \text{angle}(E, A, B) = 108. \\ & ? - \text{pentagon}(A, B, C, D, E). \end{aligned}$$

На рис. 7 представлен результат идентификации регистрационного номера транспортного средства по правилу

$$\begin{aligned} & \text{template}(A, B, C, D) : -\text{line}(A, B), \text{line}(B, C), \text{line}(C, D), \text{line}(D, A), \text{dist}(A, B) = 40, \\ & \text{dist}(B, C) = 145, \text{dist}(C, D) = 40, \text{dist}(D, A) = 145, \text{angle}(A, B, C) = 90, \\ & \text{angle}(B, C, D) = 90, \text{angle}(C, D, A) = 90. \\ & ? - \text{template}(A, B, C, D). \end{aligned}$$

На рис. 8 представлен результат идентификации книг по правилу

$$\begin{aligned} & \text{book}(A, B, C, D) : -\text{line}(A, B), \text{line}(B, C), \text{line}(C, D), \text{line}(D, A), \text{dist}(A, B) > 75, \\ & \text{dist}(B, C) > 110, \text{dist}(C, D) > 75, \text{dist}(D, A) > 110, \text{dist}(A, B) < 95, \text{dist}(B, C) < 135, \\ & \text{dist}(C, D) < 95, \text{dist}(D, A) < 135, \text{angle}(A, B, C) > 85, \text{angle}(B, C, D) > 85, \\ & \text{angle}(C, D, A) > 85, \text{angle}(A, B, C) < 95, \text{angle}(B, C, D) < 95, \text{angle}(C, D, A) < 95. \\ & ? - \text{book}(A, B, C, D). \end{aligned}$$

На рис. 8 продемонстрированы достоинства и недостатки предложенного интерпретатора. Так, на рис. 8, *a* показана его работоспособность. Из рис. 8, *б* следует, что интерпретатор может находить частично перекрытые объекты (выделены красным цветом), а из рис. 8, *в* — что он не способен идентифицировать объекты, границы которых выходят за пределы изображения, и объекты, у которых перекрыта большая часть контура.

## Заключение

Серия экспериментов показала работоспособность интерпретатора языка SOQL для обработки растровых изображений. Основным достоинством данного подхода является то, что при помощи языка SOQL можно описать форму, положение, текстурные и спектральные характеристики идентифицируемых объектов без изменения исходного текста программы. Язык SOQL позволяет строить описание в виде иерархий объектов и свойств, наличие дизъюнкций помогает компактно представлять знания.

В существующих логических методах распознавания и идентификации признаки объектов, извлекаемые из изображения, заранее сохраняются в базе фактов. Размер базы фактов влияет на скорость работы и требование к объемам оперативной памяти ЭВМ, поэтому подобные методы хорошо работают на задачах с малой размерностью пространства признаков.

В предложенном методе информация из изображения формируется динамически по требованию в соответствии с базой знаний посредством унификации встроенных предикатов. Данный подход позволяет эффективно обрабатывать большие объемы входных данных без необходимости хранить всю базу фактов в оперативной памяти в явном виде. Дальнейшее развитие данного подхода будет связано с добавлением новых встроенных предикатов, разработкой готовых правил и ограничений и т. д.

**Благодарности.** Работа выполнена при финансовой поддержке РФФИ (грант № 16-07-00554).

## Список литературы / References

- [1] **Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.** Elastically deformable models // Computer Graphics. 1987. Vol. 21, No 4. P. 205–214.
- [2] **Williams, D.J., Shah, M.** A fast algorithm for active contours and curvature estimation // CVGIP: Image Understanding. 1992. Vol. 55, No 1. P. 14–26.
- [3] **Amini, A.A., Weymouth, T.E., Jain, R.C.** Using dynamic programming for solving variational problems in vision // IEEE Trans. Pattern Anal. Machine Intell. 1990. Vol. 12, No 9. P. 855–867.
- [4] **Ganebnykh, S.N., Lange, M.M., Stepanov, D.Yu.** Metric classifier using multilevel network of templates // Pattern Recognition and Image Analysis. 2012. Vol. 22, No 2. P. 265–277.
- [5] **Bell, B., Pau, L.F.** Context knowledge and search in object-oriented Prolog-based image understanding // Pattern Recognition Letters. 1992. Vol. 13. P. 276–290.
- [6] **Schockaert, S., De Cock, M., Kerre, E.E.** Spatial reasoning in a fuzzy region connection calculus // Artificial Intelligence. 2009. Vol. 173, No 2. P. 258–298.
- [7] **Munoz-Velasco, E., Burrieza, A., Ojeda-Aciego, M.** A logic framework for reasoning with movement based on fuzzy qualitative representation // Fuzzy Sets and Systems. 2014. Vol. 242. P. 114–131.
- [8] **Colmerauer, A.** An introduction to Prolog III // Communications of the ACM. 1990. Vol. 33, No 7. P. 69–90.
- [9] **Fedorov, R., Shigarov, A.** Unification of the predicate line in the logic-syntactic recognition of raster images // Pattern Recognition and Image Analysis. 2011. Vol. 21, No 2. P. 242–246.
- [10] **Fedorov, R., Shigarov, A., Ruzhnikov, G.** Algorithm A\* in logical-structure recognition of objects // Pattern Recognition and Image Analysis. 2013. Vol. 23, No 4. P. 435–439.
- [11] **Hogger, G.J.** Introduction to logic programming. San Diego, Ca, USA: Academic Press Professional, 1984. 278 p.
- [12] **Harris, C., Stephens, M.** A combined corner and edge detector // 4th Alvey Vision Conf., University of Manchester, 31st August – 2nd September. United Kingdom: Plessey Research Roke Manor, 1988. P. 147–151.
- [13] **Ojala, T., Pietikinen, M., Harwood, D.** A comparative study of texture measures with classification based on featured distributions // Pattern Recognition. 1996. Vol. 29, No 1. P. 51–59.
- [14] **Joachims, T.** SVMlight support vector machine. Available at: <http://svmlight.joachims.org> (accessed 30.06.2015).
- [15] GEOS geometry engine. Available at: <http://trac.osgeo.org/geos> (accessed 30.06.2015).

*Поступила в редакцию 16 ноября 2015 г.,  
с доработки — 10 декабря 2015 г.*

**Interpreter of Spatial Object Query Language for raster image processing**

BYCHKOV, IGOR V., RUZHNIKOV, GENNADIY M., FEDOROV, ROMAN K.\*,  
AVRAMENKO, YURIY V.

Matrosov Institute for System Dynamics and Control Theory of SB RAS, Irkutsk, 664033,  
Russia

\*Corresponding author: Fedorov, Roman K., e-mail: [fedorov@icc.ru](mailto:fedorov@icc.ru)

**Purpose.** To expand the class of identifiable objects on raster images a priori knowledge is necessary. The required set of knowledge depends on the task. The existing identification methods are characterized by the use of a fixed set of knowledge and that is not enough. So further study in terms of formalization of knowledge and its application in the identification process is required.

**Methodology.** On the basis of the deformable models and Prolog III we propose calculus of spatial objects. The calculus of spatial objects allows a user to express the knowledge and requirements to identifiable objects. The effectiveness of the proposed algorithms for the interpretation of images has been proven. The interpretation is based on the classic Prolog mechanism with backtracking, discarding unpromising branches, checking the spatial limitations and the use of A\* algorithm and the multi start.

**Findings.** The proposed interpreter differs from the existing ones by the feature, which allows the user to turn on and use a variety of knowledge in the recognition process. A series of experiments showed the efficiency of the interpreter spatial calculation objects for raster image processing.

**Originality/value.** The main advantage of this method is the usage of the language, which can describe the shape, position, texture spectral characteristics of identifiable objects without changing the code recognition algorithm. In the logical methods the object features are extracted from images and stored in a knowledge base as the facts. The size of the knowledge base influences on the speed and the requirement for the amount of the computer RAM. Therefore, these methods work well on tasks with low dimensional feature space. In the proposed method, the information of the image is generated dynamically on demand, in accordance with the knowledge base, by the unification of built-in predicates.

*Keywords:* prolog, image identification, image analysis, pattern recognition, deformable templates.

**Acknowledgements.** This research was partly supported by RFBR (grant No. 16-07-00554).

*Received 16 November 2015*

*Received in revised form 10 December 2015*