

## ОБРАБОТКА ИЗОБРАЖЕНИЙ

УДК 528.9, 681.3.016

С.В. Абламейко, С.П. Боричев

### ФОРМИРОВАНИЕ ПРОСТРАНСТВЕННО-ЛОГИЧЕСКИХ СВЯЗЕЙ ОБЪЕКТОВ ЦИФРОВЫХ КАРТ В ГЕОИНФОРМАЦИОННЫХ СИСТЕМАХ

Рассматривается проблема ускорения процесса формирования пространственно-логических связей примыкания и пересечения объектов цифровых карт в геоинформационных системах. Отмечается, что наиболее трудоемкой вычислительной процедурой в этом процессе является процедура установления пространственно-логических связей объектов, возможность ускорения которой ограничивается последовательным характером выполнения операций ее алгоритмов на ПЭВМ. Для решения проблемы предлагается алгоритм, позволяющий реализовать эту последовательность операций параллельно при помощи команд технологии MMX/SSE ПЭВМ Pentium. Приведенные расчеты показывают его явное преимущество в быстродействии над существующими алгоритмами, а также потенциальные возможности по увеличению быстродействия на современных ПЭВМ типа Pentium, поддерживающих технологии параллельных вычислений SSE1/SSE2/SSE3/SSE4.

#### Введение

Пространственно-логические связи (ПЛС) представляют собой совокупность данных о топологических отношениях между объектами цифровой карты (ЦК), описывающих их взаимное пространственное расположение и логику взаимодействия друг с другом [1–3].

В объектно-ориентированных геоинформационных системах (ГИС), поддерживающих топологические отношения типа примыкания (стыка, касания) и пересечения между объектами ЦК, ПЛС образуют семантическую сеть – связный граф, состоящий из помеченных объектами ЦК вершин и помеченных типами отношений объектов дуг. Это представление ПЛС объектов в ЦК позволяет ГИС наиболее эффективно решать транспортные и коммуникационные задачи, такие, например, как поиск кратчайшего или оптимального пути, в том числе в условиях ограничений на передвижение, расчет зон обслуживания и оптимизация пунктов обслуживания, моделирование физических процессов, протекающих в инженерных сетях, и др. [4, 5].

ПЛС объектов в исходной ЦК<sub>исх</sub>, помещаемой в базу данных ЦК ГИС при ее создании, формируются при помощи следующих последовательно выполняемых процедур:

- установления ПЛС объектов ЦК<sub>исх</sub>, когда находятся координаты их общих точек (точек ПЛС) и определяется тип ПЛС;
- занесения полученной информации о ПЛС объектов в ЦК<sub>исх</sub>.

В зависимости от специфики решаемых задач в ГИС ПЛС объектов в ЦК<sub>исх</sub> могут формироваться [6] на двух уровнях:

– семантическом, если информация о ПЛС объектов ЦК<sub>исх</sub> используется в ГИС при необходимости. В этом случае информация о ПЛС объекта включается в его семантическое описание. Она содержит тип ПЛС (примыкание или пересечение) и идентификатор объекта, с которым эта связь образована. Получение же координат точек ПЛС, необходимых для решения некоторой прикладной задачи, осуществляется динамически путем использования процедуры установления ПЛС объектов по имеющейся информации о ПЛС, сформированной в ЦК<sub>исх</sub> на семантическом уровне;

– метрическом, если ГИС специализируется на решении транспортных или коммуникационных задач. В этом случае информация о ПЛС объектов в ЦК<sub>исх</sub> представляется в виде объектов со своими идентификаторами и классификационными кодами. Метрическое описание

таких объектов содержит координаты точки ПЛС, а семантическое описание – информацию о типе ПЛС и идентификаторах объектов, образующих эту ПЛС.

В обновленных ЦК<sub>обн</sub>, полученных из ЦК<sub>исх</sub> в результате введения в них новых объектов, удаления устаревших, а также изменения пространственного положения и геометрической формы объектов, информация о ПЛС объектов, оставшаяся от ЦК<sub>исх</sub> на соответствующих уровнях, нуждается в уточнении. Поэтому способ формирования ПЛС объектов в ЦК<sub>обн</sub> по результатам их обновления в отличие от способа формирования ПЛС объектов в ЦК<sub>исх</sub> включает в себя следующие последовательно выполняемые группы процедур:

1) процедуры, обеспечивающие занесение новых ПЛС объектов в ЦК<sub>обн</sub>:

– установление ПЛС объектов;

– идентификацию и занесение новых ПЛС объектов в ЦК<sub>обн</sub> на соответствующий уровень по результатам сравнения полученной информации о ПЛС с уже имеющейся в ЦК<sub>обн</sub>;

2) процедуры, обеспечивающие удаление ПЛС объектов из ЦК<sub>обн</sub>, утративших свою актуальность:

– установление ПЛС объектов по имеющейся информации о ПЛС в ЦК<sub>обн</sub>;

– удаление ПЛС объектов из ЦК<sub>обн</sub>, по которым не установлены общие точки и их координаты в результате выполнения предыдущей процедуры.

Можно заметить, что наиболее трудоемкая в вычислительном отношении процедура установления ПЛС объектов в способе формирования ПЛС объектов ЦК<sub>обн</sub> используется дважды, что снижает производительность этого способа по сравнению со способом формирования ПЛС объектов ЦК<sub>исх</sub>, где эта процедура используется лишь один раз.

Объемы работ и трудовые затраты по созданию и обновлению баз данных ЦК в ГИС постоянно растут [7, 8]. Оперативная поддержка баз данных ЦК в актуальном состоянии для многих ГИС в настоящее время – постоянно решаемая задача в процессе их эксплуатации [9, 10]. Поэтому большое внимание уделяется совершенствованию существующих способов, методов и алгоритмов не только в части ускорения процессов создания и обновления баз данных ЦК, но и в части формирования ПЛС объектов ЦК.

В настоящей статье на основе анализа существующих методов установления ПЛС объектов ЦК и реализованных на их базе алгоритмов, а также вычислительных возможностей современных ПЭВМ предложен алгоритм, позволяющий значительно ускорить выполнение процедуры установления ПЛС объектов ЦК и тем самым в целом повысить производительность процесса формирования ПЛС объектов ЦК в ГИС.

### 1. Существующие методы и алгоритмы установления ПЛС объектов ЦК

**Установление ПЛС объектов ЦК, представленных в векторной форме.** Для формализованного описания ПЛС примыкания и пересечения объектов используем теоретико-множественный подход. Обозначим множество всех объектов ЦК через  $O = \{O_1, O_2, \dots, O_n\}$ .

**ПЛС примыкания объектов (Prim)** определяется из условия, что один из пары объектов является незамкнутой линией и его начальная или конечная точка принадлежит границе другого объекта:

$$Prim = \{(O_1, O_2): O_1 \in O, O_2 \in O, O_2(t), t \in [0, 1], O_2(0) \in grO_1 \vee O_2(1) \in grO_1\} \dots, \quad (1)$$

где  $O_2$  – незамкнутая линия;

$grO_1$  – граница множества  $O_1$ .

**ПЛС пересечения объектов (Peres)** определяется из условия, что множество общих точек пересекающихся объектов не совпадает ни с одним из множеств точек, образующих эти объекты, при отсутствии условий примыкания объектов:

$$Peres = \{(O_1, O_2): O_1 \in O, O_2 \in O, O_1 \cap O_2 \neq \emptyset, O_1 \cap O_2 \neq O_1, O_1 \cap O_2 \neq O_2\} \neq Prim. \quad (2)$$

На рис. 1 показаны объекты ЦК, имеющие ПЛС примыкания и пересечения объектов.

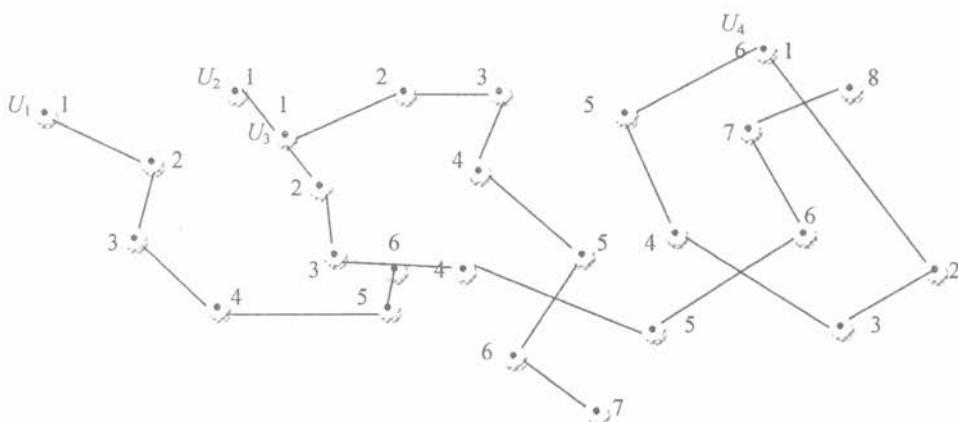


Рис. 1. Объекты ЦК, имеющие ПЛС примыкания объектов \$U\_1\$ и \$U\_2\$, \$U\_2\$ и \$U\_3\$; пересечения объектов \$U\_2\$ и \$U\_3\$, \$U\_2\$ и \$U\_4\$

В основе определения точек примыкания объектов лежит принцип поиска «ближайшего соседа» и пороговый метод принятия решения об обнаружении точки примыкания. Это означает, что между точками начала и конца каждого объекта в ЦК осуществляется поиск минимального расстояния \$R\$ до каждого сегмента остальных объектов, которое сравнивается с порогом \$\Omega\$. Если \$R \leq \Omega\$, то данная точка начала или конца объекта имеет примыкание к рассматриваемому сегменту соответствующего объекта.

Расчет координат точки примыкания \$x\_{\text{прим}}\$, \$y\_{\text{прим}}\$ объектов и расстояния \$R\$ осуществляется по формулам [11]

$$\begin{aligned}
 d1 &= |x2 - x01| * |x2 - x1| + |y2 - y01| * |y2 - y1|; \\
 d2 &= |x01 - x1| * |x2 - x1| + |y01 - y1| * |y2 - y1|; \\
 x_{\text{прим}} &= (d1 * x1 + d2 * x2) / (d1 + d2); \\
 y_{\text{прим}} &= (d1 * y1 + d2 * y2) / (d1 + d2); \\
 R &= [(x01 - x_{\text{прим}})^2 + (y01 - y_{\text{прим}})^2]^{1/2},
 \end{aligned} \tag{3}$$

где \$x01, y01\$ – координаты начальной или конечной точки объекта;

\$x1, y1, x2, y2\$ – координаты начальной и конечной точек сегмента проверяемого на примыкание объекта;

\$d1, d2\$ – вспомогательные коэффициенты.

Поиск точек пересечения объектов ЦК осуществляется путем проверки на пересечение каждого из сегментов выбранного для проверки объекта со всеми сегментами остальных объектов при помощи формул [11]

$$\begin{aligned}
 \pm s1 &= (x02 - x01) * (y1 - y01) - (x1 - x01) * (y02 - y01); \\
 \pm s2 &= (x02 - x01) * (y2 - y01) - (x2 - x01) * (y02 - y01);
 \end{aligned} \tag{4}$$

$$\begin{aligned}
 x_{\text{перес}} &= (|s1| * x2 + |s2| * x1) / (|s1| + |s2|); \\
 y_{\text{перес}} &= (|s1| * y2 + |s2| * y1) / (|s1| + |s2|),
 \end{aligned} \tag{5}$$

где \$x01, y01, x02, y02\$ – координаты начальной и конечной точек сегмента выбранного объекта;

\$x1, y1, x2, y2\$ – координаты начальной и конечной точек сегмента объекта, проверяемого на пересечение с выбранным;

\$\pm s1, \pm s2\$ – вспомогательные коэффициенты;

$x_{\text{перес}} \cup_{\text{перес}}$  – координаты точки пересечения сегмента выбранного объекта и сегмента объекта, проверяемого на пересечение.

Если значение коэффициента  $s1$  отрицательно, а  $s2$  – положительно, то существует пересечение сегментов объектов. Во всех остальных случаях комбинаций знаков коэффициентов  $s1$  и  $s2$  пересечений нет.

Эти алгоритмы, определяющие примыкание или пересечение всех ( $N/2$ ) пар векторных объектов ЦК, характеризуются вычислительной сложностью  $O(N^2)$ , что вызывает необходимость использования оптимизирующих средств решения рассмотренных задач.

Одним из методов ускорения поиска ближайших соседей является разбиение изображения ЦК на клетки (сегменты) [2] и создание клеточных структур (вычислительная сложность  $O(Nob)$ ,  $Nob$  – число объектов в клетке). В пределах каждой клетки осуществляется упорядочение конечных точек объектов (вычислительная сложность  $O(N \log N)$  или  $O(N \log N + k \log N)$ , где  $k$  – число точек пересечений объектов со сторонами клетки). Подобная структура данных позволяет по координатам сегмента выбранного объекта непосредственно определить клетку, в которой могут находиться сегменты других объектов, имеющие с ним примыкание или пересечение, и тем самым ускорить процесс установления ПЛС объектов.

**Установление ПЛС объектов ЦК в результате совместной обработки их векторного и растрового представлений.** Появление в ГИС технологий обработки цифровой картографической информации в виде векторных и растровых слоев [12] позволило оптимальным образом сочетать автоматизированные методы ее обработки и анализа и тем самым ускорить процесс установления ПЛС объектов. Среди таких методов можно отметить метод совмещенного анализа клеточных структур и их матричного (растрового) представления при установлении пересечений объектов [13], а также метод автоматического формирования ПЛС объектов ЦК, основанный на совместной обработке ее структурного и растрового изображений [14]. Установление ПЛС объектов в последнем методе осуществляется путем анализа результатов одномерного кодирования длин серий значащих элементов на  $j$ -строке и  $(j+1)$ -строке на растровом изображении ЦК. Одномерное кодирование длин серий значащих элементов на растровом изображении ЦК осуществляется построчно слева направо и сверху вниз [15]. При сканировании обнаруживаются непрерывные серии значащих элементов. Каждая из этих серий на  $j$ -строке растровой матрицы образует группу элементов (ГЭ), принадлежащих некоторому объекту, с координатами ее начала и конца  $\Gamma\mathcal{E}_{U_n}[(x_i, x_{i+k})_j]$ . Принадлежность ГЭ этому объекту на  $(j+1)$ -строке производится посредством проверки межстрочных связей ГЭ, расположенных на  $j$ -строке и  $(j+1)$ -строке. Для этого вводится понятие заявочной области  $Z$   $j$ -строки на  $(j+1)$ -строку.

Если некоторая ГЭ на  $j$ -строке  $\Gamma\mathcal{E}_{U_n}[(x_i, x_{i+k})_j]$  образует заявочную область  $Z_{U_n}[(x_{i-1}, x_{i+k+1})_{j+1}]$  на  $(j+1)$ -строку, то  $\Gamma\mathcal{E}_{U_n}[(x_i, x_{i+k})_{j+1}]$   $(j+1)$ -строки имеет с ней связь при выполнении условия

$$\Gamma\mathcal{E}_{U_n}[(x_i, x_{i+k})_{j+1}] \wedge Z_{U_n}[(x_{i-1}, x_{i+k+1})_{j+1}] \neq 0. \quad (6)$$

Если условие (6) не выполняется, то ГЭ на  $j$ -строке и  $(j+1)$ -строке не имеют между собой связи и принадлежат разным объектам. Если в заявочную область  $Z_{U_n}$  на  $(j+1)$ -строке попадают две ГЭ  $(j+1)$ -строки, которые принадлежат двум разным объектам, то возникает ситуация ветвления, которая анализируется на соответствие типу ПЛС примыкание или пересечение.

Еще один метод, позволяющий ускорить процесс установления ПЛС объектов ЦК, условно называется методом «помеченных скелетонов и областей» [16]. Он прост в реализации, обладает линейной вычислительной сложностью  $O(N)$  и не предполагает разбиения изображения ЦК на клетки и просмотра растровых изображений объектов в выбранных клетках ЦК  $[L-2]^2$ , где  $L$  – размер матрицы, проходами локальных операторов, а также растрового преобразования изображения ЦК при помощи одномерного кодирования единичных серий его элементов. Этот метод исследует лишь конкретные объекты ЦК в процессе их векторно-растрового преобразования и некоторую их 8-связную окрестность локальным оператором размером  $[3 \times 3]$  элемента растра на наличие точек ветвления (узловых элементов) [17].

## 2. Повышение быстродействия алгоритмов установления ПЛС объектов ЦК при их выполнении на современных ПЭВМ

Практика показывает, что быстродействие алгоритмов программ зависит не только от используемых методов, но и от вычислительных возможностей и производительности применяемых для их реализации современных ПЭВМ (компьютеров), созданных, например, на основе нового поколения микропроцессоров (МП) Pentium фирмы Intel. В процессе эволюции МП повышение производительности вычислений происходило не только за счет увеличения тактовой частоты, но и путем ускорения доступа к оперативной памяти, расширения возможностей конвейерной обработки команд, включения в систему команд групповых операций, реализующих технологии MMX/SSE на основе метода SIMD [18]. Групповые операции позволяют максимально использовать вычислительные возможности МП и ускорить выполнение задачи за счет одновременной обработки нескольких данных. Наибольший выигрыш по сравнению с командами общего назначения получается при использовании групповых команд в циклах, выполняющих обработку больших массивов чисел, различные манипуляции с матрицами и т. п., если реализуемый алгоритм при этом допускает параллельные вычисления. Появление многоядерных процессоров открывает новые перспективы в повышении производительности вычислений путем их распараллеливания с помощью технологий SSE1/SSE2/SSE3/SSE4 [19].

С целью повышения быстродействия алгоритмов установления ПЛС объектов, методы которых рассмотрены выше, была проанализирована возможность их реализации на ПЭВМ Pentium с использованием команд групповых операций технологии MMX [20]. Анализ алгоритмов [11, 13, 14], показал, что применение в них параллельных вычислений затруднено из-за ярко выраженного последовательного характера выполнения команд, обусловленного необходимостью проверки многочисленных условий перехода от одной ветви алгоритма к другой при обнаружении ветвлений и определении наличия примыканий и пересечений объектов. Исследования показывают, что быстродействие подобных алгоритмов может быть увеличено немногим более чем в 1,5 раза независимо от количества конвейеров, применяемых в МП [18]. В то же время алгоритм [16], несмотря на то, что состоит из последовательных операций обработки и проверки наличия примыкания и пересечения каждого элемента изображения, допускает использование параллельных вычислений, позволяющих обрабатывать несколько элементов одновременно. Преобразование последовательного алгоритма [16] в параллельный было осуществлено на основе методов, изложенных в работах [21].

## 3. Алгоритм, использующий параллельные вычисления для установления ПЛС объектов (параллельный алгоритм)

Представим исходное изображение в виде множества элементов  $G = \{G(i, j)\}$ , где  $G(i, j)$  принимает значения 0 или 1,  $j$  изменяется от 1 до  $N$ ,  $i$  – от 1 до  $M$ . Множество элементов  $G$  хранится в матричной памяти и занимает объем  $[N \times M]$  строк и столбцов элементов.

Каждый элемент множества  $G(i, j)$ , где  $i = 2 \div M - 1, j = 2 \div N - 1$ , имеет восемь соседей, которые нумеруются в соответствии со схемой, показанной на рис. 2.

$a_1$	$a_2$	$a_3$
$a_4$	$a_{ij}$	$a_5$
$a_6$	$a_7$	$a_8$

Рис. 2. Схема нумерации элементов множества

Используя эту схему в обработке бинарных уточненных изображений объектов локальным оператором  $[3 \times 3]$  элемента, определим понятие связности и характеристические числа [17].

Количество единичных элементов в  $S_8$  (8-связной окрестности) выбранного элемента для последовательного алгоритма определяется по формуле

$$A_8(a_{ij}) = \sum_{k=1}^8 a_k. \quad (7)$$

Анализ результатов обработки множества G по формуле (7) показывает, что в процессе суммирования участвуют следующие подмножества элементов множества G:

$$\begin{aligned} P_{a1}(i, j), i = 2 \div M - 1, j = 3 \div N; \\ P_{a2}(i, j), i = 1 \div M - 2, j = 3 \div N; \\ P_{a3}(i, j), 1 \div M - 2, j = 2 \div N - 1; \\ P_{a4}(i, j), i = 1 \div M - 2, 1 \div N - 2; \\ P_{a5}(i, j), i = 2 \div M - 1, j = 1 \div N - 2; \\ P_{a6}(i, j), i = 3 \div M, j = 1 \div N - 2; \\ P_{a7}(i, j), i = 3 \div M, j = 2 \div N - 1; \\ P_{a8}(i, j), i = 3 \div M, j = 3 \div N. \end{aligned} \quad (8)$$

Поэтому, выделив эти подмножества из исходного множества G и параллельно просуммировав их соответствующие элементы, получим подмножество, содержащее искомый результат  $A^*8(a_{ij})$ ,  $i = 2 \div M - 1, j = 2 \div N - 1$ , для всех элементов:

$$A^*8(a_{ij}) = P_{a1}(i, j) + P_{a2}(i, j) + P_{a3}(i, j) + P_{a4}(i, j) + P_{a5}(i, j) + P_{a6}(i, j) + P_{a7}(i, j) + P_{a8}(i, j). \quad (9)$$

Аналогично можно определить количество единичных элементов в  $S_4$  (4-связной окрестности) по формулам:

– для последовательного алгоритма

$$A_4(a_{ij}) = \sum_{k=1}^4 a_{2k-1}; \quad (10)$$

– для параллельного алгоритма

$$A^*4(a_{ij}) = P_{a1}(i, j) + P_{a3}(i, j) + P_{a5}(i, j) + P_{a7}(i, j). \quad (11)$$

Количество единичных двоек в  $S_8$  (8-связной окрестности) определяется по формулам:

– для последовательного алгоритма

$$B_8(a_{ij}) = \sum_{k=1}^8 a_k a_{k+1}; \quad (12)$$

– для параллельного алгоритма

$$\begin{aligned} B^*8(a_{ij}) = & [P_{a1}(i, j) \wedge P_{a2}(i, j)] + [P_{a2}(i, j) \wedge P_{a3}(i, j)] + [P_{a3}(i, j) \wedge P_{a4}(i, j)] + \\ & + [P_{a4}(i, j) \wedge P_{a5}(i, j)] + [P_{a5}(i, j) \wedge P_{a6}(i, j)] + [P_{a6}(i, j) \wedge P_{a7}(i, j)] + \\ & + [P_{a7}(i, j) \wedge P_{a8}(i, j)] + [P_{a8}(i, j) \wedge P_{a1}(i, j)]. \end{aligned} \quad (13)$$

Количество единичных троек в  $S_8$  (8-связной окрестности) определяется по формулам:

– для последовательного алгоритма

$$C_8(a_{ij}) = \sum_{k=1}^4 a_{2k-1} a_{2k} a_{2k+1}; \quad (14)$$

– для параллельного алгоритма

$$\begin{aligned} C^*_8(a_{ij}) = & [P_{a1}(i,j) \wedge P_{a2}(i,j) \wedge P_{a3}(i,j)] + [P_{a3}(i,j) \wedge P_{a4}(i,j) \wedge P_{a5}(i,j)] + \\ & + [P_{a5}(i,j) \wedge P_{a6}(i,j) \wedge P_{a7}(i,j)] + [P_{a7}(i,j) \wedge P_{a8}(i,j) \wedge P_{a1}(i,j)]. \end{aligned} \quad (15)$$

Число 8-связности определяется по формулам:

– для последовательного алгоритма

$$N_c(a_{ij}) = \overline{A_4}(a_{ij}) - \overline{C_8}(a_{ij}), \quad (16)$$

где  $\overline{A_4}(a_{ij}) = 1 - A_4(a_{ij})$ ,  $\overline{C_8}(a_{ij}) = 1 - C_8(a_{ij})$ ;

– для параллельного алгоритма

$$N^*_c(a_{ij}) = [1 - A_4(a_{ij})] - [1 - C_8(a_{ij})]. \quad (17)$$

Число переходов определяется по формулам:

– для последовательного алгоритма

$$C_n(a_{ij}) = A_8(a_{ij}) - B_8(a_{ij}); \quad (18)$$

– для параллельного алгоритма

$$C^*_n(a_{ij}) = A^*_8(a_{ij}) - B^*_8(a_{ij}). \quad (19)$$

С использованием этих чисел можно определить узловые элементы по формулам:

– для последовательного алгоритма

$$\{N_c, C_n\} \geq 3; \quad (20)$$

– для параллельного алгоритма

$$\{N^*_c, C^*_n\} \geq 3. \quad (21)$$

По координатам узловых элементов определяются идентификаторы объектов, которым они принадлежат. Если эти элементы являются начальными или конечными точками объекта (объектов), то ПЛС классифицируется как примыкание, если ни один из элементов не является начальной или конечной точкой соответствующего объекта, то ПЛС классифицируется как пересечение.

Оценим быстродействие параллельного алгоритма, сравнивая его с быстродействием других алгоритмов. Быстродействие алгоритмов будем измерять количеством тактов, за которое выполняются операции, входящие в соответствующие реализуемые алгоритмами формулы, на компьютерах типа Pentium/MMX с двухконвейерной обработкой команд.

#### 4. Расчет быстродействия алгоритмов по установлению ПЛС объектов

Расчет быстродействия алгоритмов по установлению ПЛС объектов ЦК, представленных в векторной и растровой формах, осуществляется по формулам (3)–(5) и (9)–(20) соответственно. Рассмотрим растровые изображения объектов  $U_1$ ,  $U_2$  и  $U_3$  на фрагменте  $[16 \times 16]$  элементов (рис. 3). Объект  $U_1$  примыкает к объекту  $U_3$  в точке с координатами  $x = 9$ ,  $y = 9$ . Объекты  $U_2$  и

$U_3$  пересекаются в точке с координатами  $x = 12, y = 6$ . Эти объекты в векторной форме представляются координатами  $U_1 (x = 2, y = 2; x = 9, y = 9)$ ,  $U_2 (x = 8, y = 2; x = 15, y = 9)$ ,  $U_3 (x = 15, y = 3; x = 3, y = 15)$ .

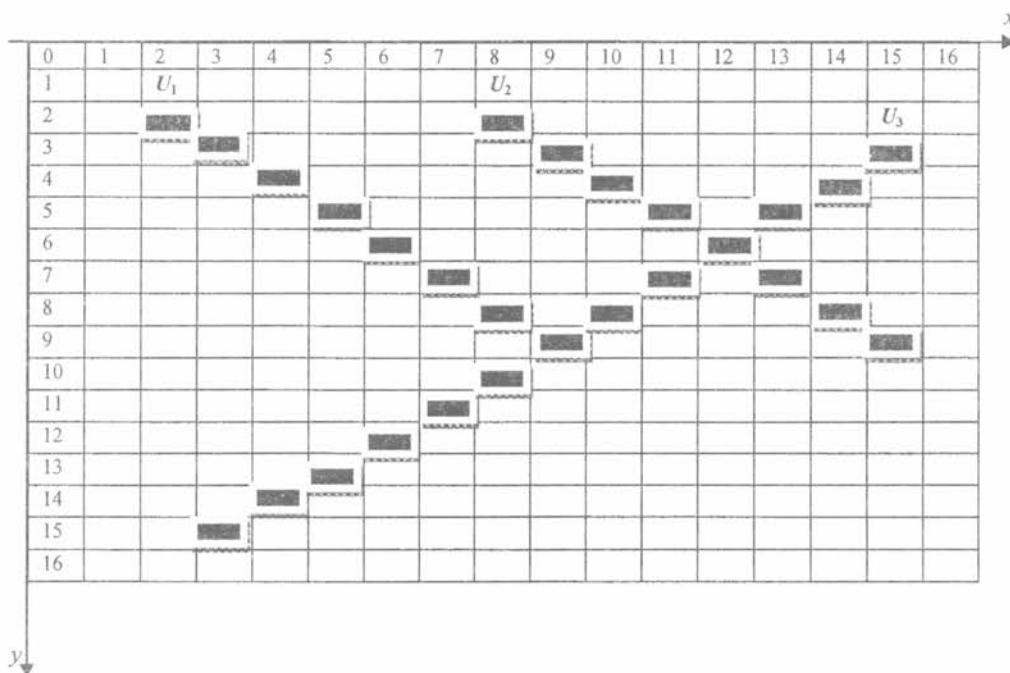


Рис. 3. Растворные изображения объектов  $U_1$ ,  $U_2$  и  $U_3$  и их ПЛС

#### 4.1. Расчет числа операций и тактов, затрачиваемых алгоритмом для установления ПЛС объектов в векторной форме (векторный алгоритм)

Число тактов, за которое выполняются базовые операции над вещественными числами двойной точности в компьютере [22], приведено в табл. 1.

Таблица 1  
Число тактов выполнения базовых операций в компьютере

Тип операции	Число тактов
Сложение/вычитание	38
Умножение	56
Деление	120
Извлечение квадратного корня	62

Расчет числа операций и тактов, затрачиваемых алгоритмом для установления ПЛС примыкания векторных объектов  $U_1$ ,  $U_2$  и  $U_3$  (табл. 2), осуществляется по формуле (3).

Таблица 2  
Число операций и тактов, затрачиваемых алгоритмом  
для установления ПЛС примыкания объектов  $U_1$ ,  $U_2$  и  $U_3$

Тип операции	Число операций	Число тактов
Сложение/вычитание	96	3648
Умножение	48	2688
Деление	12	1440
Извлечение квадратного корня	6	372
Всего	162	8148

Расчет числа операций и тактов, затрачиваемых алгоритмом для установления ПЛС пересечения векторных объектов  $U_1$  и  $U_2$ ,  $U_1$  и  $U_3$  (табл. 3) осуществляется по формулам (4).

Таблица 3  
Число операций и тактов, затрачиваемых алгоритмом  
для установления ПЛС пересечения объектов  $U_1$  и  $U_2$ ,  $U_1$  и  $U_3$

Тип операции	Число операций	Число тактов
Сложение/вычитание	20	760
Умножение	8	448
Всего	28	1208

Расчет числа операций и тактов, затрачиваемых алгоритмом для установления ПЛС пересечения векторных объектов  $U_2$  и  $U_3$  (табл. 4), осуществляется по формулам (4), (5).

Таблица 4  
Число операций и тактов, затрачиваемых алгоритмом  
для установления ПЛС пересечения объектов  $U_2$  и  $U_3$

Тип операции	Число операций	Число тактов
Сложение/вычитание	14	532
Умножение	8	448
Деление	2	240
Всего	24	1220

Таким образом, суммарное число операций и тактов, затрачиваемых алгоритмом для установления ПЛС примыкания и пересечения объектов  $U_1$ ,  $U_2$  и  $U_3$  в векторной форме, составляет 214 и 10 576 соответственно.

Быстродействие данного алгоритма может быть повышенено за счет конвейерной обработки команд в 1,5 раза. Таким образом, предположительно, быстродействие данного алгоритма может составить 7051 такт.

#### 4.2. Расчет числа операций и тактов, затрачиваемых последовательным алгоритмом [16] для установления ПЛС растровых объектов

Количество анализируемых элементов во фрагменте – 27. Каждый элемент кодируется одним байтом. Число тактов, за которое выполняются операции над целыми числами в компьютере [18, 23], приведено в табл. 5.

Таблица 5  
Число тактов выполнения операций  
над целыми числами в компьютере

Тип операции	Число тактов
Сложение/ вычитание	4
Логическое умножение	2
Умножение	34
Деление	58
Сравнение	3

Число операций и тактов, затрачиваемых алгоритмом для установления ПЛС примыкания и пересечения растровых объектов по соответствующим формулам, приведено в табл. 6.

Таблица 6

Число операций и тактов, затрачиваемое алгоритмом  
для установления ПЛС примыкания и пересечения растровых объектов

Формула вычислений	Число и типы операций	Число тактов
(7)	7 слож./элем.	$7 \times 4 \times 27 = 756$
(10)	3 слож./элем.	$3 \times 4 \times 27 = 324$
(12)	7 слож./элем., 8 лог. умн./элем.	$[(7 \times 4) + (8 \times 2)] \times 27 = 1188$
(14)	3 слож./элем., 8 лог. умн./элем.	$[(3 \times 4) + (8 \times 2)] \times 27 = 756$
(16)	3 выч./элем.	$3 \times 4 \times 27 = 324$
(18)	1 выч./элем.	$1 \times 4 \times 27 = 108$
(20)	2 сравн./элем.	$2 \times 3 \times 27 = 162$
Итого	42 опер./элем.	$3618 / 1,5 = 2412$ такта

Быстродействие данного алгоритма повышенено в 1,5 раза за счет конвейерной обработки команд и предположительно составит 2412 такта.

#### 4.3. Расчет числа операций и тактов, затрачиваемых параллельным алгоритмом для установления ПЛС растровых объектов

Алгоритмом обрабатываются подмножества (8)  $P_{a1}(i, j), P_{a2}(i, j), P_{a3}(i, j), P_{a4}(i, j), P_{a5}(i, j), P_{ab}(i, j), P_{at}(i, j)$  и  $P_{as}(i, j)$  множества  $G(i, j)$  размером  $[16 \times 16]$  элементов (см. рис. 3). Каждый элемент подмножества кодируется одним байтом. Обработка подмножеств осуществляется построчно слева направо и сверху вниз и ведется с использованием команд групповых операций технологии MMX. Каждая строка включает в себя две подстроки по 8 байт каждая, которые заносятся в соответствующие регистры MMX (двойное слово – 64 бит). В компьютере две операции над подстроками соответствующих строк подмножеств выполняются за один такт [20], т. е. одновременно обрабатываются соответствующие строки двух подмножеств. Количество обрабатываемых строк – 16.

Число операций и тактов, затрачиваемых параллельным алгоритмом для установления ПЛС примыкания и пересечения объектов, приведено в табл. 7.

Таблица 7

Число операций и тактов, затрачиваемое параллельным алгоритмом  
для установления ПЛС примыкания и пересечения объектов

Формула вычислений	Число и типы операций	Число тактов
(9)	7 слож./стр.	$7 \times 1 \times 16 = 112$
(11)	3 слож./стр.	$3 \times 1 \times 16 = 48$
(13)	7 слож./стр., 8 лог. умн./стр.	$[(7 \times 1) + (8 \times 1)] \times 16 = 240$
(15)	3 слож./стр., 8 лог. умн./стр.	$[(3 \times 1) + (8 \times 1)] \times 16 = 176$
(17)	3 выч./стр.	$3 \times 1 \times 16 = 48$
(19)	1 выч./стр.	$1 \times 1 \times 16 = 16$
(21)	2 сравн./стр.	$2 \times 1 \times 16 = 32$
Итого	42 опер./стр.	672 такта

#### 4.4. Оценка быстродействия алгоритмов

Быстродействие алгоритмов в тактах приведено в табл. 8, соотношение быстродействий различных алгоритмов – в табл. 9. Можно заметить, что быстродействие параллельного алго-

ритма для выбранного фрагмента  $[16 \times 16]$  элементов будет величиной постоянной и не зависящей от количества значащих элементов в нем. В то же время быстродействие последовательного алгоритма будет уменьшаться с каждым новым значащим элементом изображения, появившимся во фрагменте. В табл. 9 соотношение быстродействий последовательного/параллельного алгоритмов приведено для 9,5 % ( $27/256 \times 100$ ) содержания значащих элементов в выбранном фрагменте. При количестве значащих элементов от общего числа элементов во фрагменте 25 % ( $64/256 \times 100$ ) это соотношение превысит 8.

Таблица 8  
Быстродействие алгоритмов в тактах  
на компьютере типа Pentium/MMX

Алгоритм	Быстродействие в тактах
Векторный	7051
Последовательный	2412
Параллельный	672

Таблица 9  
Соотношение быстродействий различных алгоритмов,  
выполняемых на компьютере типа Pentium/MMX

Алгоритмы	Значение
Векторный/последовательный	3
Последовательный/параллельный	3,6
Векторный/параллельный	10,5

Рассмотрим диаграмму, показывающую количество тактов, затрачиваемых последовательным и параллельным алгоритмами на обработку значащих элементов, число которых выражено в процентах от общего числа элементов в выбранном фрагменте  $[16 \times 16]$  элементов (рис. 4). Быстродействия последовательного и параллельного алгоритмов будут приблизительно равны при 5% -м (5% – 13 значащих элементов) содержании значащих элементов в выбранном фрагменте. Далее быстродействие параллельного алгоритма будет возрастать в два раза с добавлением во фрагмент каждого пяти значащих элементов за счет снижения быстродействия последовательного алгоритма.

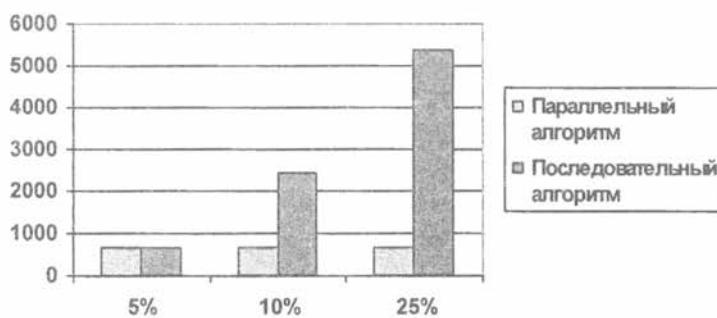


Рис. 4. Количество тактов, затрачиваемых последовательным и параллельным алгоритмами на обработку значащих элементов

Приведенные предварительные расчеты наглядно показывают преимущество в быстродействии параллельного алгоритма по сравнению с последовательным при установлении ПЛС объектов. Быстродействие параллельного алгоритма может быть значительно повышенено при выполнении вычислений на современных компьютерах, поддерживающих технологии параллельных вычислений SSE1/SSE2/SSE3/SSE4. Так, при применении компьютеров типа

Pentium 4 быстродействие параллельного алгоритма по сравнению с последовательным может быть увеличено на выбранном фрагменте для 9,5% значащих элементов в 16 раз, поскольку его пиковая производительность составляет четыре операции за один такт над двумя 128-битовыми строками [24].

### Заключение

Формирование ПЛС примыкания и пересечения объектов ЦК является актуальным при создании и обновлении баз данных ЦК ГИС, решающих транспортные и коммуникационные задачи. Все возрастающий объем информации, который необходимо при этом обрабатывать, предъявляет повышенные требования к быстродействию используемых алгоритмов. К сожалению, последовательный характер выполнения операций в существующих алгоритмах установления ПЛС объектов (определения координат точек ПЛС и их типа) в этих условиях приводит к снижению производительности процесса формирования ПЛС объектов на современных ПЭВМ. В этой связи предлагается алгоритм, позволяющий распараллелить вычисления путем включения в него команд групповых операций, реализующих технологию MMX/SSE на основе метода SIMD. Приведенные расчеты показывают явное преимущество в быстродействии предложенного алгоритма над существующими, а также его потенциальные возможности по увеличению быстродействия на современных ПЭВМ типа Pentium, поддерживающих технологии параллельных вычислений SSE1/SSE2/SSE3/SSE4.

### Список литературы

1. Халугин, Е.И. Цифровые карты / Е.И. Халугин, Е.А. Жалковский, Н.Д. Жданов. – М. : Недра, 1992. – 419 с.
2. Горбачев, В.Г. Что такое «топологические» отношения в цифровой картографии или для чего топологические отношения нужны в геоинформатике? / В.Г. Горбачев [Электронный ресурс]. – Режим доступа : [http://www.integro.ru/metod/topo\\_relations.htm](http://www.integro.ru/metod/topo_relations.htm). – Дата доступа : 30.10.2004.
3. Государственный стандарт Российской Федерации. Карты цифровые топографические. Правила цифрового описания картографической информации. Общие требования : ГОСТ Р 51607–2000. – Введ. 01.01.01. – М. : Госстандарт России : ИПК Издательство стандартов, 2001. – 8 с.
4. Шумова, О.В. Топология на электронных тематических картах / О.В. Шумова // Геодезия и картография. – 2003. – № 5. – С. 35–39.
5. Михеева, Т.И. Геоинформационная составляющая объектной модели улично-дорожной сети города / Т.И. Михеева, Р.В. Демьяненко // Самарский гос. аэрокосм. ун-т [Электронный ресурс]. – Режим доступа : [http://www.unn.ru/nelin\\_mir/files/miheeva\\_demyanenko.doc](http://www.unn.ru/nelin_mir/files/miheeva_demyanenko.doc). – Дата доступа : 16.09.2007.
6. Абламейко, С.В. Географические информационные системы. Создание цифровых карт / С.В. Абламейко, Г.П. Апарин, А.Н. Крючков. – Минск : Ин-т техн. кибернетики НАН Беларуси, 2000. – 276 с.
7. Хабаров, В.Ф. Деятельность федеральной службы геодезии и картографии России (1966–2003) / В.Ф. Хабаров // Первый конгресс геодезистов и картографов : доклады. – М. : Картгоцентр – Геодезиздат, 2003. – С. 5–15.
8. Верещака, Т.В. Цифровая технология обновления топографических карт масштабов 1:25 000 – 1:200 000 по космическим снимкам и ее концептуальные положения / Т.В. Верещака, А.С. Вахтанов // Геодезия и картография. – 2004. – № 4. – С. 33–42.
9. Кононов, В. Данные дистанционного зондирования Земли как источник информации для решения задач землеустройства и кадастра / В. Кононов, С. Марков // ЦАКИЗ ИГН НАН Украины [Электронный ресурс]. – Режим доступа : <http://www.geomatika.kiev.ua/training/DataCapture?RemoteSensing/chapter102/html>. – Дата доступа : 16.09.2007.
10. Хорев, А.Г. Опыт применения геоинформационных технологий в автодорожной отрасли / А.Г. Хорев [и др.] // Ин-т вычислительных технологий СО РАН [Электронный ресурс]. – Режим доступа : [http://www-sbras.nsc.ru/win/gis\\_new/lib/publ/b10/hor.htm](http://www-sbras.nsc.ru/win/gis_new/lib/publ/b10/hor.htm). – Дата доступа : 16.09.2007.

11. Булаев, А.В. Модели и алгоритмы вычисления топологических отношений в геоинформационных системах / А.В. Булаев, Д.Е. Андрианов // Междунар. журн. «Программные продукты и системы» [Электронный ресурс]. – 2006. – № 3. – Режим доступа : <http://swsys.ru/index.php?page=article&id=457>. – Дата доступа : 16.09.2007.
12. Дзюба, А.К. Методы подготовки и использования растровых картографических материалов / А.К. Дзюба // АО «ЦКМ» [Электронный ресурс]. – 2000. – Режим доступа : <http://masters.donntu.edu.ua/2003/ggeo/pilicheva/library/metod.htm>. – Дата доступа : 16.09.2007.
13. Каминский, В.И. Определение точек взаимного пересечения объектов местности, представленных в цифровом виде / В.И. Каминский, В.Г. Плешков, И.В. Сидоров // Геодезия и картография. – 1994. – № 2. – С. 50–51.
14. Абламейко, С.В. Автоматическое формирование пространственно-логических связей между объектами цифровой модели карты, основанное на совместной обработке ее структурного и растрового изображений / С.В. Абламейко, С.П. Боричев, А.Н. Крючков // Анализ цифровых изображений. Вып. 2. – Минск : ОИПИ НАН Беларуси, 2003. – С. 135–140.
15. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2006. – 1072 с.
16. Шавров, С.А. Метод автоматического построения сети отношений между объектами структурных изображений / С.А. Шавров // Методы и средства обработки сложной графической информации : тез. докл. II Всесоюзн. конф., Горький, сент. 1985 г. / Горьк. гос. ун-т им. Н.И. Лобачевского ; редкол. : Ю.Г. Васин [и др.]. – Горький, 1985. – С. 138–139.
17. Буча, В.В. Выделение и векторизация линейных объектов на цветных картографических изображениях / В.В. Буча, С.В. Абламейко // Вес. Нац. акад. навук Беларусі. Сер. фіз.-мат. навук. – 2005. – № 1. – С. 98–105.
18. Соколенко, П. Pentium глазами программиста / П. Соколенко [Электронный ресурс]. – 2002. – Режим доступа : <http://www.wasm.ru/article.php?article=1011001>. – Дата доступа : 16.09.2007.
19. Многоядерные процессоры [Электронный ресурс]. – Режим доступа : [http://www.knowaretech.com/processory\\_mnogoyadernye-processory.html](http://www.knowaretech.com/processory_mnogoyadernye-processory.html). – Дата доступа : 16.09.2007.
20. Команды MMX (начиная с Pentium MMX, K6) [Электронный ресурс]. – Режим доступа : <http://www.realcoding.net/article/view/1042>. – Дата доступа : 16.09.2007.
21. Абламейко, С.В. Алгоритм фильтрации изображений, использующий параллельные операции сдвига и сложения при свертке изображений с масками преобразования / С.В. Абламейко, С.П. Боричев, А.Н. Крючков // Анализ цифровых изображений. – Минск : ОИПИ НАН Беларуси, 2002. – Вып. 1. – С. 7–13.
22. Захаров, А.В. Методы вычислений с гарантированной точностью на платформе «Мультикор» : автореф. дис. ... канд. техн. наук : 05.13.11 / А.В. Захаров ; Рос. акад. наук, Ин-т проблем управления [Электронный ресурс]. – М., 2007. – Режим доступа : <http://www.botik.ru/PSI/autoref.doc>. – Дата доступа : 16.09.2007.
23. Кравченко, Т. Выполнение арифметических операций с целыми числами / Т. Кравченко // ООО «ЭФО» [Электронный ресурс]. – Режим доступа : <http://www.atmel.ru/Articles/Atmel21.htm>. – Дата доступа : 2.02.2009.
24. Соколенко, П. Заметки о технологии Hyper-Threading. Ч. 2. Программирование / П. Соколенко [Электронный ресурс]. – 2004. – Режим доступа : [http://www.macro.aaanet.ru/apnd\\_10.html](http://www.macro.aaanet.ru/apnd_10.html). – Дата доступа : 16.09.2007.

Поступила 09.10.08