УДК 004 **DOI** 10.15622/ia.22.5.8

А.В. Воробьев, А.Н. Лапин, Г.Р. Воробьева

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ АВТОМАТИЗИРОВАННОГО РАСПОЗНАВАНИЯ И ОЦИФРОВКИ АРХИВНЫХ ДАННЫХ ОПТИЧЕСКИХ НАБЛЮДЕНИЙ ПОЛЯРНЫХ СИЯНИЙ

Воробьев А.В., Лапин А.Н., Воробьева Г.Р. **Программное обеспечение для автоматизированного распознавания и оцифровки архивных данных оптических наблюдений полярных сияний.**

Аннотация. Одним из основных инструментов регистрации полярных сияний является оптическое наблюдение небосвода в автоматическом режиме с помощью камер всего неба. Результаты наблюдений фиксируются в специальных мнемонических таблицах, аскаплотах. Аскаплоты предоставляют суточную информацию о наличии или отсутствии облачного покрова и полярных сияний в различных частях небосвода и традиционно используются для исследования суточного распределения полярных сияний в заданном регионе, а также для расчета вероятности их наблюдения в других регионах в соответствии с уровнем геомагнитной активности. Обработка аскаплотов в настоящее время осуществляется вручную, что сопряжено с существенными временными затратами и высокой долей ошибок, возникающих по причине человеческого фактора. Для повышения эффективности обработки аскаплотов авторами предложен подход, обеспечивающий автоматизацию распознавания и оцифровки данных оптических наблюдений полярных сияний. Предложена формализация структуры аскаплота, применяемая для обработки его изображения, а также извлечение соответствующих результатов наблюдений и формирование результирующего набора данных. Подход предусматривает использование алгоритмов машинного зрения (в частности, в данном случае имеет место применение алгоритма классификации по правилам) и применение специализированной маски – отладочного изображения для оцифровки, представляющего собой цветное изображение, в котором задано общее положения ячеек аскаплотов. Предложенный подход и соответствующие алгоритмы реализованы в форме программного обеспечения для распознавания и оцифровки архивных данных оптических наблюдений полярных сияний. Решение представляет собой однопользовательское настольное программное обеспечение, позволяющее пользователю в пакетном режиме выполнять преобразование изображений аскаплотов в таблицы, доступные для последующей обработки и анализа. Результаты проведенных вычислительных экспериментов показали, что применение предложенного программного обеспечения позволит избежать ошибок при оцифровке аскаплотов, с одной стороны, и существенно повысить скорость соответствующих вычислительных операций, с другой. В совокупности это позволит повысить эффективность обработки аскаплотов и проведения исследований в соответствующей области.

Ключевые слова: обработка данных, оцифровка данных наблюдений, аскаплоты, программное обеспечение.

1. Введение. В условиях интенсивного освоения космического пространства и совершенствования систем наземной и космической навигации все большее значение приобретает исследование характеристик полярных сияний. Основанием тому является подтвержденная многими исследованиями взаимосвязь между

характеристиками полярных сияний и различными геофизическими процессами, протекающими в высокоширотной ионосфере Земли и способными негативно повлиять на различные объекты и системы техносферы [1].

В настоящее время основным инструментом оптического наблюдения полярных сияний являются автоматические камеры всего неба (All-Sky Camera), снабженные объективами с полем зрения в 180 [2]. При этом для исследования полярных сияний, зарегистрированных камерами всего неба, необходима первичная обработка проведенных наблюдений, что в значительной степени повышает эффективность соответствующих научных исследований [3].

Для первичной обработки наблюдений камер всего неба используются так называемые аскафильмы (all sky camera films – кадры непрерывной регистрации небосвода камерой всего неба [4]). В результате обработки аскафильмов составляются мнемонические таблицы, называемые аскаплотами (all sky camera plots [4]). В соответствии с предопределенной нотацией аскаплоты предоставляют суточную информацию о наличии или отсутствии облачного покрова и полярных сияний в различных частях небосвода [4, 5].

Результаты обработки аскаплотов используются для исследования суточного распределения полярных сияний в заданном пространственном регионе и для расчета вероятности наблюдения в других регионах в зависимости от уровня локальной геомагнитной активности. При этом обработка полученных из аскафильмов аскаплотов осуществляется преимущественно вручную, что крайне негативно сказывается на оперативности соответствующих исследований. Немаловажно отметить и высокую вероятность ошибок в результатах интерпретации аскаплотов, обусловленную человеческим фактором. Перечисленные недостатки практикуемого в настоящее время подхода [4] существенно снижают эффективность применения результатов анализа аскаплотов для исследования пространственной и временной анизотропии полярных сияний.

В этой связи ожидается, что переход к автоматической обработке аскаплотов позволит существенно повысить эффективность исследований в обозначенной области. Многоэтапная автоматическая проверка промежуточных результатов обработки аскаплотов позволит избежать или существенно сократить число ошибок интерпретации данных, характерных для неавтоматического разбора результатов наблюдений полярных сияний. Кроме того, возможность пакетной

обработки суточных аскаплотов в автоматическом режиме ожидаемо повысит скорость и удобство получения результатов наблюдений в виде, приемлемом для последующего анализа сторонними программными системами и библиотеками.

Таким образом, обработка результатов оптического наблюдения полярных сияний практически невозможна без создания системы автоматической интерпретации аскаплотов. Известные подходы к анализу аскаплотов не предполагают существенной автоматизации, но в то же время предоставляют методический базис для разработки соответствующей информационной системы [5 – 7]. При этом регламентированная отраслевыми спецификациями структура аскаплота позволяет разработать универсальное решение, не привязанное к результатам наблюдений конкретных научных организаций. Создание и внедрение обозначенной системы позволит повысить эффективность научных изысканий с исследовательской точки зрения, а также существенно повысить реактивность сопутствующего программного обеспечения с точки зрения инженерной реализации.

2. Постановка задачи. Для достижения поставленной цели представляется целесообразным решить ряд задач научного и прикладного характера. На первом этапе необходимо формализовать структуру аскаплота таким образом, чтобы полученная в итоге математическая модель могла быть положена в основу соответствующих методов обработки и анализа данных. Далее на имеющихся тестовых данных необходимо разработать алгоритм оцифровки аскаплотов, предусматривающий последовательную обработку данных, а также проверку и корректировку промежуточных результатов. На завершающем этапе предполагается разработка исследовательского прототипа соответствующего программного обеспечения, чему предшествует определение его архитектуры, инфраструктуры и стека используемых технологий с учетом особенностей обрабатываемой информации.

Существующие подходы к оцифровке табличных данных применяют методы интеллектуального анализа [8], такие как графовые нейронные сети [9 – 11], рекуррентные нейронные сети [12 – 14], семантическая сегментация [15] и сверточные нейронные сети [16, 17], что позволяет выделять сложную табличную архитектуру [18], а также корректировать дефекты изображения, вызванные фотокамерой. Для последующей оцифровки символов внутри ячеек таблицы, как правило, применяются методы оптического распознавания символов [19 – 21]. Проведенный авторами анализ показал, что основными

недостатками рассмотренных методов являются высокие требования к используемым вычислительным ресурсам, а также необходимость проведения дополнительных этапов предобработки и постобработки данных.

При этом важно отметить, что в данной статье в качестве входных данных используются аскаплоы, которые имеют табличную структуры и представлены в виде матрицы размером 5 х 48 ячеек. Также стоит отметить, что в отличие от классических табличных данных, которые представлены в виде текстовых символов, в ячейках аскаплота находятся геометрические фигуры, которым свойственны графические неточности, такие как сдвиг относительно центра самой ячейки и выход фигуры за пределы одной ячейки. Немаловажным является и тот факт, что одной рассматриваемых фигур является полностью закрашенный прямоугольник, что накладывает определенные трудности при определении границ столбцов таблицы, поскольку не представляется возможным локально определить раздел между фигурой и границей ячейки.

С учетом сказанного, применение перечисленных и иных подобных им методов оцифровки табличных данных в рассматриваемом случае не представляется авторам возможным. В этой связи представляется целесообразным разработать специализированный алгоритм для оцифровки аскаплотов с применением машинного зрения.

Для формализации и реализации решений поставленных задач предполагается использовать модели и методы распознавания образов, элементы теоретико-множественного базиса для описания структуры аскаплотов, подходы к построению схем алгоритмов, а также технологии обработки и анализа информации.

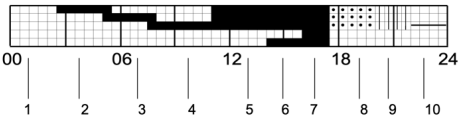
В качестве информационного обеспечения, а также используемой для эмпирических исследований входной информации, в рамках настоящей работы выступают результаты оптических наблюдений полярных сияний, зарегистрированные камерами всего неба на Кольском полуострове, в обсерватории Ловозеро [22 – 24]. Указанные данные оформлены в аскаплоты, которые, в свою очередь, представлены в виде документов формата pdf, содержащих соответствующие таблицы данных, которые построены по спецификациям описания таких данных.

3. Описание и формализация исходных данных. В качестве исходных данных используются результаты оптических наблюдений полярных сияний, зарегистрированные камерами всего неба. При этом важным параметром является так называемый зенит обсерватории,

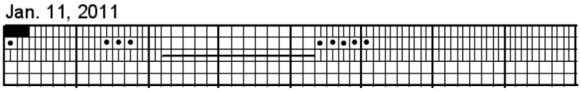
характеризующий направление вертикального подъема над точкой наблюдения [4]. Для корректной записи аскафильмов зенит обсерватории должен располагаться в центре кадра [25]. Исправленный геомагнитный меридиан обсерватории располагается вдоль линии, проходящей в направлении сверху вниз через центр кадра и указывающей на север. Соответственно при этом запад и восток находятся в правой и левой частях кадра соответственно.

Каждый аскаплот описывает результаты суточного наблюдения полярных сияний камерой всего неба. Данные фиксируются в пятистрочной таблице, столбцы которой соответствуют последовательным получасовым временных интервалам (рисунок 1). Первые три строки показывают факт наличия полярного сияния в северной, зенитной и южной частях неба соответственно [5]. Четвертая и пятая строки стандартного пятистрочного аскаплота характеризуют интенсивность полярного сияния в зенитном диапазоне.

Кроме того, непосредственно в таблице аскаплота результаты наблюдений определенным образом маркируются. Специальные обозначения используются, в частности, в пяти различных случаях: при отсутствии сияния, наблюдении сияния, в условиях частичной или сплошной облачности, отсутствия наблюдений и пр. Такая детальная нотация позволяет с высокой степенью информативности описать результаты наблюдения полярных сияний с учетом их видимости и интенсивности в соответствующие периоды времени [4, 5].



а)



б)

Рис. 1. Формат представления данных в виде аскаплота: а) 1 – сияние не наблюдаются; 2 – сияние наблюдается в северной области; 3 – сияние в зените; 4 – сияние на юге; 5 – сияние наблюдается в зените, северной и южной областях; 6 – в зените наблюдаются умеренное сияние, кроме этого свечение присутствует в северной и южной областях; 7 – в зените наблюдается сильное сияние, кроме этого свечение присутствует в северной и южной областях; 8 – частичная облачность; 9 – сплошная облачность; 10 – регистрация не проводилась; б) пример аскаплота ГС LOZ за 11.01.2011 г. [22]

Представляется целесообразным формализованное описание структуры аскаплота в теоретико-множественном базисе. Пусть суточные оптические наблюдения полярных сияний представлены в аскаплоте A, составленном из подмножеств H:

$$A = \bigcup_{i=0}^{47} H_i, \tag{1}$$

где каждому подмножеству H_i соответствует получасовой интервал наблюдений (таким образом, $i = 0,\ldots,47$).

Каждый интервал H_i представляет собой последовательность из 5 значений в соответствии с пятистрочной структурой аскаплота, описанной выше. Каждое из 5 значений характеризует определенный параметр наблюдения в соответствии с наличием и интенсивностью полярного сияния в соответствующем пространственном направлении относительно камеры всего неба. Структуру интервала с учетом сказанного можно представить следующим образом:

$$H_i = \{h_0^i, h_1^i, \ldots, h_4^i\}, \tag{2}$$

где h_j^i – результат оптического наблюдения полярного сияния в H_i-й получасовой интервал, характеризующий значение j-го параметра ($j = 0,\ldots,4$).

Каждый из пяти обозначенных параметров, представленных в интервале аскаплота, может принимать одно из 5 предопределенных значений. При этом непосредственно в исходном аскаплоте соответствующее значение помечается посредством заданной графической нотации (заливка цветом, вертикально или горизонтально перечеркнутая ячейка, пустая ячейка и пр.). Поскольку параметры результатов наблюдений в аскаплоте могут принимать строго определенные значения, представляется целесообразным формализовать их в виде соответствующего домена.

Домен допустимых значений элементов аскаплота может быть описан в виде множества, состоящего из пяти возможных значений:

$$D = \bigcup_{j=0}^{4} d_j, \tag{3}$$

где d_j – атомарное значение, доступное для использования при описании результатов наблюдений полярных сияний.

Каждому элементу аскаплота ставится в соответствие один и только один элемент из домена D. При этом количество элементов с одинаковыми значениями в составе одного аскаплота не ограничено, равно как и не требуется наличия каждого из предусмотренных доменом значений:

$$h_j^i = d;\ h_j^i \in H_i, d\ \in D, \tag{4}$$

где d – значение j-го параметра i-го элемента аскаплота H.

Кроме того, представляется целесообразным отметить, что в составе аскаплота не допускаются параметры с отсутствующими значениями. В этой связи во избежание возможных коллизий в аскаплоте используются значения по умолчанию из имеющегося домена, выбор которого определяется разработчиками:

$$h_j^i = d;\ h_j^i \in H_i, d\ \in D, d \neq \emptyset, \tag{5}$$

где d – значение j-го параметра i-го элемента аскаплота H.

Важно отметить, что в одном наборе результатов оптических наблюдений за заданный временной интервал могут присутствовать как уникальные, так и повторяющиеся аскаплоты, что, в свою очередь, свидетельствует о вариациях соответствующих анализируемых параметров. При этом возможны ситуации, при которых в течение заданного временного периода в наборе данных присутствуют такие аскаплоты, которые содержат как все доступные в домене значения, так и их подмножества. Последний вариант развития событий является наиболее часто встречаемым в реальных наблюдениях полярных сияний.

Кроме того, ввиду различных факторов техногенного и естественного происхождения в течение определенного аскаплотом периода (одного или нескольких получасовых, либо полного суточного временного интервала) оптические наблюдения полярных сияний могут отсутствовать. Соответствующие аскаплоты помечаются предусмотренной нотацией, характеризующей отсутствие наблюдений, а также частичную или полную облачность в анализируемые периоды времени.

Рассмотрим далее изображение страницы аскаплота как монохромное изображение, описываемое в формате $I(x, y)$, где x, y – координаты пикселя. При этом изображения аскаплота и

маски могут быть рассмотрены как множество пикселей с координатами x, y и значениями (R, G, B) для цветного изображения.

Тогда пусть изображение маски представляет собой цветное отмасштабированное изображение $I_{mask}(x, y)$ с красным, зеленым и синим каналами (RGB). Маска имеет размер M x N, где M и N – количество столбцов и строк изображения соответственно.

С учетом сказанного множество пикселей, относящихся ячейкам табличного представления аскаплота, может быть представлено как множество C вида:

$$c_j^i = \{I(x + \Delta x, y + \Delta y) | I_{mask}(x, y) = (50j, 5i, 0)\},$$

$$c_j^i \in C, C \to H, c_j^i \neq \emptyset, \quad (6)$$

где $\Delta x, \Delta y$ – смещение маски относительно начала координат изображения аскаплота; c_j^i – подмножество пикселей монохромного изображения страницы аскаплота, которые соответствуют ячейке в i-м столбце и j-й строке; C – множество пикселей, относящихся ячейкам табличного представления аскаплота; $I(x + \Delta x, y + \Delta y)$ – множество пикселей монохромного изображения страницы аскаплота с учетом смещения маски; $I_{mask}(x, y)$ – изображение маски; x, y – координаты, изменяемые от 0 до M и N соответственно.

Смещение $\Delta x, \Delta y$ определяется на предварительном этапе оцифровки. Также на этом этапе определяется и масштабирование маски так, чтобы ее края соответствовали краям изображения аскаплота. Поэтому значения M и N не будут постоянными и будут изменяться в зависимости от аскаплота.

При этом условие проверки принадлежности пикселя маски к конкретному цвету ячейки таблицы может быть сформулировано как:

$$R_j^i = (50j, 5i, 0), \quad (7)$$

где R_j^i – уникальный для каждой ячейки цвет, который можно идентифицировать по индексу строки j = 0,…,4 и столбца i = 0,…,47.

Еще одной важной особенностью рассматриваемого подхода к представлению результатов наблюдений является как совместное применение нескольких аскаплотов, характеризующее данные за несколько суток, так и фрагментарное использование тех же данных, относящееся к исследованию соответствующих характеристик полярного сияния в течение одного или нескольких получасовых

интервалов. Комбинирование перечисленных подходов позволяет достаточно гибко формировать наборы обрабатываемых данных за различные периоды времени для последующего анализа.

4. Характеристика решения. Для повышения эффективности обработки и анализа аскаплотов авторами было предложено программное средство, автоматизирующее указанные информационные процессы. Предусловием автоматизированной обработки является размещение pdf-документов или png-изображения с анализируемыми аскаплотами в одной директории, именованной последовательностью латинских символов.

При этом при первом запуске разработанного приложения пользователю необходимо напрямую указать целевую директорию с соответствующими аскаплотами. Кроме того, для упрощения понимания пользователем рекомендуется соблюдать предусмотренную разработчиками схему именования файлов с аскаплотами, предполагающую следующий формат:

$$A = YYYY\text{-}MM\_DD.[pdf \mid png], \quad (8)$$

где A – имя документа, YYYY – четырехсимвольное обозначение года, MM – двухсимвольное обозначение порядкового номера месяца, DD – двухсимвольное обозначение порядкового номера дня в соответствующем месяце.

На основании параметров расположения указанной пользователем директории с аскаплотами приложение начинает их последовательную загрузку и обработку. При этом для повышения реактивности разработанного программного обеспечения выполняется распараллеливание выполнения соответствующих вычислительных процессов в соответствии с вычислительными мощностями пользовательского компьютера.

При работе с предложенным приложением пользователь может дополнительно верифицировать промежуточные результаты, получаемые на различных этапах обработки рассматриваемых данных. Так, к примеру, одна из таких пользовательских проверок доступна после загрузки и первичной обработки файлов аскаплотов, что позволяет еще на ранних этапах при необходимости скорректировать действия используемого программного обеспечения.

Предложенное приложение по обработке аскаплотов также выполняет предварительную проверку значений анализируемого временного интервала, автоматически удаляя некорректные даты и исправляя имеющиеся последовательности значений. Это также

позволяет избежать некорректной интерпретации аскаплотов, что, в свою очередь, может негативно сказаться на результатах анализа соответствующих оптических наблюдений полярных сияний. При этом важно отметить, что некорректные даты появляются в результате ошибок работы оптического распознавания текста, а также ошибок записи строкового представления даты в исходных данных.

Результатом работы приложения является один или несколько (в зависимости от заданных конечным пользователем настроек) документов в csv-подобном формате. Такое представление результатов обработки позволяет продолжить анализ соответствующими инструментально-программными средствами, либо посредством специализированных программных библиотек в составе сторонних информационных систем. Кроме того, для первичного просмотра и анализа полученных в ходе работы приложения результатов конечному пользователю достаточно использовать стандартные офисные пакеты для работы с электронными таблицами, что также, в свою очередь, призвано повысить доступность соответствующих данных.

5. Архитектура решения. Предлагаемое решение на программном уровне предполагает декомпозицию на четыре взаимосвязанных модуля:

1) Оцифровка датасета.
2) Первичное восстановление дат.
3) Проверка последовательности дат.
4) Совмещение первичного датасета и восстановленного списка дат.

Модуль оцифровки датасета предусматривает обработку (в том числе параллельную) N страниц, где N – натуральное число, максимальное количество ядер процессора на компьютере или заданное пользователем значение. В ходе выполнения модуля из исходных данных (файлов для оцифровки) создается первичный датасет, используемый последующими программными модулями.

Первичный датасет представляет собой таблицу со следующими столбцами:

– date – оцифрованное изображение даты в строковом формате “DD.MM.YYYY”. Если произошла ошибка оцифровки даты, то значение применяется равным «NaT».

– time – время съемки.

– add_data – путь к изображению, на котором присутствует данный аскаплот.

– North, Zenith, South, medium, strong – оцифрованные ячейки аскаплота.

Для группировки аскаплотов по датам из первичного датасета удаляется столбец с временем и оставляется каждая 48-я строка. Приложение проверяет и исправляет пропуски, которые стоят внутри линейной последовательности дат с шагом в 1 день.

Модуль первичного восстановления дат предполагает создание списка дат из первичного датасета, а также восстановление очевидных последовательностей. При этом на начальном этапе предусмотрена замена NaT значений вручную.

Модуль проверки последовательности дат отвечает за вывод на экран дат, нарушающих последовательность, а также их замену в случае необходимости. Например, в данной последовательности дат (19, 20, 21, NaT, 23, 24, 25, 26), «NaT» является очевидным пропуском и его можно заменить на 22, чтобы восстановить последовательность. В целом пользователь проверяет полученный набор дат и исправляет значения «NaT».

Модуль совмещения первичного датасета и восстановленного списка дат предусматривает сохранение финального датасета в формате csv.

В общем виде соответствующая диаграмма компонентов может быть представлена так, как показано на рисунке 2.

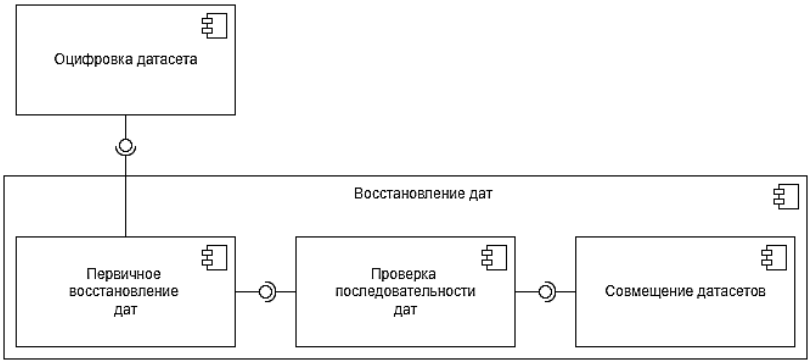


Рис. 2. Диаграмма компонентов разработанной системы

Высокая сложность пользовательского сценария связана с нетривиальностью задачи и необходимостью вносить пользовательские правки на разных стадиях работы приложения. На тестовом датасете аскаплотов пользователю вручную необходимо

проверить и/или исправить 5 значений, что составляет 0.004% от количества строк конечного датасета.

6. Алгоритм решения. В общем виде алгоритм предложенного решения декомпозируется на несколько последовательных процессов (рисунок 3). Каждый из процессов представляет собой этап оцифровки аскаплотов и верификации полученных результатов. По завершению каждого из перечисленных процессов конечному пользователю доступны для проверки и корректировки (в случае необходимости) соответствующие промежуточные результаты. При этом по выбору пользователя соответствующие действия могут быть проигнорированы.

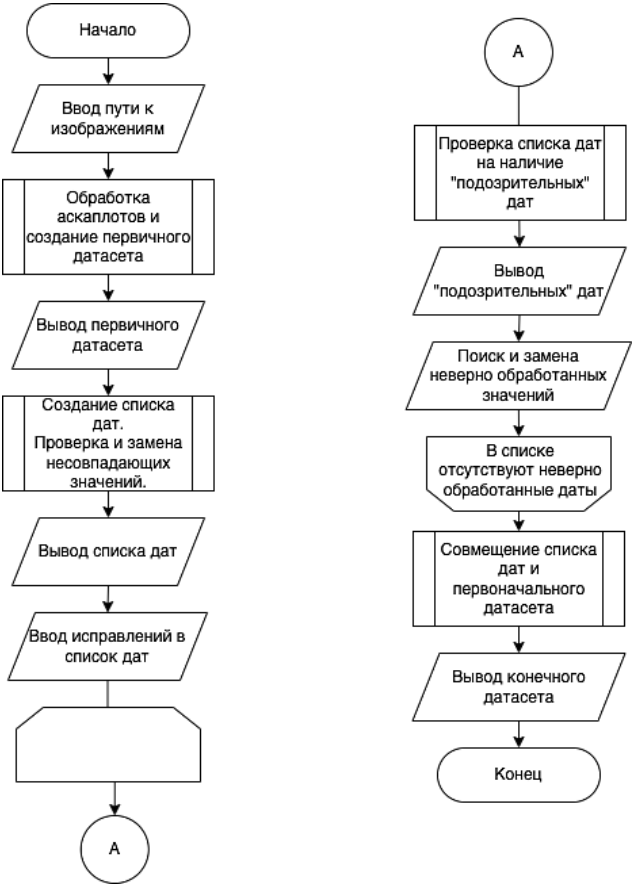


Рис. 3. Обобщенная схема алгоритма, реализующего предлагаемый подход

На первом этапе выполнения алгоритма осуществляется загрузка и обработка исходных данных на основании параметров физического расположения директории, содержащей оцифровываемые аскаплоты (рисунок 4). Пользователь вручную или посредством соответствующих интерфейсных элементов управления выбирает искомую директорию, абсолютный путь к которой фиксируется в соответствующей переменной для последующего применения по мере выполнения обозначенного алгоритма.

На втором этапе выполняется непосредственно обработка аскаплотов и формирование первичного датасета с полученными при этом промежуточными результатами выполнения алгоритма. Схема соответствующего алгоритма представлена на рисунке 4.

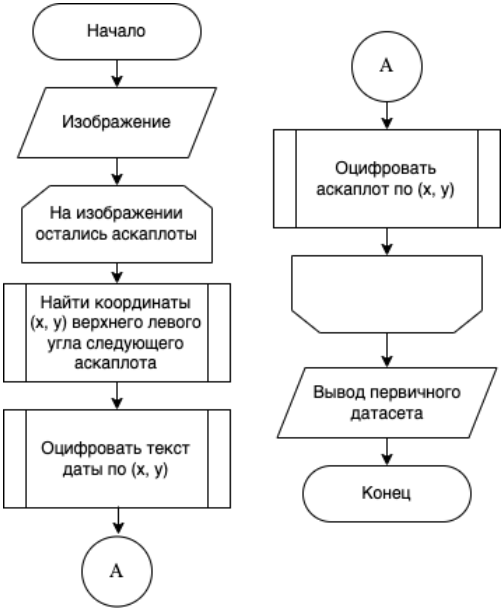


Рис. 4. Схема алгоритма формирования первичного датасета

Первичный датасет представляет собой структурированную таблицу, содержащую временную метку и соответствующее ей значение, извлеченное из аскаплота. Сформированный датасет выводится для обзора конечным пользователем. При этом основной акцент делается на проверку сформированных в ходе выполнения алгоритма временных меток. Алгоритм предусматривает автоматический поиск и замену несовпадающих значений. При этом

непосредственно конечный пользователь может также внести соответствующие исправления вручную.

Далее пользователю демонстрируются даты, разделенные не характерным для датасета промежутком времени. Так, к примеру, если за данными, зарегистрированными 26 февраля, следуют результаты оптических наблюдений за 1 марта, соответствующая информация становится доступна пользователю. В интерфейсе приложения соответствующие потенциально некорректные временные метки помечаются дополнительно отличными от основного цветом и начертанием используемого шрифта. Кроме того, формируется соответствующее сообщение по типу записи в журнале ошибок [26].

Дальнейшие действия полностью определяются конечным пользователем. В случае, если нарушение периодичности в последовательности временных меток соответствует действительности (к примеру, соответствующих аскаплотов нет в распоряжении пользователя или по каким-то иным причинам), то выявленное несоответствие не считается ошибкой и игнорируется. В противном случае пользователь исправляет выделенную последовательность и запускает процедуру выполнения соответствующих этапов алгоритма заново.

По завершении проверки и корректировки выводимых в ходе выполнения алгоритма последовательностей дат соответствующая процедура запускается повторно. Пользователю снова демонстрируются соответствующие выделенные из заданных для обработки аскаплотов списки дат, содержащие, возможно, только те несоответствия в последовательности, которые были одобрены пользователем и помечены как корректные.

На следующем этапе алгоритм предусматривает сопоставление первичного датасета, сформированного на начальных этапах, и восстановленного списка дат из предшествующего этапа выполнения алгоритма. Полученный в результате выполнения обозначенного этапа окончательный датасет передается конечному пользователю в csv-формате для последующей обработки, анализа и визуализации, в том числе сторонними программными системами и библиотеками.

Важный момент касательно предусмотренного в алгоритме предопределенного подпроцесса оцифровки аскаплотов сопряжен с применением специализированной маски поиска значимых данных в файлах аскаплотов, имеющих характерную и описанную соответствующими профильными спецификациями структуру. Анализ спецификаций, а также результаты проведенных вычислительных экспериментов показали, что данные для оцифровки в файле аскаплота

представлены в сегменте шириной 420 и длиной 75 пикселей соответственно (само изображение маски составляет 1668x183 пикселей). При этом весь документ и каждая его составляющая для оцифровки сопровождается координатной системой с параметрами x и y соответственно.

Представляется целесообразным отметить ряд важных для оцифровки параметров. На начальном этапе задается так называемый параметр x_0 – смещение (отступ поиска) по оси x, значение которого было получено экспериментально и составило 1900 пикселей. По сути, данный параметр ограничивает виртуальную линию поиска аскаплотов в анализируемом документе, на протяжении которой не встречается название страницы, номер страницы и время замера.

Кроме того, на предварительном этапе оцифровки алгоритм предусматривает поиск границ рассматриваемой области обрабатываемого документа / изображения на предмет определения верхнего левого угла аскаплота. Проведенные вычислительные эксперименты показали, что для анализируемых данных (для изображения размером 2480x3507 пикселей) соответствующие параметры составляют (x, y) = (350, 200) и (x, y) = (350 <= x <= 700, 200 <= y <= 3307) соответственно.

После обнаружения искомого верхнего левого угла начинается непосредственно оцифровка аскаплота. Создается двумерный массив для обработки каждой ячейки и считывания ее значения. Попиксельно анализируется содержимое ячейки аскаплота и в соответствии с заданной на этапе предварительной настройки алгоритма маски определяется положение соответствующего пикселя. Для этого последовательно анализируется каждый из трех цветовых каналов, и в совокупности эти значения позволяют определить цвет и его интенсивность в соответствующей ячейке.

При этом во избежание коллизий отдельно рассматриваются несколько случаев расположения анализируемого пикселя относительно ячейки аскаплота в целом:

– пиксель граничит с верхней стороной ячейки;
– пиксель граничит с правой стороной ячейки;
– пиксель граничит с левой стороной ячейки;
– пиксель граничит с нижней стороной ячейки;
– пиксель не граничит ни с одной из сторон ячейки.

Принципы классификации каждого символа могут быть сформулированы следующим образом:

– отсутствие символа можно определить по отсутствию черных пикселей в ячейке;

– в случае, если в ячейке все символы закрашены черным, то символ является черным квадратом;

– в случае, если черные пиксели есть только у верхнего и нижнего края ячейки, но их нет у левого и нижнего края, то символом является вертикальная линия;

– в случае, если черные пиксели есть только у левого и правого края ячейки, но их нет у верхнего и нижнего края, то символом является горизонтальная линия;

– в иных случаях символом ячейки считается закрашенная окружность.

В общем виде схема алгоритма оцифровки аскаплота и классификации типа ячеек представлена на рисунке 5.

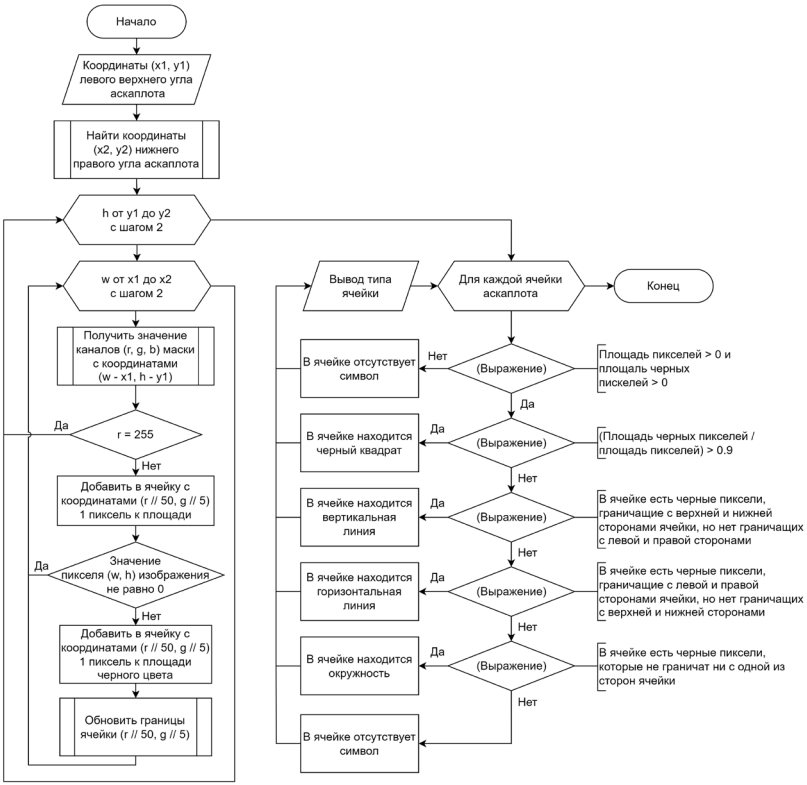


Рис. 5. Схема алгоритма оцифровки аскаплота

Важным этапом оцифровки аскаплотов является маскирование. При этом маска представляет собой цветное изображение в формате png, на котором изображено общее положение ячеек аскаплотов. При этом каждый цветовой канал соответствует определенному компоненту данных о соответствующей маске.

Так, красный канал соответствует индексу строки ячейки аскаплота, который равен результату целочисленного деления значения красного канала на 50. Зеленый канал сопоставлен с индексом столбца ячейки аскаплота, который определяется как результат целочисленного деления значения зеленого канала на 5. И, наконец, синий канал соответствует разделителю столбцов в сетке ячеек аскаплота. Кроме того, отдельно помечается белый пиксель, характеризующий пропуск данной строки в цикле изображения маски.

В общем виде процедуру построения маски можно описать следующим образом. На начальном этапе проводится выборка случайных аскаплотов из общего набора данных. Далее выбранные аскплоты масштабируются таким образом, чтобы их внешние границы совпадали по горизонтали. При этом изображения аскаплотов переводятся в бинарный формат и накладываются друг на друга с применением логического оператора «или». Полученное в результате изображение представляет собой генерализированный образ аскаплотов. Далее найденные общие границы увеличиваются и все изображение переводится обратно в цветной формат. Каждая ячейка маски закрашивается в цвет, соответствующий ее координатам, а остальная часть маски закрашивается в соответствии с вышеописанными правилами.

В общем виде пример отладочного изображения при оцифровке аскаплотов представлен на рисунке 6. Здесь может быть выделено несколько важных для оцифровки составляющих:

1. Непосредственно описание оцифрованной ячейки, состоящее из двух символов. Первый из них характеризует результат наблюдения в соответствии с принятой классификацией (например, 0 – отсутствие сияния, 1 – наблюдение сияния, 2 – полная облачность, 3 – отключенная камера, 4 – частичная облачность и т.д.). Второй символ содержит булево значение (T / F), показывающее, есть ли в анализируемой ячейке данные.

2. Обрабатываемые пиксели закрашены в цвет, который означает соседство с краем рассматриваемой области ячейки. Например, желтый цвет означает соседство пикселя с нижней границей ячейки.

3. Область наложения маски на начальное изображение аскаплота.
4. Отображение времени суток для данного столбца.
5. Обнаруженные границы аскаплота.
6. Области, соответствующие левой верхней и правой нижней границ аскаплота.

Получаемое отладочное изображение тождественно описанной маске.

Здесь представляется целесообразным отметить, что в подавляющем большинстве случаев исходные аскаплоты представляют собой pdf-документы с результатами многодневных оптических наблюдений полярных сияний. Поскольку непосредственно алгоритм оцифровки ориентирован на работу с единичными аскаплотами, предварительная обработка исходных данных может включать в себя извлечение таблиц аскаплотов из страниц наблюдений.

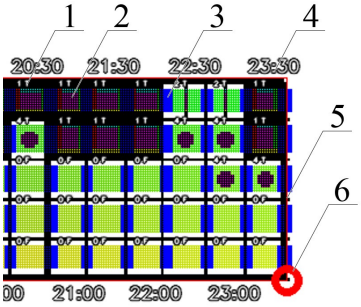


Рис. 6. Пример отладочного изображения для оцифровки аскаплота

При этом непосредственно дата наблюдений, зафиксирована в метаданных соответствующего аскаплота в тексте исходного pdf-документа. Для повышения эффективности работы и во избежание задания этих данных вручную самим пользователем, дата также оцифровывается и ставится в соответствие полученному в ходе работы алгоритма результату.

В ряде случаев возникает необходимость масштабирования маски под оцифровываемые данные. В этом случае алгоритм предусматривает дополнительную обработку и подготовку маски, интерполируя данные на основе метода ближайшего соседа.

7. Стек технологий. Предложенный алгоритм реализован с помощью языка программирования Python [28, 29, 30], при этом

конечный результат представляет собой настольное однопользовательское приложение. Сохранение пользовательских данных между сеансами работы пользователя с приложением не предусмотрено ни задачей, ни соответствующим алгоритмом. По этой причине в архитектуре приложения не предусмотрена база данных, а источником исходных данных выступает пользовательская файловая система.

При выборе языка программирования для реализации предложенных подхода и алгоритма в качестве возможных альтернатив были рассмотрены следующие языки: C++ [31], C# [32], Python [33] и JavaScript [33, 34]. Выбранные альтернативы сравнивались по следующим критериям:

– кроссплатформенность.
– скорость выполнения идентичного кода.
– наличие необходимых для реализации приложения библиотек.
– сложность разработки.
– читаемость кода.

Сравнение выделенных альтернатив было выполнено с использованием метода анализа иерархий [35, 36] по указанным критериям. Результаты проведенной сравнительной оценки показали, что наилучшие характеристики по всем заявленным критериям свойственны языку программирования Python.

8. Описание исследовательского прототипа программного обеспечения. В общем виде разработанное на основе предложенных подхода и алгоритма оцифровки аскаплотов программное обеспечение представляет собой настольное приложение, которое должно быть развернуто и применено с использованием вычислительных мощностей компьютера конечного пользователя.

При этом в зависимости от выбора пользователя предусмотрены два варианта развертывания. Первый из них связан с формированием исполняемого exe-файла, который тем или иным способом передается пользователю. В свою очередь, пользователь, получив указанный исполняемый файл, запускает его на выполнение, получая доступ к искомой функциональности.

Второй вариант сопряжен с передачей пользователю вместе с файлом программы также сопровождающих его интерпретатора Python соответствующей версии с необходимыми для работы приложения библиотеками. Для сохранения зависимостей компонент используется контейнеризация приложения через Docker, что по умолчанию обеспечивает его высокую портируемость. Докеризированная

программа размещается в Docker Hub, откуда любой пользователь может получить к ней доступ.

Следует отметить, что второй из перечисленных вариантов распространения разработанного настольного приложения является предпочтительным для передачи программы конечным пользователям. В первую очередь, это сопряжено с соблюдением требований информационной безопасности. Известно, что исполняемые файлы могут содержать код или выполнять произвольные команды, заложенные злоумышленниками, которые перехватили передаваемый пользователю файл.

Прототип приложения, реализующего предложенный алгоритм, реализован в однооконном формате, предполагающем разделение интерфейсной составляющей на две вертикальные секции. В одной из них (слева) имитируется структура директории с оцифровываемыми аскаплотами: задается непосредственно имя директории в корневом узле иерархии, а соответствующие обрабатываемые файлы представляют собой узлы иерархии в виде имени и расширения каждого файла.

В оставшейся секции пользователь может наблюдать за ходом выполнения процесса оцифровки аскаплотов. Здесь же доступна, в частности, интерактивная составляющая при работе с полученным на этапе предварительной обработки списком дат. Так, к примеру, в случае автоматического обнаружения выделенные некорректные даты могут быть помечены пользователем как корректные.

В интерфейсе приложения предусмотрены и стандартные для оконного приложения элементы управления. В частности, присутствует меню, пункты которого позволяют пользователю открыть диалоговое окно выбора директории для обработки данных, а также аналогичное окно для указания размещения результирующего документа с данными оцифровки. В дальнейшем планируется в элементы управления добавить возможность настройки внешнего вида страницы, вывод на печать и прочие стандартные опции.

При работе с приложением на предварительном этапе пользователь формирует директорию с подлежащими оцифровке аскаплотами. После запуска приложения осуществляется начальная оцифровка исходных данных: определяется тип файла, при необходимости извлекается изображение, которое содержит аскаплоты для обработки. При этом представляется целесообразным отметить, что в существующей версии приложения предполагается централизованное хранение всех оцифровываемых аскаплотов.

В дальнейшем предполагается реализовать поддержку выборочного набора обрабатываемых файлов с аскаплотами.

При запуске последующей процедуры проверки дат из обработанных аскаплотов извлекаются соответствующие параметры. При этом формируемый приложением набор дат автоматически проверяется, а даты, нарушающие последовательность, визуально выделяются в результирующем списке. Пользователь может выбрать действия с выделенными датами, после чего процесс обработки аскаплотов запускается на выполнение с обновленным набором параметров.

Каждый из этапов обработки регистрируется конечным пользователем в режиме реального времени. На завершающем этапе формируется результирующий датасет, который выводится в табличный файл, доступный для последующего просмотра, обработки и анализа сторонними средствами. Для последующих операций и новых итераций оцифровки пользователь указывает целевую директорию и повторяет все перечисленные выше действия.

В перспективе развития предложенного решения планируется расширение приложения на веб-ориентированную архитектуру. Это позволит расширить круг его пользователей, с одной стороны, и снизить требования к их вычислительным мощностям, с другой.

9. Вычислительный эксперимент. Для оценки эффективности предложенного решения был проведен ряд вычислительных экспериментов, в ходе которых осуществлялась оцифровка pdf-документов с аскаплотами. Исходные данные были взяты из документов «PGI Geophysical Data» за период с 2012 по 2020 гг, в котором помимо непосредственно данных оптических наблюдений полярных сияний представлены и другие результаты измерений, в частности наблюдения геомагнитного поля и космических лучей.

Выбранные для оцифровки аскаплоты обрабатывались двумя способами. В первом случае контрольная группа выполняла оцифровку вручную, последовательно перебирая соответствующие документы и формируя строки результирующей таблицы с результатами наблюдений. Во втором случае оцифровка выполнялась одним пользователем с применением разработанного приложения, реализующего предложенные подходы и алгоритмы.

В ходе проведенных вычислительных экспериментов в каждом из обоих случаев были оцифрованы 1 035 аскаплотов за девятилетний период наблюдений. При использовании практикуемого в настоящее время подхода обработки вручную, формирование результирующей таблицы в совокупности заняло 5 100 человеко-минут. При этом

последующий анализ показал, что в ходе оцифровки было допущено более 200 ошибок в 76 аскаплотах (это составляет ~7.34 % от общего числа аскаплотов).

При оцифровке тех же аскаплотов с применением разработанного программного обеспечения скорость обработки одного аскаплота составила в среднем 0,5 с, что в совокупности для всех аскаплотов заняло ~8,6 минуты (технические характеристики вычислительной машины: CPU: Intel Core I7-9700KF, частота на момент замеров составляла 4.57 ГГц). При этом проведенный последующий анализ результатов оцифровки не выявил ошибок обработки результатов оптических наблюдений.

Для ускорения процесса оцифровки с использованием предложенного программного обеспечения было применено распараллеливание на 8 процессов [37], каждый из которых обрабатывал отдельную страницу. Благодаря этому время оцифровки удалось снизить более чем в 5.2 раза (технические характеристики вычислительной машины: CPU: Intel Core I7-9700KF, частота на момент замеров составляла 4.57 ГГц). В результате общее время оцифровки (с учетом вывода отладочных изображений) ~ 1,7 мин.

По данным проведенных вычислительных экспериментов в 1 035 обработанных аскаплотах ошибок оцифровки не обнаружено. По сравнению с ручным методом разработанное программное обеспечение позволило ускорить процесс оцифровки аскаплотов примерно в 3000 раз.

10. Заключение. В настоящее время задача оптического наблюдения полярных сияний успешно решается с применением камер всего неба, которые ведут непрерывную фото- и / или видеорегистрацию небосвода в режиме реального времени. Результаты наблюдений фиксируются в специализированных документах, известных как «аскаплоты». Каждый из аскаплотов в заданной нотации характеризует результаты оптических наблюдений, фиксируя их в пятистрочной таблице, столбцы которой соответствуют последовательным получасовым временным интервалам.

При этом на данный момент времени все аскаплоты обрабатываются вручную. Такой способ регистрации наблюдений используется на всех этапах. На первом из них записываемые камерами аскафильмы переводятся в аскаплоты и далее, на втором этапе, так же вручную строятся таблицы с результатами наблюдений. Соответствующие процессы сопряжены с высокими человеческими и временными затратами, а также существенной долей ошибок, возникающих в ходе обработки результатов наблюдений.

В этой связи в настоящей работе предложен подход, обеспечивающий автоматизацию распознавания и оцифровки данных оптических наблюдений полярных сияний. В рамках обозначенного подхода предложен вариант формализации аскаплота, используемый в дальнейшем для последовательного попиксельного считывания данных регистрации и формирования итоговой таблицы наблюдений в текстовом csv-подобном формате, доступном для последующей обработки, анализа и визуализации сторонними средствами.

Отличительной особенностью предложенного подхода является применение специального вспомогательного изображения для оцифровки, используемого в качестве маски для обработки данных. В общем виде маска представляет собой цветное изображение, в котором заданы общие положения ячеек аскаплотов. Каждый цветовой канал при этом соответствует определенному компоненту данных.

В ходе выполнения исследований был предложен ряд алгоритмов, обеспечивающих процесс оцифровки аскаплотов, с одной стороны, и поддерживающих взаимодействие соответствующего приложения с пользователем, с другой. На каждом этапе выполнения алгоритма пользователю доступны действия, позволяющие скорректировать последующую обработку.

На основе предложенных подхода и алгоритмов было разработано соответствующее настольное программное обеспечение. Проведенные вычислительные эксперименты показали, что применение предложенного программного обеспечения для обработки аскаплотов позволит избежать ошибок при оцифровке. Кроме того, время, затрачиваемое на обработку каждого аскаплота, существенно сокращается и для одного аскаплота составляет в среднем 0,1 с (с учетом распараллеливания выполнения).

В контексте перспективы развития предложенного подхода планируется его перевод на веб-ориентированную платформу с поддержкой многопользовательского режима работы. Это позволит, с одной стороны, расширить круг потенциальных пользователей приложения, и снизит нагрузку на клиентские вычислительные мощности, с другой.

Элементы предложенных решений в настоящее время используются авторами статьи в том числе и для решения задач локальной диагностики наличия полярных сияний [38].

Литература

1. Kozyreva O.V., Pilipenko V.A., Bland E.C., Baddeley L.J., Zakharov V.I. Periodic modulation of the upper ionosphere by ULF waves as observed simultaneously by

SuperDARN radars and GPS/TEC technique // Journal of Geophysical Research: Space Physics. 2020. vol. 125(7). no. e2020JA028032. DOI: 10.1029/2020JA028032.
2. Klimov P., Kozelov B., Roldugin A., Sigaeva K. Joint Recording of Pulsating Auroras on Board the Lomonosov Satellite and by All-Sky Cameras on the Kola Peninsula // Bulletin of the Russian Academy of Sciences: Physics. 2022. vol. 86. no. 3. pp. 300–304. DOI: 10.3103/S106287382203011X.
3. Yang X., Shang Zh., Hu K., Hu Y., Ma B., Wang Y., Wang W. Cloud cover and aurora contamination at dome A in 2017 from KLCAM // Monthly Notices of the Royal Astronomical Society. 2021. vol. 501. no. 3. pp. 3614–3620. DOI: 10.1093/mnras/staa3824.
4. Ягодкина О.И., Воробьев В.Г., Шекунова Е.С. Наблюдения полярных сияний над Кольским полуостровом // Труды Кольского научного центра РАН. 2019. Т. 10. № 8(5). С. 43–55.
5. Nakamura J., Kitamura T., Fukushima S. Auroral ASCAPLOT at Syowa Station in 1959 and 1960 // Antarctic record. 1962. no. 16. pp. 1339–1360.
6. Feldstein Y.I. The discovery and the first studies of the auroral oval: A review // Geomagnetism and Aeronomy. 2016. vol. 56. pp. 129–142. DOI: 10.1134/S0016793216020043.
7. Feldstein Y.I., Vorobjev V.G., Zverev V.L. Planetary features of aurorae: Results of the IGY (a review) // Geomagnetism and Aeronomy. 2010. vol. 50. pp. 413–435. DOI: 10.1134/S0016793210040018.
8. Hashmi K.A., Liwicki M., Stricker D., Afzal M.A., Afzal M.A., Afzal M.Z. Current Status and Performance Analysis of Table Recognition in Document Images With Deep Neural Networks // IEEE Access. 2021. vol. 9. pp. 87663–87685. DOI: 10.1109/ACCESS.2021.3087865.
9. Namysł M., Esser A.M., Behnke S., Kohler J. Flexible Hybrid Table Recognition and Semantic Interpretation System // SN Computer Science. 2023. vol. 4. no. 246. DOI: 10.1007/s42979-022-01659-z.
10. Lee E., Park J., Koo H.I., Cho N.I. Deep-learning and graph-based approach to table structure recognition // Multimedia Tools and Applications. 2022. vol. 81. no. 4. pp. 5827–5848. DOI: 10.1007/s11042-021-11819-7.
11. Li X.H., Yin F., Dai H.S., Liu C.L. Table Structure Recognition and Form Parsing by End-to-End Object Detection and Relation Parsing // Pattern Recognition. 2022. vol. 132. no. 108946. DOI: 10.1016/j.patcog.2022.108946.
12. Sage C., Aussem A., Elghazel H., Eglin V., Espinas J. Recurrent Neural Network Approach for Table Field Extraction in Business Documents // International Conference on Document Analysis and Recognition (ICDAR). 2019. pp. 1308–1313. DOI: 10.1109/ICDAR.2019.00211.
13. Khan S.A., Khalid S.M.D., Shahzad M.A., Shafait F. Table Structure Extraction with Bi-Directional Gated Recurrent Unit Networks // International Conference on Document Analysis and Recognition (ICDAR). 2019. pp. 1366–1371. DOI: 10.1109/ICDAR.2019.00220.
14. Hochreiter S., Schmidhuber J. Long Short-Term Memory // Neural computation. 1997. vol. 9. no. 8. pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
15. Paliwal S.S., Vishwanath D., Rahul R., Sharma M., Vig L. TableNet: Deep Learning Model for End-to-end Table Detection and Tabular Data Extraction from Scanned Document Images // International Conference on Document Analysis and Recognition (ICDAR). 2019. pp. 128–133. DOI: 10.1109/ICDAR.2019.00029.
16. Tensmeyer C., Morariu V.I., Price B., Cohen S., Martinez T. Deep Splitting and Merging for Table Structure Decomposition // International Conference on Document Analysis and Recognition (ICDAR). 2019. pp. 114–121. DOI: 10.1109/ICDAR.2019.00027.

17. Siddiqui S.A., Fateh I.A., Rizvi S.T.R., Dengel A., Ahmed S. DeepTabStR: Deep Learning based Table Structure Recognition // International Conference on Document Analysis and Recognition (ICDAR). 2019. pp. 1403–1409. DOI: 10.1109/ICDAR.2019.00226.
18. Couasnon B., Lemaitre A. Recognition of Tables and Forms // Handbook of Document Image Processing and Recognition. Chapter Recognition of Tables and Forms. 2014. pp. 647–677. DOI: 10.1007/978-0-85729-859-1_20.
19. Zucker A., Belkada Y., Vu H., Nguyen V.N. ClusTi: Clustering Method for Table Structure Recognition in Scanned Images // Mobile Networks and Applications. 2021. vol. 26. no. 4. pp. 1765–1776. DOI: 10.1007/s11036-021-01759-9.
20. Nguyen Q.D., Le D.A., Phan N.M., Zelinka I. OCR error correction using correction patterns and self-organizing migrating algorithm // Pattern Analysis and Applications. 2021. vol. 24. pp. 701–721. DOI: 10.1007/s10044-020-00936-y.
21. Patel C., Patel A., Patel D. Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study // International Journal of Computer Applications. 2014. vol. 55(10). pp. 50–56. DOI: 10.5120/8794-2784.
22. Vorobjev V. PGI Geophysical data. 2015. October, November, December. Murmansk, Apatity: PGI KSC RAS, 2016.
23. Vorobjev V.G., Roldugin V.C., Yagodkina O.I. Large Amplitude Undulations of Evening Site Diffuse Aurorae. Optical Characteristics and Conditions of Generation // Geomagnetism and Aeronomy. 2015. vol. 55. pp. 45–50. DOI: 10.1134/S0016793215010132.
24. Vorobjev V.G., Yagodkina O.I., Antonova E.E. Ion Pressure in Different Regions of the Dayside Auroral Precipitation // Geomagnetism and Aeronomy. 2020. no. 60. pp. 727–736. DOI: 10.1134/S0016793220060146.
25. Popov L.N., Krakovetskiy Yu.K., Gokhberg M.B., Pilipenko V.A. Terrogenic effects in the ionosphere: a review // Physics of the Earth and Planetary Interiors. 1989. vol. 57. no. 1-2. pp. 115–128.
26. Zhang T., Qiu H., Castellano G., Rifai M., Chen C.S., Pianese F. System Log Parsing: A Survey // IEEE Transactions on Knowledge and Data Engineering. 2023. pp. 8596–8614. DOI: 10.1109/TKDE.2022.3222417.
27. Patil O., Chavan U. Rule Based Expert System for Error Log Analysis // International Journal of Innovative Technology and Exploring Engineering. 2020. vol. 9. no. 10. pp. 188–192. DOI: 10.35940/ijitee.J7466.0891020.
28. Peta S. Python- An Appetite for the Software Industry // International Journal of Programming Languages and Applications (IJPLA). 2022. vol. 12. DOI: 10.5121/ijpla.2022.12401.
29. Singh B.P. Python and Its Future Scope // International Journal of Advanced Research in Science, Communication and Technology. 2022. pp. 400–403. DOI: 10.48175/IJARSCT-4829.
30. Dr U., Patkar U. Python for web development // International Journal of Computer Science and Mobile Computing. 2022. vol. 11. no. 4. pp. 36–48. DOI: 10.47760/ijcsmc.2022.v11i04.006.
31. Rong W., Xu T., Sun, Z., Sun, Z., Ouyang, Y., Xiong, Z. An Object Tuple Model for Understanding Pointer and Array in C Language // IEEE Transactions on Education. 2023. pp. 1–12. DOI: 10.1109/TE.2023.3236027.
32. Peta S. C Programming Language–Still Ruling the World // Global Journal of Computer Science and Technology. 2022. vol. 22(1). pp. 9–13.
33. Park H., Kim S., Bae B. Dynamic code compression for JavaScript engine // Software: Practice and Experience. 2023. vol. 53. no. 5. pp. 1196–1217. DOI: 10.1002/spe.3186.

34. Wang Z., Bu D., Wang N., Yu S., Gou S., Sun A. An empirical study on bugs in JavaScript engines // Information and Software Technology. 2023. vol. 155. no. 107105. DOI: 10.1016/j.infsof.2022.107105.
35. Romanchuk V.M. The Problem of Adequacy of the Analytic Hierarchy Process // Modelling and Data Analysis. 2022. vol. 10. no. 4. pp. 79–87. DOI: 10.17759/mda.2020100407.
36. Polat T.K. An Application of Analytic Hierarchy Process and Fuzzy Analytic Hierarchy Process to the Case Type Selection Problem // Academic Perspective Procedia. 2018. vol. 1. no. 1. pp. 1179–1188. DOI: 10.33793/acperpro.01.01.188.
37. Vorobev A.V., Pilipenko V.A., Enikeev T.A., Vorobeva G.R. Geoinformation system for analyzing the dynamics of extreme geomagnetic disturbances from observations of ground stations // Computer Optics. 2020. vol. 44(5). pp. 782–790. DOI: 10.18287/2412-6179-CO-707.
38. Vorobev A.V., Soloviev A.A., Pilipenko V.A., Vorobeva G.R., Gainetdinova A.A., Lapin A.N., Belakhovsky V.B., Roldugin A.V. Local diagnostics of aurora presence based on intelligent analysis of geomagnetic data // Solar-Terrestrial Physics. 2023. vol. 9(2). pp. 22–30. DOI: 10.12737/stp-92202303.

Воробьев Андрей Владимирович — д-р техн. наук, профессор, кафедра геоинформационных систем факультета информатики и робототехники, Уфимский университет науки и технологий; старший научный сотрудник, Геофизический центр РАН. Область научных интересов: геоинформационные технологии, цифровая обработка сигналов. Число научных публикаций — 164. geomagnet@list.ru; улица Карла Маркса, 12, 450007, Уфа, Россия; р.т.: +7(917)345-2299.

Лапин Александр Николаевич — студент, кафедра геоинформационных систем факультета информатики и робототехники, Уфимский университет науки и технологий. Область научных интересов: математическое и компьютерное моделирование, цифровые двойники, машинное обучение. Число научных публикаций — 13. meccos160@yandex.ru; улица Карла Маркса, 12, 450008, Уфа, Россия; р.т.: +7(917)439-6040.

Воробьева Гульнара Равилевна — д-р техн. наук, профессор, кафедра вычислительной математики и кибернетики факультета информатики и робототехники, Уфимский университет науки и технологий. Область научных интересов: геоинформационные и веб-технологии, системы хранения и обработки информации. Число научных публикаций — 153. gulnara.vorobeva@gmail.com; улица Карла Маркса, 12, 450008, Уфа, Россия; р.т.: +7(917)417-4111.

Поддержка исследований. Исследование выполнено при финансовой поддержке РНФ, проект № 21-77-30010.

DOI 10.15622/ia.22.5.8

A. VOROBEV, A. LAPIN, G. VOROBEVA

SOFTWARE FOR AUTOMATED RECOGNITION AND DIGITIZATION OF ARCHIVE DATA OF AURORA OPTICAL OBSERVATIONS

Vorobev A., Lapin A., Vorobeva G. **Software for Automated Recognition and Digitization of Archive Data of Aurora Optical Observations.**

Abstract. One of the main tools for recording auroras is the optical observation of the sky in automatic mode using all-sky cameras. The results of observations are recorded in special mnemonic tables, ascaplots. Ascaplots provide daily information on the presence or absence of cloud cover and auroras in various parts of the sky and are traditionally used to study the daily distribution of auroras in a given spatial region, as well as to calculate the probability of their observation in other regions in accordance with the level of geomagnetic activity. At the same time, the processing of ascaplots is currently carried out manually, which is associated with significant time costs and a high proportion of errors due to the human factor. To increase the efficiency of ascaplot processing, we propose an approach that automates the recognition and digitization of data from optical observations of auroras. A formalization of the ascaplot structure is proposed, which is used to process the ascaplot image, extract the corresponding observation results, and form the resulting data set. The approach involves the use of machine vision algorithms and the use of a specialized mask - a debug image for digitization, which is a color image in which the general position of the ascaplot cells is specified. The proposed approach and the corresponding algorithms are implemented in the form of software that provides recognition and digitization of archival data from optical observations of auroras. The solution is a single-user desktop software that allows the user to convert ascaplot images into tables in batch mode, available for further processing and analysis. The results of the computational experiments have shown that the use of the proposed software will make it possible to avoid errors in the digitization of ascaplots, on the one hand, and significantly increase the speed of the corresponding computational operations, on the other. Taken together, this will improve the efficiency of processing ascaplots and conducting research in the relevant area.

Keywords: data processing, digitization of observational data, ascaplots, software.

References

1. Kozyreva O.V., Pilipenko V.A., Bland E.C., Baddeley L.J., Zakharov V.I. Periodic modulation of the upper ionosphere by ULF waves as observed simultaneously by SuperDARN radars and GPS/TEC technique. Journal of Geophysical Research: Space Physics. 2020. vol. 125(7). no. e2020JA028032. DOI: 10.1029/2020JA028032.
2. Klimov P., Kozelov B., Roldugin A., Sigaeva K. Joint Recording of Pulsating Auroras on Board the Lomonosov Satellite and by All-Sky Cameras on the Kola Peninsula. Bulletin of the Russian Academy of Sciences: Physics. 2022. vol. 86. no. 3. pp. 300–304. DOI: 10.3103/S106287382203011X.
3. Yang X., Shang Zh., Hu K., Hu Y., Ma B., Wang Y., Wang W. Cloud cover and aurora contamination at dome A in 2017 from KLCAM. Monthly Notices of the Royal Astronomical Society. 2021. vol. 501. no. 3. pp. 3614–3620. DOI: 10.1093/mnras/staa3824.
4. Yagodkina O.I., Vorobyov V.G., Shekunova E.S. [Observations of auroras over the Kola Peninsula]. Proceedings of the Kola Scientific Center of the Russian Academy of

Sciences – Trudy Kol'skogo nauchnogo tsentra RAN. 2019. vol. 10. no. 8(5). pp. 43–55. (in Russ.).
5. Nakamura J., Kitamura T., Fukushima S. Auroral ASCAPLOT at Syowa Station in 1959 and 1960 // Antarctic record. 1962. no. 16. pp. 1339–1360.
6. Feldstein Y.I. The discovery and the first studies of the auroral oval: A review. Geomagnetism and Aeronomy. 2016. vol. 56. pp. 129–142. DOI: 10.1134/S0016793216020043.
7. Feldstein Y.I., Vorobjev V.G., Zverev V.L. Planetary features of aurorae: Results of the IGY (a review). Geomagnetism and Aeronomy. 2010. vol. 50. pp. 413–435. DOI: 10.1134/S0016793210040018.
8. Hashmi K.A., Liwicki M., Stricker D., Afzal M.A., Afzal M.A., Afzal M.Z. Current Status and Performance Analysis of Table Recognition in Document Images With Deep Neural Networks. IEEE Access. 2021. vol. 9. pp. 87663–87685. DOI: 10.1109/ACCESS.2021.3087865.
9. Namysł M., Esser A.M., Behnke S., Kohler J. Flexible Hybrid Table Recognition and Semantic Interpretation System. SN Computer Science. 2023. vol. 4. no. 246. DOI: 10.1007/s42979-022-01659-z.
10. Lee E., Park J., Koo H.I., Cho N.I. Deep-learning and graph-based approach to table structure recognition. Multimedia Tools and Applications. 2022. vol. 81. no. 4. pp. 5827–5848. DOI: 10.1007/s11042-021-11819-7.
11. Li X.H., Yin F., Dai H.S., Liu C.L. Table Structure Recognition and Form Parsing by End-to-End Object Detection and Relation Parsing. Pattern Recognition. 2022. vol. 132. no. 108946. DOI: 10.1016/j.patcog.2022.108946.
12. Sage C., Aussem A., Elghazel H., Eglin V., Espinas J. Recurrent Neural Network Approach for Table Field Extraction in Business Documents. International Conference on Document Analysis and Recognition (ICDAR). 2019. pp. 1308–1313. DOI: 10.1109/ICDAR.2019.00211.
13. Khan S.A., Khalid S.M.D., Shahzad M.A., Shafait F. Table Structure Extraction with Bi-Directional Gated Recurrent Unit Networks. International Conference on Document Analysis and Recognition (ICDAR). 2019. pp. 1366–1371. DOI: 10.1109/ICDAR.2019.00220.
14. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural computation. 1997. vol. 9. no. 8. pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
15. Paliwal S.S., Vishwanath D., Rahul R., Sharma M., Vig L. TableNet: Deep Learning Model for End-to-end Table Detection and Tabular Data Extraction from Scanned Document Images. International Conference on Document Analysis and Recognition (ICDAR). 2019. pp. 128–133. DOI: 10.1109/ICDAR.2019.00029.
16. Tensmeyer C., Morariu V.I., Price B., Cohen S., Martinez T. Deep Splitting and Merging for Table Structure Decomposition. International Conference on Document Analysis and Recognition (ICDAR). 2019. pp. 114–121. DOI: 10.1109/ICDAR.2019.00027.
17. Siddiqui S.A., Fateh I.A., Rizvi S.T.R., Dengel A., Ahmed S. DeepTabStR: Deep Learning based Table Structure Recognition. International Conference on Document Analysis and Recognition (ICDAR). 2019. pp. 1403–1409. DOI: 10.1109/ICDAR.2019.00226.
18. Couasnon B., Lemaitre A. Recognition of Tables and Forms. Handbook of Document Image Processing and Recognition. Chapter Recognition of Tables and Forms. 2014. pp. 647–677. DOI: 10.1007/978-0-85729-859-1_20.
19. Zucker A., Belkada Y., Vu H., Nguyen V.N. ClusTi: Clustering Method for Table Structure Recognition in Scanned Images. Mobile Networks and Applications. 2021. vol. 26. no. 4. pp. 1765–1776. DOI: 10.1007/s11036-021-01759-9.

20. Nguyen Q.D., Le D.A., Phan N.M., Zelinka I. OCR error correction using correction patterns and self-organizing migrating algorithm. Pattern Analysis and Applications. 2021. vol. 24. pp. 701–721. DOI: 10.1007/s10044-020-00936-y.
21. Patel C., Patel A., Patel D. Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study. International Journal of Computer Applications. 2014. vol. 55(10). pp. 50–56. DOI: 10.5120/8794-2784.
22. Vorobjev V. PGI Geophysical data. 2015. October, November, December. Murmansk, Apatity: PGI KSC RAS, 2016.
23. Vorobjev V.G., Roldugin V.C., Yagodkina O.I. Large Amplitude Undulations of Evening Site Diffuse Aurorae. Optical Characteristics and Conditions of Generation. Geomagnetism and Aeronomy. 2015. vol. 55. pp. 45–50. DOI: 10.1134/S0016793215010132.
24. Vorobjev V.G., Yagodkina O.I., Antonova E.E. Ion Pressure in Different Regions of the Dayside Auroral Precipitation. Geomagnetism and Aeronomy. 2020. no. 60. pp. 727–736. DOI: 10.1134/S0016793220060146.
25. Popov L.N., Krakovetskiy Yu.K., Gokhberg M.B., Pilipenko V.A. Terrogenic effects in the ionosphere: a review. Physics of the Earth and Planetary Interiors. 1989. vol. 57. no. 1-2. pp. 115–128.
26. Zhang T., Qiu H., Castellano G., Rifai M., Chen C.S., Pianese F. System Log Parsing: A Survey. IEEE Transactions on Knowledge and Data Engineering. 2023. pp. 8596–8614. DOI: 10.1109/TKDE.2022.3222417.
27. Patil O., Chavan U. Rule Based Expert System for Error Log Analysis. International Journal of Innovative Technology and Exploring Engineering. 2020. vol. 9. no. 10. pp. 188–192. DOI: 10.35940/ijitee.J7466.0891020.
28. Peta S. Python- An Appetite for the Software Industry. International Journal of Programming Languages and Applications (IJPLA). 2022. vol. 12. DOI: 10.5121/ijpla.2022.12401.
29. Singh B.P. Python and Its Future Scope. International Journal of Advanced Research in Science, Communication and Technology. 2022. pp. 400–403. DOI: 10.48175/IJARSCT-4829.
30. Dr U., Patkar U. Python for web development. International Journal of Computer Science and Mobile Computing. 2022. vol. 11. no. 4. pp. 36–48. DOI: 10.47760/ijcsmc.2022.v11i04.006.
31. Rong W., Xu T., Sun, Z., Sun, Z., Ouyang, Y., Xiong, Z. An Object Tuple Model for Understanding Pointer and Array in C Language. IEEE Transactions on Education. 2023. pp. 1–12. DOI: 10.1109/TE.2023.3236027.
32. Peta S. C Programming Language–Still Ruling the World. Global Journal of Computer Science and Technology. 2022. vol. 22(1). pp. 9–13.
33. Park H., Kim S., Bae B. Dynamic code compression for JavaScript engine. Software: Practice and Experience. 2023. vol. 53. no. 5. pp. 1196–1217. DOI: 10.1002/spe.3186.
34. Wang Z., Bu D., Wang N., Yu S., Gou S., Sun A. An empirical study on bugs in JavaScript engines. Information and Software Technology. 2023. vol. 155. no. 107105. DOI: 10.1016/j.infsof.2022.107105.
35. Romanchuk V.M. The Problem of Adequacy of the Analytic Hierarchy Process. Modelling and Data Analysis. 2022. vol. 10. no. 4. pp. 79–87. DOI: 10.17759/mda.2020100407.
36. Polat T.K. An Application of Analytic Hierarchy Process and Fuzzy Analytic Hierarchy Process to the Case Type Selection Problem. Academic Perspective Procedia. 2018. vol. 1. no. 1. pp. 1179–1188. DOI: 10.33793/acperpro.01.01.188.
37. Vorobev A.V., Pilipenko V.A., Enikeev T.A., Vorobeva G.R. Geoinformation system for analyzing the dynamics of extreme geomagnetic disturbances from observations of

ground stations. Computer Optics. 2020. vol. 44(5). pp. 782–790. DOI: 10.18287/2412-6179-CO-707.

38. Vorobev A.V., Soloviev A.A., Pilipenko V.A., Vorobeva G.R., Gainetdinova A.A., Lapin A.N., Belakhovsky V.B., Roldugin A.V. Local diagnostics of aurora presence based on intelligent analysis of geomagnetic data. Solar-Terrestrial Physics. 2023. vol. 9(2). pp. 22–30. DOI: 10.12737/stp-92202303.

Vorobev Andrei — Ph.D., Dr.Sci., Professor, Geoinformation systems department of computer science and robotics faculty, Ufa University of Science and Technology; Senior researcher, Geophysical Center of RAS. Research interests: geoinformation technologies, digital signal processing. The number of publications — 164. geomagnet@list.ru; 12, Karl Marx St., 450007, Ufa, Russia; office phone: +7(917)345-2299.

Lapin Alexander — Student, Geoinformation systems department of computer science and robotics faculty, Ufa University of Science and Technology. Research interests: mathematical and computer modeling, digital twins, machine learning. The number of publications — 13. meccos160@yandex.ru; 12, Karl Marx St., 450008, Ufa, Russia; office phone: +7(917)439-6040.

Vorobeva Gulnara — Ph.D., Dr.Sci., Professor, Computational mathematics and cybernetics department of computer science and robotics faculty, Ufa University of Science and Technology. Research interests: geoinformation and web technologies, systems of information storing and processing. The number of publications — 153. gulnara.vorobeva@gmail.com; 12, Karl Marx St., 450008, Ufa, Russia; office phone: +7(917)417-4111.

Acknowledgements. The reported study was funded by RSF, project number 21-77-30010.