

УДК 004.93

© М. В. Кучуганов, А. В. Кучуганов

## ДЕСКРИПЦИОННАЯ ЛОГИКА НА ГРАФАХ ИЗОБРАЖЕНИЙ

В работе предлагается для формального описания и структурного анализа изображений использовать расширение  $ALC(GI)$  дескрипционной логики  $ALC$ . Концепты и роли логики  $ALC(GI)$  интерпретируются на графе изображения и его подграфах. Описана модель изображения в виде многослойного атрибутивного графа. Граф изображения содержит слой цветowych сегментов, слой границ, слой скелетонов. Каждый слой представляет собой планарный граф, слои связаны между собой отношениями «предок–потомок». Переход от пиксельного представления изображения к графовому позволяет существенно увеличить эффективность его анализа. Приведены примеры предметных терминологических аксиом, определяющих структурные элементы изображения и составленные из них буквы, а также результаты эксперимента, проведенного на задаче распознавания букв в слитном рукописном тексте.

*Ключевые слова:* дескрипционная логика, граф, распознавание изображений, рукописный текст.

DOI: [10.20537/vm180410](https://doi.org/10.20537/vm180410)

### Введение

Традиционно задачи анализа изображений решаются путем сравнения и распознавания конструкций, составленных из морфологических элементов — особых точек и отрезков, т. е. путем поиска таких сочетаний элементов, которые удовлетворяют заданным правилам. Однако чем сложнее синтаксис искомых конструкций, тем быстрее возрастает сложность [1] алгоритмов. Дальнейшее повышение эффективности систем анализа графической информации, на наш взгляд, требует взаимодействия методов распознавания образов и методов моделирования рассуждений.

Очевидно, для задач извлечения знаний из изображений наиболее адекватным является аппарат теории графов.

Задача анализа и сравнения (сопоставления) информации, представленной в виде графов, достаточно хорошо проработана в проблеме понимания и оценки ситуаций. В работе [2] ситуация представляется в виде ориентированного атрибутивного графа отношений (DARG — Directed Attributed Relational Graph), который представляет собой граф, содержащий множества атрибутов вершин и ребер. В качестве описания изображений в задачах поиска в последнее время стали применяться атрибутивные графы отношений (ARG), нечеткие атрибутивные графы (FARG). Однако задачи структурного анализа изображений объектов, выявления сходства и отличий при этом не ставились. Соответственно, описания менее структурированы и содержат ограниченный набор атрибутов.

Типичный подход к представлению изображений в виде атрибутивных графов заключается в следующем. Выполняется цветовая сегментация изображения. Каждый цветовой сегмент (регион) — это узел графа. Узел помечается цветом, плотностью и атрибутами формы. Ребра графа — отношения между соседними регионами. При этом форма региона, представленного узлом графа, является важнейшей характеристикой. В работе [3] для этого использованы алгебраические инварианты М. Ху [4] — центральные моменты и моменты, независимые от преобразований масштаба, переноса, поворота, зеркального отображения на плоскости. Таким образом, каждый узел графа представляет собой вектор в пространстве 11 атрибутов. В работе [5] в качестве атрибутов формы используют дескрипторы НОГ (Histogram of Oriented Gradients) — гистограммы ориентации (градиентов направления) отрезков, углы между скелетными линиями, длина региона. Здесь необходимо отметить, что показатели, основанные

на статистических вычислениях, как правило, недостаточны для задач анализа и выявления отличий в форме объектов.

В работах [6, 7] рассматривается применение элементов теории графов в задаче распознавания рукописных математических формул. Вводятся двухуровневый орграф и правила отображения на него основных классов математических формул. Описывается метод решения задачи распознавания путем построения семантического дерева формулы и решения обратной задачи — генерации графа изображения формулы. Для предупреждения и исправления ошибок распознавания вводится метод, основанный на ручном управлении процессом. Сложность в данном подходе, на наш взгляд, представляет формальное описание в терминах теории графов множества конструкций рукописного алфавита — терминальных символов, применяемых в математических формулах, контроль корректности описаний, заданных экспертом, и, соответственно, трудоемкость исправления ошибок в программе.

Дескрипционные логики [8–10] нашли широкое применение в различных аналитических и поисковых системах. Достоинство дескрипционных логик заключается в том, что с их помощью эксперт описывает термины и определения предметной области, а машина вывода (reasoner) системы осуществляет контроль корректности определений и поиск соответствующих экземпляров в разделе данных.

В работах [11, 12] представлены нечеткие дескрипционные логики, с помощью которых авторы описывают такие объекты на изображениях, как море, волны, песок, пляж, строения, город, ландшафт, лес, дорога и т. п., с целью индексации и поиска картинок в хранилищах данных. Рассматриваются пространственные отношения: слева, справа, выше, ниже. Принадлежность областей изображения тому или иному классу вычисляется на основе цветовых характеристик пикселей. Задача поиска объектов по более сложным отношениям или по форме при этом не ставилась.

Испанские ученые [13] выделяют на изображении цветовые области, формируют их описание (форма, цвет) и пространственные характеристики (ориентация, топология/положение), а потом это описание переводят в дескрипционные логики с целью получения формального и явного смысла и классификации с их помощью. Для описания концептов в дескрипционной логике в качестве атомарных они берут эти характеристики областей изображения, т. е. алгоритмы вычисления характеристик областей определяют интерпретацию соответствующих атомарных концептов и ролей. В качестве примера анализируют ситуации, возникающие перед роботом, ориентирующимся в пространстве коридоров и комнат. Недостатком работы следует считать ограниченный набор характеристик изображения.

Французские ученые [14] для описания структуры объектов на изображении используют расширение дескрипционной логики  $\mathcal{ALC}$  (Attributive Concept Language with Complements) с конкретной областью  $F$ , представляющей результаты вычисления специальных функций, анализирующих пространственные отношения между элементами изображения. Авторы основываются на своей работе [15], где использовали математическую морфологию для описания пространственных отношений.

Общий принцип применения дескрипционных логик для описания изображений заключается в создании специализированных наборов атомарных концептов и ролей.

Мы предлагаем для формального описания и структурного анализа изображений использовать дескрипционную логику  $\mathcal{ALC}(GI)$  с атомарными концептами и ролями, областью интерпретации которых является граф изображения и его подграфы.

В § 1 описывается дескрипционная логика  $\mathcal{ALC}(G)$  — расширение  $\mathcal{ALCOIQf}[\sqcup, \sqcap, \circ, *, id]$  дескрипционной логики  $\mathcal{ALC}$  с атомарными концептами и ролями для описания графов. Определяется терминология графов  $\mathcal{T}_G$  (теория графов в  $\mathcal{ALC}(G)$ ) и рассматриваются ее основные свойства. Приводятся примеры определений некоторых графовых понятий (связный подграф, цепочка, цикл) в логике  $\mathcal{ALC}(G)$ .

В § 2 описываются графовая модель изображения в виде многослойного атрибутивного графа и дескрипционная логика  $\mathcal{ALC}(GI)$  — расширение  $\mathcal{ALCOIQf}[\sqcup, \sqcap, \circ, *, id]$  дескрипционной логики  $\mathcal{ALC}$  с дополнительными атомарными концептами и ролями, представляющими гео-

метрические и прочие нужные атрибуты элементов изображения.

Граф изображения содержит слой цветowych сегментов, слой границ, слой скелетов. Каждый слой представляет собой планарный граф, слои связаны между собой отношениями «предок–потомок». Переход от пиксельного представления изображения к графовому позволяет существенно увеличить эффективность его анализа.

В § 3 приведены примеры предметных терминологических аксиом, определяющих структурные элементы изображения и составленные из них буквы, а также результаты эксперимента, проведенного на задаче распознавания букв в слитном рукописном тексте.

## § 1. Дескрипционная логика на графах

Дескрипционные логики (ДЛ) используются для описания и анализа понятий (концептов) и отношений между ними (ролей), но в языке ДЛ, в отличие от языка логики первого порядка, не используются переменные. Это сближает их с пропозициональной логикой и существенно упрощает анализ формул.

Будем рассматривать графы общего вида — ориентированные, с изолированными вершинами, кратными дугами и петлями.

Граф  $G$  — это кортеж  $\langle V, E, src, trg \rangle$ , где  $V$  — непустое множество вершин графа,  $E$  — множество дуг графа такое, что  $V \cap E = \emptyset$ ,  $src, trg$  — функции из  $E$  в  $V$ , сопоставляющие каждой дуге  $e \in E$  ее начало  $src(e)$  и конец  $trg(e)$ .

Множество разбиений множества  $E$  дуг графа  $G$  будем обозначать через  $Pg(G)$ .

Для описания графов будем использовать дескрипционную логику  $\mathcal{ALC}(G)$  — расширение  $\mathcal{ALCQI}Qf[\sqcup, \sqcap, \circ, *, id]$  дескрипционной логики  $\mathcal{ALC}$ , использующее, кроме прочих, следующие атомарные одноместные (концепты) и двуместные (роли) отношения:

CA1)  $Node(x)$  —  $x$  является вершиной графа;

CA2)  $Edge(x)$  —  $x$  является дугой графа;

CA3)  $Graph(x)$  —  $x$  является подграфом графа;

RA1)  $hasSource(x : Edge, y : Node)$  — вершина  $y$  является началом дуги  $x$ , функциональное отношение (атрибут дуги);

RA2)  $hasTarget(x : Edge, y : Node)$  — вершина  $y$  является концом дуги  $x$ , функциональное отношение (атрибут дуги);

RA3)  $hasGraph(x : Edge, y : Graph)$  — подграф  $y$  содержит дугу  $x$ , функциональное отношение (атрибут дуги).

Концепты, роли и терминологии, использующие только эти понятия, будем называть *графовыми*.

Терминология (теория) графов  $\mathcal{T}_G$  имеет следующие аксиомы:

A1)  $Node \sqcap Edge \sqsubseteq \perp$ ;

A2)  $\exists hasSource \equiv Edge, \exists hasSource^- \sqsubseteq Node, \exists hasSource \sqsubseteq = 1 hasSource$ ;

A3)  $\exists hasTarget \equiv Edge, \exists hasTarget^- \sqsubseteq Node, \exists hasTarget \sqsubseteq = 1 hasTarget$ ;

A4)  $Graph \sqcap (Node \sqcup Edge) \sqsubseteq \perp$ ;

A5)  $\exists hasGraph \equiv Edge, \exists hasGraph^- \equiv Graph, \exists hasGraph \sqsubseteq = 1 hasGraph$ .

Поясним взаимосвязи между моделями терминологии графов  $\mathcal{T}_G$  и графами.

Пусть  $CN, RN$  — конечные непустые множества атомарных концептов и атомарных ролей,  $IN$  — конечное множество индивидов.

Сигнатура  $\Sigma$  — это кортеж  $\langle CN, RN, IN \rangle$ .

Интерпретация сигнатуры  $\Sigma$  — это кортеж  $\langle \Delta, \cdot^{\mathcal{I}} \rangle$ , где  $\Delta$  — непустое множество,  $\cdot^{\mathcal{I}}$  — интерпретирующая функция, которая сопоставляет, соответственно, концептам  $C \in CN$  — множества  $C^{\mathcal{I}} \subseteq \Delta$ , ролям  $R \in RN$  — множества  $R^{\mathcal{I}} \subseteq \Delta \times \Delta$ , индивидам  $a \in IN$  — элементы  $a^{\mathcal{I}} \in \Delta$ .

Любая модель  $\mathcal{I}$  теории  $\mathcal{T}_G$  однозначно определяет граф

$$G^{\mathcal{I}} = \langle Node^{\mathcal{I}}, Edge^{\mathcal{I}}, hasSource^{\mathcal{I}}, hasTarget^{\mathcal{I}} \rangle$$

и разбиение множества его дуг  $P^{\mathcal{I}} \in Pg(G^{\mathcal{I}})$ :

$$P^{\mathcal{I}} = \{p(y) \mid p(y) = \{x \mid \langle x, y \rangle \in hasGraph^{\mathcal{I}}\}, y \in Graph^{\mathcal{I}}\}.$$

Любая пара  $\langle G, P \rangle$ , где  $G = \langle V, E, src, trg \rangle$  — граф,  $P \in Pg(G)$ , однозначно определяет модель  $I(G, P) = \langle \Delta, \cdot^{\mathcal{I}} \rangle$  теории  $\mathcal{T}_G$ , в которой  $\Delta = V \cup E \cup P$ ,  $Node^{\mathcal{I}} = V$ ,  $Edge^{\mathcal{I}} = E$ ,  $Graph^{\mathcal{I}} = P$ ,  $hasSource^{\mathcal{I}} = src$ ,  $hasTarget^{\mathcal{I}} = trg$ ,  $hasGraph^{\mathcal{I}} = \{\langle x, y \rangle \mid x \in y, y \in P\}$ .

Очевидно, что  $\mathcal{I} = I(G^{\mathcal{I}}, P^{\mathcal{I}})$ ,  $G = G^{I(G, P)}$ ,  $P = P^{I(G, P)}$ .

Терминология  $\mathcal{T}_{G_0} \subseteq \mathcal{T}_G$  с аксиомами A1, A2, A3 является *универсальной* в следующем смысле.

Пусть сигнатура  $\Sigma = \langle CN, RN, \emptyset \rangle$ ,  $\nu$  — нумерация множества  $2^{CN} \cup 2^{RN}$  (произвольная инъекция из  $2^{CN} \cup 2^{RN}$  в  $N$ ).

Функция перевода  $TL_{\Sigma}^{\nu}$  сопоставляет концептам  $C \in CN$  и ролям  $R \in RN$  графовые концепты и роли:

$$TL_{\Sigma}^{\nu}(C) = \bigsqcup_{\sigma \in 2^{CN}, C \in \sigma} LNode(\nu(\sigma)),$$

$$TL_{\Sigma}^{\nu}(R) = \bigsqcup_{\sigma \in 2^{RN}, R \in \sigma} LEdge(\nu(\sigma)),$$

где

$$LNode(n) = (=n hasSource^{-}.(hasSource = hasTarget)),$$

$$LEdge(n) = (hasSource^{-} \circ hasTarget \circ id(LNode(n)) \circ hasSource^{-} \circ hasTarget), \quad n \in N.$$

Концепты  $LNode(n)$  представляют вершины графа с  $n$  петлями, а роли  $LEdge(n)$  — пары вершин графа, связанных отношениями «существует вершина с  $n$  петлями, в которой заканчивается дуга из первой вершины пары и начинается дуга во вторую».

Функция перевода позволяет задавать интерпретации сигнатуры  $\Sigma$  с помощью графов. При этом количество петель у вершины графа однозначно определяет, в соответствии с нумерацией  $\nu$ , каким концептам (либо ролям) принадлежит она либо соответствующие ей связи.

Переводы концептов, ролей и терминологий сигнатуры  $\Sigma = \langle CN, RN, \emptyset \rangle$  получаются из исходных путем замены всех вхождений атомарных концептов  $C \in CN$  и ролей  $R \in RN$  на переводы  $TL_{\Sigma}^{\nu}(C)$  и  $TL_{\Sigma}^{\nu}(R)$  соответственно.

**Теорема 1** (графовое представление терминологий). Пусть  $\mathcal{T}$  — терминология,  $C$  — концепт сигнатуры  $\Sigma = \langle CN, RN, \emptyset \rangle$ ,  $\nu$  — нумерация множества  $2^{CN} \cup 2^{RN}$ . Концепт  $C$  выполним в терминологии  $\mathcal{T}$  тогда и только тогда, когда перевод концепта  $TL_{\Sigma}^{\nu}(C)$  выполним в терминологии  $\mathcal{T}_{G_0} \cup TL_{\Sigma}^{\nu}(\mathcal{T})$ , где  $TL_{\Sigma}^{\nu}(\mathcal{T})$  — перевод  $\mathcal{T}$ .

**Доказательство.** Докажем, что если концепт  $C$  выполним в терминологии  $\mathcal{T}$ , то концепт  $TL_{\Sigma}^{\nu}(C)$  выполним в терминологии  $TL_{\Sigma}^{\nu}(\mathcal{T})$ , обратное очевидно.

Любая интерпретация  $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$  концептов и ролей определяет, с точностью до изоморфизма, граф интерпретации  $G(\mathcal{I})$  следующим образом.

Пусть для любого непустого  $\sigma \in 2^{CN}$

$$C_{\sigma}^{\mathcal{I}} = \bigcup \{C^{\mathcal{I}} \mid C \in \sigma\} \setminus \bigcup \{C^{\mathcal{I}} \mid C \notin \sigma\}$$

и

$$C_{\emptyset}^{\mathcal{I}} = \Delta \setminus \bigcup CN.$$

Пусть для любого  $\sigma \in 2^{RN}$

$$R_{\sigma}^{\mathcal{I}} = \bigcup \{R^{\mathcal{I}} \mid R \in \sigma\} \setminus \bigcup \{R^{\mathcal{I}} \mid R \notin \sigma\}.$$

Множество вершин  $V_{G(\mathcal{I})}$  графа  $G(\mathcal{I})$  — это множество  $\Delta \cup \bigcup RN$ . Множество дуг  $E_{G(\mathcal{I})}$  графа  $G(\mathcal{I})$  — это минимальное подмножество множества  $V_{G(\mathcal{I})} \times V_{G(\mathcal{I})} \times N$ , которое содержит:

- 1) для каждого  $\sigma \in 2^{\text{CN}}$ , каждой вершины  $a \in C_\sigma^{\mathcal{I}} - \nu(\sigma)$  петель с началом и концом в вершине  $a$ ;
- 2) для каждого  $\sigma \in 2^{\text{RN}}$ , каждой вершины  $\langle a, b \rangle \in R_\sigma^{\mathcal{I}} - \nu(\sigma)$  петель с началом и концом в вершине  $\langle a, b \rangle$ ;
- 3) для каждой вершины  $\langle a, b \rangle \in \bigcup \text{RN}$  — дугу с началом в вершине  $a$  и концом в вершине  $\langle a, b \rangle$  и дугу с началом в вершине  $\langle a, b \rangle$  и концом в вершине  $b$ .

Очевидно, что если  $\mathcal{I}$  — модель терминологии  $\mathcal{T}$ , в которой выполняется концепт  $C$ , то для произвольного  $P \in \text{Pg}(G(\mathcal{I}))$  интерпретация  $I(G(\mathcal{I}), P)$  является моделью терминологии  $\mathcal{T}_{G0} \cup \text{TL}_{\Sigma}^{\nu}(\mathcal{T})$ , в которой выполняется концепт  $\text{TL}_{\Sigma}^{\nu}(C)$ .  $\square$

**Следствие 1.** Проблема выполнимости концептов логики  $\text{ALC}(G)$  в терминологии  $\mathcal{T}_{G0}$  неразрешима.

Система определений на графах — это ацикличная система определений с графовыми базовыми концептами и ролями.

Следующий пример системы определений на графах содержит определения концептов и ролей, которые будут использоваться в дальнейшем.

**Пример 1.** Некоторые производные роли и концепты на графах.

RB1)  $\text{hasIncident}(x : \text{Edge}, y : \text{Node}) \equiv \text{hasSource} \sqcup \text{hasTarget}$  — вершина  $y$  инцидентна дуге  $x$ ;

RB2)  $\text{hasEdge}(x : \text{Graph}, y : \text{Edge}) \equiv \text{hasGraph}^-$  — дуга  $y$  является дугой подграфа  $x$ ;

RB3)  $\text{hasNode}(x : \text{Graph}, y : \text{Node}) \equiv \text{hasEdge} \circ \text{hasIncident}$  — вершина  $y$  является вершиной подграфа  $x$ ;

RB4)  $\text{hasNear}(x : \text{Edge}, y : \text{Edge}) \equiv \text{hasGraph} \circ \text{hasGraph}^-$  — дуги  $x$  и  $y$  принадлежат одному подграфу;

RB5)  $\text{hasNearSource}(x : \text{Edge}, y : \text{Edge}) \equiv (\text{hasSource} \circ \text{hasIncident}^-) \sqcap \text{hasNear}$  — дуга  $y$  инцидентна началу дуги  $x$  и находится в том же подграфе;

RB6)  $\text{hasNearTarget}(x : \text{Edge}, y : \text{Edge}) \equiv (\text{hasTarget} \circ \text{hasIncident}^-) \sqcap \text{hasNear}$  — дуга  $y$  инцидентна концу дуги  $x$  и находится в том же подграфе;

RB7)  $\text{hasNearAdjacent}(x : \text{Edge}, y : \text{Edge}) \equiv \text{hasNearSource} \sqcup \text{hasNearTarget}$  — дуги  $x$  и  $y$  смежны и принадлежат одному подграфу;

CB1)  $\text{Connected}(x : \text{Graph}) \equiv \neg \exists \text{hasEdge} \circ (\text{hasNear} \sqcap \neg(\text{hasNearAdjacent}^*))$  — подграф  $x$  является связным;

CB2)  $\text{NearSource}_n(x : \text{Edge}) \equiv \leq n \text{hasNearSource}$  — началу дуги  $x$  инцидентны не более  $n$  дуг ее подграфа,  $n \in \mathbb{N}$ ;

CB3)  $\text{NearTarget}_n(x : \text{Edge}) \equiv \leq n \text{hasNearTarget}$  — концу дуги  $x$  инцидентны не более  $n$  дуг ее подграфа,  $n \in \mathbb{N}$ ;

CB4)  $\text{Chain}(x : \text{Graph}) \equiv \text{Connected} \sqcap \forall \text{hasEdge}.(\text{NearSource}_2 \sqcap \text{NearTarget}_2) \sqcap \exists \text{hasEdge}.(\text{NearSource}_1 \sqcup \text{NearTarget}_1)$  — подграф  $x$  является (простой) незамкнутой цепочкой;

CB5)  $\text{Cycle}(x : \text{Graph}) \equiv \text{Connected} \sqcap \forall \text{hasEdge}.(\text{NearSource}_2 \sqcap \text{NearTarget}_2) \sqcap \neg \exists \text{hasEdge}.(\text{NearSource}_1 \sqcup \text{NearTarget}_1)$  — подграф  $x$  является (простой) замкнутой цепочкой, циклом.

Терминологии  $T_G(\mathcal{T}) = \mathcal{T}_G \cup \mathcal{T}$ , где  $\mathcal{T}$  — система определений на графах, будем называть терминологиями на графах (с системой определений  $\mathcal{T}$ ).

Пусть  $C$  — концепт терминологии  $T_G(\mathcal{T})$ ,  $\mathcal{I}$  — модель теории  $\mathcal{T}_G$ .

Каждому элементу  $e \in C^{\mathcal{I}} \subseteq \text{Node}^{\mathcal{I}} \cup \text{Edge}^{\mathcal{I}} \cup \text{Graph}^{\mathcal{I}}$  соответствует подграф  $gr(e)$  графа  $G^{\mathcal{I}}$ :

$gr(e) = \langle \{e\}, \emptyset, \emptyset, \emptyset \rangle$ , если  $e \in \text{Node}^{\mathcal{I}}$ ;

$gr(e) = \langle \text{src}(E') \cup \text{trg}(E'), E', \text{src}|E', \text{trg}|E' \rangle$ , где  $E' = \{e\}$ , если  $e \in \text{Edge}^{\mathcal{I}}$ , либо  $E' = \{x \mid \langle x, e \rangle \in \text{hasGraph}^{\mathcal{I}}\}$ , если  $e \in \text{Graph}^{\mathcal{I}}$ .

**Утверждение 1** (выполнимость концептов в  $T_G(\mathcal{T})$  и существование графов). *Концепт  $C$  терминологии на графах  $T_G(\mathcal{T})$  выполним тогда и только тогда, когда существует граф  $G$ , в котором имеются подграфы, соответствующие концепту  $C$ .*

Часто направление дуг графа не имеет значения. В этом случае получаем *неориентированные* графы. Не имеет смысла вводить для таких случаев новые концепты и роли.

Каждое определяемое понятие (концепт или роль) имеет свое описание — формулу, построенную из атомарных понятий и операций над ними.

Два понятия будем называть *двойственными*, если определение одного получается из определения другого заменой в определении роли *hasSource* на роль *hasTarget* и наоборот.

Иногда оказывается, что двойственные понятия эквивалентны. Их интерпретация не меняется при изменении направления дуг. Такие самодвойственные понятия будем называть *инвариантными* (относительно изменения направления дуг). Они позволяют работать с орграфом как с неориентированным.

В примере 1 роли RB5, RB6 и концепты CB2, CB3 двойственные, остальные понятия инвариантные.

**Утверждение 2** (достаточное условие инвариантности понятий).

- (1) Если  $A$  и  $B$  — двойственные понятия, то  $A \sqcap B$  и  $A \sqcup B$  — инвариантные понятия.
- (2) Если понятие построено из инвариантных, то оно инвариантное.

Пусть  $C$  — концепт терминологии  $T_G(\mathcal{T})$ ,  $G$  — граф.

Графу  $G$  соответствует множество  $\mathcal{M}(G) = \{I(G, P) \mid P \in Pg(G)\}$  моделей терминологии графов  $\mathcal{T}_G$ . В каждой модели  $\mathcal{I} \in \mathcal{M}(G)$  концепту  $C$  соответствует множество  $gr(C^{\mathcal{I}}) = \{gr(e) \mid e \in C^{\mathcal{I}}\}$  подграфов графа  $G$ .

Множество подграфов графа  $G$ , соответствующих концепту  $C$  терминологии  $T_G(\mathcal{T})$ , — это множество

$$Cgr(G, \mathcal{T}, C) = \bigcup_{\mathcal{I} \in \mathcal{M}(G)} gr(C^{\mathcal{I}}).$$

Далее определения концептов используются для поиска соответствующих им подграфов в конечных графах.

Для проверки выполнимости, непересекаемости и других логических свойств концептов в терминологиях на графах можно использовать уже существующие программы логического вывода для дескрипционных логик.

## § 2. Графы изображений

Особенность предлагаемого подхода к анализу изображений заключается в том, что с помощью дескрипционной логики описываются не характеристики отдельных пикселей изображения, а его *сегменты* — связные множества пикселей с «близкими» характеристиками, их границы и скелетоны.

В ходе обработки изображения с целью структурного анализа изображенных объектов формируется *графовая модель изображения*, представляемая в виде многослойного атрибутивного графа, содержащего несколько информационных слоев, связанных пространственными отношениями.

*Слой* — это граф, вершины и дуги которого представляют объекты одной природы, то есть имеют фиксированные для слоя наборы атрибутов.

*Многослойный атрибутивный граф* изображения (*MLAG* — Multi Layered Attributive Graph) — это структура вида

$$MLAG = \langle V_0, G_{SEGM}, G_{CONT}, G_{SKEL}, R \rangle,$$

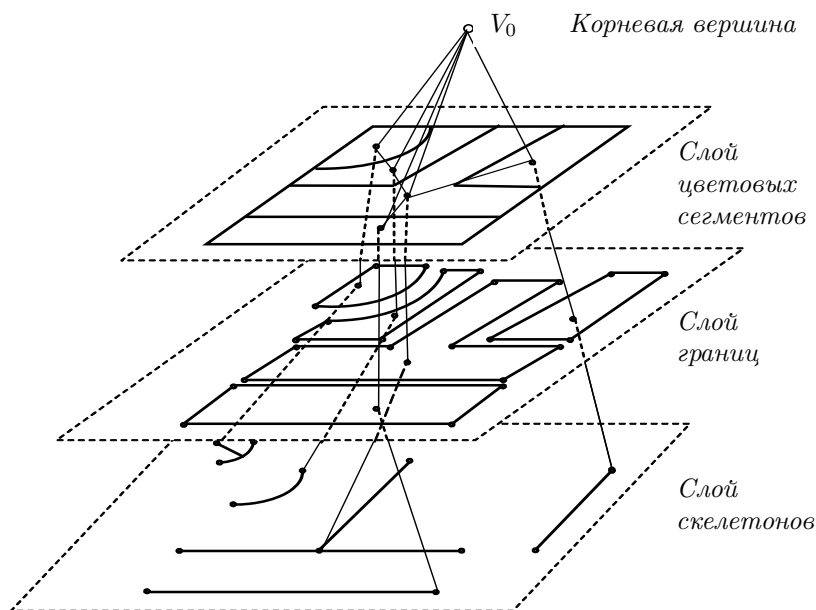


Рис. 1. Многослойный атрибутивный граф изображения

где

$V_0$  — корневая вершина с общими атрибутами изображения;

$G_{SEGM}$  — слой цветowych сегментов;

$G_{CONT}$  — слой границ (контуров) сегментов;

$G_{SKEL}$  — слой скелетов сегментов;

$R$  — множество отношений «предок–потомок» между слоями графа (отношений между вершиной или ребром из одного слоя и вершиной или ребром из другого слоя).

Как и в случае с тематическими слоями в геоинформационных системах, многослойный атрибутивный граф облегчает анализ информации.

$MLAG$  изображения строится специальной программой — *интерпретатором изображений* [16], которая осуществляет цветовую сегментацию исходного изображения, векторизацию границ сегментов и построение их скелетов. Элементами границ и скелетов сегментов являются отрезки прямых и дуги окружностей, которым в  $MLAG$  сопоставляются вершины и дуги соответствующих графов, нагруженные атрибутами с геометрической и прочей информацией, необходимой для анализа исходного изображения без использования пикселей.

На рис. 1 показан принцип построения  $MLAG$  изображения.

Для описания изображений будем использовать дескрипционную логику  $\mathcal{ALC}(GI)$  — расширение  $\mathcal{ALCOIQf}[\square, \sqcap, \circ, *, id]$  дескрипционной логики  $\mathcal{ALC}$ , использующее, кроме графовых и прочих, следующие атомарные концепты и роли для описания  $MLAG$  изображения:

CA4)  $Angle(x : Node)$  — вершина  $x$  имеет степень 2 и (наименьший) угол между инцидентными ей дугами больше заданной пороговой величины  $\delta$  (по умолчанию  $\delta = \pi/4$ );

CA5)  $Segment(x : Graph)$  — подграф  $x$  является множеством границ соответствующего сегмента;

RA4)  $hasContour(x : Segment, y : Cycle)$  — цикл  $y$  является одной из границ (контуров) сегмента  $x$ ;

RA5)  $hasSkel(x : Segment, y : Connected)$  — подграф  $y$  является скелетом сегмента  $x$ , функциональное отношение (атрибут);

RA6)  $hasColor(x : Segment, y : Number)$  — число  $y$  является цветом сегмента  $x$ , функциональное отношение (атрибут);

RA7)  $hasX(x : Node, y : Number)$  — число  $y$  является координатой  $X$  вершины  $x$ , функциональное отношение (атрибут);



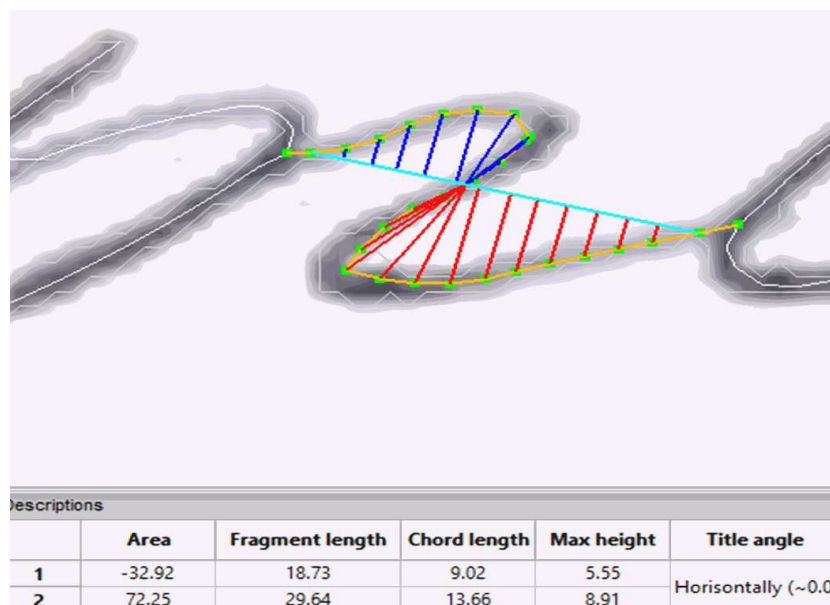


Рис. 2. Вычисление атрибутов цепочки

RA8)  $hasY(x : Node, y : Number)$  — число  $y$  является координатой  $Y$  вершины  $x$ , функциональное отношение (атрибут).

Пусть

*хорда* цепочки — это направленный отрезок, соединяющий её концы;

*начало* цепочки — тот ее конец, который лежит левее (выше, если хорда вертикальна) другого;

*сегмент цепочки* — это часть кривой, соответствующей цепочке, ограниченная соседними точками пересечения кривой с хордой; сегмент цепочки называется *левым* (*правым*) если он находится слева (справа) от ее хорды;

*высота* сегмента цепочки — максимальное расстояние от хорды до точек сегмента цепочки;

*площадь* сегмента цепочки — площадь фигуры, образованной сегментом цепочки и соответствующим отрезком хорды.

На рис. 2 показан принцип работы геометрических функций, вычисляющих атрибуты цепочек. Концы цепочки соединяются хордой. Подсчитываются ее длина, угол наклона, а также высота и площадь каждого сегмента.

Для описания цепочек и циклов используются следующие атомарные атрибуты:

RA9)  $hasSegmentNumber(x : Chain, y : Number)$  — число  $y$  — это количество сегментов цепочки  $x$ ;

RA10)  $hasChordLength(x : Chain, y : Size)$  — вычисляет лингвистическое значение {Small, Medium, Large} длины хорды, соединяющей концы цепочки;

RA11)  $hasChordAngle(x : Chain, y : Size)$  — вычисляет лингвистическое значение {Small, Medium, Large} угла между прямой, на которой лежит хорда цепочки, и (какой-либо) горизонтальной прямой;

RA12)  $hasMaxHeight(x : Chain, y : Size)$  — вычисляет лингвистическое значение {Small, Medium, Large} максимальной высоты сегментов цепочки  $x$ ;

RA13)  $hasArea(x : Chain, y : Size)$  — вычисляет лингвистическое значение {Small, Medium, Large} суммарной площади всех сегментов цепочки  $x$ ;

RA14)  $hasLeftArea(x : Chain, y : Size)$  — вычисляет лингвистическое значение {Small, Medium, Large} суммарной площади левых сегментов цепочки  $x$ ;

RA15)  $hasRightArea(x : Chain, y : Size)$  — вычисляет лингвистическое значение {Small, Medium, Large} суммарной площади правых сегментов цепочки  $x$ ;

RA16)  $hasLeft(x : Chain, y : Node)$  — вершина  $y$  является левым концом цепочки  $x$ ;



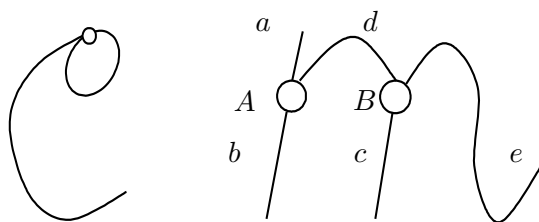


Рис. 3. Пример написания рукописных букв *c* и *m*

RA17)  $hasRight(x : Chain, y : Node)$  — вершина  $y$  является правым концом цепочки  $x$ ;

RA18)  $hasTop(x : Chain, y : Node)$  — вершина  $y$  является верхним концом цепочки  $x$ ;

RA19)  $hasBottom(x : Chain, y : Node)$  — вершина  $y$  является нижним концом цепочки  $x$ ;

RA20)  $hasCycleArea(x : Cycle, y : Size)$  — вычисляет лингвистическое значение {Small, Medium, Large} площади, ограниченной циклом  $x$ ;

RA21)  $hasLeftNode(x : Cycle, y : Node)$  — вершина  $y$  расположена левее «центра тяжести» цикла  $x$ ;

RA22)  $hasRightNode(x : Cycle, y : Node)$  — вершина  $y$  расположена правее «центра тяжести» цикла  $x$ ;

RA23)  $hasTopNode(x : Cycle, y : Node)$  — вершина  $y$  расположена выше «центра тяжести» цикла  $x$ ;

RA24)  $hasBottomNode(x : Cycle, y : Node)$  — вершина  $y$  расположена ниже «центра тяжести» цикла  $x$ .

Логическое описание графа изображения  $GI$  является расширением соответствующей терминологии  $\mathcal{T}(GI)$  описаниями атомарных концептов и ролей, присущих  $MLAG$ .

Как правило, в задачах анализа изображений не все вершины графов границ и скелетов сегментов одинаково важны. Поэтому имеет смысл выделить те, что интересны, и цепочки, их соединяющие:

CB6)  $Selected(x : Node) \equiv (Node \sqcap \neg(=2 hasIncident^-)) \sqcup Angle$  — вершина  $x$  является выделенной, опорной для анализа изображения;

CB7)  $Link(x : Chain) \equiv Chain \sqcap =2 hasNode.Selected$  — цепочка  $x$  является выделенной (особой), связывает выделенные вершины;

CB8)  $Loop(x : Cycle) \equiv Cycle \sqcap <2 hasNode.Selected$  — цепочка  $x$  является петлей.

Это понадобится в дальнейшем для более надежного разделения объектов на изображении (в нашем примере — некоторых рукописных букв, написанных слитно).

### § 3. Пример: распознавание рукописного текста

В качестве примера использования логики  $\mathcal{ALC}(GI)$  для описания и анализа изображений рассмотрим задачу распознавания слитного рукописного текста. Основной проблемой в этой задаче является получение достаточно общего и в то же время достаточно точного описания букв, легко проверяемого неким алгоритмом. Покажем, что язык  $\mathcal{ALC}(GI)$  достаточен (может использоваться) и удобен для этих целей.

В данном примере в графе  $MLAG$  нас интересует только слой  $G_{SKEL}$  — граф скелетов (см. выше).

Исходное изображение — отсканированная страница рукописного текста.

На рис. 3 показаны описанные ниже варианты написания рукописных букв *c* и *m*.

Пусть

CB9)  $RightArc(x : Link) \equiv Link \sqcap (hasChordLength = Medium) \sqcap (hasSegmentNumber = 1) \sqcap (hasLRightArea = Medium)$  — выделенная цепочка, имеющая хорду средней длины и один сегмент справа (по ходу) от хорды, площадь которого средняя (левая дуга);

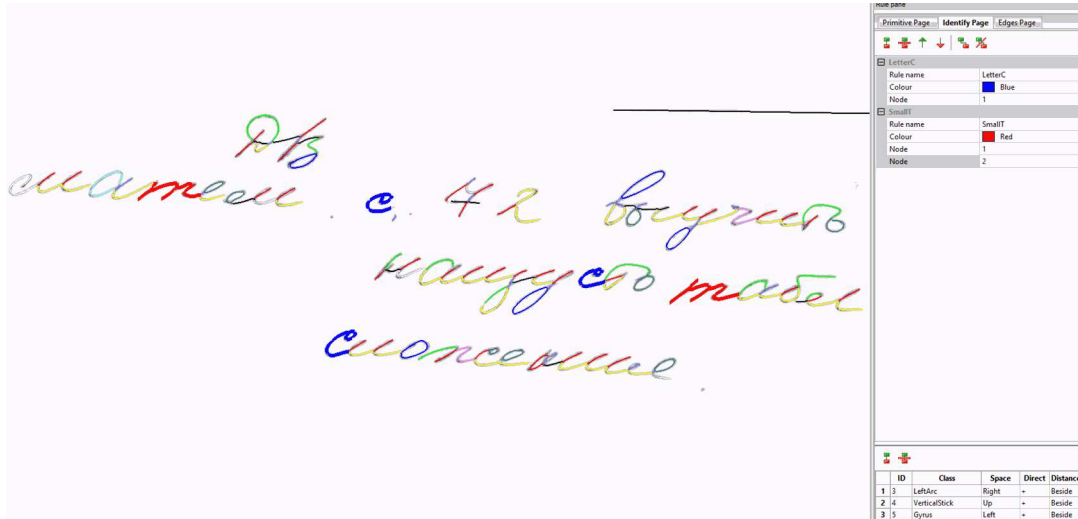


Рис. 4. Пример распознавания рукописных букв *c* и *m*

CB10)  $SmallLoop(x : Loop) \equiv Loop \sqcap (hasCycleArea = Small)$  — петля, имеющая малую площадь.

Тогда строчную букву *c* (один из вариантов написания) можно определить как  $LetterSmallC \equiv RightArc \sqcap \exists hasTop \circ (hasLeftNode \sqcap hasTopNode)^- . SmallLoop$ .

Пусть

CB11)  $Stick(x : Link) \equiv Link \sqcap (MaxHeight = Small)$  — выделенная цепочка, имеющая малое отклонение от хорды (палочка);

CB12)  $VerticalStick(x : Link) \equiv Stick \sqcap (hasChordLength = Medium) \sqcap (Angle = Large)$  — палочка, имеющая среднюю длину и вертикальное положение (вертикальная палочка);

CB13)  $ShortVerticalStick(x : Link) \equiv Stick \sqcap (hasChordLength = Small) \sqcap (Angle = Large)$  — палочка, имеющая малую длину и вертикальное положение (короткая вертикальная палочка);

CB14)  $LeftArc(x : Link) \equiv Link \sqcap (hasChordLength = Medium) \sqcap \sqcap (hasSegmentNumber = 1) \sqcap (hasLeftArea = Medium)$  — выделенная цепочка, имеющая хорду средней длины и один сегмент слева (по ходу) от хорды, площадь которого средняя (левая дуга);

CB15)  $Gyrus(x : Link) \equiv Link \sqcap (hasSegmentNumber = 2) \sqcap (hasArea = Medium)$  — выделенная цепочка, имеющая два сегмента (один слева, другой справа от хорды) и площадь каждого сегмента средняя (извилина).

Тогда строчную букву *m* (один из вариантов написания) можно определить так:

$$LetterSmallT \equiv ShortVerticalStick \sqcap \exists hasBottom \circ hasTop^- . VerticalStick \sqcap \exists hasBottom \circ hasLeft^- . LeftArc( \sqcap \exists hasRight \circ hasTop^- . VerticalStick \sqcap \exists hasRight \circ hasLeft^- . Gyrus).$$

На рис. 4 показан результат эксперимента по распознаванию описанных вариантов написания рукописных букв *c* и *m*.

### Заключение

Цель данной статьи — показать возможности использования дескрипционных логик на графах для описания и эффективного анализа изображений.

Модель изображения в виде предложенного многослойного атрибутивного графа легко описывается в логике  $\mathcal{ALCC}(GI)$  и существенно расширяет возможности структурного анализа

изображений с помощью дескрипционных логик. Например, вместо перебора множества пикселей [11, 12], проверяются только цепочки графа изображения, соединяющие пары выделенных вершин.

Формальный язык логики  $\mathcal{ALC}(GI)$  характерен наличием множества функций, анализирующих геометрическую форму и относительное расположение структурных элементов изображения, что позволяет легко строить описания объектов, содержащихся в нем.

Пример, выполненный на достаточно сложных для распознавания объектах — рукописных буквах, показывает, на наш взгляд, перспективность данного подхода.

#### СПИСОК ЛИТЕРАТУРЫ

1. Фу К. Структурные методы в распознавании образов. М.: Мир, 1977. 319 с.
2. Stotz A., Nagi R., Sudit M. Incremental graph matching for situation awareness // 2009 12th International Conference on Information Fusion. Seattle, WA, USA. IEEE, 2009. P. 452–459.  
<https://ieeexplore.ieee.org/document/5203773/>
3. Luqman M.M., Ho H.N., Burie J.-C., Ogier J.-M. Automatic indexing of comic page images for query by example based focused content retrieval // 10th IAPR International Workshop on Graphics Recognition. Lehigh University. Bethlehem, PA, USA, 2013. <https://hal.archives-ouvertes.fr/hal-00944427>
4. Hu M.-K. Visual pattern recognition by moment invariants // IEEE Transactions on Information Theory. 1962. Vol. 8. Issue 2. P. 179–187. DOI: [10.1109/TIT.1962.1057692](https://doi.org/10.1109/TIT.1962.1057692)
5. Zhang Q., Song X., Shao X., Zhao H., Shibasaki R. Attributed graph mining and matching: an attempt to define and extract soft attributed patterns // 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA. IEEE, 2014. P. 1394–1401. DOI: [10.1109/CVPR.2014.181](https://doi.org/10.1109/CVPR.2014.181)
6. Сапаров А.Ю., Бельтюков А.П. Математическое моделирование изображений формул с целью их распознавания // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2013. Вып. 1. С. 153–167. DOI: [10.20537/vm130114](https://doi.org/10.20537/vm130114)
7. Сапаров А.Ю., Широкова И.Ю. Разработка пользовательского интерфейса для управления процессом распознавания рукописных математических формул // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2016. Т. 26. Вып. 1. С. 141–152. DOI: [10.20537/vm160111](https://doi.org/10.20537/vm160111)
8. Baader F., Calvanese D., McGuinness D.L., Nardi D., Patel-Schneider P.F. (Eds.). The description logic handbook. Theory, implementation, and applications. Cambridge: Cambridge University Press, 2003. xviii+555 p.
9. Lutz C. Description logics with concrete domains — a survey // Advances in modal logic. Vol. 4 / Balbiani P., Suzuki N.-Y., Wolter F., Zakharyashev M. (Eds.). London: King's College Publications, 2003. P. 265–296.
10. Золин Е.Е. Дескрипционная логика (лекции). <http://lpcs.math.msu.su/~zolin/dl/>
11. Simou N., Athanasiadis Th., Stoilos G., Kollias S. Image indexing and retrieval using expressive fuzzy description logics // Signal, Image and Video Processing. 2008. Vol. 2. Issue 4. P. 321–335. DOI: [10.1007/s11760-008-0084-1](https://doi.org/10.1007/s11760-008-0084-1)
12. Dasiopoulou S., Kompatsiaris I., Strintzis M.G. Applying fuzzy DLs in the extraction of image semantics // Journal on Data Semantics XIV. Lecture Notes in Computer Science. Vol. 5880 / Spaccapietra S., Delcambre L. (Eds.). Berlin: Springer, 2009. P. 105–132. DOI: [10.1007/978-3-642-10562-3\\_4](https://doi.org/10.1007/978-3-642-10562-3_4)
13. Falomir Z., Jiménez-Ruiz E., Escrig M.T., Museros L. Describing images using qualitative models and description logics // Spatial Cognition & Computation. 2011. Vol. 11. Issue 1. P. 45–74. DOI: [10.1080/13875868.2010.545611](https://doi.org/10.1080/13875868.2010.545611)
14. Hudelot C., Atif J., Bloch I.  $\mathcal{ALC}(\mathbf{F})$ : a new description logics for spatial reasoning in images // Computer Vision — ECCV 2014 Workshops. Part II. Lecture Notes in Computer Science. Vol. 8926 / Agapito L., Bronstein M., Rother C. (Eds.). Cham: Springer, 2015. P. 370–384. DOI: [10.1007/978-3-319-16181-5\\_26](https://doi.org/10.1007/978-3-319-16181-5_26)
15. Hudelot C., Atif J., Bloch I. Integrating bipolar fuzzy mathematical morphology in description logics for spatial reasoning // European Conference on Artificial Intelligence ECAI 2010. Lisbon, Portugal, 2010. P. 497–502. DOI: [10.3233/978-1-60750-606-5-497](https://doi.org/10.3233/978-1-60750-606-5-497)
16. Кучуганов А.В. Биоинспирированные алгоритмы выделения информативных признаков изображений // Известия Томского политехнического университета. 2012. Т. 321. № 5: Управление, вычислительная техника и информатика. С. 141–145. <http://earchive.tpu.ru/handle/11683/4528>

Поступила в редакцию 05.05.2018

Кучуганов Михаил Валерьевич, Удмуртский государственный университет, 426034, Россия, г. Ижевск, ул. Университетская, 1.

E-mail: [qmikle1@yandex.ru](mailto:qmikle1@yandex.ru)

Кучуганов Александр Валерьевич, к. т. н., Ижевский государственный технический университет им. М. Т. Калашникова, 426069, Россия, г. Ижевск, ул. 30 лет Победы, 2.

E-mail: [Aleks\\_KAV@udm.ru](mailto:Aleks_KAV@udm.ru)

***M. V. Kuchuganov, A. V. Kuchuganov***

**Description logic on image graphs**

**Citation:** *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2018, vol. 28, issue 4, pp. 582–594 (in Russian).

*Keywords:* description logic, graph, image recognition, handwritten text.

MSC2010: 03B70, 68T10

DOI: [10.20537/vm180410](https://doi.org/10.20537/vm180410)

In this paper, it's proposed to use the extension  $\mathcal{ALC}(GI)$  of description logic  $\mathcal{ALC}$  for the formal description and structural analysis of images. Concepts and the roles of  $\mathcal{ALC}(GI)$  are interpreted on an image graph and its subgraphs. The model of image in the form of multi-layered attribute graph is presented. It contains a layer of color segments, a layer of boundaries and a layer of skeletons. Each layer is a planar graph, layers are linked by means of “ancestor–descendant” relations. The transition from the pixel representation of an image to the graph one allows us to increase the effectiveness of image analysis. Examples of terminological axioms that define structural elements of an image and letters composed of them, as well as the results of an experiment of recognizing letters in a cursive handwritten text are presented.

## REFERENCES

1. Fu K.S. *Syntactic methods in pattern recognition*, New York–London: Academic Press, 1974, xi+295 p.
2. Stotz A., Nagi R., Sudit M. Incremental graph matching for situation awareness, *2009 12th International Conference on Information Fusion*, Seattle, WA, USA, IEEE, 2009, pp. 452–459. <https://ieeexplore.ieee.org/document/5203773/>
3. Luqman M.M., Ho H.N., Burie J.-C., Ogier J.-M. Automatic indexing of comic page images for query by example based focused content retrieval, *10th IAPR International Workshop on Graphics Recognition*, Lehigh University, Bethlehem, PA, USA, 2013. <https://hal.archives-ouvertes.fr/hal-00944427>
4. Hu M.-K. Visual pattern recognition by moment invariants, *IEEE Transactions on Information Theory*, 1962, vol. 8, issue 2, pp. 179–187. DOI: [10.1109/TIT.1962.1057692](https://doi.org/10.1109/TIT.1962.1057692)
5. Zhang Q., Song X., Shao X., Zhao H., Shibasaki R. Attributed graph mining and matching: an attempt to define and extract soft attributed patterns, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, IEEE, 2014. P. 1394–1401. DOI: [10.1109/CVPR.2014.181](https://doi.org/10.1109/CVPR.2014.181)
6. Saparov A.Yu., Bel'tyukov A.P. Mathematical modeling of formula images for their recognition, *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2013, issue 1, pp. 153–167 (in Russian). DOI: [10.20537/vm130114](https://doi.org/10.20537/vm130114)
7. Saparov A.Yu., Shirobokova I.Yu. User interface development to manage the process of handwritten mathematical formula recognition, *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2016, vol. 26, issue 1, pp. 141–152 (in Russian). DOI: [10.20537/vm160111](https://doi.org/10.20537/vm160111)
8. Baader F., Calvanese D., McGuinness D.L., Nardi D., Patel-Schneider P.F. (Eds.). *The description logic handbook. Theory, implementation, and applications*, Cambridge: Cambridge University Press, 2003, xviii+555 p.
9. Lutz C. Description logics with concrete domains — a survey, *Advances in modal logic. Vol. 4*, Eds.: Balbiani P., Suzuki N.-Y., Wolter F., Zakharyashev M. London: King's College Publications, 2003, pp. 265–296.

10. Zolin E.E. Description logic (lectures). (In Russian). <http://lpcs.math.msu.su/~zolin/dl/>
11. Simou N., Athanasiadis Th., Stoilos G., Kollias S. Image indexing and retrieval using expressive fuzzy description logics, *Signal, Image and Video Processing*, 2008, vol. 2, issue 4, pp. 321–335. DOI: [10.1007/s11760-008-0084-1](https://doi.org/10.1007/s11760-008-0084-1)
12. Dasiopoulou S., Kompatsiaris I., Strintzis M.G. Applying fuzzy DLs in the extraction of image semantics, *Journal on Data Semantics XIV. Lecture Notes in Computer Science. Vol 5880*, Eds.: Spaccapietra S., Delcambre L. Berlin: Springer, 2009, pp. 105–132. DOI: [10.1007/978-3-642-10562-3\\_4](https://doi.org/10.1007/978-3-642-10562-3_4)
13. Falomir Z., Jiménez-Ruiz E., Escrig M.T., Museros L. Describing images using qualitative models and description logics, *Spatial Cognition & Computation*, 2011, vol. 11, issue 1, pp. 45–74. DOI: [10.1080/13875868.2010.545611](https://doi.org/10.1080/13875868.2010.545611)
14. Hudelot C., Atif J., Bloch I.  $\mathcal{ALC}(\mathbf{F})$ : a new description logics for spatial reasoning in images, *Computer Vision – ECCV 2014 Workshops. Part II. Lecture Notes in Computer Science. Vol 8926*, Eds.: Agapito L., Bronstein M., Rother C. Cham: Springer, 2015, pp. 370–384. DOI: [10.1007/978-3-319-16181-5\\_26](https://doi.org/10.1007/978-3-319-16181-5_26)
15. Hudelot C., Atif J., Bloch I. Integrating bipolar fuzzy mathematical morphology in description logics for spatial reasoning, *European Conference on Artificial Intelligence ECAI 2010*, Lisbon, Portugal, 2010, pp. 497–502. DOI: [10.3233/978-1-60750-606-5-497](https://doi.org/10.3233/978-1-60750-606-5-497)
16. Kuchuganov A.V. Bioinspired algorithms for extracting informative features of images, *Proceedings of Tomsk Polytechnic University*, 2012. vol. 321, issue 5: Management, Computer Science and Informatics. pp. 141–145 (in Russian). <http://earchive.tpu.ru/handle/11683/4528>

Received 05.05.2018

Kuchuganov Mikhail Valer'evich, Udmurt State University, ul. Universitetskaya, 1, Izhevsk, 426034, Russia.  
E-mail: [qmikle1@yandex.ru](mailto:qmikle1@yandex.ru)

Kuchuganov Aleksandr Valer'evich, Candidate of Engineering, Kalashnikov Izhevsk State Technical University, ul. 30 let Pobedy, 2, Izhevsk, 426069, Russia.  
E-mail: [Aleks\\_KAV@udm.ru](mailto:Aleks_KAV@udm.ru)