

**Сжатие бинарной графики
в информационных и телекоммуникационных системах.**

Евгений Игоревич Агеенко

*Аспирант, начальник бюро сетевых технологий НИИ ИТС,
Ульяновский Государственный Университет
E-mail: ageson@cs.joensuu.fi*

Pasi Fränti

*Доктор наук, исследователь Академии Наук Финляндии.
Университет в Joensuu (Финляндия),
E-mail: franti@cs.joensuu.fi*

1. Технология «Document Imaging» и графические информационные системы.

Технология «Document Imaging» (DI) получила свое наибольшее распространение в последнее десятилетие [11]. DI-технология является основной составной частью концепции «Безбумажная технология» и включает в себя: преобразование документов в электронную форму, а также последующее хранение и управление их электронными образами. Технология предназначена для автоматизации таких задач и видов деятельности, как факсимильные коммуникации; документооборот и управление делопроизводством (document management, workflow); информационные банки данных и информационные системы; печать по запросу (print-on-demand); а также различные задачи, связанные с поиском, получением и обработкой документов, включая просмотр, коррекцию и вывод на печать (document browsing, retrieval and processing application). Более того, DI-технология должна в будущем стать основой таких бумажно-интенсивных отраслей как банковское дело и правительство. Результат внедрения технологии очевиден: значительное повышение производительности труда, простой, а, следовательно, и дешевый доступ к информации, значительное уменьшение стоимости на ее хранение и управление.

Одним из характерных примеров применения DI-технологии являются *графические информационные системы (ГИС)*, такие как, например, системы управления инженерным документооборотом (*EDM*), географические информационные системы (*GIS*) и проч. системы и банки данных [1,2,3]. В ГИС системах бумажные документы преобразуются посредством сканирования в электронную форму и хранятся в сжатом виде с целью уменьшения стоимости их хранения и передачи. Впоследствии документы (здесь и далее под словом *документы* подразумеваются электронные документы - изображения реальных бумажных документов) могут быть пролистаны пользователем и запрошены для детального просмотра, печати или дальнейшей обработки. После запроса документа хост-система (сервер) передает его в пользовательскую систему (клиент) в сжатом виде, где он впоследствии восстанавливается (декодируется) и обрабатывается. Примерами современных систем, построенных с использованием принципов DI, являются: персональный сканер/ксерокс/факс *Visioner's Paperport*, приложения для групповой обработки данных на базе *Lotus Notes*, базы данных и информационные системы на базе *Cornerstone/Oracle InerOffice*.

Процесс сканирования является в настоящее время относительно недорого стоящей процедурой. В то же время сканированные электронные документы представляются в растровом виде и занимают огромные объемы памяти. Так, например, стандартное факсимильное изображение (черно-белое, формата А4, с относительно низким оптическим разрешением 200 DPI, размера 1728x2376 точек) занимает примерно 0.5 Mb, машиностроительный чертеж (А1, 400 DPI) требует для своего хранения уже 16 Mb; и это еще не предел, т.к. типичные изображения, используемые в GIS, занимают порядка 100 Mb и более. Таким образом, узким местом системы, а соответственно и основной задачей, является уменьшение объема хранимой/передаваемой информации.

Объем занимаемой памяти оказывает влияние почти на все характеристики DI-системы; при его уменьшении экономия средств наблюдается во многих областях: меньше ресурсов требуется для хранения документов, соответственно меньший объем информации передается по каналам связи, следовательно, меньше загрузка телекоммуникационных систем и быстрее передача информации. Последнее влечет увеличение производительности труда персонала, потому как делает более быстрым и интерактивным доступ к распределенной в локальных/глобальных сетях и Internet информации и позволяет избежать нежелательных задержек, - т.о. меньше времени и сетевых ресурсов тратится на обращение (запрос + получение) к информации.

Возможность быстрого предварительного просмотра документов (*preview*) в информационной системе также имеет большое значение, потому как позволяет пользователю *листать (browse)* документы в архиве без необходимости получения и восстановления всего документа на

пользовательской системе. Быстрый просмотр осуществляется за счет передачи лишь небольшой порции данных, позволяющих построить изображение документа, достаточное для идентификации его содержания пользователем. Качество построенного изображения должно быть достаточно высоким, чтобы с некоторой долей уверенности определить содержание документа. В то же время изображение должно быть построено достаточно быстро. Количество передаваемой информации, необходимой для построения этого изображения, должно быть достаточно мало, чтобы не вызвать нежелательных временных задержек при передаче либо восстановлении изображения.

Итак, сжатие информации уменьшает объем передаваемой информации и уже позволяет ускорить доступ к информации. В свою очередь, восстановление информации само по себе, должно быть достаточно быстрым (как минимум быстрее, чем передача информации) и не сложным процессом, чтобы не вызывать дополнительных временных задержек и потерю интерактивности при работе с системой, и избежать необходимости использовать более сложное, следовательно, более дорогостоящее оборудование.

В настоящее время, почти все изображения, используемые в DI-системах, вводятся в бинарной форме (обычно черно-белой) и сжимаются при помощи быстрого алгоритма MMR, более известного как *CCITT/ITU Group IV* (далее G4), который является международным стандартом на сжатие информации в факсимильных системах с 1984 года [5]. Стандарт G4, базирующийся на кодировании бегущих битовых строк по Хаффману (МНС) и относительном адресном кодировании (READ), является усовершенствованным вариантом стандарта *Group III* (G3), используемым почти в каждом факсимильном аппарате [4]. Почти все существующие на настоящий момент ГИС поддерживают алгоритмы G3 и G4. Одним из основных препятствий к внедрению DI-технологий, как было отмечено выше, является высокая стоимость хранения и передачи электронных документов. К сожалению G4, не смотря на его быстрдействие, не может решить вышеуказанной проблемы: коэффициент сжатия - *КС* доставляемый алгоритмом G4, является явно недостаточным для современных задач, к тому же, из-за ряда технических проблем G4 никогда не заменит G3 в системах факсимильной связи [11]. (*КС* определяется здесь как отношение объема памяти, занимаемого оригинальным изображением, к объему памяти, занимаемому сжатым представлением изображения, достаточным для его однозначного восстановления [12]).

С целью решения проблемы, группой ведущих компаний (IBM, AT&T, Mitsubishi Electric) был разработан новый алгоритм для сжатия бинарной графики *JBIG* (*Joint Bilevel Image Expert Group*). С 1993 г. *JBIG* является последним полностью определенным международным промышленным стандартом (ISO/IEC 1154, CCITT/ITU T.82) и обеспечивает сжатие бинарных изображений с коэффициентом сжатия, превосходящим G4 на 20-60% [7]. Наилучшие результаты наблюдаются для сложных изображений, содержащих плотный или рукописный текст, фотографии или рисунки, а также фотокопий - основных документов, используемых в ГИС. К тому же *JBIG* был изначально разработан для факсимильных коммуникаций, и первые факсимильные аппараты, использующие *JBIG*, появятся в продаже в 1997 году.

К сожалению, несмотря на все преимущества, *JBIG* имеет один серьезный недостаток, препятствующий его широкому использованию: восстановление сжатых при помощи *JBIG* изображений в 2-3 раза медленнее по сравнению с G4, и требует быстрдействующего, следовательно, дорогостоящего оборудования. В настоящей работе предлагается новый алгоритм *БАРС*, базирующийся на *JBIG*, но превосходящий его по быстрдействию в 2.5 раза.

2. Алгоритм сжатия бинарной графики JBIG

В алгоритме JBIG изображение сжимается пиксель за пикселем в растровом порядке, используя арифметическое кодирование и контекстно-зависимое вероятностное моделирование [8,9]. Комбинация уже закодированных пикселей, окружающих кодируемый на настоящий момент пиксель, составляет контекст (см. рис 1). Вероятности появления черных и белых пикселей в изображении определяются адаптивно для каждого контекста, начиная с равномерного распределения в начале кодирования изображения. После определения контекста и вероятности пиксель кодируется при помощи арифметического кодирования, после чего значения вероятностей обновляются. Адаптивное кодирование в отличие от статического позволяет кодеру адаптироваться к локальным особенностям изображения в процессе кодирования, доставляя большее значение коэффициенту сжатия [9].

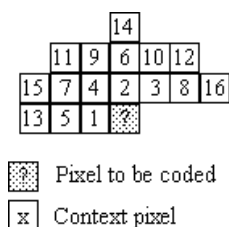


Рис. 1 Пример локального контекста из 16 пикселей. Пиксели включаются в контекст в порядке нумерации. Кодируемый пиксель отмечен знаком «?».

Чем больше размер локального контекста (число составляющих его пикселей), тем более точно предсказание (вероятностное моделирование) кодируемого пикселя, тем более отличается от равномерного распределение вероятностей, тем меньше значение энтропии, и, следовательно, больше КС. Однако для контекстов большого размера адаптация к статистическим свойствам изображения занимает большее время. Например, для контекста из 10 пикселей количество возможных комбинаций достигает $2^{10}=1024$.

Арифметическое кодирование является оптимальным кодированием по отношению к вероятностному моделированию [8]. В среднем каждый пиксель сжимается до размера $-\log_2(p)$ бит, где p - оценка вероятности кодируемого пикселя (в контекстном моделировании - это оценка условной вероятности в зависимости от контекста). В JBIG используется бинарный арифметический кодер, известный как QM-кодер [10]. Кодер суб-оптимален, но это в полной мере компенсируется быстротой кодирования (декодирования) и быстрой адаптацией к локальным особенностям изображения. Не смотря на быстрое действие QM-кодера, арифметическое кодирование продолжает оставаться довольно медленной процедурой.

Зависимость КС_{JBIG} от размера контекста изображена на рис. 2. Испытания показывают, что наибольшее значение КС доставляет контекст из 14 пикселей, в то же время наиболее эффективным значением является 10 (принятое в стандарте). Значения КС для набора тестовых изображений комитета ССИТТ (стандартные документы, используемые в факсимильных коммуникациях, формат А4, разрешение 200 DPI, размер изображения 1728x2376 пикселей) для различных методов сжатия приведены в табл. 1. Как видно БАРС незначительно превосходит JBIG по коэффициенту сжатия. При увеличении оптического разрешения значения КС

увеличиваются, т.к. увеличивается число равномерных черных и белых участков на изображении, при этом также увеличивается разница между KC_{G4} (достигает 40) и KC_{JBIG} (достигает 70), и между KC_{BAPC} и KC_{JBIG} .

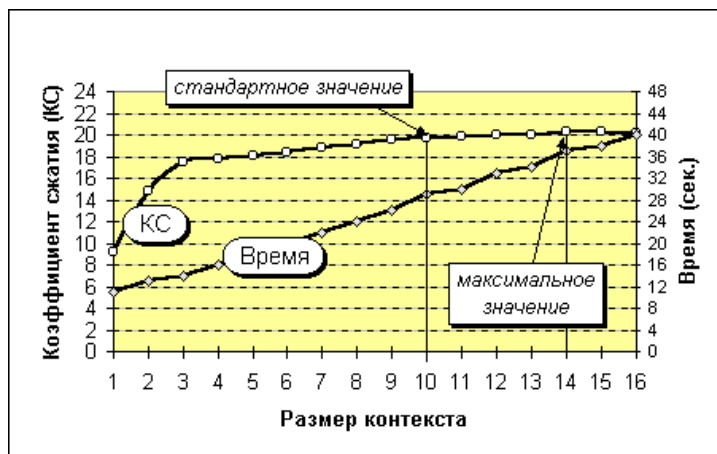


Рис. 2 Коэффициент сжатия и время восстановления как функции от размера контекста.

Таблица 1: Коэффициенты сжатия для алгоритмов: G3, G4, JBIG, BAPC, максимальные теоретические значения в статическом и динамическом случаях.

	G3	G4	$KC_{\text{стат., макс.}}$	$KC_{\text{динам., макс.}}$	$JBIG_{10}$	BAPC
ССИТТ 1	21.30	28.35	32.90	38.67	34.95	35.22
ССИТТ 2	26.50	47.51	60.30	67.17	60.45	60.17
ССИТТ 3	13.80	17.88	22.14	25.73	23.36	23.41
ССИТТ 4	6.80	7.41	8.76	10.38	9.45	9.53
ССИТТ 5	12.60	15.93	18.91	21.89	19.87	19.98
ССИТТ 6	18.60	30.82	39.78	45.32	40.89	40.51
ССИТТ 7	6.40	7.41	9.25	10.04	9.12	9.21
ССИТТ 8	15.90	26.87	33.63	39.86	36.07	35.30
ИТОГО:	12.19	15.55	18.95	21.72	19.72	19.78

3. Алгоритм сжатия бинарной графики БАРС

Алгоритм сжатия БАРС является комбинацией *блочного моделирования* и алгоритма JBIG. Блочное моделирование, впервые предложенное и детально изученное М. Кунтом и О. Джонсоном в 1980 г. [6], основывается на равномерном разбиении изображения на прямоугольные (а для простоты реализации - *квадратные*) блоки небольшого размера со стороной, равной 2-32 пикселя. После разбиения изображения блоки кодировались *по Хаффману* - кодирование битовыми цепочками переменной длины - в зависимости от их частотных характеристик. С разработкой арифметического кодирования и контекстного моделирования вопрос о возможности применения блочного моделирования для сжатия бинарных изображений поднимался редко. После длительного перерыва в 1993-1994 гг. такая возможность была рассмотрена П.Франти [12,13]. Им была предложена комбинация блочного моделирования с JBIG: в процессе сжатия изображение разбивается на квадратные блоки, которые классифицируются как черные, белые или смешанные, после чего, более медленному процессу - кодированию по JBIG, подвергаются пиксели, принадлежащие только смешанным блокам. Быстродействие достигается за счет того, что количество смешанных блоков в документах составляет не более 40% [12]. Естественно, блочная классификация является дополнительной информацией, которую требуется сохранять вместе со сжатым по JBIG изображением. Другими словами, с уменьшением размера блока, увеличивается объем дополнительной информации, что приводит к уменьшению значения КС. В то же время увеличивается доля черных и белых блоков, что приводит к увеличению быстродействия (см. рис. 3)

Основными отличиями алгоритма БАРС от описанного в [12,13] метода являются:

1. *Использование QM-кодера для сжатия блоковой классификации* - сначала блоковые коды представляются следующим образом: белый блок - одним «0»-битом, черный блок - последовательностью «1,0», смешанный блок - «1,1»; затем это представление кодируется QM-кодером, используя динамическую модель 2-го порядка (контекст составляют коды двух блоков, расположенных слева и сверху от кодируемого). Это позволяет не только приблизить $КС_{\text{БАРС}}$ к $КС_{\text{JBIG}}$, но и превзойти его.
2. *Блоковые коды размещаются в начале файла сжатого документа и используются для предварительного просмотра документов.* Предварительное изображение строится с использованием только блоковых кодов: каждый пиксель предварительного изображения представляет 1 блок оригинального изображения и принимает белый цвет, если блок белый, и черный цвет в противном случае. Такой способ вполне достаточен для успешной идентификации изображения (см. рис 4). Как показывают испытания, блоковые коды составляют не более **2%** от размера сжатого файла (см. табл. 2). Таким образом, для построения предварительного изображения на пользовательскую систему необходимо передавать в **50** раз меньшее количество информации, что значительно увеличивает быстродействие и снижает загрузку телекоммуникационных каналов.

Таблица 2: Соотношение блоковых и пиксельных кодов в ССИТТ-документах (бит.)

	Блоковые коды	Пиксельные коды
ССИТТ 1	2584	113469
ССИТТ 2	3200	64560
ССИТТ 3	4136	170792
ССИТТ 4	4176	426328
ССИТТ 5	4208	200856
ССИТТ 6	4104	96768
ССИТТ 7	6488	438816
ССИТТ 8	6248	109576
ВСЕГО:	2%	98%

В результате практических испытаний (см. рис. 3) определено наиболее оптимальное значение размера блока: **16x16** пикселей. Это значение, во-первых, является хорошим компромиссом между значением КС (уменьшается при уменьшении размера блока) и быстродействием метода, во-вторых, кратно 8, что упрощает реализацию алгоритма. В то же время меньший размер блока (например, 8x8) способствует повышению качества предварительного изображения и рекомендуется для использования с документами небольшого размера (меньше 1000x1000) или невысокого разрешения (менее 200 DPI).

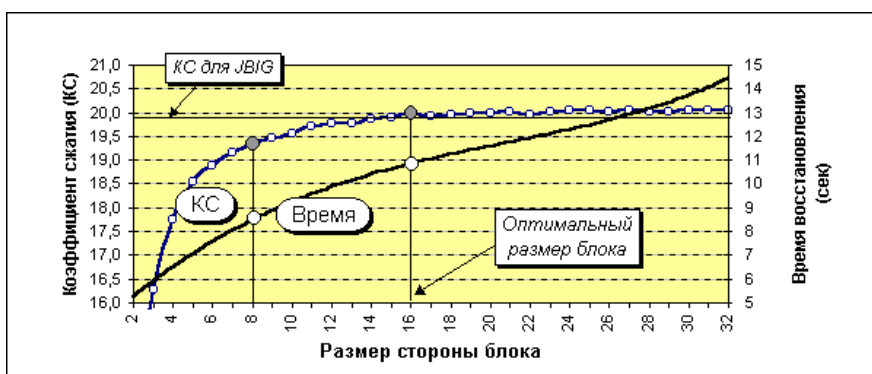


Рис 3. Зависимость коэффициента сжатия и времени восстановления от размера блока.

Теоретические исследования, подтвержденные практическими испытаниями, показали, что алгоритм БАРС превосходит JBIG по времени сжатия и восстановления изображений более чем в **2.5** раза (см. табл. 3).

Таблица 3: *Время сжатия/восстановления изображений для JBIG и БАРС (сек.)
Тестовая среда: Pentium-100 (80 mips), MS-Windows 95.*

	Сжатие			Восстановление		
	JBIG	БАРС	БАРС/JBIG	JBIG	БАРС	БАРС/JBIG
ССИТТ 1	31	9	29%	29	6	21%
ССИТТ 2	31	7	23%	28	5	18%
ССИТТ 3	32	12	38%	28	11	39%
ССИТТ 4	31	19	61%	29	21	72%
ССИТТ 5	33	12	36%	28	11	39%
ССИТТ 6	31	9	29%	29	7	24%
ССИТТ 7	32	17	53%	28	19	68%
ССИТТ 8	31	10	32%	28	9	32%
В среднем	31.5	11.9	38%	28.4	11.1	39%

4. Заключение.

В настоящей работе представлен алгоритм БАРС для сжатия бинарной графики без потери информации, основывающийся на комбинации блочного и контекстно-зависимого предиктивного моделирования и арифметического кодирования. Алгоритм БАРС превосходит алгоритм JBIG (алгоритм сжатия, включенный в существующий в настоящее время международный индустриальный стандарт для сжатия бинарных изображений с одноименным названием) в **2.5** раза по времени восстановления изображений, и не уступает ему по коэффициенту сжатия. Алгоритм позволяет строить изображения для предварительного просмотра документов в ГИС, используя лишь **2%** информации, хранящейся в начале файла сжатого изображения. Вышесказанное позволяет преодолеть затруднения, связанные с использованием JBIG в графических информационных системах и «Document Imaging» технологии в целом, а также значительно повысить производительность труда и снизить загрузку телекоммуникационных систем.

Приложение.

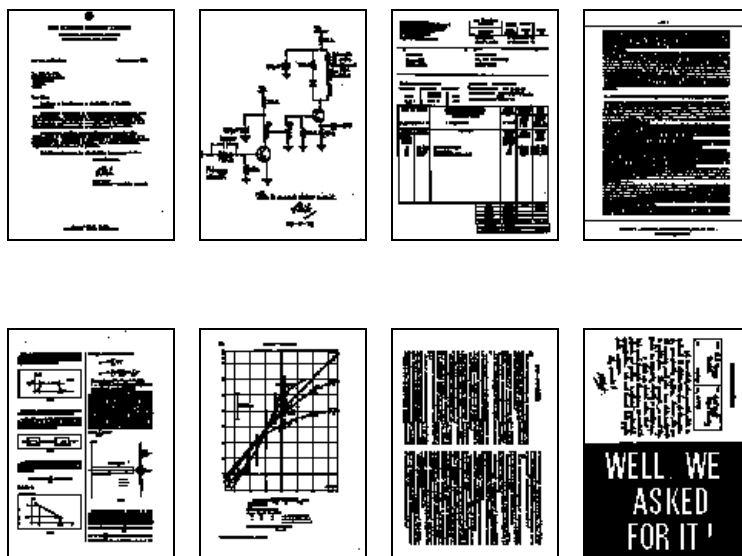


Рис 4. Предварительные изображения ССИТ документов (размер блока 16x16).

Литература

- [1] **Е.И. Агеенко**, К вопросу о сжатии инженерной графики, *Фундаментальные проблемы математики и механики*, **1**, Изд-во УлГУ, 1996.
- [2] **Е.И. Агеенко**, Сжатие чертежной графической информации, *Фундаментальные проблемы математики и механики*, **2**, Изд-во УлГУ, 1996.
- [3] **E.I. Ageenko and P. Fränti**, Enhanced JBIG-based compression for satisfying objectives of engineering document management system, *Optical Engineering*, March, 1998, to appear (University of Joensuu, Dept. of Computer Science. *Technical report A-97-2*).
- [4] **CCITT**, Standardization of Group 3 Facsimile Apparatus for Document Transmission, ITU Recommendation T.4, 1980.
- [5] **CCITT**, Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus, ITU Recommendation T.6, 1984.
- [6] **JBIG**, Progressive Bi-level Image Compression, ISO/IEC International Standard 11544, ITU Recommendation T.82, 1993.
- [7] **M. Kunt, O. Johnsen**, Block Coding: A Tutorial Review. *Proceedings of the IEEE*, Vol. **68** (7), pp. 770-786, July 1980.
- [8] **G.G. Langdon, J. Rissanen**, Compression of Black-White Images with Arithmetic Coding. *IEEE Transactions on Communications*, Vol. 29 (6), pp. 358-367, June 1981.
- [9] **A. Moffat**, Two-level Context Based Compression of Binary Images. *IEEE Proceedings Data Compression Conference*, Snowbird, Utah, 1991, pp. 382-391.
- [10] **W.B. Pennebaker, J.L. Mitchell**, *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, 1993.
- [11] *The JBIG Compression Standard: The Future for Document Imaging*, InfoTrends Research Group, Inc, September, 1996.
- [12] **P. Franti and O. Nevalainen**, A Two-Stage Modeling Method for Compressing Binary Images by Arithmetic Coding, *The Computer Journal*, **36**, 1993, pp. 615-622.
- [13] **P. Franti**, A Fast and Efficient Compression Method for Binary Images, *Signal Processing: Image Communication*, **6**, 1994, pp. 69-76.