
УДК 681.42.2, 681.327.22(07),004.021

А. М. Башкатов, канд. тех. наук, **А. О. Соловцов**

Инженерно-технический ин-т Приднестровского госуниверситета

(Молдова, MD-3300, г. Тирасполь, ул. Восстания, 2а,

тел. (0373777) 16642, 60375, e-mail: tiras_bash@mail.ru, alexey@forpost.md)

Использование алгоритмов сортировки в задачах количественной оценки изменяемых растров

Предложена технология количественного анализа растровых изменений посредством определения характеристик пикселей и применения к ним алгоритмов сортировки.

Запропоновано технологію кількісного аналізу растрових змін визначенням характеристик пікселів та використання щодо них алгоритмів сортування.

К л ю ч е в ы е с л о в а: растр, пиксел, интенсивность, цветовая компонента, порог, сортировка.

Растровое изображение объекта является одной из наиболее распространенных форм представления информации, отражающей ход какого-либо процесса или результаты наблюдений. Изображение формируется массивом пикселей по матричной схеме, что обеспечивает возможность максимально полного описания с требуемой степенью детализации. Исследователь может оценить объект с необходимой точностью и возможностью изменения разрешения, цветовой схемы или иных характеристик. Положительный эффект очевиден, особенно при статичном изображении.

Однако при проведении мониторинга [1] требуется не только визуально отобразить картинку, серию изображений, но и провести сравнительный анализ возможных изменений. Задача усложняется тем, что даже локальный фрагмент может иметь неоднозначно выраженные границы, а под воздействием сторонних факторов геометрия растровых зон может изменяться, их форма и целостность отличаться от первоначальных. Это затрудняет анализ, так как в ряде случаев необходима количественная оценка растровых изменений.

Алгоритмы сортировки. Вывод динамических растровых изображений обусловлен использованием соответствующих графических форматов, учитывающих особенности построения сцен и аппаратные ресурсы компьютера. Качество в значительной степени зависит от ресурсов памяти и быстродействия процессора, что не является сдерживающим фактором в

настоящее время. В этой связи более существенный интерес представляют способы оптимизации растра, варианты выбора эффективных схем его формирования и обработки.

Для практических целей полученный растр может быть представлен как в оттенках черно-белого (256 бит), так и в цветном решении (с применением цветовых компонент RGB). Количественно каждая точка растра характеризуется набором из трех значений, соответствующих красной, зеленой и синей составляющей спектра. В процессе построчного сканирования в системе создается массив значений, характеризующих пиксели растра в виде последовательного дисплейного списка. В результате получается набор дискретных значений с образованием однородных или смешанных цветовых зон, которые должны сформировать изображение в целом. Поэтому ставится задача ускорения процессов локализации и сбора фрагментов.

Для поиска нужной графической информации разработан ряд методов, среди которых особый практический интерес представляют методы, основанные на кластеризации изображений [2—6]. Базовые операции при таком подходе детально описаны в работе [7], а их очередность в общем виде схематически представлена на рис. 1.

Оперативное определение размеров контролируемых областей связано с определенными трудностями. Во-первых, линия границы часто имеет извилистую форму и не всегда четко выражена графически. Во-вторых, форма зоны подвержена геометрическим искажениям (с течением времени) и может иметь внутри участки с другими растровыми характеристиками. Это затрудняет исследование процессов динамики переноса и не дает возможности рассчитать дрейф растровых зон (например, при проведении геомониторинга).

Предлагается оптимизировать анализ исследуемого изображения, рассматривая наборы групп пикселей, обладающих сходными атрибутами и применив известный метод группового кодирования [8], а затем выполнить расчет зон, просуммировав пиксели, относящиеся к своим цвето-яркостным кластерам. Диапазон идентичности пикселей растра задается величинами порогов, характеризующих установленные интервалы (min—max) значений прозрачности A и цветовых компонент RGB. Затем аналогичная операция повторяется для следующего снимка (растра) с сохранением полученных результатов в массиве последовательных списков-строк.

Таким образом, в ходе ряда итераций и последующего анализа сформированных массивов значений (например, с использованием сдвиговых регистров [9]) можно получить данные, отражающие увеличение или уменьшение числа пикселей заданной интенсивности, что позволяет аргументи-

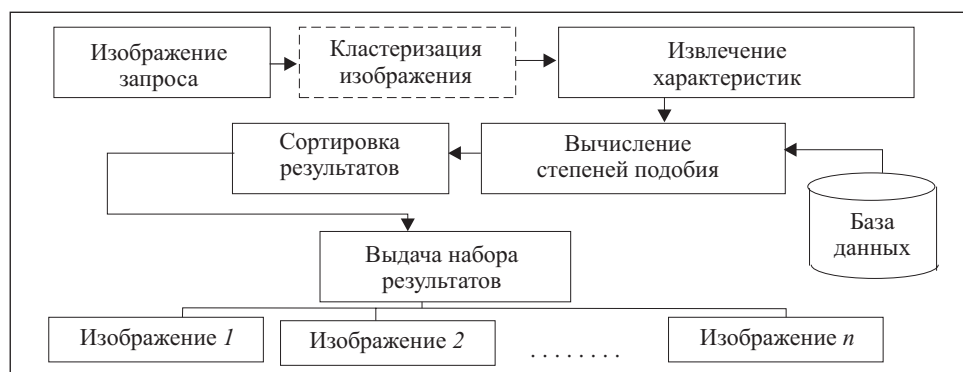


Рис. 1. Общая схема контекстного поиска изображений в базах данных на примитивном уровне абстракции [7]

ровано утверждать, что процесс развивается, площадь зоны увеличивается или, наоборот, процесс затухает.

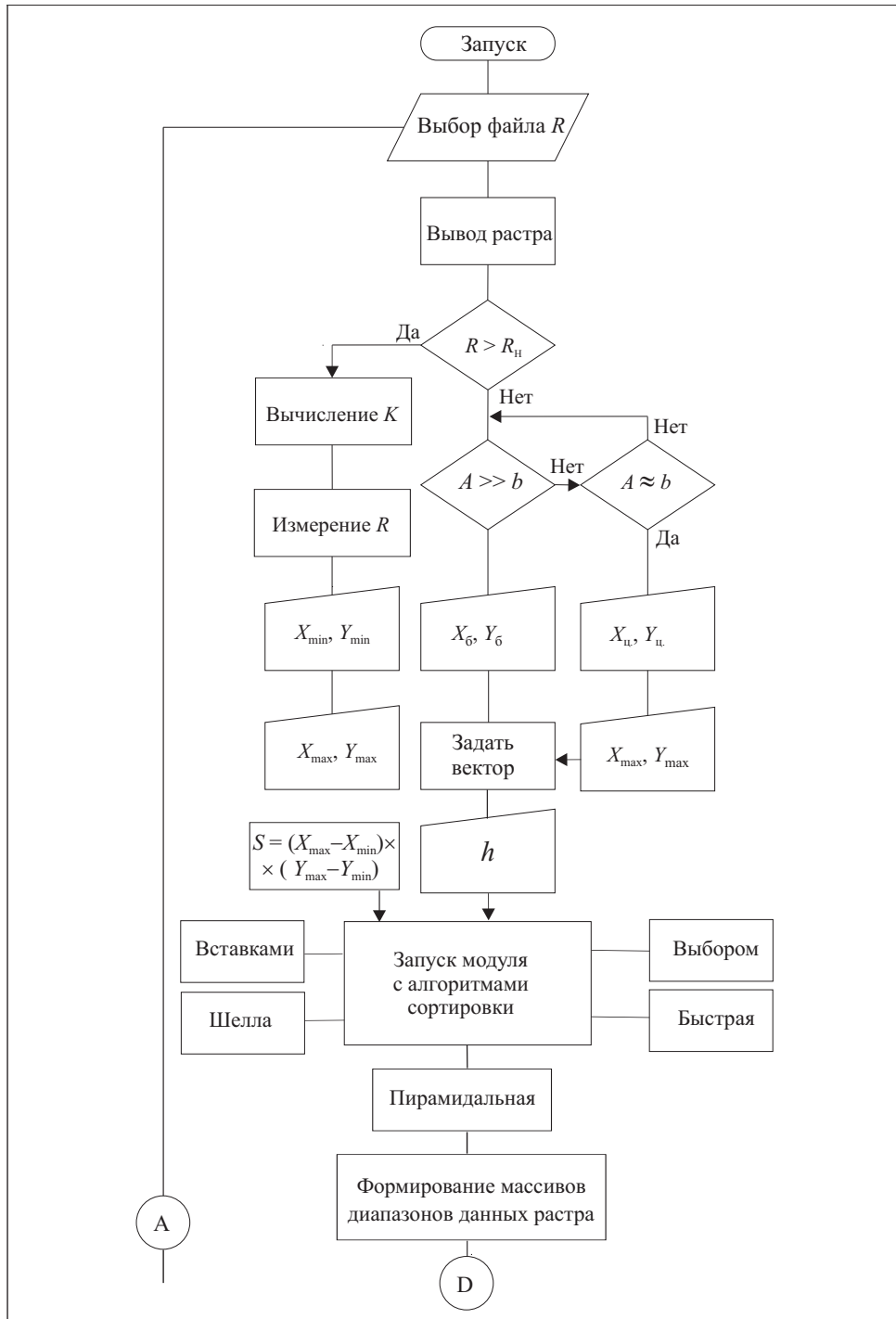
Практически эффект состоит в уменьшении времени селекции сходных по критериям идентичности пикселей. Поэтому поставлена цель — ускорить процесс формирования групп однородных кластеров вне зависимости от типа исследуемого растра.

Один из путей уменьшения временных потерь — применение алгоритмов сортировки [10]. На практике применяются следующие технологии: сортировка вставками [11]; сортировка Шелла [12]; пирамидальная сортировка [13]; быстрая сортировка [14]; сортировка выбором [15].

Для выбора наиболее эффективного алгоритма из приведенного списка следует учесть особенности его реализации и используемые параметры. Среди универсальных критериев можно выделить следующие [16, 17]: время сортировки; ресурсы памяти; устойчивость выполнения процедуры; естественность поведения; сфера применения.

Специалисты разделяют сортировки на внутренние (работа с данными в оперативной памяти с произвольным доступом) и внешние (упорядочивание информации, расположенной на внешних носителях).

Это накладывает дополнительные технические ограничения на работу алгоритма, так как, во-первых, доступ к носителю осуществляется последовательно, т.е. в каждый момент времени можно считать или записать только элемент, следующий за текущим, во-вторых, объем больших массивов данных (больших растров) превышает размер оперативного запоминающего устройства (ОЗУ). Кроме того, следует учитывать, что доступ к данным на носителе осуществляется значительно медленнее, чем при выполнении операций с оперативной памятью.



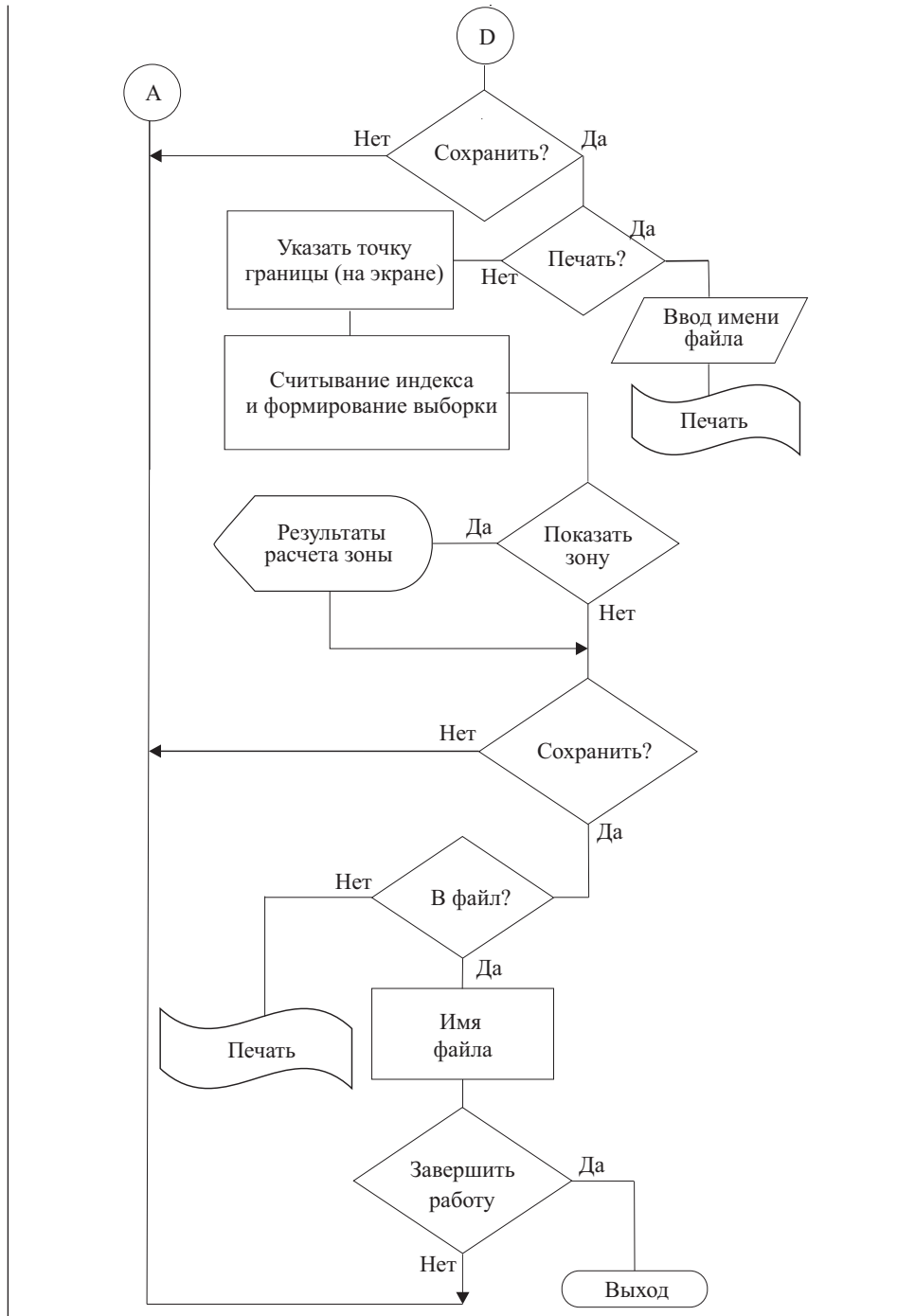


Рис. 2. Алгоритм программы оценки раstra с применением функций сортировки

Алгоритмы сортировки условно делятся на два основных подкласса.

1. Основанные на внутренней сортировке данных, которые сортируются в ОЗУ без дополнительных затрат; оперируют с массивами, целиком помещающимися в оперативной памяти с произвольным доступом к любой ячейке; применимы для работы с небольшими изображениями.

2. Использующие механизмы внешней сортировки, что обуславливает необходимость точного выбора метода сортировки, определения ресурсов дополнительной внешней памяти; оперируют с запоминающими устройствами большого объема с последовательным доступом для сортировки файлов.

Программное решение данной задачи с применением алгоритмов сортировки возможно с помощью алгоритма, приведенного на рис. 2, где K — тип кластера, $K = R/R_n$; R — разрешение исходного объекта; R_n — установленное разрешение экрана; h — ширина коридора; A и b — размеры адресного пространства (длина и ширина анализируемого участка растра); $X_{ц}$, $Y_{ц}$ — центр исследуемой зоны.

Работа программы начинается с загрузки файла растрового изображения, содержащегося на диске. Если изображение имеет недостаточный уровень детализации или требуется увеличить интересующий фрагмент в пределах окна, то пользователь должен курсором указать диагональные точки зоны растра (с координатами x_{\min} , y_{\min} , x_{\max} , y_{\max}). Если разрешение растра достаточное для контроля, то потребуется лишь задать базовую точку $x_б$, $y_б$ с указанием вероятного направления вектора поиска h и подтвердить действие. Расчетный параметр S определяет площадь коридора растровой зоны для выполнения сортировочных операций.

Имеется возможность выбора категории исследуемого растра — прямоугольная или вытянутая форма ареала. В последнем случае, ввиду явной несоразмерности длин сторон растра, определяется ориентация участка (вектором средней линии) и задается граница коридора (расстояние отклонения в одну сторону от средней линии). С учетом установленного разрешения программа рассчитывает число пикселей между крайними точками и запишет эти значения в соответствующие переменные среды. Если различие сторон рассматриваемого фрагмента незначительно (соотношение сторон $\geq 1:3$), то необходимо задать базовый адрес (левого нижнего пикселя) и вектор диагонали к правому верхнему пикселю участка.

Данные операции являются начальным подготовительным этапом и позволяют сократить время расчета, исключив обработку данных по другим пикселям растра. Определение в программе адреса точки растра выполняется по формуле [18]

$$Адрес_{пик} = (X_{\max} - X_{\min})(Y - Y_{\min}) + (X - X_{\min}) + Адрес_{баз}, \quad (1)$$

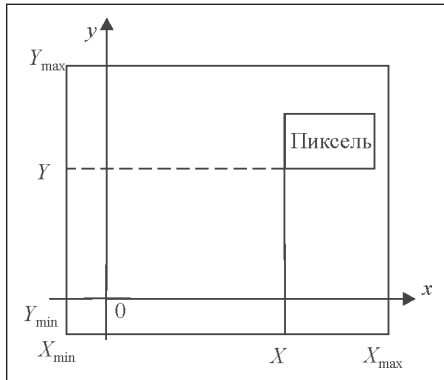


Рис. 3. Схема нахождения адреса пикселя [18]

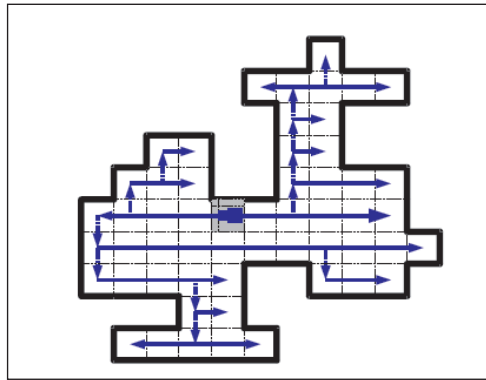


Рис. 4. Схема сканирования пикселей строк после анализа их соседства [20]

где базовый адрес пикселя в строке определяется с учетом правила знаков в виде

$$\text{Адрес}_{\text{баз}} = (X_0 - X_{\min})(Y_0 - Y_{\min}),$$

а значения остальных параметров поясняет геометрическая схема на рис. 3. Установленные адресаты пикселей через указатели связываются с соответствующими цвето-яркостными данными, формируя списки групп [2, 19]. После этого выполняется подсчет пикселей с индивидуальными значениями атрибутов (интенсивностью цветовых компонент или заданными в градациях серого). Образуются кластеры пикселей с одинаковыми значениями свойств, дискретно распределенные в строках растрового фрагмента.

Порядок выбора идентичных пикселей осуществляется исходя из условий их соседства и установленной связности (рис. 4). Для связных (четыре или восемь) пикселей выполняется расчет расстояний между базовыми цветами *ARGB* согласно следующей формуле [5]:

$$\rho(x, y) = \sqrt{\alpha(x_A - y_A)^2 + \beta(x_R - y_R)^2 + \gamma(x_G - y_G)^2 + \varepsilon(x_B - y_B)^2},$$

где x и y — цвета с компонентами $\{x_A, x_R, x_G, x_B\}$, $\{y_A, y_R, y_G, y_B\}$; $\alpha, \beta, \gamma, \varepsilon$ — параметры, обеспечивающие визуальную равномерность и подбираемые экспериментально; A определяет очень важную компоненту растра — альфа-канал или прозрачность; G и B — зеленая и синяя цветовые компоненты. Суммирование в строке данных кластеров происходит по формуле

$$S = \left[\sum_{k=1}^K A_k \left(\sum_{r=1}^R I_r \sum_{g=1}^G I_g \sum_{b=1}^B I_b \right) \right] m,$$

где m — число однотипных кластеров в строке; I — интенсивность составляющей пикселя.

Таблица 1

Входные данные	Циклы сортировки				
	Вставками	Шелла	Пирамидалная	Быстрая	Выбором
216 142 98 82 66 56 243 106 59 237	159	120	415	43	206
Растр 1 (100 элементов)	10 447	2 534	14 523	313	15 551
Растр 2 (1000 элементов)	987 283	40 058	755 557	3 016	1 505 501

Таблица 2

Идентичные значения матриц									
<i>Упорядоченный растр 1 из 100 элементов</i>									
0	0	13	15	25	28	35	37	43	48
49	52	55	56	57	59	62	64	65	65
66	67	74	77	78	80	82	84	94	94
102	104	105	110	112	113	114	115	115	117
122	126	126	128	129	129	135	136	140	141
142	143	148	151	153	155	163	168	169	172
174	176	177	181	183	183	189	189	190	192
193	196	197	199	202	204	204	206	210	215
215	217	217	220	221	221	221	222	227	230
232	233	234	234	240	242	243	243	247	253
<i>Упорядоченный растр 2 из 1000 элементов</i>									
0	0	0	0	1	1	1	2	2	2
2	2	2	2	3	3	3	4	4	5
5	5	5	5	5	5	6	6	6	7
7	7	7	7	7	8	8	8	9	9
9	9	10	10	10	11	11	12	12	13
13	14	14	14	14	14	14	15	15	16
16	16	16	16	17	17	18	18	18	18
18	19	19	19	19	21	21	21	21	21
21	22	22	22	23	23	23	23	23	23
23	24	24	24	24	25	25	25	26	26
27	27	27	27	28	28	28	28	28	28
29	29	29	29	29	30	30	31	31	31
31	32	32	32	33	33	33	33	33	33
34	34	34	34	34	34	34	35	35	35
36	36	36	36	37	37	37	37	37	38
38	39	39	39	40	40	40	41	41	41

Продолжение табл. 2

Идентичные значения матриц									
41	42	42	42	42	42	42	43	43	43
43	44	44	44	44	44	44	45	45	46
46	46	46	46	46	46	47	47	47	47
48	48	50	50	50	50	51	51	52	52
52	52	53	53	53	53	53	54	54	54
54	55	55	55	55	55	55	56	56	56
57	57	57	57	57	58	58	59	59	59
59	59	60	60	60	60	60	61	61	61
61	62	62	62	62	62	62	64	64	64
64	64	64	65	65	65	66	66	67	67
67	67	68	68	68	68	68	68	69	69
69	69	69	69	69	69	70	70	70	71
71	71	71	71	72	72	72	72	72	73
73	73	73	74	74	74	75	76	76	77
77	77	77	78	78	78	78	78	79	79
79	79	80	80	80	80	80	81	82	82
82	82	83	83	83	83	84	84	84	85
85	85	85	85	85	86	86	86	87	87
87	87	88	88	88	88	88	88	89	89
89	89	89	89	90	90	90	91	91	91
91	92	92	92	92	92	93	93	93	93
93	94	94	94	94	94	95	95	96	96
96	96	96	96	97	97	97	97	97	97
98	98	98	98	99	99	99	99	99	99
99	99	100	100	100	100	100	100	101	101
101	101	102	102	102	102	102	103	103	103
103	103	103	104	104	104	104	105	105	105
105	105	105	105	105	106	106	107	107	107
107	108	108	109	109	109	110	110	110	111
111	111	112	112	112	112	112	112	113	113
113	113	113	114	114	114	114	114	114	115
115	115	115	115	116	116	116	117	117	117
118	118	118	119	119	120	120	120	120	121
121	121	121	121	121	124	124	124	124	125
125	125	126	126	126	126	126	126	126	126
127	127	127	127	127	128	128	128	128	128
128	128	129	129	129	129	130	130	130	130
130	131	131	131	131	131	131	132	133	133
133	134	134	135	135	135	135	135	136	136
136	136	136	137	137	137	137	137	137	138
138	139	139	139	139	139	139	139	139	140
140	140	140	140	141	141	141	141	141	141
142	142	142	143	143	143	143	143	143	144

Идентичные значения матриц									
144	144	144	145	145	145	145	146	146	146
146	147	147	147	147	147	148	148	148	148
149	149	149	149	149	150	150	152	152	152
152	152	152	153	154	154	154	155	155	155
156	156	156	156	157	158	158	158	159	159
159	159	159	159	159	160	160	160	160	160
161	161	161	161	161	162	162	162	163	163
163	163	163	164	164	164	164	166	166	166
166	167	168	168	169	170	170	170	170	170
170	170	170	170	171	171	171	171	172	172
173	173	173	173	173	173	174	174	174	174
175	175	175	175	176	176	176	177	177	177
177	178	178	179	179	180	180	180	180	180
181	181	181	181	181	182	182	182	183	184
184	184	185	185	185	185	185	186	186	186
186	186	187	187	187	187	187	187	188	188
188	188	189	189	189	189	190	190	190	190
190	190	190	190	190	191	191	191	191	192
192	192	193	193	193	194	194	194	194	194
195	195	195	195	195	196	196	196	197	197
198	198	198	198	198	198	199	199	199	199
199	200	200	200	201	201	201	201	201	202
202	203	203	203	203	203	203	203	203	204
204	204	204	205	205	205	205	205	205	205
206	207	207	207	208	209	209	210	210	210
210	210	211	212	212	213	213	214	214	214
214	215	215	215	216	216	217	217	218	218
218	218	218	218	219	220	220	220	220	220
220	221	222	222	222	222	223	223	223	223
223	223	223	224	224	225	225	225	225	225
225	225	225	225	225	226	226	227	227	227
228	228	228	228	228	228	229	229	230	230
230	230	231	231	231	231	231	232	232	232
232	232	233	233	233	234	234	234	234	235
235	235	235	235	235	235	236	236	236	236
236	236	237	237	238	238	238	239	239	239
240	240	240	240	240	241	242	242	242	243
243	243	243	244	244	244	245	245	246	246
246	246	246	246	246	246	246	246	246	247
247	248	248	249	249	249	249	250	250	250
251	252	252	252	253	254	254	254	254	254

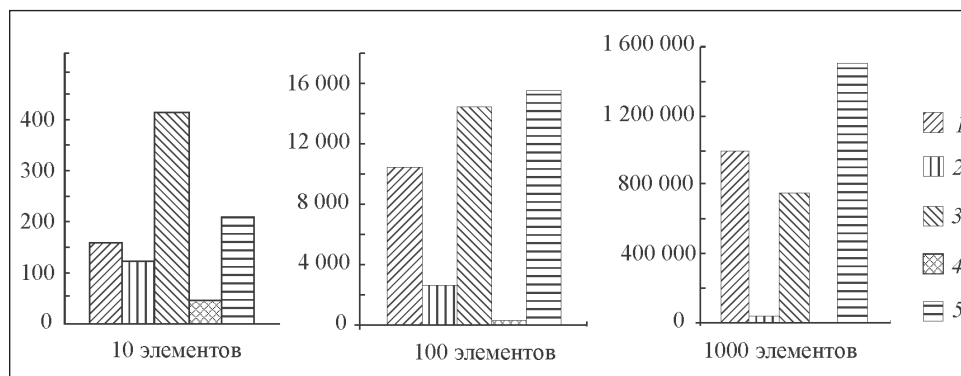


Рис. 5. Сравнительная диаграмма работы алгоритмов сортировки: 1 — вставками; 2 — Шелла; 3 — пирамидальная; 4 — быстрая; 5 — выбором

Сформированные массивы имеют диапазоны значений однородной интенсивности, из которых пользователь может выделить (указанием пикселя или вводом числа по индексной диаграмме) нужный для анализа тип. Программа запускает процедуру поиска и сбора всех пикселей выбранного растра в один блок [2]. В результате получаем количественную характеристику растра заданной интенсивности.

Дальнейшая группировка структур по строкам выполняется модулем сортировки с учетом их адресного расположения. Вариантов сортировки может быть несколько, поэтому задача исследователя выбрать наиболее эффективный из имеющихся в списке. Затем выбирается следующий растр, и процесс повторяется уже без указания пикселя. Число выбранных итераций, задание порога (допуска) устанавливаются в соответствии с поставленной задачей и уровнем точности, который необходимо обеспечить.

В качестве экспериментальной оценки данного подхода предлагаемые алгоритмы сортировки были применены для растровых объектов, заданных в градациях серого (с индексами 0-255). Для этого сформированы тестовые таблицы и проведена сортировка идентичных значений матриц (табл. 1, 2). Результаты эксперимента представлены на рис. 5.

Выводы

Визуальный контроль динамически изменяемых объектов на основе растровых снимков применяется в различных сферах. Это томограммы сканирования головного мозга, цветовые диаграммы распределения силовых полей, изменение спектров распределения температур, графики изменения свойств материалов и ряд других приложений. Качество таких изображений зависит не только от используемого оборудования, но и от

используемой технологии обработки данных. При обработке растров существует проблема длительности проведения расчетных операций, обусловленная дискретным разделением пикселей и наличием большого числа разнородных групп. Нужные элементы необходимо найти, выделить из общего списка адресатов, объединить в микро- или макрогруппы с себе подобными и провести итоговую калькуляцию. Поэтому перспективным развитием данной технологии будет систематизация растровых изображений по характеристикам на основе серии замеров в предметной области и создание квалитетрической шкалы для оптимизации процедур оценки растров в будущем.

The article offers the technology of quantitative analysis of raster changes by means of determining pixel characteristics and applying the sorting algorithm to them.

1. *Мониторинг* — энциклопедия [Электронные данные] / Режим доступа — <http://dic.academic.ru>
2. *Елизаров А. И., Есипова В. А., Калайда В. Т., Климкин В. М.* Кластеризация филиграней по численным характеристикам изображений (проект № 00-06-80460). — Томск : изд. Томского ун-та, 2007. — С. 223—234.
3. *Башков Е. А., Вовк О. Л.* Статистическая кластеризация для выделения регионов изображений // *Мат. V междунар. конф. «Интеллектуальный анализ информации ИАИ-2005»*. — Киев : НТТУ «КПИ», 2005. — С. 50—59.
4. *Методы кластеризации («Машинное обучение (курс лекций, К.В.Воронцов)»)* [Электронные данные] / Режим доступа — <http://www.MachineLearning.ru/wiki>
5. *Кластеризация палитры изображения и сжатие в формате PNG* [Электронные данные] / Режим доступа — <http://habrahabr.ru>
6. *Кластерный анализ* — Википедия [Электронные данные] / Режим доступа — <http://ru.wikipedia.org>
7. *Вовк О. Л.* Особенности контекстного поиска кластеризованных изображений // *Мат. VII междунар. конф. «Интеллектуальный анализ информации ИАИ-2007»*. — Киев : НТТУ «КПИ», 2007. — С. 22—31.
8. *Метод группового кодирования RLE* [Электронные данные] / Режим доступа — <http://www.studfiles.ru>
9. *Метод сдвиговых регистров* [Электронные данные] / Режим доступа — <http://caitocroy.ru>
10. *Кнут Д.* Сортировка и поиск: Искусство программирования на ЭВМ. Т.3. — М. : Наука, 1981.
11. *Сортировка вставками* [Электронные данные] / Режим доступа — <http://algotlist.manual.ru>
12. *Сортировка Шелла* [Электронные данные] / Режим доступа — <http://www.codenet.ru>
13. *Пирамидальная сортировка* [Электронные данные] / Режим доступа — <http://algotlist.manual.ru>
14. *Быстрая сортировка (Структуры и алгоритмы)* [Электронные данные] / Режим доступа — <http://www.structur.h1.ru>
15. *Сортировка выбором, общий подход* [Электронные данные] / Режим доступа — <http://www.codelab.ru>
16. *Кантор И.* Алгоритмы сортировки [Электронные данные] / Режим доступа — <http://algotlist.manual.ru>

17. Сундукова Т. О., Ванькина Г. В. Структуры и алгоритмы компьютерной обработки данных [Электронные данные] / Режим доступа — <http://www.intuit.ru>
18. Роджерс Д. Алгоритмические основы машинной графики — М.: Мир, 1989. — 512 с.
19. Байгарова Н. С., Бухитаб Ю. А., Евтеева Н. Н., Корягин Д. А. Некоторые подходы к организации содержательного поиска изображений и видеоинформации.— М. : ИПМ им. М. В. Келдыша РАН, 2002 (проект № 01-01-00267).
20. Недзьведь А. М., Белоцерковский А. М., Абламейко С. В., Рязанов В. В. Вычисление объемных характеристик изображений компьютерной томографии //Искусственный интеллект. — 2009. — № 3. — С. 582—589.

Поступила 04.11.11;
после доработки 22.03.12

БАШКАТОВ Александр Майорович, канд. техн. наук, доцент кафедры программного обеспечения вычислительной техники и автоматизированных систем Инженерно-технического ин-та Приднестровского госуниверситета им. Т. Г. Шевченко. В 1986 г. окончил Днепропетровский госуниверситет. Область научных исследований — оценка экологической безопасности воздушной среды города методами статистического и графического моделирования.

СОЛОВЦОВ Алексей Олегович, студент Инженерно-технического ин-та Приднестровского госуниверситета им. Т. Г. Шевченко. Область научных исследований — компьютерная графика, разработка WEB-приложений.

