

УДК 514.1:744

И.Л. Бильчук

ВЫЧИСЛИТЕЛЬНАЯ ГЕОМЕТРИЯ ДЛЯ ТЕХНИЧЕСКИХ ЧЕРТЕЖЕЙ

Для моделирования зданий разрабатывается новая топологическая и геометрическая концепция, использующая для моделирования ячейку и избегающая ряда недостатков, связанных с моделированием зданий исключительно с помощью помещений или строительных элементов. Предпосылками пригодности новой концепции является разбиение технических чертежей на простые полигоны и эффективное выполнение над парами полигонов из этого множества логических операций объединение, пересечение, разность и симметричная разность. Кроме этого необходима быстрая геометрическая идентификация полигонов с помощью нажатия мыши на графических интерфейсах.

A new topological and geometrical conception aimed at building modeling is being developed. This conception uses a cell for building modeling and avoids a set of drawbacks occurring when the building modeling is carried out on the base of rooms and structural members. The fragmentation of the technical drawings into simple polygons as well as the efficient fulfillment of the logical operations such as union, intersection, difference and symmetric difference over pairs of polygons are the prerequisites for the new conception's appropriateness. Besides, fast geometrical identification of polygons by clicking on graphic interfaces is necessary.

Характерным для полигонов технических чертежей является то, что они могут касаться сами себя и других полигонов. Это приводит к значительным трудностям при разработке надежных алгоритмов для логических операций, так как действительные числа не могут быть точно изображены в компьютере и арифметические операции над действительными числами не могут быть точно выполнены. Результаты предварительных исследований обычных алгоритмов были неудовлетворительными.

После проверки девяти алгоритмов для логических операций над полигонами был выбран метод Леонова и исследован более подробно. Этот метод строго разделяет геометрические узлы и сегменты линий полигона от его топологических станций и граней. Нормализованная пара операнд логической операции переводится с помощью геометрических операций в стандартную форму, из которой с помощью топологических операций выводятся контуры и полигоны результата. Выяснилось, что этот метод с точки зрения топологии является надежным и эффективным. Основные затраты по времени приходятся на проверку всех точек пересечения всех граней пары операнд.

Благодаря специальному определению полигонов и организации множества полигонов с помощью R-деревьев (region trees) стало возможным настолько сократить затраты по времени на выполнение логических операций в технических чертежах, что метод Леонова теперь может быть эффективно применен в чертежах со многими тысячами полигонов.

Эффективность R-деревьев для осуществления поиска во множестве полигонов с целью геометрической идентификации полигонов была систематично и всесторонне исследована на основе собственной объектно-ориентированной Java-реализации. В результате этого исследования был определен оптимальный минимальный и максимальный размер контейнера узла R-дерева. Было доказано, что операции над R-деревьями выполняются на

порядок быстрее в сравнении со способностью человека воспринимать реакцию компьютера с помощью органов чувств. Следовательно, полигоны и R-деревья хорошо пригодны в качестве основных компонентов новой концепции моделирования зданий, основанной на использовании ячеек.

Цели исследования. Способность моделировать на компьютере возведение строительного сооружения, его отделку, техническое оснащение и эксплуатацию является предпосылкой для целесообразного использования информационной техники в строительстве. До сих пор модели основывались или на помещениях, или на строительных элементах. В концепции, основывающейся на помещениях, объем здания геометрически разбивается на помещения. Строительные элементы здания располагаются относительно помещений. В концепции, основывающейся на строительных элементах, субстанция здания рассматривается как структурированное множество строительных элементов. Помещения описываются с помощью ограничивающих их строительных элементов. В обеих концепциях техническое оснащение и эксплуатация описываются относительно помещений.

Здания обладают свойствами, которые невозможно хорошо отобразить с помощью этих двух концепций. Если строительные элементы, например, перекрытия, балки и колонны проникают друг в друга, как это обычно бывает в железобетонном строительстве, то присвоение возникающего таким образом общего строительного объема отдельным строительным элементам не может быть однозначным. Также в строительных сооружениях бывает объем, который при моделировании целесообразно рассматривать независимо от отдельных строительных элементов, например, сцена и зона зрительского зала или различные климатические зоны большого офисного помещения. Технические службы и зоны, используемые персоналом, могут затрагивать не все помещение, а только его часть. Описание элемента здания, как например, стены, с помощью геометрии ее поверхностей является недостаточным, так как из этого описания не следует информация о внутренней структуре стены. Обе концепции приводят к огромным множествам объектов с многочисленными связями, управление которыми является чрезвычайно затруднительным.

Целью исследования была разработка новой концепции для моделирования зданий, избегающей названных проблем при отображении реальных свойств зданий. Исследование было ограничено исследованием плана этажа здания, изображенного на одном чертеже. Полученные таким образом знания и разработанные методы могут быть позже обобщены для любых строительных сооружений.

Решение состоит в том, чтобы полностью разбить чертеж на ячейки, не пересекающие друг друга. Каждое помещение является объединением некоторого числа ячеек. Персонал, техническое оборудование, предметы интерьера, а также функции снабжения и эксплуатации присваиваются ячейкам. Ячейка является центральным топологическим элементом данной концепции. Множество ячеек разбивается на два класса: внутренние ячейки и внешние ячейки. Множество внутренних ячеек образует план здания, множество внешних ячеек образует среду вокруг здания. Внешними ячейками могут быть также описаны проёмы в плане здания.

Значение терминов «внутреннее» и «внешнее», используемых в связи с ячейками, значительно отличается от их значения при использовании их в

связи с площадями. На границе площади всегда происходит переход между внутренней и внешней областью. Это приводит к правилам, как например «winding rule» в Java, которые позволяют геометрически различать внутреннее и внешнее. Чисто геометрическое различие внутренних и внешних ячеек невозможно. Две соседствующие ячейки могут принадлежать одному и тому же классу. Поэтому положение внутри или снаружи является атрибутом ячейки, значение которому присваивает пользователь.

Так как чертежный лист содержит большое число нерегулярных ячеек, то эффективная геометрическая идентификация ячеек с помощью нажатия мыши на экране компьютера является значительной предпосылкой пригодности новой концепции. Описанная в литературе концепция R-деревьев (region trees) кажется для этого особенно пригодной. В предварительных исследованиях, выполненных до проведения данного исследовательского проекта, эффективность этой концепции не была оценена достоверно. В литературе также не встречаются надежные сведения о том, растет ли число операций, необходимых для занесения, удаления и поиска по R-дереву линейно в зависимости от числа прямоугольников в дереве, с большим числом прямоугольников, независимо от их геометрических свойств.

Центральным геометрическим элементом новой концепции является полигон. Наружная граница полигона — это замкнутая последовательность прямых граней, направленных против часовой стрелки. Последовательность граней называется контуром полигона и ограничивает множество точек, лежащих при обходе контура в положительном направлении слева от контура. Внешний контур может содержать один или многие контуры, которые не пересекаются и направлены по часовой стрелке. Каждый из этих контуров ограничивает множество точек, которые при обходе контура в положительном направлении лежат слева от контура и покрывают всю оставшуюся площадь чертежной поверхности. Площадь полигона состоит из пересечения множеств точек наружного и внутренних контуров. Чертеж содержит большое число полигонов.

Полигоны эффективно соединяются друг с другом с помощью логических операций: объединение, пересечение, разница и симметричная разность. По определению эти операции выполняются над множествами точек полигонов. Так как эти множества являются бесконечными, то операции выполняются над контурами полигонов. Результатом является множество полигонов, которое может быть пустым или содержать более чем один полигон. Эти полигоны должны быть пригодны для того, чтобы выступить в качестве операндов дальнейших логических операций.

Логические операции над полигонами имеют центральное значение для отображения строительных сооружений и для других задач, например, в геоинформационных системах. Несмотря на это никакой из известных нам из предварительных исследований методов не мог быть успешно использован для выполнения операций над нетривиальными полигонами в случае, когда контуры касались себя или друг друга и проникали друг в друга.

В соответствии с данной ситуацией были сформулированы следующие цели исследований:

- 1) исследование эффективности R-деревьев;
- 2) исследование эффективности и надежности операций над полигонами;

3) разработка тестовой платформы для исследования затрат времени на выполнение алгоритмов.

Исследование концепции R-деревьев. Концепция R-деревьев для управления геометрическими формами в плоскости была опубликована Антонином Гутманом [7] в 1984 г. Согласно этой концепции каждая геометрическая форма заключается в прямоугольник с гранями, параллельными осям координат, таким образом, чтобы прямоугольник как можно плотнее охватывал форму (minimal bounding rectangle). Такой прямоугольник называется боксом формы и служит ее представителем. Все операции R-дерева выполняются над боксами геометрических форм. Основными операциями при использовании R-дерева являются:

- найди все боксы, содержащие данную точку;
- найди все боксы, пересекающиеся с поисковым боксом.

Дополнительно для управления R-деревом необходимы функции «добавь бокс в дерево» и «удали бокс из дерева».

Множество боксов R-дерева разбивается на подмножества. Задается минимальное и максимальное число боксов в подмножестве. Каждое подмножество как можно плотнее описывается дополнительным боксом, именуемым боксом узла дерева. Боксы внутри бокса узла могут пересекаться. В отличие от боксов форм, каждый из которых содержит только лишь одну геометрическую форму, боксы узлов являются элементами управления дерева. Все боксы форм расположены на одном уровне в дереве, имеющем номер 0. Боксы узлов 1-го уровня в свою очередь разбиваются на подмножества, которые также описываются боксами. Таким образом, возникают дальнейшие уровни в R-дереве. Верхний уровень дерева содержит только лишь один бокс — корень дерева.

Логика алгоритмов управления и использования R-дерева описана в [1, 7, 8]. Гутман реализовал ее со структурой данных, использующей страницы памяти, и сильно схожей с концепцией реляционных банков данных.

Приведённые в литературе [9, 10] результаты исследований числа операций для функций добавления, удаления и поиска в R-деревьях показывают, что они значительно зависят от числа боксов, последовательности занесения и удаления форм в дереве, размера узлов дерева, отношения ширины бокса к его высоте и отношения размеров бокса к размерам всего чертежа. Указанные в литературе стандартные отклонения замеров времени настолько велики, что не позволяют делать надежные выводы о пригодности использования R-деревьев для строительных задач.

Для того чтобы достоверно определить факторы, влияющие на оценку R-деревьев, и максимально сократить их, сначала была выполнена объектно-ориентированная версия метода Гутмана на Java. Более ранние исследования показали, что структура дерева при добавлении в него и удалении из него форм часто изменяется и, что такие изменения затрагивают большую часть дерева. Поэтому был реализован показанный на рис. 1 графический интерфейс для визуальной инспекции изменений в структуре дерева. С использованием этого инструмента могут быть исследованы деревья, содержащие до 81 бокса геометрических форм. Полученный опыт позволил оптимизировать объектно-ориентированную реализацию и подготовиться к тестированию больших R-деревьев.



Рис. 1. Тестовая платформа для визуальной инспекции структуры R-деревьев

Для исследования поведения больших R-деревьев (до 10 000 узлов) был разработан тестовый генератор. Каждый тест проводился для одной выбранной функции: добавление, удаление или поиск для заданного числа боксов форм. В качестве результата теста рассчитывалось среднее арифметическое значение времени выполнения функции и стандартное отклонение замеров на заданном числе станций. Операции, выполняемые на станции, зависят от типа тестируемой функции. Так, например, тестирование функции «добавь бокс формы в дерево» требует выполнения следующих операций:

1) сконструируй заданное число боксов форм с помощью генератора случайности;

2) с помощью генератора случайности определи перестановку индексов боксов форм и замерь время, необходимое для добавления боксов форм в измененной последовательности;

3) повтори шаг 2 заданное число раз и посчитай среднее значение затрат времени, необходимого для добавления бокса формы в дерево;

4) повтори шаги с 1-го по 3-й на каждой станции и посчитай статистику.

Определенные таким способом тестовые значения для функций R-деревьев зависят от выбора следующих параметров:

тип функции: добавление, удаление формы в дереве, поиск по дереву;
 число боксов форм дерева;
 минимальное и максимальное число детей узла дерева;
 минимальное и максимальное отношение ширины бокса к ширине чертежа;

минимальное и максимальное отношение высоты бокса к высоте чертежа.

Из-за большого числа проводимых замеров и необходимости графического изображения результатов тестирования был разработан показанный на рис. 2 тестовый редактор. На левой стороне редактора могут быть заданы параметры для проведения тестов. На правой стороне редактора может быть настроена презентация результатов тестирования.

		horizontal variable	u	
mean time per operation		vertical variable	w	polygons
standard deviation		make curve	number	mean time
number of polygons	11500	remove point	number	4
number of searches	100	annotation of current curve		8/15
number of removes	100	color of the points		red
observations per series	4	color of the lines		black
series per station	2	clear curve	number	
stations per test	3	sheet width		260.00
width of drawing mm	230.00	sheet height		170.00
height of drawing mm	170.00	user units per mm horizontal		50.000
max. box / drawing width	0.200	user units per mm vertical		0.200
max. box / drawing height	0.200	value of u at left edge		0.00
min. box width mm	2.00	value of w at bottom edge		0.00
min. box height mm	2.00	shapes per R-tree node min		2
shapes per R-tree node min	2	shapes per R-tree node max		3
min	max	increment	message field	
500.00	5000.00	500.00		

Рис. 2. Редактор тестовой платформы

С помощью генератора тестов и редактора тестовой платформы была систематически исследована пригодность R-деревьев для новой концепции [1]. В отличие от примеров, приведенных в литературе, были получены тестовые результаты с незначительным стандартным отклонением. Рис. 3 показывает влияние числа боксов форм в дереве на время, необходимое для удаления бокса формы, для различных минимальных и максимальных размеров контейнера узла, указанных в аннотации к кривым. Для функции удаления формы из дерева минимальный размер 8 и максимальный размер 15 детей в узле оказался самым оптимальным. Затраты по времени почти линейны.

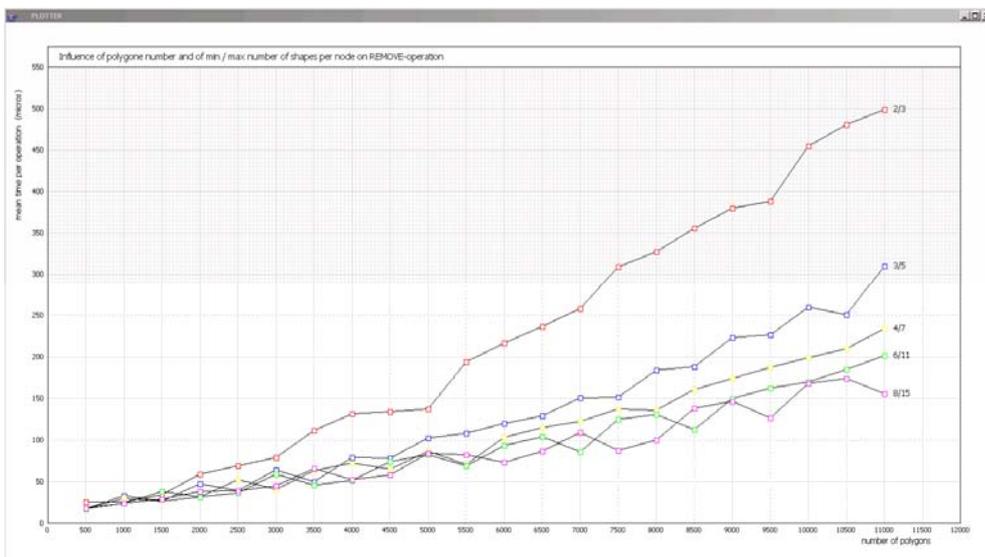


Рис. 3. Влияние числа боксов форм в дереве на время необходимое для удаления формы из дерева

Рис. 4 показывает влияние числа боксов форм N в дереве на время необходимое для поиска по дереву для различных минимальных и максимальных размеров контейнера узла, указанных в аннотации к кривым. Для функции поиска по дереву минимальный размер 8 и максимальный размер 15 детей в узле также оказался самым оптимальным. Затраты по времени приблизительно пропорциональны $N \log N$.

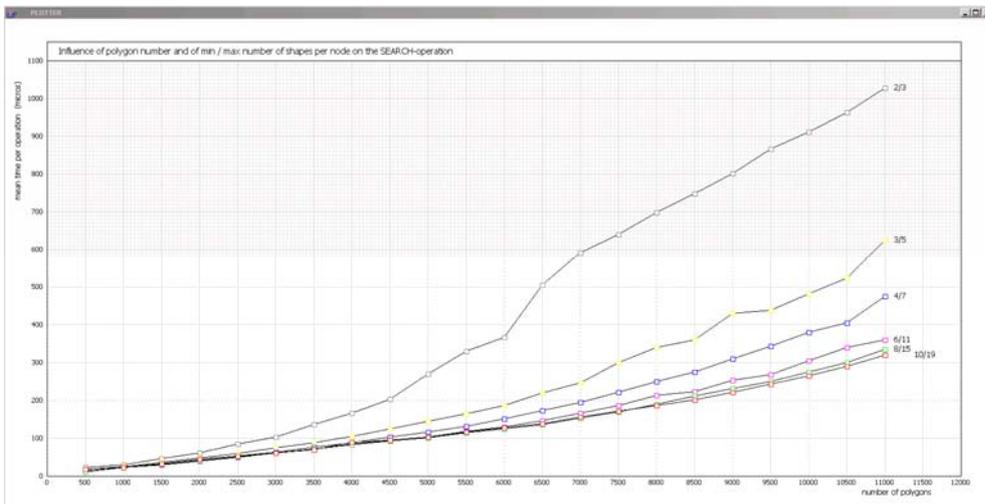


Рис. 4. Влияние числа боксов форм в дереве на время необходимое для поиска по дереву

Было установлено, что все функции над R-деревьями с 10 000 боксов форм, выполненные на обычном ноутбуке, требовали времени на порядок меньшего, чем время человеческой реакции на экран компьютера. Следова-

тельно, R-деревья хорошо пригодны для использования в технических чертежах. Особенно следует отметить следующее:

- 1) эффективность всех основных операций значительно увеличивается при увеличении минимального числа детей узла от 2 до 10;
 - 2) оптимальное максимальное число детей узла дерева — от 10 до 30;
 - 3) при оптимальном размере узла дерева время, необходимое для поиска, примерно в 30 раз превышает время, необходимое для добавления формы в дерево;
 - 4) время, необходимое для функции удаления формы из дерева, составляет примерно половину от времени, необходимого для выполнения поиска по дереву;
 - 5) время, необходимое для добавления формы в R-дерево и для удаления формы из R-дерева линейно зависит от размера R-дерева.
6. Время необходимое для поиска по дереву, при оптимальном размере узла является слабо нелинейным;
- 7) время, необходимое для поиска по дереву и удаления формы из дерева, растет при увеличении отношения максимальных размеров бокса формы к размеру чертежа.

Логические операции над полигонами. Так как предварительные исследования показали, что существующие алгоритмы для логических операций над полигонами в топологически и геометрически сложных случаях, типичных для технических чертежей, не дают удовлетворительных результатов, то сначала был сделан обзор [2] описанных в литературе методов. Из девяти исследованных методов предпочтение было отдано подходу «Эффективный алгоритм для замкнутого множества логических операций над полигональными регионами в плоскости» Леонова и Никитина [11]. Этот метод пригоден для обобщенных полигонов. Затраты по времени составляют $O(n \log n)$, где n — сумма числа узлов, граней и точек пересечения полигонов, которые являются операндами в логических операциях. Более того, ожидалось, что данный метод является устойчивым в условиях неизбежной неточности при сохранении и обработке действительных чисел на компьютере.

Основная концепция метода Леонова/Никитина состоит в систематичном разделении геометрических и топологических аспектов логических операций. Геометрия полигона описывается глобальными координатами узловых точек и прямыми ненаправленными сегментами линий, являющимися связью между парами узлов. Топология контура описывается с помощью циклической последовательности станций (vertices) и граней. Каждой станции соответствует определенный узел, однако каждому узлу могут быть присвоены несколько станций. Каждая грань соответствует одному сегменту линии и направлена от начальной станции в конечную. Одному сегменту линии могут соответствовать несколько граней. Каждая станция имеет одну входящую и одну исходящую грань. Этим способом можно описывать контуры, которые касаются сами себя в узлах и на гранях или пересекают сами себя.

Каждая логическая операция разбивается на следующие этапы.

1. Каждый контур, определенный пользователем, переводится в нормальную форму. Для этого, если необходимо, конструируются дополнительные узлы во внутренней области существующих сегментов линий, определяются станции и расщепляются грани. В нормальной форме две грани одного

контур могут касаться только в узлах их начальных и конечных станций. Более того, контур не может пересекать сам себя в узле. Если пользователь задал контур, который не удовлетворяет этим условиям, то такой контур отклоняется. Контур, возникающие как результат логических операций по методу Леонова/Никитина, уже находятся в нормальной форме и поэтому не нуждаются в дополнительной проверке и пригодны в качестве составляющих операнд.

2. Каждый заданный пользователем полигон переводится в нормальную форму. Для этого, если необходимо, конструируются дополнительные узлы во внутренней области существующих сегментов линий, определяются станции и расщепляются грани. В нормальной форме две грани различных контуров одного полигона могут касаться только в узлах их начальных и конечных станций. Более того, контуры одного нормализованного полигона не могут пересекаться. Если пользователь задал полигон, который не удовлетворяет этим условиям, то такой полигон отклоняется. Полигоны, возникающие как результат логических операций по методу Леонова/Никитина, уже находятся в нормальной форме и поэтому не нуждаются в дополнительной проверке и пригодны в качестве операнд.

3. Оба операнда одной логической операции являются полигонами, чьи контуры в общем случае пересекаются. Сначала пара операнд переводится в стандартную форму. Для этого вводятся дополнительные узлы, станции и грани. В стандартной форме пара граней обоих полигонов может пересекаться только в тех узлах, которые присвоены их станциям. Если пара граней накладывается на одну общую линию, то эти грани обладают одинаковыми узлами, но однако возможно, что при этом грани направлены в противоположные стороны.

4. На следующем этапе определяется положение каждой грани одного операнда по отношению к площади другого операнда. Принадлежащий какой-либо грани сегмент линии, за исключением его узлов, должен в стандартной форме находиться полностью внутри или полностью снаружи площади другого операнда. Во всех узлах, в которых полигоны пересекаются, входящие грани описываются с помощью этикеток (descriptors). Эти этикетки хранятся в списке, упорядоченном согласно возрастанию угла, образованного гранью и осью x в направлении против часовой стрелки.

5. Собираются контуры результата. При этом последовательно рассматриваются все контуры обоих операнд. Тип логической операции и положение грани определяют, принадлежит ли грань контуру результата или нет. После того, как была найдена первая грань нового контура, собираются граничащие грани и так далее, пока не будет достигнута начальная грань. В точке пересечения следующая грань определяется через поиск в списке этикеток. Если у контура нет точек пересечения, то весь контур или принадлежит или не принадлежит результату операции.

6. Множество новых контуров разбивается на полигоны.

Рис. 5 показывает стандартную форму примера Леонова, а также список этикеток узла этого примера.

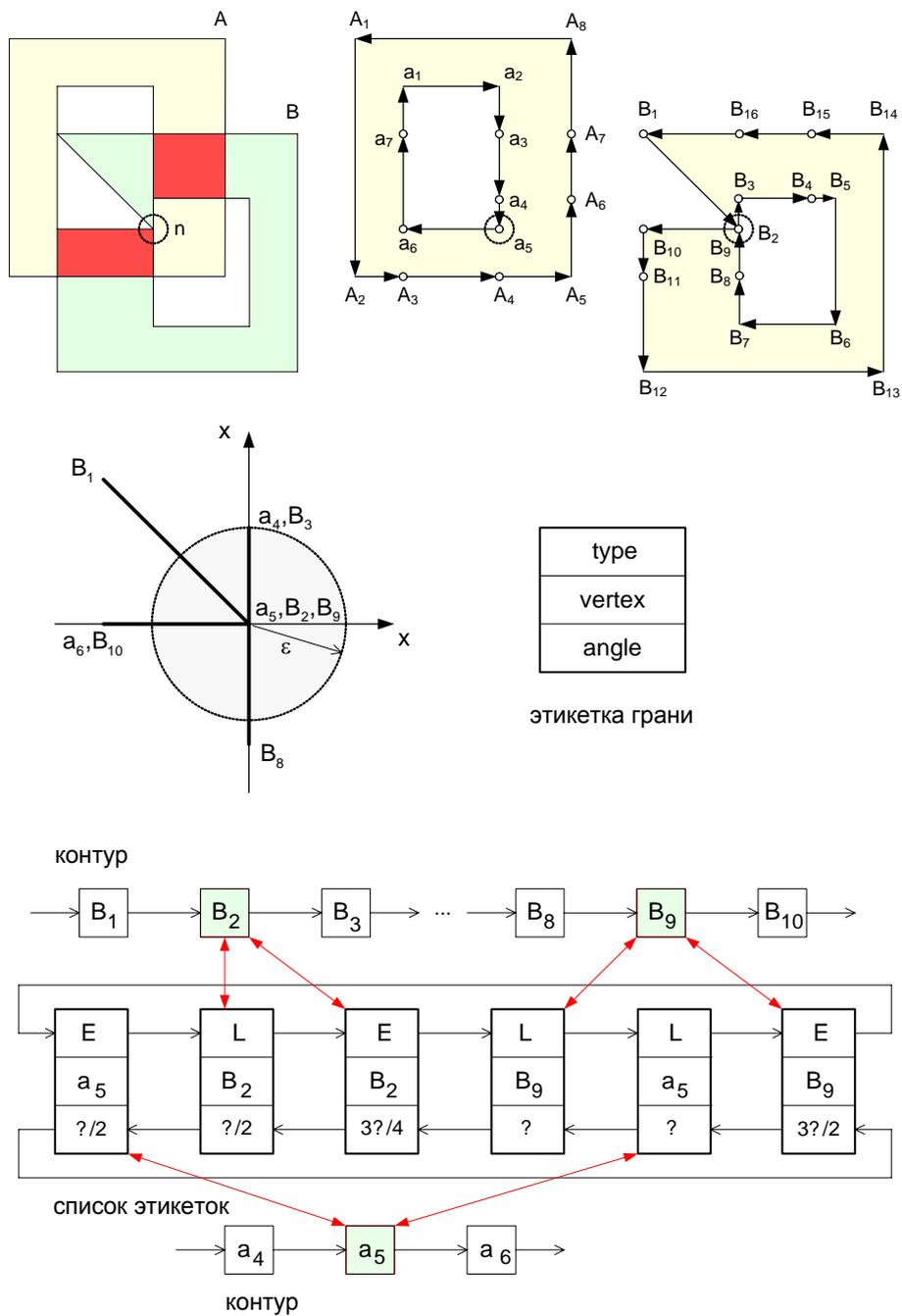


Рис. 5: Стандартная форма пары операнд

Метод Леонова/Никитина был реализован на Java-платформе с циклическими двунаправленными списками в качестве основной структуры данных. Рис. 6 показывает структуру классов реализации. Подробности алгоритмов и структур данных изложены в [3, 4].

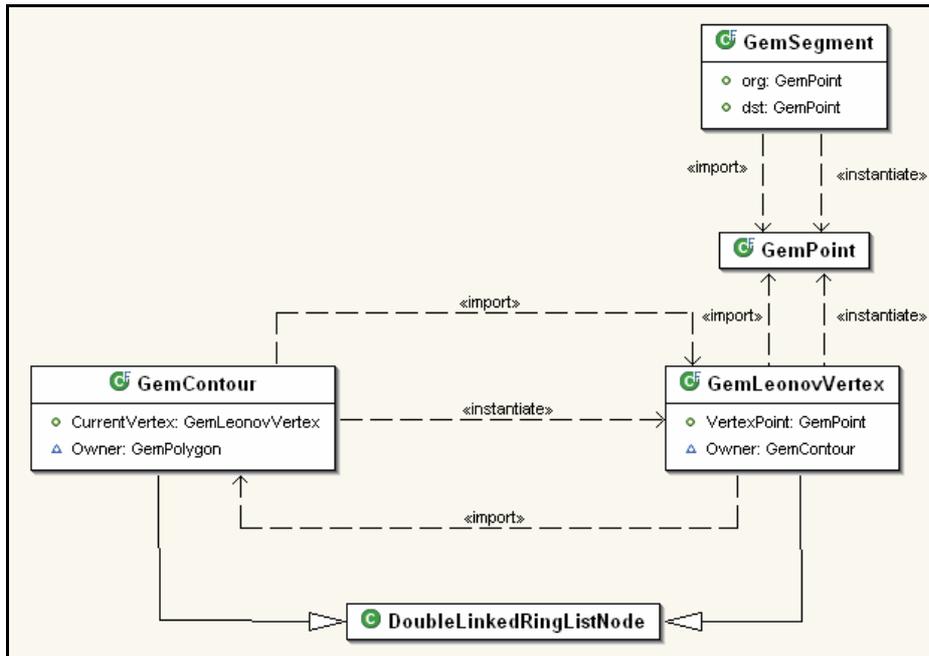


Рис. 6. Реализация метода Леонова/Никитина

Во время реализации метода выяснились очевидные противоречия в теоретических основах метода. В то время, как у Леонова/Никитина правила приема контура, не имеющего точек пересечения, в результате не зависят от направления контура по или против часовой стрелки, для успешного выполнения использованных Леоновым/Никитиным примеров это различие было необходимо.

Компьютерная реализация метода была использована для проведения тестирования затрат времени, необходимого для выполнения метода Леонова/Никитина, на двух примерах. Первый пример состоит из двух кругоподобных полигонов, состоящих из прямых сегментов, чье число варьировалось от 6 до 192. Второй пример представляет собой наложение двух гребенок, у которых число зубцов варьируется от 4 до 40. При этом число точек пересечения равно квадрату числа граней.

Результаты обоих примеров показывают, что топологическая часть алгоритма является надежной, устойчивой и быстрой. Затраты по времени геометрической части в сравнении с затратами по времени топологической части значительно доминируют, начиная с полигонов с 10 гранями. Рис. 7 показывает, что это объясняется растущими квадратично затратами по времени, необходимыми для расчета точек пересечения в стандартной форме пары операнд. Улучшение может быть достигнуто путем изменения геометрической части алгоритма. Однако известные методы, как например, алгоритм «sweep line» или «red and blue line intersections» не могут быть просто одним к одному использованы в технических чертежах, так как особые случаи касания и самокасания должны надежно распознаваться. Эта проблема является актуальной темой международных исследований.

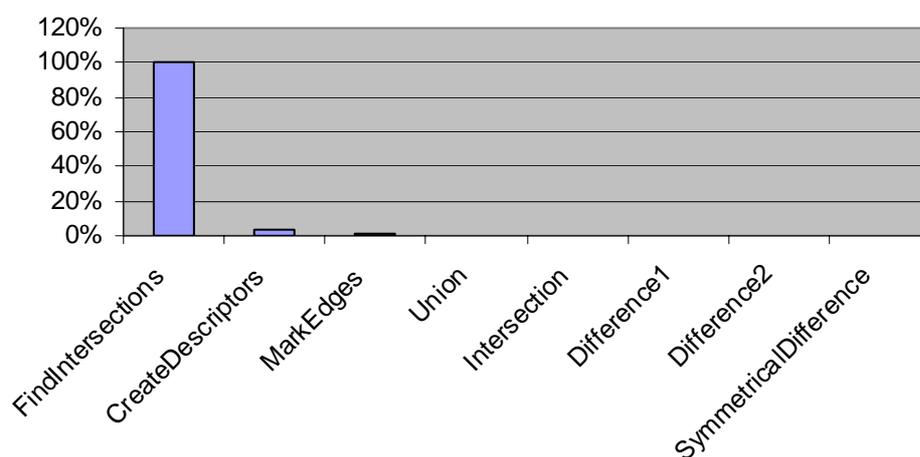
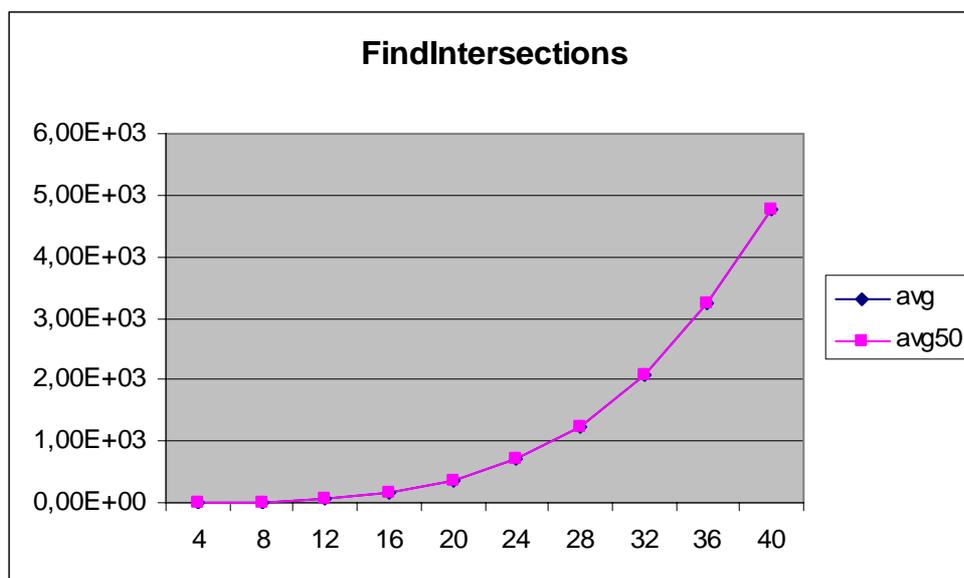


Рис. 7. Затраты по времени в миллисекундах для определения точек пересечения в прим. 2 как функция от числа граней

Для разъяснения противоречий, обнаруженных во время реализации, были систематично переработаны теоретические основы логических операций над полигонами и оформлены в виде методического пособия [6]. При этом выяснилось, что термины «регион» и «полигон» в [11] не достаточно точно определены и ограничены для использования их в технических чертежах. Для овладения сложностью чертежей их содержание должно быть описано объектно-ориентировано с помощью по возможности единообразных и простых единиц, чье число может быть большим.

Желаемое обобщение достигается тем, что каждый изображенный на чертеже объект разбивается на полигон и оформление полигона. Полигон описывает размеры объекта, а оформление — его семантику. Например, шкаф обозначается на плане этажа прямоугольником с перекрестьем, а бе-

тонная стена обозначается цветной штриховкой. Допустимые составляющие полигона показаны на рис. 8:

- 1) один наружный контур, направленный против часовой стрелки и непересекающий сам себя;
- 2) любое число внутренних контуров, направленных по часовой стрелке и непересекающих ни себя, ни другие контуры полигона;
- 3) площадь полигона, состоящая из точек пересечения площадей его контуров.

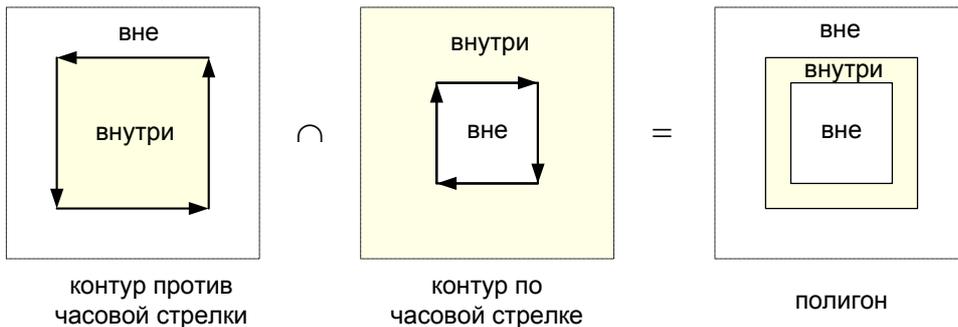


Рис. 8. Компоненты полигона

Согласно этой концепции чертеж раскладывается на многочисленные простые полигоны. Каждый из этих полигонов расположен вне другого полигона чертежа. Полигон состоит, как правило, из небольшого числа граней. С помощью некоторых предварительных геометрических тестов, делающих во многих случаях часть сложных операций ненужными, затраты по времени могли быть значительно улучшены.

Дополнительно полигоны чертежа управляются R-деревом, так что потенциально пересекающиеся полигоны чертежа могут быть найдены с незначительными затратами. Благодаря этому можно избежать выполнения тех логических операций, чей результат отличается от нуля, если операнды пересекаются.

Обобщенно можно утверждать, что метод Леонова/Никитина с учетом реализованных изменений хорошо пригоден для новой концепции.

Выводы. В рамках проведенных исследовательских работ были разработаны надежные научные основы для моделей и представления технических чертежей. Поэтому сформулированные теории, методы и процессы были опубликованы в виде методического пособия. На основе этого пособия и программного пакета, реализованного в течение проекта на Java, был проведен компактный курс для студентов старших курсов ВолгГАСУ. Учебный материал войдет в содержание магистровской программы «Computational Civil Engineering», которая будет предложена в ВолгГАСУ, начиная с октября 2007 г. Дальнейшие исследовательские работы по вычислительной геометрии следует проводить в связи с Computer-aided Facility Management.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Bilchuk, I.* Geometrical Identification of Objects in Civil Engineering Applications. IKM. Weimar, 2006.

2. *Chanturidze, A.* A Review of Algorithms for Constructing Overlays of Polygons / A. Chanturidze, V. Galishnikova. Research Report, VolgGASU, 2006.
3. *Chanturidze, A.* Implementation of the Leonov Algorithm/ A. Chanturidze, V. Galishnikova. Research Report, VolgGASU, 2006.
4. *Chanturidze, A.* Robustness and Speed of the Leonov Algorithm / A. Chanturidze, V. Galishnikova. Research Report, VolgGASU, 2006.
5. *Chanturidze, A.* Runtime Testing of the Leonov Algorithm / A. Chanturidze, V. Galishnikova. Research Report, VolgGASU, 2006.
6. *Bilchuk, I.* Computational Geometry in Civil Engineering. Chapter 4: Polygons. Monograph in preparation / I. Bilchuk, P.J. Pahl.
7. Guttman, A. R-Trees: A dynamic index structure for spatial searching // SIGMOND '84 : Proceedings of Annual Meeting, Boston. ACM Press 1984.
8. *Manolopoulos, Y.* et al. R-Trees: Theory and Applications. Springer-Verlag, London, 2006.
9. *Günther, O.* Efficient Structures for Geometric Data Management. Springer-Verlag, Berlin Heidelberg, 1988.
10. *Van Kreveld, M.* Algorithmic Foundations of Geographic Information Systems / Van M. Kreveld, J. Nievergelt, T. Roos. Springer-Verlag, Berlin Heidelberg, 1997.
11. *Леонов, М.В.* Эффективный алгоритм для замкнутого множества логических операций над полигональными регионами в плоскости / М.В. Леонов, А.Г. Никитин ; Сибирское отделение Российской академии наук. Новосибирск, 1997.

© Бильчук И.Л., 2007