

Алгоритмы Повышения Эффективности Логического Метода Идентификации Объектов

Программные системы, предназначенные для идентификации объектов на растровых изображениях, ориентированы на решение узкого круга задач. Часто пользователю не хватает функциональных возможностей и знаний, заложенных в программные системы, для идентификации объекта заданной формы и текстуры. Для решения этой задачи необходимо позволить пользователю сформулировать запрос и предоставить механизм идентификации объектов в соответствии с этим запросом. В статье предлагается язык пространственных запросов SOQL, объединяющий два подхода: применение логического вывода и деформируемых моделей, что позволяет использовать выразительность языков первого порядка и эффективно обрабатывать нечеткость признаков. Основная идея алгоритма идентификации заключается в поиске удовлетворяющей комбинации признаков на изображении в виде логического вывода, где каждая найденная комбинация оценивается функцией энергии. Учитывая нечеткость признаков, количество возможных признаков может быть значительным. Формирование признаков до начала логического вывода приведет к чрезмерному разрастанию объема базы фактов. Поэтому извлечение признаков, удовлетворяющих текущим требованиям, производится в процессе логического вывода с помощью унификации встроенных предикатов. Логические методы мало используются для решения задач, связанных с обработкой изображений, т.к. обладают высокой вычислительной сложностью. В данной статье описываются алгоритмы повышения эффективности логического метода идентификации объектов.

Ключевые слова: идентификация объектов, распознавание образов, логические методы, Пролог

Algorithms of increasing efficient of logical method for object identification

There are program systems for object identification on raster images that are focused to solve bounded task range. Sometimes the program system capabilities are not enough to identify object with special contour and texture features desired by users. In order to resolve this task it needs two things. First is a language to allow user to formulate the query and second is an interpreter of the query. In this article the language SOQL for query of spatial object is proposed. It combines two approaches: logic method and deformable models. The logic method allows using expressiveness of the first-order languages. The deformable models allow effectively processing fuzzy features. Main idea of identification algorithm is search of satisfying combination of features on image as inference, where each combination is estimated by energy function. Number of possible features may be very large because they are fuzzy. Extraction of features before starting inference can be the cause of huge fact base. Therefore extraction of features satisfying user's query is in the progress of inference by unification inbuilt predicates. Logical methods have high computations complexity; therefore they rarely apply to processing images. In this article the algorithms of increasing efficient of logical method for object identification are described.

Keywords: object identification, object recognition, logical methods, Prolog

Введение

На текущий момент в области компьютерного зрения активно разрабатываются методы идентификации объектов на растровых изображениях, которые обычно имеют направленность на решение узких задач. Ограниченностю методов вытекает из набора знаний, используемых при идентификации объектов. На разных изображениях большое значение могут получить те или иные свойства объектов или их совокупность. Одной из актуальных задач является идентификация объектов определенной формы с заданными текстурными и спектральными свойствами. Решение этой задачи может быть получено при использовании формализованных знаний об объекте в процедуре идентификации.

Гибкое использование априорной информации при идентификации объектов на изображении позволяет повысить качество работы методов. При

этом большинство методов, где используется априорная информация, склоняются к тому, что необходим некоторый механизм формализации знаний. В качестве одного из таких механизмов можно использовать переборный алгоритм языка Пролог. Рассмотрим некоторые из работ, использующих переборный алгоритм языка Пролог для идентификации объектов на растровых изображениях.

В работе [2] задача идентификации объектов сводится к решению задачи логического вывода формул логики первого порядка. Идентифицируемый объект задается описанием образующих его частей. В статье [4] механизм логического вывода языка Пролог дополнен предикатами для обработки изображений. Авторы предлагают использовать принципы объектно-ориентированного программирования в логическом методе идентификации объектов. Идентификация объектов происходит в три

этапа. На первом этапе происходит обнаружение линий, углов. На втором происходит детектирование контура объекта. На третьем происходит сопоставление полученной комбинации признаков и их взаимного расположения заданному описанию. В [6] предлагается для ускорения процесса логического вывода решать задачу удовлетворения ограничений (УО). В работе доказывается, что время, требуемое для решения задачи УО, окупается за счет существенного уменьшения времени необходимого для логического вывода. Авторы используют эвристики при решении задачи УО.

Основным преимуществом данных подходов является возможность описания пользователем идентифицируемых объектов, их частей, структуры и взаимных отношений с другими объектами.

Логические методы представляют мощные и гибкие средства для решения задачи идентификации объектов на растровых изображениях. В процессе логического вывода происходит поиск возможных решений. Приведем общий принцип рассмотренных подходов получения решений в процессе логического вывода. Идентификация объектов на изображении в логических методах обычно происходит в соответствии с правилом вида: $P : -P_1, \dots, P_n$, где P – голова правила, а P_1, \dots, P_n – тело правила. В теле правила перечислены признаки объекта. Решение считается найденным, если каждый предикат в теле правила принимает значение истина. Принимая во внимание размытость и неоднозначность изображения, шумы возможно будет найдено несколько пересекающихся решений положения одного и того же объекта. В этом случае необходимо предусмотреть способ выбора наиболее подходящих решений из большого числа альтернативных. Производится оценка альтернативных решений различными способами [3, 9, 10, 12].

В целом необходимо отметить, что использование логических методов, позволяющих гибко использовать априорные знания, приводит к увеличению вычислительной сложности процесса идентификации. Поэтому является актуальным повышение эффективности логических методов.

Логический метод идентификации объектов на растровых изображениях

Приведем краткое описание предлагаемого логического метода идентификации объектов на растровых изображениях [1]. В данном методе предлагается идентифицировать объекты по заданному описанию текстурных и спектральных признаков, а также описанию формы контура объекта. Описание объектов задается при помощи разработанного авторами языка пространственных запросов Spatial Object Query Language (SOQL). Язык SOQL базируется на языке Пролог. Язык SOQL до-

полнен специальными встроенными предикатами, термом, функциями и пространственными ограничениями, необходимыми для идентификации объектов на изображениях. Терм $p(x, y)$ – задает положение точки на плоскости изображения. Встроенные предикаты $line(p_1, p_2)$, $circle(p, r)$ – позволяют искать на изображении прямолинейные части контура и окружности соответственно. Предикат $texture([p_1, \dots, p_n], Model)$ – определяет соответствие текстурных и спектральных признаков объекта заданной модели. Реализованы следующие функции: $angle(p_1, p_2, p_3)$ – для определения значения угла, образованного тремя точками; $dist(p_1, p_2)$ – для определения расстояния между двумя точками. Реализована возможность задавать пространственные ограничения на взаимное расположение частей контура в виде $x\theta y$, где x, y – это функции или числовые константы, а $\theta = \{=, <, >, \leq, \geq\}$. Так же реализовано одно из наиболее часто употребляемых ограничений $parallel(p_1, p_2, p_3, p_4)$ – для задания параллельности двух отрезков.

Оценка найденного решения производится на основе функции энергии. Рассмотрим более подробно ее построение. С каждым встроенным предикатом связана функция принадлежности $\mu(s_{i_1}, \dots, s_{i_a}) \in [0; 1]$ – мера соответствия признака изображению, где $s_i \in S$ – переменные, соответствующие положению точек на плоскости изображения; S – множество точек на плоскости. Пусть $InBP$ – 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 предикат в теле правила P , C_j – j пространственное ограничение в теле правила P , $i \in [1; n]$, $j \in [1; m]$. Значение функции энергии вычисляется по формуле:

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

где: n – количество предикатов в теле правила P ; $\mu_i(s_{i_1}, \dots, s_{i_a})$ – значение функции принадлежности встроенного предиката P_i ; L_i – значение функции энергии P_i предиката в теле правила P . Значение функции 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 (набору правил с пространственными ограничениями), и приводящих функцию энергии к локальному минимуму $L_i(s_1, \dots, s_k) \rightarrow \min_{s_i \in S}$. Объекты, для которых значение функции энергии достигает своего минимума в локальной области, образуют множество решений

задачи идентификации объектов. Учитывая наличие множества локальных минимумов, требуется организация полного перебора возможных положений объекта в соответствии с заданной логической моделью.

Приведем пример правила для идентификации объектов прямоугольной формы $\text{rectangle}(A, B, C, D)$

```
:- line(A,B), line(B,C), line(C,D), line(D,A), dist(A,B) > 25, dist(A,B) < 40, dist(B,C) > 25, dist(B,C) < 40, angle(A,B,C) > 85, angle(A,B,C) < 95, parallel(A,B,C,D), parallel(B,C,D,A).
```

При помощи данного правила можно идентифицировать объекты, имеющие прямоугольный контур, длина ребра которых составляет от 25 до 40 пикселей. Обратите внимание на то, что длина ребра и величина углов между ними задана с допуском, это сделано для того чтобы учесть различные варианты положения камеры, из-за которых контур объекта может отличаться от требуемого.

Для идентификации объекта необходимо произвести полный перебор, чтобы найти комбинацию возможных положений графических примитивов, которые бы удовлетворяли заданному описанию на языке SOQL. Количество графических примитивов для одного изображения может измеряться тысячами, а всевозможные их комбинации миллионами. Поэтому задача идентификации является сложной комбинаторной задачей, требующей применения методов уменьшения вычислительной сложности.

Работа интерпретатора языка SOQL основывается на классическом переборе с возвратами языка Пролог, дополненного механизмом отсечения неперспективных ветвей логического вывода путем проверки пространственных ограничений, отсечения по порогу. В процессе поиска в глубину производится накопление ограничений и связывание переменных ограничений со слотами, выделенных для хранения значений переменных правил. Для каждого найденного решения (ветки дерева перебора в глубину) вычисляется функция энергии. Оставляются решения, которые имеют значения функции энергии меньше заданного порога. Идентификация объектов ведется в предположении, что границы идентифицируемых объектов не пересекаются. При этом часто существуют решения со значением функции энергии меньше порога, которым соответствуют пересекающие альтернативы положения объекта. Среди всех альтернативных решений выбирается вариант с наименьшим значением функции энергии.

Алгоритмы повышения эффективности логического метода

Детектор углов Харриса

Последовательно перебирать всевозможные значения точек весьма дорого в вычислительном

плане. Поэтому более эффективно будет выделить стартовые точки, с которых необходимо начинать перебор. Предполагается, что к такими относятся угловые точки, т.е. которые являются вершиной угла, образованного двумя сегментами контура. Для обнаружения таких точек используется детектор углов Харриса [5].

Результат работы детектора Харриса – множество особенных точек, на изображении. Каждая особенная точка характеризуется значением функции отклика. Полученное множество сортируется по убыванию значения функции отклика. Относительно каждой точки данного множества запускается процедура идентификации. Сортировка, обеспечивает идентификацию объектов с тех положений, где вероятность обнаружить объект выше.

Классификация изображения методом опорных векторов

Классификатор на основе метода опорных векторов [11] используется в роли фильтра, применяемого к изображению. После обработки изображения фильтром происходит разделение изображения на участки, в которых могут и не могут находиться идентифицируемые объекты, что приводит к уменьшению пространства перебора. Для работы классификатора множество допустимых значений (точек изображения) представлено в виде вектора признаков. Вектор признаков в нашем случае это характеристика частоты вхождения в ячейку изображения размера $n \times n$ различных элементов. Элементами выступают текстурные и спектральные признаки.

Текстурные признаки вычисляются методом LBP [8]. Обычно в подходах, где применяется метод LBP, разработчики используют не весь набор возможных значений шаблонов, а только часть информативных шаблонов, подходящих для конкретной задачи идентификации объектов. В зависимости от признаков идентифицируемого объекта в качестве информативных могут использоваться различные шаблоны. Каждой точке изображения присваивается один из 256 бинарных шаблонов, характеризующих ее. Спектральные признаки соответствуют значениям яркости по каждому из каналов цветовой модели RGB, которые так же принимают одно из 256 возможных значений. Компонентами вектора признаков будут значения частоты встречаемости текстурных или спектральных признаков в ячейке.

Отсечение неперспективных ветвей на основе алгоритма A*

Переборный механизм модифицирован с целью отсечения альтернативных решений с заведомо худшим решением на основе алгоритма A*. Смысл алгоритма заключается в том, что в процессе поиска в глубину, если для текущей ветки

лучшее возможное решение хуже альтернативного, то продолжать эту ветку нет смысла. Допустим, что в процессе логического вывода в теле правила имеем последовательность $\{P_1, \dots, P_{j-1}\}$ — унифицированных и $\{P_j, \dots, P_n\}$ — не унифицированных 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: $h_{L_i}(s_{i_1}, \dots, s_{i_b}) = -1$. Т.е. 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 следующее 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.

Анализ пространственных ограничений

Montanari [7] предложил использовать модели удовлетворения ограничений (УО) для описания ряда комбинаторных задач, возникающих при компьютерной обработке изображений, и назвал эти задачи УО «сетями ограничений». Это связано с тем, что систему ограничений можно представить в виде неориентированного графа с вершинами — переменными и ребрами, соответствующими ограничениям между двумя переменными. Сети ограничений являются графовым представлением, используемым для поиска стратегий решения задач УО.

Теория УО предлагает удобный аппарат и простую формальную схему для представления и решения комбинаторных задач. Целью решения задачи УО является нахождение значений переменных, удовлетворяющих определенным ограничениям. Парадигма УО является обобщением пропозициональной логики. УО предлагает общность, гибкость, среду моделирования высокого уровня, управление поиском, компактность представления задачи, распространение ограничений, быстрые методы нахождения решения. На основе аппарата теории УО в логическом методе реализован механизм анализа пространственных ограничений для

управления поиском решений задачи идентификации объектов.

Рассмотрим реализованный механизм анализа пространственных ограничений подробнее. При унификации встроенных предикатов производится перебор возможных значений переменных и проверка накопленных ограничений. Учитывая размер домена переменных (все пиксели на изображении), 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: $triangle(A, B, C) :- line(A, B), line(B, C), line(C, A), angle(A, B, C) = 135, dist(A, B) = 10, dist(B, C) = 20$. На рис. 1 показан процесс анализа пространственных ограничений для встроенных предикатов. Фиксируется некоторая точка изображения и сопоставляется с одной из переменных предиката (в данном примере с A). В результате рассмотрения ограничения $dist(A, B) = 10$ перебор значений переменной B производится в точках окружности радиуса 10 с центром в точке A (рис. 1.а). При унифицированных переменных A, B и ограничениях $dist(B, C) = 20, angle(A, B, C) = 135$ значение переменной C можно вычислить (рис. 1.б). Так как к моменту унификации последнего предиката в теле правила, все переменные имеют конкретные значения, то его унификация определяется однозначно (рис. 1.в).

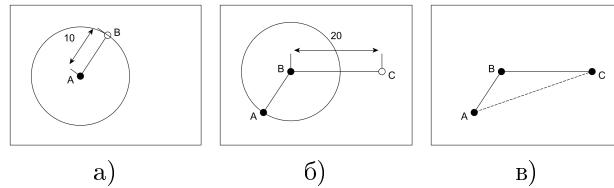


Рис. 1. Анализ пространственных ограничений.

В таблице 1 приводится общее время работы алгоритма поиска решений до и после применения методов уменьшения вычислительной сложности на изображениях различного размера.

Результаты идентификации объектов

В [6] по описанию объекта авторы производят поиск возможных решений (2100 вариантов возможных решений для одного объекта) задачи идентификации объектов. Большое количество решений объясняется тем, что под описание объекта попадает множество альтернатив. Далее приводятся результаты идентификации объектов различной формы. На рис. 2.а показан результат идентификации объекта на изображении из работы [6]. На рис. 2.б показан результат идентификации объектов с круглым контуром, рис. 2.в — прямоугольным

Таблица 1. Сравнение времени работы алгоритма идентификации до и после применения методов уменьшения вычислительной сложности.

Размер изображения (пикс.)	Время работы (мин)	
	После	До
150 × 200	00 : 26	01 : 10
600 × 300	00 : 28	05 : 03
500 × 400	00 : 27	09 : 10
1200 × 750	00 : 40	18 : 54
1500 × 1500	01 : 15	—
3000 × 3000	02 : 45	—

контуром, рис. 2.г – контуром в форме равностороннего пятиугольника.

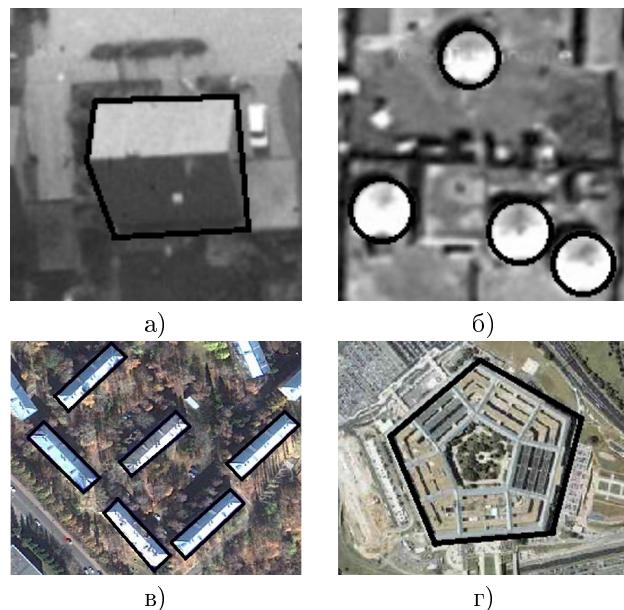


Рис. 2. Результаты идентификации объектов с различной формой контура.

Выводы

В данной работе представлен логический метод идентификации объектов на растровых изображениях, в котором удалось добиться приемлемого времени работы за счет повышения эффективности процедуры логического вывода различными алгоритмами, применения эвристик и методов предварительной обработки изображения. Основным достоинством данного подхода является то, что при помощи языка SOQL можно формализовать такие знания об объекте как контур, положение, текстурные и спектральные характеристики без изменения исходного кода программы. Язык SOQL позволяет строить описание в виде иерархий объектов и свойств, наличие дизъюнкций помогают компактно представлять знания. В предложенном методе информация из изображения формируется динамиче-

ски 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. Предполагается дальнейшее развитие данного подхода путем добавления новых встроенных предикатов, разработки готовых правил, ограничений и т.д.

Литература

- [1]
- [2] Косовская Т.М., Власова М.А. Использование языков семейства Prolog для распознавания изображений // Тр. СПИИРАН. – 2013. – Т.25, №1. – С.277-293.
- [3] Amini AA., Weymouth TE., Jain RC. Using dynamic programming for solving variational problems in vision // IEEE Trans Pattern Anal Machine Intell, 1990. – Vol.12, No.9. – pp.855-867.
- [4] Bell B., Pau L.F. Context knowledge and search in object-oriented Prolog-based image understanding // Pattern Recognition Letters, 1992. – Vol.13, – pp.276-290.
- [5] Harris C., Stephens M. A combined corner and edge detector // Plessey Research Roke Manor, 1988. – pp.147-151.
- [6] Kolbe T.H., Plumer L., Cremers A.B. Using Constraints for the Identification of Buildings in Aerial Images // Proceedings of the 2nd International Conference on the Practical Application of Constraint Technology, 1996.
- [7] Montanari U. Networks of constraints: Fundamental properties and applications to picture processing // Information Sciences, 1974. Vol.7, No.2. – pp.95-132.
- [8] 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. pp.51-59.
- [9] Terzopoulos D., Platt J., Barr A., Fleischer K. Elastically Deformable Models // Computer Graphics, 1987. Vol.21, No.4. – pp.205-214.
- [10] Shet V., Singh M., Bahlmann C., Ramesh V., Neumann J., Davis L. Predicate Logic Based Image Grammars for Complex Pattern Recognition // International Journal of Computer Vision, 2011. – Vol.93. pp.141-161.
- [11] <http://svmlight.joachims.org> – SVMlight Support Vector Machine.
- [12] Williams DJ., Shah M. A fast algorithm for active contours and curvature estimation // CVGIP: Image Understanding, 1992. – Vol.55, No.1. pp.14-26.