

РЕДУКЦИЯ СТРУКТУРНЫХ ЭЛЕМЕНТОВ ИЗОБРАЖЕНИЯ НА ДИСКРЕТНЫЙ РАСТР

Лепихов Ю.Н.¹, Лялин В.Е.², Григорьев Е.В.¹

¹Тулский филиал ОАО «Центр Телеком»

²ГОУ ВПО «Ижевский государственный технический университет»

На интуитивном уровне оператор редукции линейных структурных элементов и границ планарных элементов представляется достаточно простым образом – при прохождении линий через ячейку раstra для каждой точки линии в этой ячейке используется оператор точечной редукции $R_0(\tilde{x}, \tilde{y})$ (1). Таким образом, наличие линий в ячейке выделяет соответствующий пиксел раstra, а вся линия представляется связной совокупностью пикселов.

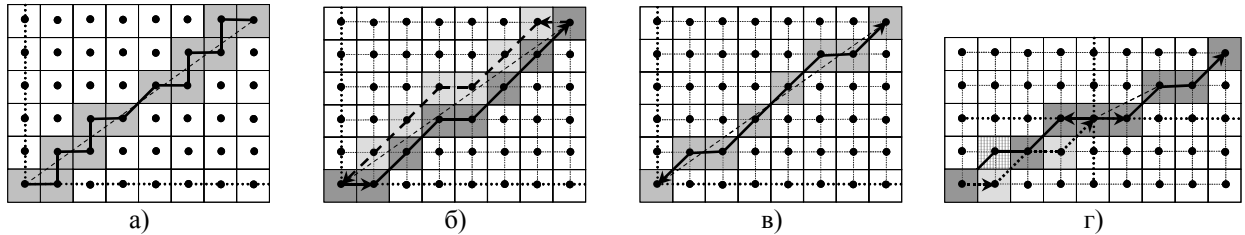


Рис. 1. Линейная редукция на основе: (а) точечной редукции, (б) 1-го алгоритма Брезенхама, (в) 2-го алгоритма Брезенхама, (г) 2-го алгоритма Брезенхама в области начала координат

Однако, как видно из рис.1.а, формируемый дискретизированный отрезок прямой линии имеет нерегулярный угловатый вид и визуально воспринимается как ломаная линия. Причина этого в том, что выделяются пикселы, в ячейках которых могут присутствовать слишком малые участки линии; при этом преобладают вертикальные и горизонтальные переходы на растре, которые и придают линии угловатый вид. Если исключить эти переходы (в этом случае связность обеспечивается за счет диагональных переходов на растре), то визуальный вид линии приобретает гораздо большее сходство с отрезком прямой.

Идея такого построения реализована в алгоритмах Брезенхама [2], в которых используются только целочисленные вычисления за счет многократного масштабирования фрагмента раstra, а также ведется подсчет и минимизация инкремента (невязки) текущей точки линии относительно узлов раstra. В зависимости от способа подсчета инкремента различаются два алгоритма Брезенхама. Первый алгоритм учитывает инкремент таким же способом, как и оператор точечной редукции $R(\tilde{x}, \tilde{y}) = ([\tilde{x}], [\tilde{y}])$ (2), как показано на рис.1.б. При этом возникает эффект, аналогичный искажениям при использовании оператора точечной редукции (1).

Как видно из рисунка для линейных элементов эффект искажения линий проявляется в несовпадении дискретных представлений отрезков прямых, прорисовываемых в прямом и обратном направлениях. Естественно, что более приемлемым вариантом подсчета инкремента является схема, соответствующая оператору точечной редукции $R(\tilde{x}, \tilde{y}) = ([\tilde{x} + 0.5\text{sign}(\tilde{x})], [\tilde{y} + 0.5\text{sign}(\tilde{y})])$ (3), результат воздействия которого показан на рис.1.в.

Здесь прорисовка отрезка в прямом и обратном направлениях дает идентичные результаты и формируемое дискретное представление отрезка симметрично относительно его центра. Однако, также как и при использовании оператора точечной редукции (3), могут возникнуть искажения в области начала координат. Это обусловлено тем, что формирование связной цепочки пикселов реализуется по восьми координатным секторам, ограничиваемым координатными осями и биссектрисами (диагоналями) квадрантных углов (см. рис.2.а) и для каждого сектора эта цепочка представляет собой последовательность диагональных и осевых переходов с идентичным порядком их чередования.

На рис.1.г показаны (сплошной линией) такие переходы для прямой линии, являющейся продолжением отрезка с начальной точкой в начале координат и конечной точкой с координатами (2,1). Из рисунка видно, что в окрестности начала координат наблюдается заметное искажение линии. Устранить указанное искажение можно, если перейти к однонаправленной системе секторов, как показано на рис.2.б. В этом случае любой отрезок отрисовывается таким образом, что-либо его начальная точка имеет значение ординаты меньше значения ординаты конечной точки, либо, если эти значения координат равны, начальная точка имеет значение абсциссы, меньше значения абсциссы конечной точки.

При этом результат действия такого линейного оператора редукции сводится к инверсии начальной и конечной точек отрезка, если они не удовлетворяют указанным выше условиям, и становится эквивалентным действию точечного оператора редукции (1). Видоизменение дискретного представления отрезка показана на рис.1.г в виде дополнительных окрашенных в более светлый тон пикселей и заштрихованного удаляемого пиксела.

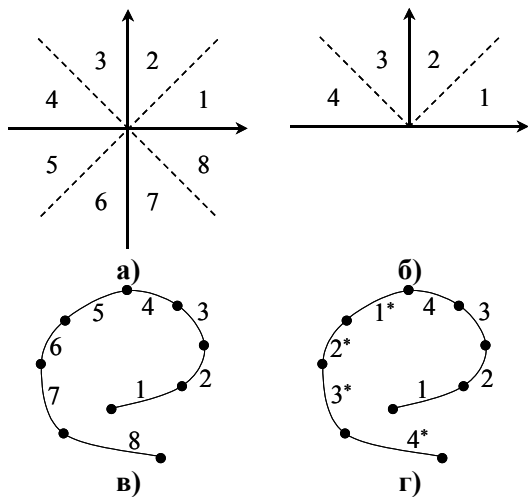


Рис.2. Индексация секторов направлений (а, б) и сегментация кривой по секторам направлений (* - инверсия направления) (в, г): (а, в) двунаправленная, (б, г) однонаправленная

Последняя модификация алгоритма Брезенхама легко обобщается на оператор линейной редукции произвольной кривой на дискретный растр так, как показано на рис.2.в и 2.г. При этом необходимо сегментировать кривую на участки, для которых текущие направления целиком находятся внутри одного из координатных секторов направлений (рис.2.в). Если некоторые сегменты кривой не удовлетворяют указанным выше условиям, то осуществляется их инверсия (на рисунке отмечена звездочкой).

При этом формируется однонаправленная сегментация кривой по секторам направлений (рис.2.г) и в каждом секторе общее количество осевых и диагональных переходов определяется размером сегмента по направлению соответствующей оси.

Линейные операторы редукции на дискретный растр самым непосредственным образом связаны с цепными кодами. Простой цепной код впервые предложил Фримен [3, 4]. Идея простого цепного кода заключается в том, что для любой связной линии на растре кодируются координаты начального пиксела, а для каждого последующего пиксела цепочки в качестве кода используется его приращение, определяющее переход на один из смежных пикселей (рис.3.а). Поскольку таких смежных пикселей всего 8, то для кодирования каждого пиксела необходимо 3 бита информации. Таким образом, оценка информационной емкости линий на растре – 3 бита на элемент, без учета служебной информации (координат начальной точки и количества пикселей в цепочке, либо признака окончания цепочки). Так для линии на рис.1.г простой цепной код будет представлять собой последовательность вида 02020102020 – всего 33 бита. К недостаткам простого цепного кода можно отнести допустимость возвратных переходов и переходов второго порядка в смежные пиксели, что приводит к эффекту образования пятен на линиях.

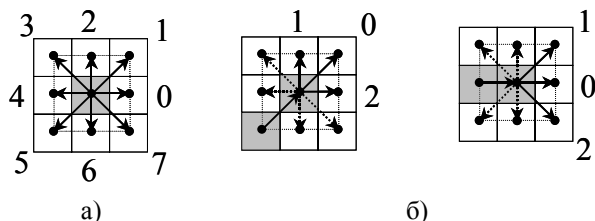


Рис.3. Простой (а) и дифференциальный (б) цепные коды

Более гибкая схема предусматривается дифференциальным цепным кодом (ДЦК) [1], в котором кодируются координаты начального пиксела, следующий пиксел кодируется простым цепным кодом, а для всех последующих пикселей кодируются не приращения, а изменения направления переходов на растре (см. рис.3.б). Если в этой схеме исключить возвратные переходы и переходы второго порядка в смежные пиксели, а также ортогональные переходы (точки,

соответствующие таким пикселям являются особыми концевыми узлами линейных сегментов типа «излом», которые сегментируют линии на отдельные гладкие сегменты), то для кодирования каждого пиксела необходимо три состояния (1.5 бита на элемент, 12 бит для линий на рис.1.г).

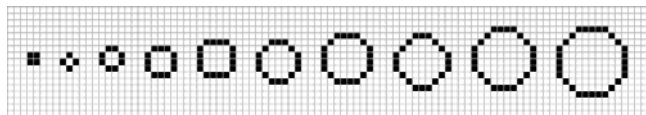


Рис.4. Окружности различного радиуса на растре

При рассмотрении ДЦК важным является вопрос о возможной максимальной кривизне таких структурных элементов изображения, как дуги и линейные сегменты переменной кривизны (ЛСПК). С учетом выказанного замечания о недопустимости в

- для окружностей чрезмерно малого радиуса (первые две окружности на рис.4) переходы в цепном коде являются ортогональными, что не допускает возможность использования корректно построенных ДЦК; однако, такие структурные образования визуально не воспринимаются как окружности, и, по сути, ничем не отличаются от квадратов минимального размера;
- для окружностей минимального радиуса (третья и четвертая окружности на рис.4) переходы в цепном коде уже не являются ортогональными, что делает возможным использование корректно построенных ДЦК; при этом, такие структурные образования визуально воспринимаются как окружности и по всем своим характеристикам ничем не отличаются от них;
- для окружностей радиуса, большего, чем минимальный радиус (пятая и последующие окружности на

рис.4), их описание полностью соответствует корректно построенным ДЦК.

Таким образом, представление обладающих кривизной линейных структурных элементов графических изображений на растре, таких как дуги и ЛСПК, полностью отвечает условиям корректности ДЦК, используемых для их описания.

Если сопоставить дифференциальный цепной код с теми цепочками пикселей, которые формируются в результате применения линейных операторов редукции Брезенхама, то нетрудно заметить, что для этих операторов возможны только два типа переходов (диагональные и осевые) в каждом из координатных секторов направлений (рис.2.а). В этом случае, имеем двунаправленный секторный цепной код (ДСЦК) и для каждого гладкого односекторного сегмента произвольной линии служебной информацией являются только код сектора (3 бита) и координаты начальной точки сегмента (возможно, с учетом инверсии). Таким образом, может быть получена оценка информационной емкости линий на растре – несколько больше, чем 1 бит на элемент.

В качестве другого варианта кодирования при использовании однонаправленной индексации секторов направлений (рис.2.б) может быть выбран однонаправленный секторный цепной код (ОСЦК). В этом случае код обеспечивает корректность линейного оператора редукции и сокращение длины кода служебной информации (номер сектора) на 1 бит.

Сопоставление различных видов цепных кодов линии, показанной на рис.1.в дает следующие результаты: простой цепной код – 1011101 – всего 21 бит; дифференциальный цепной код – $(1)_3 210021$ – всего 12 бит; двунаправленный секторный цепной код – $(1)_3 1011101$ – всего 10 бит; однонаправленный секторный цепной код – $(1)_2 1011101$ – всего 9 бит.

Результаты сравнения различных разновидностей цепных кодов с учетом служебной информации для линейного сегмента, показанного на рис.1.в, приведены в табл.1. Здесь показаны коды каждого из переходов на растре, а служебная информация помещена в указании кодов начальных точек линейного сегмента. Для простого цепного кода используется 3 бита на один переход на растре, для дифференциального цепного кода – 2 бита на один переход на растре, для двусекторного и односекторного цепных кодов – по 1 биту на один переход на растре.

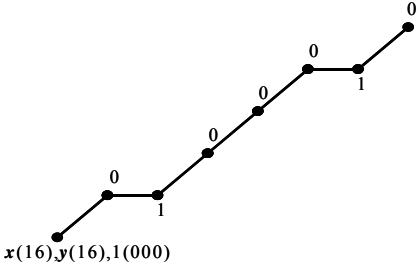
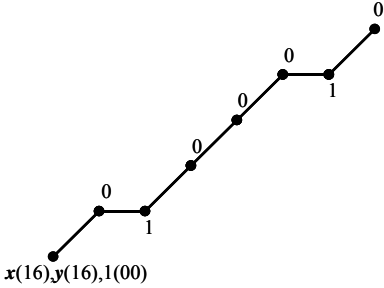
В отличие от приведенного выше примера из таблицы наглядно видно, какую долю занимает информационная емкость кода служебной информации: для цепного кода – 32 бита; для ДЦК – 35 бит; для ДСЦК - 35 бит; для ОСЦК - 34 бита.

Несмотря на незначительную разницу емкости ДСЦК и ОСЦК (всего 1 бит), ОСЦК в отличие от ДСЦК обеспечивает корректный оператор редукции, поэтому его использование является предпочтительным.

Таким образом, применение корректных операторов редукции структурных элементов изображений на дискретный растр позволяет построить эффективное представление пространственной структуры изображения и, при использовании различных разновидностей цепных кодов, получить экономную схему кодирования этой структуры, что, в конечном счете, повышает эффективность передачи цифровой графической информации по каналам связи телекоммуникационных систем.

Таблица 1

Сравнение различных цепных кодов		
Вид кода	Код	Емкость (b)
Цепной код		50
ДЦК		47

<p>ДСЦК</p>		<p>42</p>
<p>ОСЦК</p>		<p>41</p>

Литература

1. Павлидис Т. Алгоритмы машинной графики и обработки изображений. - М.: Радио и связь, 1986. - 400с.
2. Энджел Э. Интерактивная компьютерная графика. – М.: Вильямс, 2001. – 592с.
3. Freeman H. On the encoding of arbitrary geometric configurations // IEEE Trans. Electron. Comput., 1961, v. 10, № 2. – P. 260-268.
4. Freeman H. Shape description via the use of critical points // Pattern recognition, 1978, v. 10, № 3. – P. 159-169.

