

- сущность “zadanie” — служит для хранения информации о выданных заданиях;
- сущность “project” — служит для хранения информации о выполняемых проектах;
- сущность “klient” — служит для хранения информации о клиентах компании;
- сущность “sost” — служит для хранения информации о состоянии выполнения задания;
- сущность “dolzn” — служит для хранения информации о должностях в компании;
- сущность “prav” — служит для хранения информации о правах доступа к приложению;
- сущность “users” — служит для хранения информации о пользователях приложением.

Созданная физическая модель базы данных представлена на рисунке 1.

Как правило, основной целью физического проектирования базы данных является описание способа физической реализации логического проекта базы данных [2].

**Заключение.** Созданный программный продукт позволит автоматизировать учет и контроль исполнения заданий сотрудниками.

#### Список цитируемых источников

1. *Гайдамакин, Н. А.* Автоматизированные информационные системы, базы и банки данных. Вводный курс : учеб. пособие / Н. А. Гайдамакин. — М. : Гелиос АРВ, 2002. — 368 с.
2. *Брауде, Э.* Технология разработки программного обеспечения / Э. Брауде. — СПб. : Питер, 2004.
3. *Петров, В. Н.* Информационные системы / В. Н. Петров. — СПб. : Питер, 2002. — 688 с.

УДК 004.92

**А. В. Корсак, М. И. Козел, М. А. Вареник**

*Учреждение образования «Барановичский государственный университет», Барановичи*

### АЛГОРИТМЫ СЖАТИЯ ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ И ПЕРСПЕКТИВЫ ИХ ИСПОЛЬЗОВАНИЯ

**Введение.** К настоящему времени разработано множество алгоритмов сжатия, ориентированных на различные типы изображений.

Сжатие — способ более эффективного представления информации. Эффективность изображения определяется уровнем его избыточности. В цифровых изображениях существуют различные типы избыточности. Первый тип обуславливается повторением в файле изображения пикселей одинаковых цветов, преобладанием какого-либо цвета (статистическая избыточность). Такая избыточность характерна для деловой графики (диаграммы, графики). Другой тип избыточности является результатом корреляции пикселей. Такая избыточность называется пространственной и чаще всего присутствует в фотореалистичных изображениях. Также для фотореалистичных изображений характерна психофизическая избыточность. Она связана с тем, что не все детали изображения одинаково воспринимаются зрителем. Ряд деталей, имеющих одновременно малые размеры и малый контраст, не видны на изображении, поэтому могут быть удалены без вреда для качества восприятия изображения. Невозможно в общем виде сказать, имеет ли алгоритм сжатия высокую или низкую эффективность. Эффективность компрессии зависит от типа сжимаемого изображения [2].

Целью данного исследования является рассмотрение основных методов и алгоритмов сжатия изображений и перспективы их использования.

**Основная часть.** Несмотря на быстрый рост ёмкости устройств хранения, по-прежнему весьма актуальными остаются различные алгоритмы сжатия изображений. Существующие методы сжатия могут быть разделены на две категории: сжатие всего файла (иногда эта процедура называется упаковкой пикселей) и «внутренние» методы сжатия, работающие с самой структурой файла. Практически все графические системы поддерживают форматы, в которых используются структурные методы сжатия, учитывающие структуру файла.

Все существующие алгоритмы можно разделить на два больших класса: алгоритмы сжатия без потерь, алгоритмы сжатия с потерями.

Для алгоритмов сжатия без потерь существует алгоритм, обратный алгоритму сжатия, позволяющий точно восстановить исходное изображение. Для алгоритмов сжатия с потерями обратного алгоритма не существует. Существует алгоритм, восстанавливающий изображение, не обязательно точно совпадающее с исходным. Алгоритмы сжатия и восстановления подбираются так, чтобы добиться высокой степени сжатия и при этом сохранить визуальное качество изображения.

Алгоритмы сжатия без потерь:

1. RLE (run-length encoding) — один из самых старых и самых простых алгоритмов архивации графики. Изображение в нем вытягивается в цепочку байт по строкам раstra. Само сжатие в RLE происходит

за счет того, что в исходном изображении встречаются цепочки одинаковых байт. Применим алгоритм для изображений с небольшим количеством цветов: деловая и научная графика. Применяется в форматах PCX, TIFF, BMP. На его принципах и комбинациях основываются более эффективные и сложные алгоритмы.

2. Алгоритм Лемпеля—Зива—Велча (Lempel—Ziv—Welch, LZW). Идея алгоритма LZW в том, что со входного потока последовательно считываются символы, далее в созданной таблице проверяются строки. Если данная строка имеется, то следующий символ считывается, а если строки нет, тогда в поток записывается код для предыдущей найденной строки, строка вносится в таблицу. В настоящий момент алгоритм применяют во многих известных программах сжатия данных — ZIP, ARJ, LHA, а также в файлах формата TIFF, PDF, GIF, PostScript и др.

3. Кодирование Хоффмана. В данном случае также применяется кодирование повторяющихся данных, где для кодирования часто повторяющихся последовательностей используют коды меньшей длины, в отличие от более редких последовательностей. Словарь кодов — это двоичное дерево, где редко встречающиеся повторяющиеся цепочки располагаются дальше от корня дерева. Тут номера веток от корня до самой цепочки и представляют собой код последовательности. В XXI в. этот алгоритм почти не применяется в чистом виде, но используется в файлах PNG, JPEG [1].

Алгоритмы сжатия с потерями.

1. Алгоритм JPEG (Joint Photographic Expert Group). В отличие от других методов сжатия, JPEG не является единым алгоритмом, а представляет собой набор методов. Схема JPEG была специально разработана для сжатия цветных и полутоновых многоградационных изображений (фотографий, телевизионных заставок, другой сложной графики), в которых различия между соседними пикселями незначительны. Алгоритм JPEG обеспечивает высокий коэффициент сжатия. Бывают ситуации, в которых алгоритм создает «ореол» вокруг резких вертикальных и горизонтальных границ в изображении. При слишком высокой степени сжатия изображение делится на блоки  $8 \times 8$  пикселей. Поддерживается алгоритм JPEG в форматах Quick Time, PostScript Level 2, Tiff 6.0.

Алгоритм JPEG используется там, где появляется необходимость хранить фотоизображения: полиграфии, в цифровых фотоаппаратах, в Интернете и т. д. Он занимает видное место в системах мультимедиа. Этот алгоритм не применяется для сжатия изображений при многоэтапной обработке, потому что искажения будут внесены в изображения при каждом этапе сохранения промежуточных результатов обработки. Для сжатия астрономических или медицинских изображений данный алгоритм не подходит [1].

2. Алгоритм JPEG-2000. Он использует технологию вейвлет-преобразования, которая основывается на представлении сигнала в виде суперпозиции базовых функций — волновых пакетов. Таким образом, изображение не только станет более четким и гладким, но и размер файла по сравнению с JPEG при таком же качестве уменьшится. Из-за применения вейвлетов, изображения в данном формате при высоких степенях сжатия устраняются недостатки более ранней версии. JPEG-2000 применяется для цифровых охранных систем, а также в разных алгоритмах распознавания и в биометрии. JPEG-2000 можно использовать для создания изображения глубиной цвета в 48 бит. Широкое распространение данного алгоритма способствует введению инновационных технологий и приложений для работы с изображениями [1].

3. Алгоритм фрактального сжатия изображения. Различные методы сжатия изображений основываются на устранении тех или иных форм избыточности, в частности, фрактальные методы рассматривают самоподобие как источник избыточности. Считается, что самоподобие является свойством почти всех природных объектов и их изображений, следовательно, устранение этой формы избыточности может значительно уменьшить объем данных, необходимых для описания природного объекта или его изображения.

Фрактал — это структура, выделенная при анализе изображения и обладающая схожей формой независимо от ее размеров. Например, в изображении кроны дерева фрактал — изображение листа. Фрактальное сжатие изображений основано на гипотезе, согласно которой в любом изображении можно обнаружить локальное самоподобие различных его частей.

При фрактальном сжатии изображение представляется в более компактной форме — с помощью коэффициентов системы итерируемых функций (IFS). IFS представляет собой набор трехмерных аффинных преобразований, переводящих одно изображение в другое. Преобразованию подвергаются точки в трехмерном пространстве ( $x$ -координата,  $y$ -координата, яркость). Фактически фрактальная компрессия — это поиск самоподобных областей в изображении и определение для них параметров аффинных преобразований.

Алгоритм фрактального сжатия несимметричен по времени — на сжатие изображения требуется примерно в 1 000—10 000 раз больше времени, чем на распаковку. Это объясняется большими затратами времени на поиск в изображении подобных областей. Алгоритм ориентирован на полноцветные изображения и изображения в градациях серого цвета, без резких переходов цветов. Фрактальное сжатие реализовано в формате FIF [2].

В настоящий момент фрактальные методы используются для приложений архивирования, например, цифровые энциклопедии, которым кодирование требуется один раз, а декодирование — многократное количество раз. Фрактальное сжатие применяется в ряде узкоспециализированных задач, таких как передача изображений со спутников. В медицине рентгеновские снимки, обработанные с помощью фрактальных алгоритмов, дают не только более качественную картинку, но и более качественную диагностику [1].

4. Рекурсивный (волновой) алгоритм (wavelet). Данный вид архивации используется очень давно и происходит из идеи применения когерентности областей. Алгоритм ориентирован на черно-белые

и цветные изображения с плавными переходами. Идеально подходит для иллюстраций типа рентгеновских снимков. Если задается слишком большой коэффициент, то на резких границах, а именно приходящихся на диагонали, появляется «лестничный эффект» — ступеньки разного размера в несколько пикселей, а также яркости. Идея алгоритма заключается в том, что в файл сохраняется разница чисел между средними значениями соседних блоков в изображении, которая обычно принимает значения, близкие к 0.

**Заключение.** Выбор того или иного метода зависит от области, где будут применяться сжатые изображения, и от характеристик этих изображений [2]. Можно отметить, что каждый из методов находит свое место в узкоспециализированных областях. Каждый алгоритм имеет свои достоинства и недостатки, которые не существенны с точки зрения их использования. На сегодня устаревают алгоритмы, а также виды информации, на которые они применимы. Популярные алгоритмы не всегда эффективны на новых типах данных, что делает актуальной проблему синтеза новых алгоритмов. Поэтому будущее за новыми алгоритмами с высокими требованиями к ресурсам и все более и более высокой степенью сжатия [1].

#### Список цитируемых источников

1. Герасимова, В. В. Существующие методы сжатия изображения и их перспективы использования / В. В. Герасимова // Молодежный научный форум: технические и математические науки : сб. ст. по материалам XL Междунар. студенч. науч.-практ. конф. — № 11 (40). — С. 61—65.
2. Туликова, М. С. Методы и алгоритмы сжатия графической информации [Электронный ресурс] // М. С. Туликова // Науч.-техн. вестн. информ. технологий, механики и оптики. — 2006. — № 29. — Режим доступа: <https://cyberleninka.ru/article/n/metody-i-algoritmy-szhatiya-graficheskoy-informatsii>. — Дата доступа: 29.04.2019.

УДК 004.415.2

А. А. Кочубей

Белорусский государственный университет, Минск

### СТАНДАРТИЗАЦИЯ ДОКУМЕНТИРОВАНИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В РЕСПУБЛИКЕ БЕЛАРУСЬ

**Введение.** Современные требования к качеству программ определяют необходимость документирования любых видов работ при их создании [1] поскольку: документы являются средством коммуникации между всеми участниками процесса разработки программного обеспечения (далее — ПО); в документах закрепляются обязанности и полномочия каждого участника процесса разработки; документы выступают в качестве контрольных пунктов, позволяющих контролировать ход разработки и отражающих положение дел; для пользователя документация выполняет обучающую и справочную функции.

Безусловно, документация облегчает обмен информацией, но в то же время большое количество создаваемой документации оказывает своеобразную нагрузку на участников процесса разработки.

**Основная часть.** В исследовании использована группа общенаучных методов. Из указанной группы методов были использованы: общелогическая (в виде классификации, обобщения, анализа, синтеза) и эмпирическая (в виде сравнения) группы.

Классификация использовалась при создании групп использованных источников и литературы исследования по тематике и направлениям.

Обобщение было использовано при формулировке выводов на основании анализа определенных документов.

Анализ был применен при рассмотрении всей базы нормативных правовых актов (НПА) и технических нормативных правовых актов (далее — ТНПА), выделении отдельных актуальных и соответствующих теме НПА и ТНПА, определении содержащихся в них документов.

Метод синтеза использовался также для формулировки выводов, для создания общей картины произведенного исследования.

Сравнение использовалось при сравнении конкретных стандартов для определения сходства или различия содержащихся в них документов.

Графическая систематизация источников исследования представлена на рисунке 1.

На основании анализа ТНПА, отраженных в достаточно большом количестве, можно сделать вывод, что они содержат требования к ПО не только общего, но и специального характера, определяют стадии жизненного цикла (ЖЦ) ПО, составляющие этих стадий в виде отдельных работ и требования как к работам, так и стадиям в целом.

НПА же, в свою очередь, регламентируют общие вопросы, определяя только основные моменты (закрепление субъектов, их прав и обязанностей, трактовки связанных с данной тематикой терминов и т. д.).