

УДК 004.932

Е.О. Богачева, Д.В. Дмитриев, О.Н. Корелин

ОСОБЕННОСТИ СЖАТИЯ ФОРМАТА DjVu ДЛЯ ЭКОНОМИИ СЕТЕВОГО ТРАФИКА И ИССЛЕДОВАНИЕ МЕТОДОВ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ

Нижегородский государственный технический университет им. Р.Е. Алексеева

Проведена сравнительная характеристика форматов JPEG и DjVu для сжатия изображений с текстовой информацией. Подробно рассмотрен алгоритм сжатия изображений с текстовой информацией DjVu. Выявлены недостатки формата DjVu. Проведено сравнение альтернативных методов выделения связанных компонент на изображении с помощью сегментации.

Ключевые слова: формат сжатия изображений, DjVu, JPEG, сегментация изображений.

Большая часть знаний в виде культурного и образовательного материала существующего сегодня доступна лишь в бумажной форме. Внедрение данного информационного богатства в цифровую область в схожем с оригиналом виде, с лёгким доступом и поиском, является основным шагом к созданию Всемирной Интернет Библиотеки.

Для экономии трафика при передаче оцифрованных изображений с текстом по вычислительным сетям могут использоваться различные форматы, одним из них является DjVu. Создание электронных архивов – качественно новый уровень хранения информации. Но в каком виде желательно хранить и передавать оцифрованные изображения, чтобы при этом были соблюдены два взаимоисключающих требования – минимальные объемы хранения и передачи с точным соответствием сохраняемого изображения оригиналу?

Один из наиболее известных форматов сжатия изображений с потерями данных JPEG, сжатие основывается на дискретно-косинусоидальном преобразовании. Получаемое после сжатия изображение имеет определенные отличия от оригинала. Если рассматривать JPEG в качестве формата для хранения текстовой информации, то основной задачей является не просто получение изображения, сжатого до требуемого значения, а подбор необходимого баланса между выходным объемом и получаемым при этом качеством сжатого изображения.

При JPEG-сжатии изображений с текстом возможно возникновение следующих проблем:

- Резкое искажение первоначальной цветовой палитры;
- Появление на изображении бесформенных цветных пятен с однородной заливкой;
- Появление каймы вокруг литер текста и других границ с резким переходом «цвет-цвет».

Таким образом, применимость формата JPEG для сжатия цветных и черно-белых изображений с текстовой информацией достаточно ограничена.

Формат сжатия изображений DjVu использует радикально отличный от JPEG принцип обработки изображений. DjVu относится к группе форматов, основанных на технологии MRC (Mixed raster content), перед сжатием изображение необходимо «разделить на слои».

DjVu (от фр. *déjà vu* – «уже виденное») – технология сжатия изображений с потерями, разработанная специально для хранения сканированных документов – книг, журналов и рукописей. Формат является эффективным решением для передачи всех нюансов оформления, например, исторических документов, где важное значение имеет не только содержание, но и цвет и фактура бумаги; дефекты пергамента: трещинки, следы от складывания; исправления, кляксы, отпечатки пальцев; следы, оставленные другими предметами и т. д. Архивы всего мира накопили огромное число исторических бумаг с неповторимыми особенностями такого рода.

Технологии, используемые в DjVu, можно разделить на две основные группы: сегментация и сжатие. Основная идея DjVu состоит в том, чтобы отделить текст от фона и изображения, а также использовать различные методы для сжатия полученных компонент. Исходное изображение разделяется на три составляющие:

1) фоновый слой содержит иллюстрации и текстуру страницы. Разрешение слоя по умолчанию понижается для экономии места до 100 точек на дюйм. Фон кодируется, используя вейвлетный алгоритм сжатия IW44;

2) маска содержит изображение текста и высоко контрастные компоненты картинок, используется как трафарет (указывает, является ли соответствующий пиксел в изображении основным или фоновым). Сохраняется с разрешением 300 точек на дюйм и сжимается с помощью алгоритма сжатия чёрно-белых изображений JB2 или DjVuBitonal;

3) передний план содержит цвет текста и рисунков (цветовую информацию о маске, как правило, большие участки смежных пикселей с почти одинаковыми цветами), разрешение обычно понижается до 25 точек на дюйм, так как символы текста имеют одинаковые или близкие друг к другу цвета. Кодируется, используя тот же алгоритм IW44.

Основная идея JB2 – найти отдельные фигуры на странице (например, символы), а также использовать алгоритм кластеризации формы, чтобы найти сходство между ними. Формы, которые являются репрезентативными для каждого кластера, кодируются как отдельные изображения методом, аналогичным JBIG1 (стандарт сжатия изображений без потерь). Слой-маска перед упаковкой JB2 претерпевает оптимизацию с целью освобождения от "случайных" штрихов и "шума сканера". Пиксель декодированного изображения строится следующим образом: если соответствующий пиксель в изображении маски равен 0, то пиксель имеет значения соответствующего пикселю фонового изображения, если маска пикселя равна 1, то цвет пикселя берется из цвета фонового слоя.

Для кодирования фона и переднего плана используется прогрессивный вейвлетный алгоритм сжатия IW44 или DjVuPhoto. Формат IW44 близок к новому стандарту JPEG2000, но менее требователен к системным ресурсам при декомпрессии изображения во время просмотра: декодер занимает меньше памяти и оптимизирован для работы в 3 раза быстрее, чем быстрый режим JPEG-2000. IW44 предлагает множество преимуществ перед существующими методами сжатия однотонных изображений.

В результате работы был разработан кодер и декодер алгоритмов сжатия изображений с потерями: из PGM (простой формат хранения полутоновых изображений portable pixmap) в JPEG и DjVu. Сравнение алгоритмов проводилось на тестовом наборе черно-белых изображений. Результаты показали, что DjVu показывает более успешные результаты сжатия, выходной файл DjVu составляет 3–7% от оригинала, JPEG: 14–217% (превышение 100% означает, что исходное изображение имеет размер меньше результирующего).

При оценке применимости JPEG важно помнить о целевой ориентации формата. Изначально он создавался для сжатия изображений полутонового типа, т.е. состоящих из большого объема плавных переходов одного цвета в другой. Для изображений «текст с иллюстрациями» характерна другая особенность – наличие заметного количества резких переходов «цвет-цвет» на границе отдельных литер текста и общего фона страницы.

DjVu показывает лучшие результаты сжатия изображений «текст с иллюстрациями» и является более приоритетным форматом при выборе формата для хранения отсканированных изображений, но проблемы, возникающие при работе с форматом DjVu, являются обратной стороной его преимуществ:

- ресурсоемкость процесса;
- качество разделения изображения на отдельные слои.

Вторая проблема вытекает из того, что одним из ключевых критериев для определения принадлежности данных к тому или иному слою принимается вид перехода между соседствующими цветами. Это могут быть как резкие переходы между литерами текста и фоном страницы, между цветами, которыми залиты отдельные участки плаката/карты, между линиями диаграмм и окружающим фоном, так и плавные переходы, характерные для фотографии, рисунка, фоновой части страницы.

Существует множество способов разделения исходного изображения на три слоя: передний план, задний план и черно-белая маска. Одни алгоритмы предлагают следующую по-

следовательность: сегментация документов на визуально различимые регионы, классификация этих регионов в тексте чертежей, изображений или полутонов. Этот подход хорошо работает для двухуровневых или цветных документов с равномерным фоном, где образы четко отделены от текста. Другие алгоритмы, следуя схеме снизу вверх, пытаются найти текст, на основе локальных особенностей. Как правило, они используют отличительные особенности текста, которые выделяют его из фона. Например, текст на переднем плане идентифицируется по определенной регулярности и ориентации, и его пространственное единство (символы из одной строки имеют аналогичную высоту и ориентацию).

Сегментация делит изображение на составные части и объекты. Степень детализации этого деления зависит от решаемой задачи. Сегментация сложных изображений является весьма нетривиальной задачей обработки изображений.

На ранней стадии DjVu сегментация была заложена в идее цветной кластеризации, которая пришла из компьютерного зрения. Сначала в основу сегментации входил двухцветный алгоритм кластеризации, он не рассматривал задачу как "сегментация" или "поиск" текста, напротив применялся гораздо более общий алгоритм классификации переднего плана / фона. Изображение разбивалось на квадратные блоки пикселей. Алгоритм кластеризации k-means находил два доминирующих цвета в каждом блоке. Далее каждый пиксель назначался к переднему плану, если его цвет ближе к прототипу кластера переднего плана, чем прототипу кластера фона. Несмотря на свою простоту, этот алгоритм находил текст (как и многие другие объекты) в сложных цветных изображениях с очень хорошей точностью. Однако могли возникать проблемы при определении принадлежности к переднему или заднему плану у неравномерно отсканированных изображений, а метод DjVu должен был обрабатывать любой тип сканируемого документа.

Сегментация опирается на следующие стратегии разложения:

- 1) извлечение компонент: определение компонент связности или групп пикселей, которые должны принадлежать к фону или переднему плану;
- 2) фильтрация: принятие решения для каждого компонента, как предпочтительней его закодировать, в качестве переднего плана или фона. Вместо того, чтобы выполнять полное сжатие, необходимо полагаться на оценку стоимости бита, полученную с помощью простых измерений по компоненте и ее окрестности.

Сейчас MDL (minimum description length) принцип управляет сегментацией в формате DjVu. Каждое принятое решение сводит к минимуму общую стоимость кодирования. В качестве простейшей параметрической модели для разделения слоев в DjVu используется двумерная модель Маркова. В шаге фильтрации конкурируют две стратегии, связанные с различными схемами кодирования. Предпочтительной схемой кодирования является та, которая дает низкую общую стоимость кодирования. MDL принцип используется, чтобы сделать только одно решение для каждого компонента, что позволяет избежать трудоемкой минимизации комплекса целевой функции.

Компонент классифицируется как передний план, если

$$C_{fg}^{col} + C_{bg|fg}^{col} + C_{mask}^{bin} - C_{bg}^{col} \leq 0$$

и в качестве фона в противном случае.

- 1) при кодировании маски стоимость бита C_{mask}^{bin} примерно пропорционально периметру компонента. Эта схема не предполагает потерь, так как в DjVu потеря при черно-белом кодировании едва заметны;

2) предполагается, что цвет переднего плана однородный, кодируемую часть C_{fg}^{col} , которая является значением этого цвета, можно считать незначительной до тех пор пока несоответствие C_{fg}^{col} составляет разницу между пикселями переднего плана и их средним значением.

3) Для фона, чтобы избежать вычислений над всем изображением, мы должны вычислить разницу в стоимости $C_{bg}^{col} - C_{bg|fg}^{col}$. Схема кодирования фона оптимизирована для однотонных изображений и обычно требует больше битов для кодирования резких переходов.

Точность сегментации во многом определяет успех сжатия конечного изображения. По этой причине повышенное внимание должно быть уделено надежности первичной сегментации.

Как правило, алгоритмы сегментации монохромных изображений основаны на одном из двух базовых свойств яркости изображения: разрывности и однородности. В первом случае подход состоит в разбиении изображения на части исходя из резких перепадов значений яркости, которые происходят, например, на границах объектов. Вторая группа методов осуществляет разделение изображений на области, однородные в смысле определенных, заранее заданных критериев.

В рамках работы были рассмотрены альтернативные способы выделения компонент:

- методы выделения краев – задача решается путем обнаружения границ областей на основе разрыва яркости;
- методы пороговой обработки – совершаются пороговые преобразования, которые используют распределение определенных характеристик пикселей;
- методы сегментации на отдельные области, основанные на прямом поиске подходящих областей.

В большинстве методов выделения контуров и основанных на этих методах алгоритмов используются линейные операторы с последующей пороговой обработкой. Основная идея обнаружения перепадов базируется на поиске мест изображения, где яркость меняется быстро с помощью следующих двух критериев:

- найти места, где первая производная превосходит по модулю некоторый заранее заданный порог;
- найти места, где вторые производные яркости имеют пересечения нулевого уровня.

Задача выделения контуров требует использования операторов над соседними элементами, которые чувствительны к изменениям и подавляют области постоянных уровней яркости. Таким образом, формируется изображение признака, в котором те части изображения, где возникают изменения, становятся светлыми, в то время как остальные части остаются темными.

Математическое обоснование методов заключается в вычислении производных, представленных для цифровых изображений в виде дискретных приближений градиента.

Градиентом двумерной функции $f(x, y)$ называется вектор $\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$.

Модуль вектора градиента равен $\nabla f = [G_x^2 + G_y^2]^{1/2}$.

Основное свойство вектора градиента заключается в том, что он указывает в сторону максимального роста изменения функции f в точке (x, y) . Угол наклона этого вектора равен $\alpha(x, y) = \arctg(G_y/G_x)$.

В качестве градиентных методов выделяют операторы, представляющие собой матричные маски:

- детектор Собела;
- детектор Превитта;
- детектор Робертса;

Производные аппроксимируются разностями значений пикселей в малой окрестности изображения. На рис. 1, а показана окрестность размерами 3×3 , где \mathbf{z} – значения пикселей. Приближение частной производной в направлении \mathbf{x} (вертикальном) по отношению к центральной точке окрестности (т.е. \mathbf{z}_5) определяется разностной формулой $G_x = (\mathbf{z}_7 + \mathbf{z}_8 + \mathbf{z}_9) - (\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{z}_3)$. Аналогично, частная производная по направлению \mathbf{y} приближается разностным выражением $G_y = (\mathbf{z}_3 + 2\mathbf{z}_6 + \mathbf{z}_9) - (\mathbf{z}_1 + 2\mathbf{z}_4 + \mathbf{z}_7)$. Обе эти величины вычисляются во всех точках изображения с двумя масками, приведенными на рис. 1, б и в. После этого приближение градиентного изображения получается сложением абсолютных величин двух фильтрованных изображений.

Детектор Превитта обнаруживает края с помощью приближений Превитта первых производных, заданных на рис. 2. Детектор краев Робертса обнаруживает края с помощью приближений Робертса первых производных, заданных на рис. 3.

а)		
z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

б)		
-1	-2	-1
0	0	0
1	2	1

в)		
-1	0	1
-2	0	2
-1	0	1

Рис. 1. Окрестность (маска) (а), маски Собела (б и в)

-1	-1	-1
0	0	0
1	1	1
$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$		

-1	0	1
-1	0	1
-1	0	1
$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$		

Рис. 2. Маски Превитта и формулы приближения первых производных

-1	-0
0	1
$G_x = z_9 - z_5$	

0	-1
1	0
$G_y = z_8 - z_6$	

Рис. 3. Маски Робертса и формулы приближения первых производных

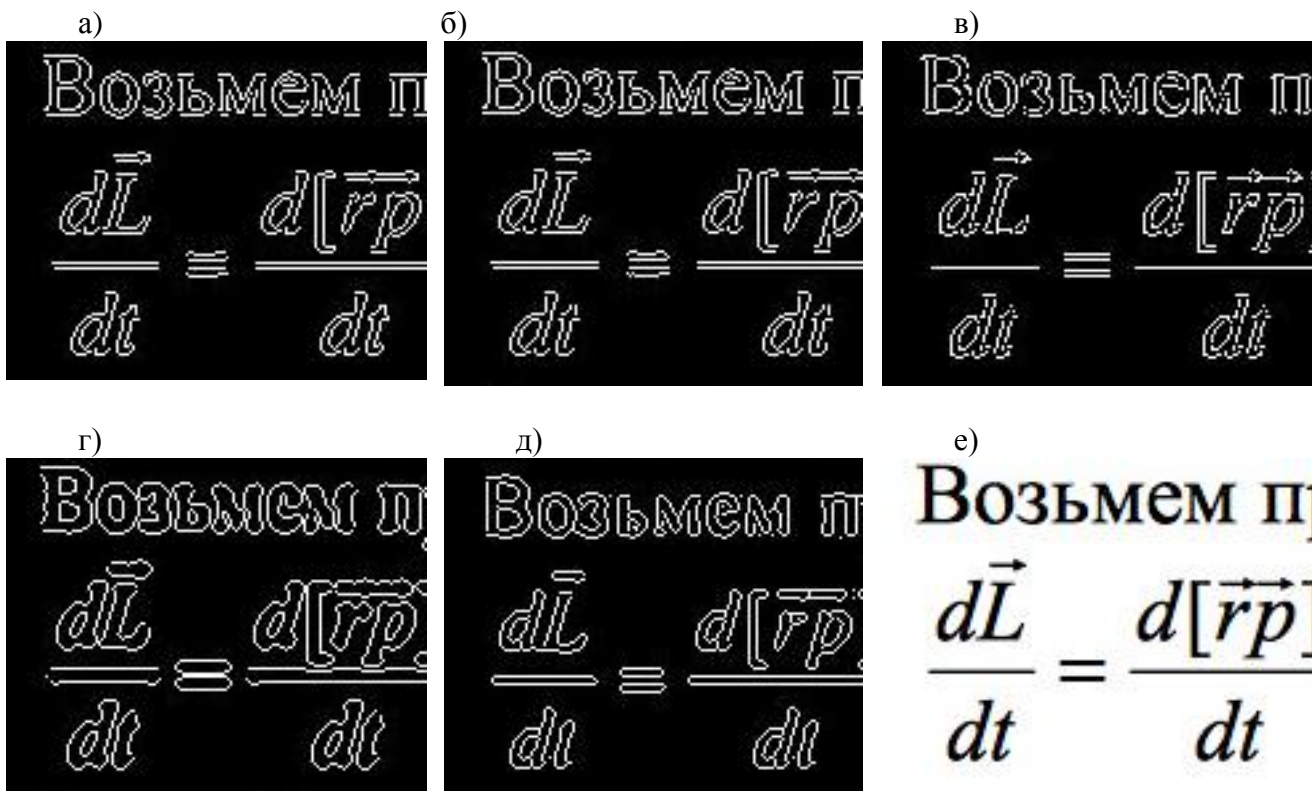


Рис. 5. Детектор краев Собела (а), детектор краев Превитта (б), детектор краев Робертса (в), детектор Лапласиан гауссиана (г), детектор Канни (д); исходное изображение (е)

В качестве методов, основанных на производной второго порядка, выделяют оператор Лапласиана. Данный оператор обнаруживает границы в местах смены знака производной функции яркости. Использование модуля лапласиана приводит к удвоению контуров, что дает нежелательный результат (рис. 5, з). Часто используют лапласиан в сочетании со сглаживанием, например по методу Гаусса, с целью уменьшения влияния шума. Такое сочетание называют детектором Лапласиан гауссиана. Детектор Лапласиан гауссиана обнаруживает края, выполняя поиск пересечений нулевого уровня после фильтрации $f(x, y)$ гауссианом.

В рассмотренных методах имеется возможность указать, к каким именно перепадам чувствительна данная оценка: к горизонтальным, вертикальным или перепадам обоих типов. Так как исследования связаны с изображениями с текстовой информацией, необходимо учитывать оба типа перепадов (текст содержит горизонтальные и вертикальные прямые).

Сравнение проводилось на шести изображениях с текстом, имеющих в своем составе формулы или рисунки. На рис. 5 показаны результаты сравнения обнаружения краев на примере одного изображения.

В результате сегментации нам необходимо получить замкнутые области для последующего выделения алфавита и обработки алгоритмом сжатия JВ2. Градиентные методы с простой пороговой обработкой уступают по качеству выделения контуров методу Канни, который обнаруживает края, выполняя поиск локальных максимумов градиента $f(x, y)$. Градиент вычисляется от гауссиана.

Детектор Канни можно описать следующим образом:

1) изображение сглаживается гауссовым фильтром с заданным стандартным отклонением σ для сохранения шума;

2) в каждой точке изображения вычисляется градиент $g(x, y) = [G_x^2 + G_y^2]^{1/2}$ и направление края (направление градиента) $\alpha(x, y) = \text{arctg}(G_y/G_x)$. Направления градиента используется для уточнения контуров. Точки перепада определяются как точки локального максимума градиента;

3) точки перепада, определенные в п.2, вызывают рост гребней на изображении модуля градиента. Алгоритм отслеживает верхнюю часть этих гребней и присваивает нулевое значение точкам, которые не лежат на гребне. В результате на выходе строится тонкая линия. Затем пиксели гребня подвергаются пороговой обработке с использованием двух порогов $T1$ и $T2$, причем $T1 < T2$. Пиксели гребня, величина которых больше $T2$, называются «сильными», а пиксели, значения которых попадают в интервал $[T1, T2]$, называются «слабыми».

4) алгоритм совершает соединение, добавляя к сильным пикселям слабые, которые δ -связаны с сильными.

В идеале рассмотренные методы должны обнаруживать только пиксели, принадлежащие краям и перепадам яркости. Однако на практике выделенные пиксели редко относятся только к этой категории в силу многих причин: воздействия шума, разрыва краев из-за неравномерного освещения и других факторов, которые вносят ложные перепады яркости в изображения. Также методы не удовлетворяют требованиям непрерывности и минимальной толщины контурных линий, поэтому за алгоритмом обнаружения краев обычно следует процедура компоновки выделенных пикселей краев в настоящие, осмысленные линии и краевые сегменты.

Методы пороговой обработки занимают центральное место в прикладных задачах сегментации изображений, благодаря интуитивно понятным свойствам и простоте реализации. Однако они не дают желаемого результата, если фон изображения сильно неоднороден по яркости. В таких случаях необходимо применять предварительную обработку для компенсации перепадов фоновой яркости, после чего можно совершать глобальное пороговое преобразование.

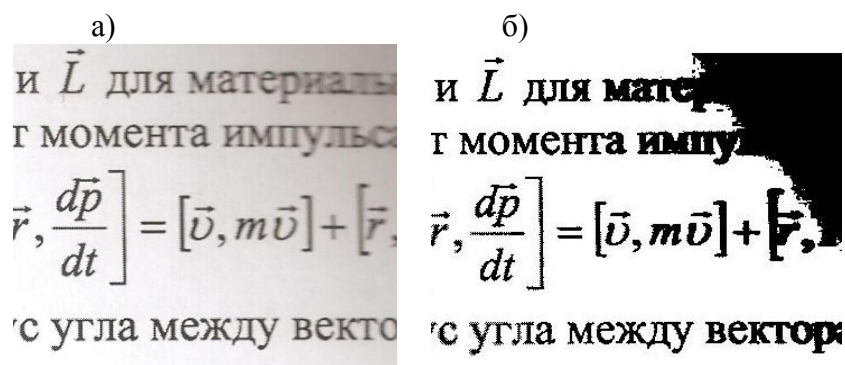


Рис. 6. Исходное изображение (а) и результат пороговой обработки (б)

Сегментация на отдельные области делит изображение таким образом, чтобы каждый пиксел принадлежал какому-либо сегменту. Существует большое число алгоритмов, конечной целью которых является разбиение изображения на области, самый распространенный метод – преобразование водораздела (рис. 7). Метод водораздела заключается в поиске линий, разделяющих локальные минимумы и максимумы значений яркостей элементов изображения. Данный вид сегментации является не пригодным для изображений с текстом, так как основной задачей является отделение переднего плана от заднего.

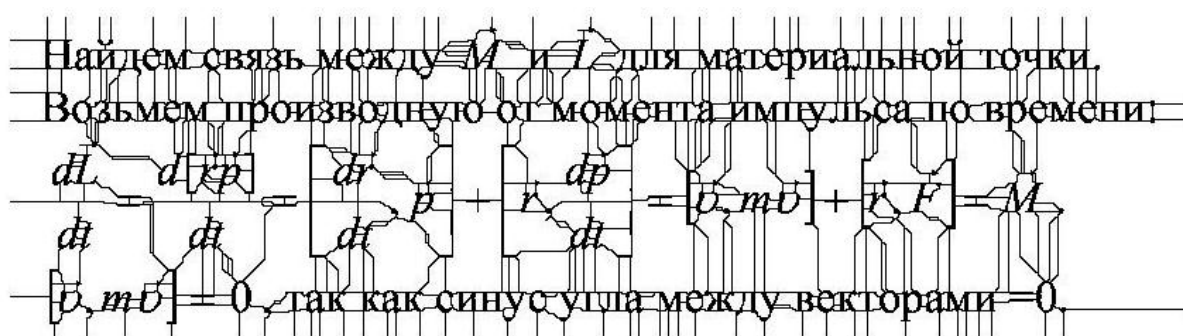


Рис. 7. Результат сегментации преобразованием водораздела

Конечный результат анализа изображений во многом определяется качеством сегментации, а степень детализации выделяемых характеристик зависит от конкретной задачи. Поэтому не существует отдельного метода или алгоритма, подходящего для решения всех типов задач сегментации, каждый из методов имеет свои достоинства и недостатки. В большинстве случаев выбирается один или несколько алгоритмов и модифицируются под специфические условия задачи.

В рамках проведенного сравнения лучшие результаты были получены после применения к изображению детектора Канни, пороговая сегментация и сегментация преобразованием водораздела не подходят для решения задачи выделения компонент из изображения с текстовой информацией.

С DjVu цветная страница отсканированного журнала в 300dpi обычно занимает от 40KB до 80KB, то есть примерно от 5 до 10 раз меньше, чем JPEG для подобного уровня четкости. Преобразование электронных документов в DjVu предлагает существенные преимущества; технология основывается на классификации каждого пикселя: на передний план (текст, графика) или на фон (картинки, текстуры бумаги и цвет), тем самым создавая сегментационные слои, которые сжимаются отдельно.

Библиографический список

1. High quality document image compression with djvu / L. Bottou [et al.] // Journal of Electronic Imaging. 1998. № 7(3). P. 410–428.
2. **Haffner, P.** A general segmentation scheme for DjVu document compression / P. Haffner, L. Vincent // International Symposium on Mathematical Morphology, 2002.
3. **Гонсалес, Р.** Цифровая обработка изображений в среде MATLAB / Р. Гонсалес, Р. Видс, С. Эддинс. – М.: Техносфера, 2006. – 616 с.
4. **Яне, Б.** Цифровая обработка изображений / Б. Яне. – М.: Техносфера, 2007. – 584 с.

*Дата поступления
в редакцию 22.04.2014*

E. Bogacheva, D. Dmitriev, O. Korelin

**FEATURES OF DjVu FORMAT COMPRESSION FOR NETWORK TRAFIFIC
SAVING AND RESEARCH OF IMAGES SEGMENTATION METHODS**

Nizhny Novgorod state technical university n.a. R.E. Alexeev

Purpose: Compare formats of image compression JPEG and DjVu with text information; Examine the algorithm and features of DjVu compression format in detail, analyze limitations of algorithm; review segmentation methods for the problem of separation foreground and background of image with text information.

Design/methodology/approach: JPEG and DjVu formats coders were developed as a result of research, segmentation methods were realized. Comparison was realized for black-white tests images with text and graphics information (pictures and formulas).

Findings: DjVu shows best results of compression of images "text with illustrations" and is the more prioritized format when selecting of format for scanned images storage than JPEG. The quality of separation to particular layers is the one of complexity of DjVu format. Alternative methods of segmentation were examined: contour, threshold and segmentation to particular parts. Best results were obtained after applying of Kanny's filter, threshold segmentation and watershed transform segmentation are not suitable for solving the problem of allocation of component images with text information.

Key words: image compression format, DjVu, JPEG, image segmentation.