

СРАВНИТЕЛЬНЫЙ АНАЛИЗ БЫСТРЫХ АЛГОРИТМОВ СЛИЯНИЯ ДЛЯ ИТЕРАТИВНОЙ ПОЛИГОНАЛЬНОЙ АППРОКСИМАЦИИ КОНТУРНЫХ ЦЕПЕЙ ПРИ РАЗЛИЧНЫХ КРИТЕРИЯХ ОПТИМИЗАЦИИ

Р.В. Хмелев hmelyoff@smr.ru

Институт систем обработки изображений РАН,

Самарский государственный аэрокосмический университет имени академика С.П. Королева

Аннотация

В статье исследуются свойства быстрых алгоритмов слияния, применяющихся для итеративной полигональной аппроксимации контурных цепей и различающихся критериями оптимизации. Исследуемые критерии оптимизации – максимум периметра, минимум максимальной и среднеквадратической погрешности. Алгоритмы сравниваются с точки зрения предпочтительности для локализации углов, по численным критериям (длине периметра, максимальной и среднеквадратической погрешности) и вычислительной сложности.

Введение

Алгоритмы полигональной аппроксимации контурных цепей традиционно используются для упрощенного представления контуров и анализа их формы. Цель полигональной аппроксимации в том, чтобы получить представление контурной цепи в виде многоугольника с числом вершин меньшим, чем число изломов цепи. Используются различные критерии оптимизации аппроксимаций, самые популярные – минимум среднеквадратической или максимальной погрешности. Одной из задач полигональной аппроксимации является возможно более точная локализация выраженных углов контура, однако это не всегда удается – алгоритмы часто «промахиваются».

Среди многообразия подходов к полигональной аппроксимации существует класс *итеративных алгоритмов слияния* [1-9], общим принципом которых является то, что на каждой итерации узлы контурной цепи исключаются из аппроксимации таким образом, чтобы происходило минимальное ухудшение выбранного критерия оптимизации. Хотя алгоритмы слияния не всегда дают аппроксимации, оптимальные с точки зрения выбранного критерия, они вычислительно наиболее просты и способны локализовать углы контура, что делает их перспективными для применения в задачах реального времени. Ранее самый быстрый алгоритм слияния со сложностью $O(N \log N)$ (где N – число изломов контурной цепи) был описан в [8]: на каждой итерации удаляется узел с минимальным расстоянием до стягивающей соседние узлы хорды, чтобы как можно меньше ухудшить максимальную погрешность аппроксимации (*минимаксный алгоритм*). Данный алгоритм обычно дает неплохие аппроксимации (с точки зрения субъективной оценки соответствия формы контура и аппроксимирующего многоугольника) и способен успешно локализовать выраженные углы.

В [1] для алгоритмов слияния в качестве критерия оптимизации был впервые предложен максимум периметра аппроксимирующего многоугольника, также были описаны два быстрых алгоритма слияния на его основе. Один из них со сложностью $O(N \log N)$ повторяет минимаксный алгоритм [8] с той лишь разницей, что на каждой итерации удаляется узел с минималь-

ным неравенством треугольника относительно соседних узлов (его удаление приводит к минимальному уменьшению периметра). Сложность $O(N \log N)$ достигается за счет того, что узлы хранятся в виде итеративно обновляемых бинарных деревьев, упорядоченных по величине неравенства треугольников.

Второй алгоритм является еще более быстрым ($O(N) - O(N \log N)$) и состоит из двух этапов. При выполнении первого этапа (сложность $O(N)$) на каждой итерации удаляется сразу все локальные минимумы неравенства треугольников, пока не останутся два узла. На втором этапе формируется очередность добавления узлов в аппроксимацию, делается это посредством сортировки узлов по убыванию максимальной величины неравенства треугольников, достигавшейся в них на какой либо итерации (сложность $O(N \log N)$). Хотя порядок сложности сортировки такой же, как у бинарных деревьев, сортировка вычислительно проще.

В [1] эти алгоритмы назывались «алгоритм итеративного удаления» и «алгоритм удаления локальных минимумов» соответственно, в данной работе было решено привести названия алгоритмов к общепринятому для данного класса виду и подчеркнуть способ выбора удаляемых узлов – здесь они называются «алгоритм треугольного слияния» и «алгоритм треугольного слияния по локальным минимумам» соответственно.

Помимо быстрых алгоритмов треугольного слияния, в [1] был описан принцип максимизации периметра аппроксимирующего многоугольника и рекурсивный алгоритм $O(N^4)$, который для любого заданного контура строит аппроксимации максимально возможного периметра для любого количества аппроксимирующих узлов.

Целью данной статьи является сравнение трех критериев оптимизации (максимума периметра, минимума максимальной и среднеквадратической погрешности) в алгоритмах, одинаковых с точностью до критерия оптимизации. Алгоритмы сравниваются с точки зрения предпочтительности для локализации углов, по численным критериям (длине периметра, максимальной и среднеквадратической погрешности) и вычислительной сложности.

1. Сравнение критериев оптимизации в алгоритмах сложности $O(N^4)$

Проанализируем, чем отличаются аппроксимации по разным критериям оптимизации и покажем, что алгоритмы, основанные на критерии максимума периметра, принципиально более ориентированы на попадание в углы контура, чем алгоритмы, основанные на критериях минимума погрешности. Рассмотрим пример на рис. 1а – пусть дан бочкообразный контур, у которого хорошо заметны четыре угла, аппроксимируем его четырехугольником при условии, что узлы аппроксимации лежат на контуре.

Как видно на рис. 1б,в, узлы аппроксимаций, оптимальных по минимумам погрешности, могут не совпадать ни с одним углом аппроксимируемого многоугольника, в то время как узлы аппроксимаций, оптимальных по максимуму периметра (рис. 1г), по лемме о максимуме периметра аппроксимирующего многоугольника (доказательство которой приведено в [1]) всегда совпадают с углами аппроксимируемого многоугольника, хотя и не всегда с самыми заметными с точки зрения человека.

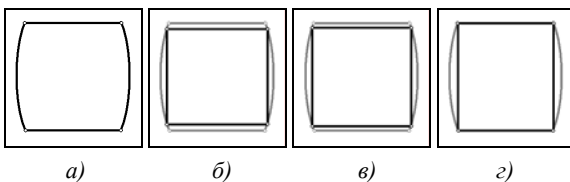


Рис. 1. а) Исходный бочкообразный контур. б,в) Аппроксимации четырехугольником, оптимальные по критерию минимума максимального и среднеквадратического отклонения соответственно. г) Аппроксимация, оптимальная по критерию максимума периметра.

Алгоритмы аппроксимации работают с контурными цепями, точки которых имеют целочисленные координаты. Во всех известных автору алгоритмах узлы аппроксимации являются подмножеством точек контурной цепи, и, следовательно, также имеют целочисленные координаты, однако оптимальные аппроксимации по любому из критериев минимума погрешности могут иметь нецелочисленные координаты. Т.е., даже самый вычислительно тяжелый алгоритм четвертой степени, аналогичный с точностью до критерия оптимизации рекурсивному алгоритму максимального периметра [1] (перебирающий в качестве узлов аппроксимации все точки контура), может не найти оптимальной аппроксимации по минимуму погрешности. По критерию же максимального периметра истинный максимум можно найти всегда, поскольку для этого достаточно перебрать точки изломов контура.

Итого: для контурных цепей наиболее точные и вычислительно тяжелые алгоритмы оптимизации по какому-либо критерию минимума погрешности могут не попадать в углы и не находить истинный минимум погрешности.

2. Сравнение пригодности быстрых алгоритмов для локализации углов

Поскольку алгоритмы сложности $O(N^4)$ слишком тяжелы вычислительно и на практике обычно не используются, сравним неоптимальные быстрые алгоритмы, которые одинаковы с точностью до критерия оптимизации. Из ранее описанных алгоритмов будем использовать алгоритмы треугольного слияния [1] и алгоритм минимаксного слияния [8]. Из ранее не описывавшихся будем использовать модификации [1]: алгоритм минимаксного слияния по локальным минимумам и два алгоритма среднеквадратического слияния (в которых удаляемые узлы выбираются таким образом, чтобы минимально увеличивалась среднеквадратическая погрешность).

Сравним критерии максимума периметра и минимума максимальной погрешности. Каждому узлу аппроксимации присваивается некоторое значение d , причем каждому значению d соответствует множество вариантов положения узла: для критерия минимакса погрешности множество эквивалентных узлов – овал (рис. 2а, кривая L_p), для критерия максимума периметра – эллипс (рис. 2б, кривая L_Δ). На рис. 2б А и В – фокусы эллипса, для любой точки С, лежащей на L_Δ : $AC+BC=\text{const}$.

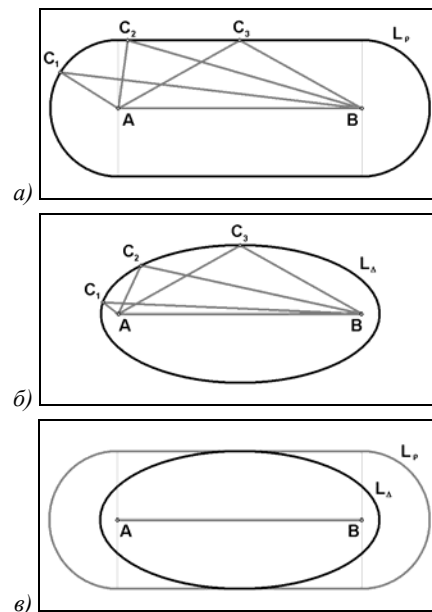


Рис. 2. Примеры кривых равных кандидатов на удаление по различным критериям. а,б) L_p и L_Δ – кривые равных кандидатов на удаления по критерию расстояния ($\rho(C, AB)=\text{const}$) и неравенства треугольника ($AC+BC-AB=\text{const}$) соответственно. в) При совмещении рисунков а) и б) видно различие между кривыми L_p и L_Δ .

Т.о., это значение d уравнивает в качестве кандидатов на удаление множество разных углов – острых и тупых, с различными длинами сторон, т.е. углов с разной степенью заметности. Формального определения заметности угла не существует, но интуитивно понятно, что заметность угла напрямую зависит от его остроты и от расстояния до ограни-

чивающего его отрезка (на рис. 2 $\rho(C, AB)$). Почему именно от $\rho(C, AB)$, а не от длины сторон или высоты? Угол, близкий к π , может иметь длинные стороны и быть при этом практически незаметным, острый и хорошо заметный угол (например, $\angle AC_1B$ на рис. 2а), может иметь малую высоту.

На рис. 2а заметно, что постоянное значение $\rho(C, AB)$ соответствует углам с очень разной остротой, в то время как на рис. 2б постоянное значение неравенства треугольников соответствует тупым углам с большим $\rho(C, AB)$ ($\angle AC_3B$) и острым углам с маленьким $\rho(C, AB)$ ($\angle AC_1B$), т.е. в неравенстве треугольников характеристики заметности равноценных кандидатов на удаление *лучше сбалансированы* – сильное значение одной характеристики заметности соответствует слабому значению другой и наоборот. На рис. 3 также приведены примеры того, что постоянные значения $\rho(C, AB)$ соответствуют углам с очень разной остротой, в то время как неравенство треугольника соответствует остроте гораздо лучше.

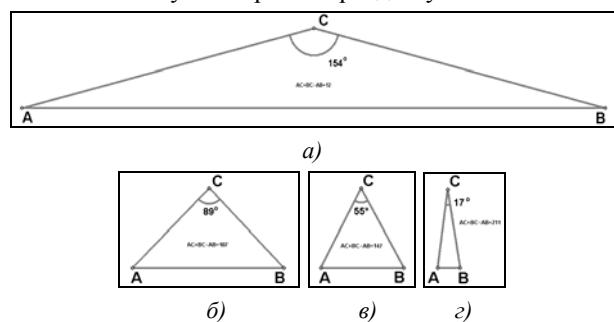


Рис. 3. Примеры того, как расстояние $\rho(C, AB)$ (на всех четырех картинках равно 120) уравнивает в качестве кандидатов на удаление узлы с существенно разной остротой и важностью, выражаемой через неравенство треугольника.

Исходя из этих свойств, можно привести пример реального контура, при аппроксимации которого быстрые алгоритмы минимаксного слияния удалят углы раньше (рис. 5б₂, б₅), чем алгоритмы треугольного слияния (рис. 5б₁, б₄).

Главным негативным свойством среднеквадратических алгоритмов слияния являются грубые ошибки локализации скругленных углов. Характерный вид контурной цепи в районе скругленного угла представлен на рис. 4а. Из трех узлов В, С и D с точки зрения минимаксного и треугольного алгоритмов предпочтительнее удалить узел D, с точки зрения среднеквадратического – узел В, поскольку при этом достигается минимальное увеличение суммы квадратов отклонений (во фрагменте контурной цепи ABC меньше точек, чем во фрагментах BCD и CDE). Из-за этого быстрые среднеквадратические алгоритмы слияния часто проигрывают треугольным и минимаксным при локализации скругленных углов (рис. 4б – аппроксимация треугольным и минимаксным алгоритмами, рис. 4в – аппроксимация среднеквадратическим алгоритмом).

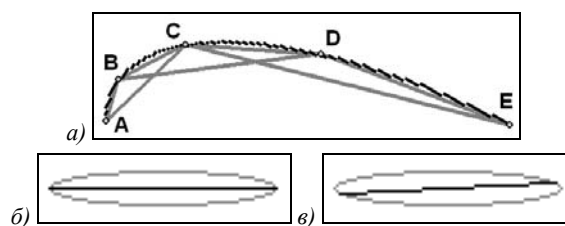


Рис. 4. Иллюстрации к свойствам среднеквадратических алгоритмов слияния.

Примеры того, что среднеквадратические алгоритмы слияния дают самые грубые ошибки локализации углов можно увидеть на рис. 5в₃, в₆, д₃, д₆, е₃, е₆. Узлы, совпадающие с углами, они также могут удалять в первую очередь по сравнению с другими быстрыми алгоритмами (рис. 5а₃, а₆, б₃, б₆).

3. Сравнение по числовым критериям

Каждый из быстрых алгоритмов решает в первую очередь свою задачу оптимизации – максимизацию периметра, минимизацию максимальной или среднеквадратической погрешности. Эти задачи не тождественны, но связаны – оптимизация по любому из этих критериев приводит в качестве побочного эффекта также к некоторой оптимизации и по остальным критериям. Интересно выяснить, насколько хорошо каждый алгоритм справляется с задачами оптимизации по сравнению с остальными алгоритмами.

Было проведено сравнение быстрых алгоритмов слияния по трем численным критериям – 1) периметр аппроксимации, 2) максимальная и 3) среднеквадратическая погрешность аппроксимации. Графики для представления результатов сравнения подходят плохо, поскольку, с одной стороны, велик диапазон сравниваемых значений, с другой стороны – во многих случаях различия между алгоритмами очень незначительные (сотые и тысячные доли величины диапазона), вследствие чего графики будут сливаться. Поэтому основным результатам сравнения являются таблицы попарной сравнительной статистики выигравшей и проигравшей.

О чем говорит эта сравнительная статистика? В алгоритмах слияния на каждой итерации после удаления узлов создаются *новые условия для следующих итераций*, и если алгоритм показывает хорошую статистику выигравшей в своем родном критерии оптимизации по сравнению с другими алгоритмами, это значит, что метод выбора удаляемых узлов *хорош с точки зрения создания благоприятных условий для следующих итераций*. Если же алгоритм часто проигрывает по родному критерию «чужакам», это значит, что данный критерий оптимизации *принципиально плох* для использования в итеративных алгоритмах слияния – удаления, приводящие на начальных итерациях к минимальному ухудшению критерия оптимизации, оборачиваются большим числом проигравшей на следующих итерациях.

Далее используются следующие обозначения для алгоритмов:

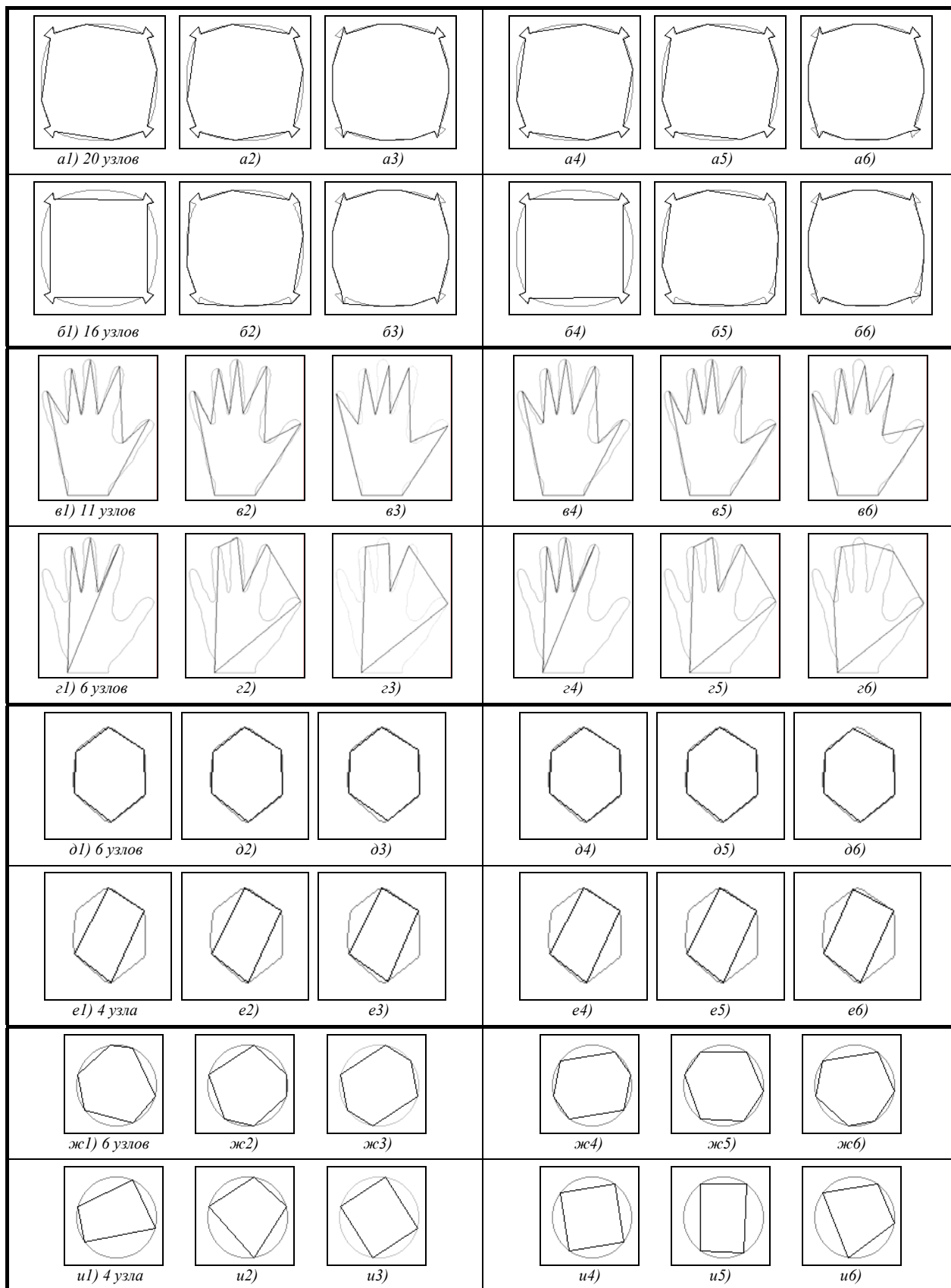


Рис. 5. Примеры применения быстрых алгоритмов итеративной аппроксимации. 1,2,3 – алгоритмы слияния по глобальным минимумам, 4,5,6 – алгоритмы слияния по локальным минимумам (1,4 – треугольный, 2,5 – минимаксный, 3,6 – среднеквадратический).

PerimGlobMin – треугольное слияние [1];
 PerimLocMin – треуг. слияние по лок. мин-ам [1];
 MaxDevGlobMin – минимаксное слияние [8];
 MaxDevLocMin – минимакс. слияние по лок. мин-ам;
 RmsDevGlobMin – среднеквадратическое слияние;
 RmsDevLocMin – с.к. слияние по лок. мин-ам.

Общепринятого набора изображений для тестирования алгоритмов полигональной аппроксимации не существует (в каждой работе используется свой собственный набор тестовых изображений [1-9]), в данной работе также проводилось сравнительное тестирования на собственном наборе контуров, вид которых охватывает разные случаи жизни (степень изломанности, криволинейные фрагменты, наличие или отсутствие явных углов, рис. 6). Для каждого контура строились все возможные аппроксимации с числом узлов от 2 до $N-1$ (где N – число изломов контурной цепи) и вычислялись значения численных критериев (периметр, максимальная и с.к. погрешность). Итоговая сравнительная статистика представлена в таблице 1.

Согласно итоговой статистике, каждый из быстрых алгоритмов чаще выигрывает у чужаков на своем поле, чем проигрывает, однако статистика выигрышей для треугольных алгоритмов (таблица 1а) лучше (75-90%), чем у алгоритмов минимизации погрешно-

сти. Кроме того, треугольный алгоритм слияния (PerimGlobMin) показывает хорошую статистику выигрышей по «неродному» среднеквадратическому критерию (43-45%) в сравнении со среднеквадратическими алгоритмами (53-55%) (таблица 1в). Самый быстрый треугольный алгоритм слияния по локальным минимумам (PerimLocMin) немного выигрывает по родному критерию у треугольного алгоритма слияния (PerimGlobMin), но у него гораздо хуже статистика выигрышей по неродным критериям.

В критерии максимальной погрешности наиболее высок процент совпадающих результатов (таблица 1б), что неудивительно, ибо величина этого критерия зависит от одного фрагмента контура и стягивающей его хорды. Минимаксные алгоритмы сравнительно редко проигрывают чужакам на своем поле (6-15%). Треугольный алгоритм слияния показывает 33-34% совпадений с минимаксными алгоритмами по их родному критерию.

Среднеквадратические алгоритмы показывают самую худшую статистику проигрышей «чужакам на своем поле» (18-45%, таблица 1в). Из этого следует, что с.к. критерий наименее хорош для алгоритмов слияния с точки зрения создания благоприятных условий для следующих итераций.

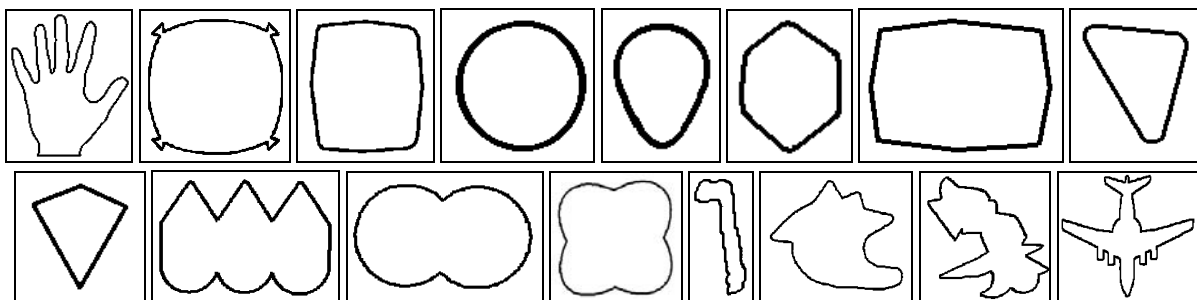


Рис. 6. Тестовые изображения для сравнения алгоритмов по критериям периметра и погрешностей аппроксимации.

Таблица 1. Итоговая статистика сравнения быстрых алгоритмов для различных критериев. Серым выделены клетки, в которых для одного из алгоритмов численный критерий является «родным». Первое из трех чисел означает процент выигрышей алгоритма, имя которого пишется в левой части таблицы; второе – процент совпадающих результатов; третье – процент выигрышей алгоритма, имя которого пишется сверху столбца.

	а) Итоговая статистика выигрышей по длине периметра				
	PerimLocMin	MaxDevGlobMin	MaxDevLocMin	RmsDevGlobMin	RmsDevLocMin
PerimGlobMin	39,69-4,01-56,28	79,87-3,02-17,10	75,82-1,96-22,21	90,54-1,89-7,55	89,84-0,86-9,28
PerimLocMin	—	81,71-1,47-16,81	80,07-1,28-18,64	91,93-0,83-7,23	92,44-0,64-6,91
MaxDevGlobMin	—	—	25,94-23,43-50,62	71,03-13,11-15,84	75,31-5,40-19,28
MaxDevLocMin	—	—	—	78,10-8,67-13,21	81,03-9,67-9,28
RmsDevGlobMin	—	—	—	—	52,20-3,31-44,48
	б) Итоговая статистика выигрышей по максимальной погрешности				
	PerimLocMin	MaxDevGlobMin	MaxDevLocMin	RmsDevGlobMin	RmsDevLocMin
PerimGlobMin	66,31-18,32-15,36	12,34-34,71-52,94	11,21-33,55-55,22	20,09-32,49-47,41	28,09-23,72-48,18
PerimLocMin	—	6,87-6,42-86,69	6,23-7,32-86,43	10,25-4,78-84,95	12,24-4,69-83,06
MaxDevGlobMin	—	—	9,57-72,13-18,28	41,62-44,71-13,66	49,53-34,94-15,52
MaxDevLocMin	—	—	—	45,22-40,53-14,23	53,19-33,20-13,59
RmsDevGlobMin	—	—	—	—	23,27-58,88-17,83
	в) Итоговая статистика выигрышей по с.к. погрешности				
	PerimLocMin	MaxDevGlobMin	MaxDevLocMin	RmsDevGlobMin	RmsDevLocMin
PerimGlobMin	75,37-3,82-20,79	48,76-3,08-48,15	47,09-1,99-50,91	45,00-2,02-52,97	43,71-0,73-55,54
PerimLocMin	—	20,12-1,41-78,46	19,15-1,22-79,62	18,73-0,77-80,48	19,47-0,67-79,84
MaxDevGlobMin	—	—	40,53-23,40-36,06	25,55-12,98-61,45	34,20-6,36-59,43
MaxDevLocMin	—	—	—	31,98-8,67-59,33	33,75-10,02-56,21
RmsDevGlobMin	—	—	—	—	50,91-3,43-45,64

4. Сравнение вычислительной сложности

Было проведено сравнение алгоритмов по скорости вычислений (на машине Athlon XP 2500+). Сложность вычисления численного критерия для одного узла минимальна у треугольных алгоритмов, чуть выше – у минимаксных, и самая высокая – у среднеквадратических. Для всех контуров (рис. 6) было найдено суммарное время построения последовательности добавления узлов в аппроксимацию. Наиболее быстрым вполне ожидаемо оказался алгоритм треугольного слияния по локальным минимумам, время вычисления для остальных алгоритмов приведено относительно него в процентах:

PerimLocMin	100,0 _{min}	PerimGlobMin	363,6
MaxDevLocMin	157,8	MaxDevGlobMin	413,8
RmsDevLocMin	304,6	RmsDevGlobMin	592,1

5. Заключение

Итоговые выводы, основанные на визуальном сравнении аппроксимаций, а также на сравнении алгоритмов по численным критериям и скорости вычислений, следующие:

1. Среднеквадратические быстрые алгоритмы слияния являются наименее предпочтительными во всех отношениях: с точки зрения локализации углов, проигрышей по родному числовому критерию и по скорости вычислений.
2. Треугольные алгоритмы слияния предпочтительнее по сравнению с минимаксными с точки зрения локализации углов и скорости вычислений. С точки зрения численных критериев сложно отдать предпочтение одному из двух подходов: у треугольных алгоритмов лучше статистика выигрышей, у минимаксных – проигрышей чужакам на своем поле.

Благодарности

Работа выполнена при поддержке российско-американской программы «Фундаментальные исследования и высшее образование» (BRHE), грантов Российского фонда фундаментальных исследований № № 04-7-90149, 05-01-08043, 06-08-01024.

Литература

1. Р.В. Хмелев. Итеративная аппроксимация последовательностей по максимуму периметра и с использованием неравенства треугольника. // *Компьютерная оптика* № 27, 2005, с. 155-164.
2. Leu J.G., Chen L., Polygonal approximation of 2d shapes through boundary merging, *Pattern Recognition Letters*, 7(4): 231-238, 1988.
3. Ku K.M., Chiu K.M., Polygonal approximation of digital curve by graduate iterative merging, *Electronics Letters*, 31: 444-446, 1995.
4. Fahn C.-S., Wang J.-F., Lee J.-Y., An adaptive reduction procedure for the piecewise linear approximation of digitized curves, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11: 967-973, 1989.
5. Wu J.-S., Leou J.-J., New polygonal approximation schemes for object shape representation, *Pattern Recognition*, 26: 471-484, 1993.
6. Boxer L., Chang C.-S., Miller R., Rau-Chaplin A., Polygonal approximation by boundary reduction, *Pattern Recognition Letters*, 14: 111-119, 1993.
7. Visvalingam M., Whyatt J., Line generalization by repeated elimination of points, *Cartographic Journal*, 30(1): 46-51, 1993.
8. Pikaz A., Dinstein I., An algorithm for polygonal approximation based on iterative point elimination, *Pattern Recognition Letters*, 16 (6): 557-563, 1995.
9. Horst J.A., Beichl I., A simple algorithm for efficient piecewise linear approximation of space curves, *Proc. Int. Conf. Image Processing-ICIP'97*, 2: 744-747, 1997.