

# Алгоритм поиска прямоугольников на векторной модели изображения

Алексей В. Губанов, Алексей В. Куликов

Нижегородский филиал фирмы “TESIS-Software BVBA”  
Нижний Новгород, Россия

## Аннотация

Описываемая задача поиска прямоугольников решалась в рамках общей задачи распознавания и точного определения местоположения составного объекта, заданного стилизованным шаблоном, на векторной модели, построенной из полутонового растрового изображения. На этапе построения векторной модели строятся контуры. Контур является кусочно-линейной аппроксимацией границ связанных множеств пикселей, обладающих одинаковыми характеристиками. В нашем случае для построения модели использовалась библиотека функций векторной обработки изображений в реальном времени [1]. Перечисляются возникающие в реальной системе ситуации и условия поиска. Поиск прямоугольников на построенных контурах состоит из нескольких этапов. Вначале производится фильтрация исходных контуров с использованием размеров шаблона. Каждый из оставшихся контуров поочередно рассматривается в качестве кандидата в искомые прямоугольники. На этом этапе производится попытка аппроксимировать контур наиболее близким к нему прямоугольником. Один из примененных способов – построение минимального описанного прямоугольника вокруг контура [2, 3]. Второй способ представляет собой вычисление главных осей инерции контура и построение прямоугольника с такими же характеристиками. Третий способ является комбинацией двух предыдущих. В случае, когда количество найденных прямоугольников недостаточно для определения местоположения объекта, к исходному набору контуров применяется разрезание. Разрезание контуров выполняется различными способами:

- отрезание выпуклых многоугольников, расположенных по краям большого контура
- разрезание всего контура на выпуклые многоугольники
- одновременное извлечение многоугольников, соответствующих выпуклым и вогнутым участкам контура

Производится сглаживание и анализ вышеописанным алгоритмом многоугольников, получившихся после разрезания. Алгоритм поиска учитывает заданные ограничения на допустимые искажения, возникающие в ходе получения изображения и построения векторной модели. Для последующего анализа вычисляется степень близости контура к искомому прямоугольнику. Приводятся результаты, достигнутые при реализации реальной системы поиска составных объектов, заданных шаблонами.

**Ключевые слова:** Поиск прямоугольников, Векторная модель изображения, Минимальный описанный прямоугольник, Главные оси инерции, Разрезание контуров.

## 1. ВВЕДЕНИЕ

Алгоритм разработан в рамках решения задачи определения местоположения электронных компонент на изображении, поступающем с видекамеры. Такая задача возникает, например, при автоматизации процесса сборки электронных устройств. Постановка задачи в этом случае выглядит следующим образом: определение (в реальном времени) положения объекта, заданного стилизованным шаблоном, на полутоновом растровом изображении. Шаблон включает описания границ элементов, из которых состоит искомый объект, аппроксимированных геометрическими примитивами – отрезками, прямоугольниками, окружностями, дугами и т.д. (Рис. 1) Поиск на изображении отдельных геометрических примитивов (в нашем случае – прямоугольников) является составной частью общей задачи.

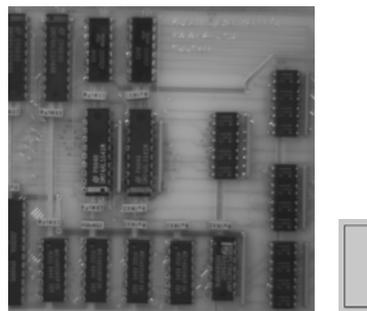


Рис. 1. Компоненты и один из вариантов их задания стилизованным шаблоном

При решении общей задачи предварительно производится построение векторной модели изображения (Рис. 2), т.е. построение контуров объектов. Контуром является кусочно-линейная аппроксимация границы связанного множества пикселей, обладающих одинаковыми характеристиками.



Рис. 2. Векторная модель изображения

Для построения векторной модели изображения используется библиотека функций векторной обработки изображений в реальном времени [1]. При этом существенно сокращается объем обрабатываемой информации, что приводит к увеличению быстродействия. После этого для поиска геометрических примитивов можно использовать алгоритмы вычислительной геометрии, например, разбиение многоугольника на выпуклые части, триангуляцию, построение минимального описанного прямоугольника и т.д. [2, 4, 5].

Качество векторной модели изображения зависит от характеристик входного изображения, способов его предобработки, алгоритмов построения модели и механизмов ее предобработки. В качестве характерных факторов, затрудняющих поиск прямоугольников, можно привести:

- наличие искажений у полученных контуров – «дрожание» границы (когда сторона прямоугольника представлена несколькими отрезками, не лежащими на одной прямой)
- наличие заведомо «малых» прямоугольников, сопоставимых по размерам с уровнем «искажений»
- отсутствие частей искомым прямоугольников
- примыкание искомым прямоугольников к более крупным объектам

В итоге, прямоугольник может быть представлен на векторной модели следующими способами:

- один контур описывает границы одного прямоугольника
- один контур описывает границы нескольких прямоугольников
- прямоугольник представлен несколькими контурами

Собственно поиск прямоугольников на полученных контурах состоит из нескольких этапов:

1. фильтрация исходных контуров по линейным размерам
2. вычисление параметров прямоугольников, аппроксимирующих контуры
3. вычисление ошибки аппроксимации контуров найденными прямоугольниками
4. разрезание контуров, потенциально содержащих границы нескольких прямоугольников
5. повторение этапов 2, 3.

Все найденные прямоугольники, имеющие ошибку аппроксимации контура, не превышающую некоторый порог, используются в дальнейшем для решения общей задачи.

## 2. ПОИСК ПРЯМОУГОЛЬНИКОВ НА ИСХОДНОМ НАБОРЕ КОНТУРОВ (БЕЗ РАЗРЕЗАНИЯ)

### 2.1 Фильтрация исходных контуров по линейным размерам

Фильтрация исходных контуров по линейным размерам является первым этапом в процессе поиска прямоугольников на исходном наборе контуров. Контур не берется в рассмотрение, если не выполнены следующие условия:

$$L_{min} * K < (H_{iu}, W_{iu}) < L_{max}, \text{ где}$$

$H_{iu}$  – высота минимального прямоугольника со сторонами, параллельными осям координат, описанного вокруг контура,

$W_{iu}$  – ширина минимального прямоугольника со сторонами, параллельными осям координат, описанного вокруг контура,

$L_{min}$  – длина минимально-возможной стороны прямоугольника с учетом допустимого изменения масштаба относительно прямоугольника, заданного шаблоном,

$K$  – коэффициент частичного присутствия прямоугольника, который может изменяться в диапазоне  $[0, 1]$ ,

$L_{max}$  – длина максимально-возможной стороны прямоугольника с учетом допустимого изменения масштаба относительно прямоугольника, заданного шаблоном.

Тем самым отбрасываются слишком малые и слишком большие контуры, которые по своим линейным размерам заведомо не могут являться искомыми прямоугольниками.

Исключение составляет случай, когда производится поиск малых прямоугольников (порог «малости» задается по площади). В этом случае контур обрабатывается, даже если он меньше наименьшего из прямоугольников шаблона. Это связано с нелинейностью величины искажений относительно размеров объекта, сильно проявляющей себя при малых величинах размеров. Величина малости зависит от разрешения изображения и способов векторизации.

### 2.2 Вычисление параметров прямоугольников, аппроксимирующих контуры

На этом этапе производится попытка аппроксимировать контур наиболее близким к нему прямоугольником, и вычисляется количественная мера близости контура к прямоугольнику. После чего производится окончательная проверка попадания вычисленных параметров в допустимый диапазон по отношению к линейным размерам текущего шаблонного прямоугольника. Каждый из оставшихся контуров рассматривается в качестве кандидата в искомый прямоугольник.

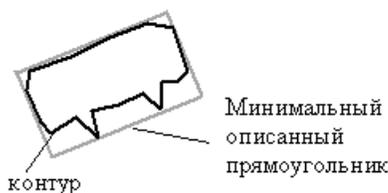
Рассмотрим способы поиска прямоугольника, наиболее близкого к контуру.

#### 2.2.1 Построение минимального описанного прямоугольника вокруг контура

Алгоритмы построения минимального описанного прямоугольника вокруг заданного многоугольника достаточно хорошо известны [2, 3]. Мы использовали для поиска минимального описанного прямоугольника алгоритм Туссена (метод «вращающихся калибров»).

Степень близости контура к аппроксимирующему прямоугольнику определяется как отношение площади контура к площади прямоугольника.

Проверенным на практике преимуществом описываемого метода является довольно точное определение угла поворота аппроксимирующего прямоугольника. Основным недостатком этого метода является большая ошибка аппроксимации при наличии на анализируемом контуре, представляющем прямоугольник, искажений типа пиков (Рис. 3).

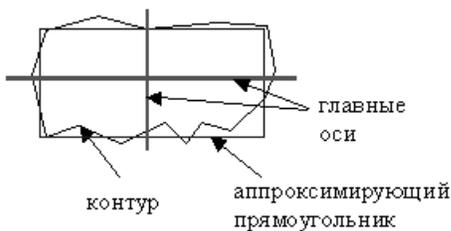


**Рис. 3.** Минимальный прямоугольник, описывающий контур

Так как степень близости в этом случае окажется небольшой, это увеличивает вероятность пропуска реальных прямоугольников на зашумленном изображении или при неудачной предобработке.

### 2.2.2 Определение моментных характеристик контура, как плоской фигуры

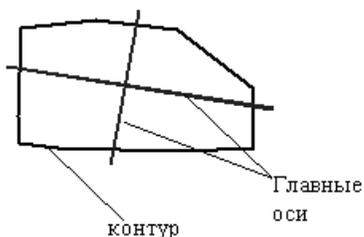
Вычисляются главные оси инерции контура, и аппроксимирующий прямоугольник строится в соответствии с их направлением и численными значениями. Фактически, на контур «накладывается» прямоугольник с такими же главными осями инерции, что и рассматриваемый контур (Рис. 4).



**Рис. 4.** Главные оси

Степень близости контура к аппроксимирующему прямоугольнику также, как и в 2.2.1, определяется как отношение площадей контура и прямоугольника.

Этот подход преодолевает недостатки предыдущего способа, но, в свою очередь, имеет свои – степень близости рассматриваемого контура к прямоугольнику получается чрезмерно завышенной, что увеличивает вероятность ложного опознавания контура как прямоугольника, даже если он в действительности таковым не является. Кроме того, при наличии искажений формы (например, после предобработки может иметь место «срезание» углов, непараллельность противоположных сторон и т.п.) направление главных осей инерции не будет совпадать с направлением сторон реального прямоугольника (Рис. 5).



**Рис. 5.** Ошибка вычисления направления сторон прямоугольника

Ошибка в определении направления особенно возрастает для контуров малого размера и контуров, описывающих границы прямоугольников, близких к квадрату.

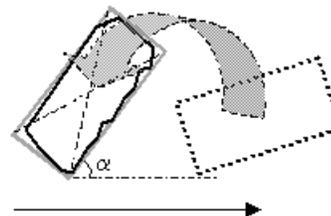
### 2.2.3 Комбинированный подход

Третий способ вычисления параметров прямоугольника является комбинацией двух предыдущих подходов. Описанными выше способами строятся два аппроксимирующих прямоугольника. Угол поворота результирующего аппроксимирующего прямоугольника (относительно оси OX) берется в соответствии с расположением минимального прямоугольника, описанного вокруг контура, а длины сторон и координаты центра определяются по прямоугольнику, построенному на главных осях инерции. Таким образом, заимствовав преимущества вышеописанных методов, удалось избавиться от их недостатков.

## 2.3 Определение местоположения

Найденный аппроксимирующий прямоугольник проверяется на сходство с исходными прямоугольниками из шаблона. Должен найтись такой прямоугольник шаблона, чтобы длины сторон найденного прямоугольника укладывались в диапазон, определяемый соответствующими сторонами искомого прямоугольника шаблона с учетом допустимого преобразования масштаба.

Если такой прямоугольник найден, вычисляется поворот, масштабирование и сдвиг прямоугольника шаблона, так, чтобы он совпал с найденным аппроксимирующим прямоугольником. (Рис. 6)



**Рис. 6.** Трансформация шаблона

Поворот определяется углом:

$$\alpha_{rect} = \alpha - \alpha_p,$$

где  $\alpha$  – угол между наибольшей стороной найденного прямоугольника и осью OX;

Масштабирование определяется коэффициентом:

$$K_{Scale} = \sqrt{\frac{S}{S_p}},$$

где  $S$  – площадь найденного прямоугольника,

$S_p$  – площадь прямоугольника шаблона.

Сдвиг определяется смещением центра прямоугольника шаблона (точки  $C$  с координатами  $C_x, C_y$ ):

$$dX = C_x - C_x^p \cdot K_{Scale} \cdot \cos(\alpha_{rect}) + C_y^p \cdot K_{Scale} \cdot \sin(\alpha_{rect})$$

$$dY = C_x - C_x^p \cdot K_{Scale} \cdot \sin(\alpha_{rect}) - C_y^p \cdot K_{Scale} \cdot \cos(\alpha_{rect})$$

где  $C_x^p, C_y^p$  – координаты точки пересечения диагоналей найденного прямоугольника.

Особым является случай поиска прямоугольника малого размера, заданного шаблоном (площадью меньше, чем

определенный на практике порог). В этом случае масштаб не вычисляется, т.к. считается, что характерные размеры такого прямоугольника уже сопоставимы с шумом, присутствующим на изображении, и вычислять масштаб нецелесообразно.

Если только одна из сторон найденного аппроксимирующего прямоугольника удовлетворяет диапазону, а перпендикулярная ей меньше нижней границы диапазона, то можно предположить, что исходный контур описывает только часть искомого прямоугольника. В этом случае прямоугольник шаблона прикладывается к контуру возможными способами (Рис. 7).



Рис. 7. Возможности приложения шаблона прямоугольника к контуру с сохранением отношения сторон

### 3. РАЗРЕЗАНИЕ КОНТУРОВ

Может иметь место ситуация, когда количество найденных прямоугольников меньше ожидаемого количества (например, меньше чем в шаблоне). Естественно предположить, что это могло произойти из-за недостатков обработки, например, когда контуры, представляющие границы искомого прямоугольника, объединились с границами других, более крупных, объектов. В этом случае производится разрезание контуров, потенциально содержащих искомые прямоугольники.

На этом этапе вначале используется дополнительное сглаживание анализируемых контуров, чтобы выделить наиболее крупные детали. В нашей задаче использовалось сглаживание по следующему критерию – площадь, заключенная между участком контура и аппроксимирующим его отрезком должна быть меньше заданного порога. Довольно сложной проблемой является при этом выбор достаточности степени сглаживания. Она выбирается как компромисс между двумя условиями: с одной стороны, сглаживание должно спрямлять длинные стороны больших прямоугольников, с другой стороны, не исказить форму так, чтобы сглаживались мелкие прямоугольники. На практике количественный порог для сглаживания был выбран в 10 раз меньше площади наименьшего прямоугольника шаблона (с учетом допустимого преобразования масштаба). Другими словами, перед разрезанием обеспечивалось отношение сигнал/шум не менее 10.

Разрезание применяется к большим относительно искомым прямоугольникам контурам и выполняется различными способами.

#### 3.1 Отрезание выпуклых многоугольников, расположенных по краям большого контура

Этот вариант является наиболее простым и быстрым, но основным его недостатком является существенная зависимость от степени предшествующего сглаживания. Предобработка контуров перед использованием алгоритма должна спрямить ложные чередующиеся вогнутые и выпуклые участки. Иначе, вместо отрезания ожидаемого кандидата в прямоугольник, могут быть отрезаны треугольные области малой площади. (Рис. 8).

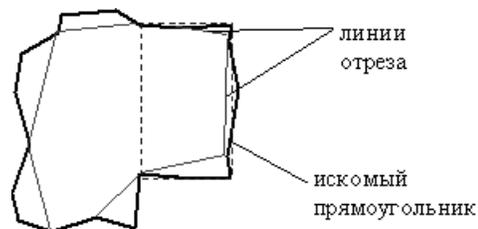


Рис. 8. Ошибки отрезания

Этот подход может быть эффективно использован в ситуациях, когда большая часть искомого прямоугольника расположена вдоль границы более крупного объекта.

Предыдущий вариант может быть усовершенствован введением дополнительного условия на отрезание. Условие накладывается на величину угла между стороной большого контура и отрезаемым многоугольником. Отрезание выполняется только при превышении этим углом некоторого порога. Это позволяет пропускать мелкие выпуклые части при разрезании.

#### 3.2 Разрезание всего контура на выпуклые многоугольники

Этот вариант является более сложным и, как следствие, более медленным. Контур разрезается на выпуклые многоугольники по определенным правилам, вдоль линий, соединяющих вогнутые точки контура (Рис. 9).

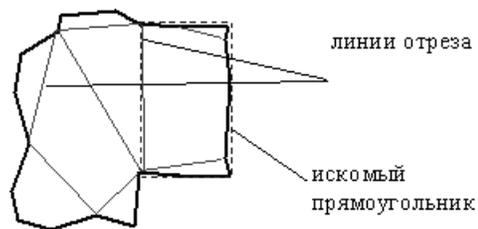


Рис. 9. Разрезание на выпуклые многоугольники

В нашей задаче дополнительно накладывались условия на длину линии отреза. После разрезания использовались только внешние из полученных многоугольников.

К положительным особенностям данного способа разрезания можно высокую вероятность правильного отрезания искомого прямоугольника (или, по крайней мере, значимой его части). Однако скорость работы алгоритма заметно снижается по мере увеличения числа точек обрабатываемого контура.

### 3.3 Одновременное извлечение выпуклых и вогнутых многоугольников

Фактически, это дальнейшее развитие варианта 3.1 (с отрезанием выпуклых многоугольников от большого контура) – одновременное извлечение при этом еще и многоугольников, соответствующих вогнутым участкам контура. Быстродействие при этом остается практически таким же. Этот подход может быть эффективен, если известно, что искомый прямоугольник может располагаться не только на выпуклой части контура, но и занимать вогнутый участок (Рис. 10).

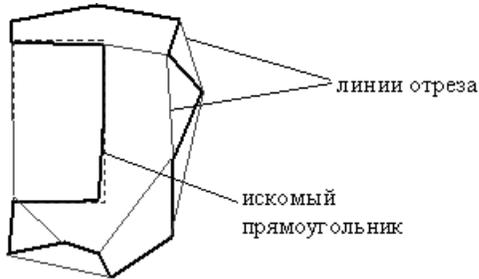


Рис. 10. Прямоугольник на вогнутом участке

Недостатком является заметное увеличение кандидатов, что может оказаться неудобным на последующих этапах анализа.

### 3.4 Выбор способа разрезания

Практически надежный и приемлемый по временным затратам результат был получен при сочетании вышеописанных вариантов разрезания. При этом в качестве основного критерия использовалось число точек в разрезаемом контуре, оставшихся после его сглаживания. Из практических целей (скорость работы алгоритма, надежность поиска и т.д.) было определено, что при количестве точек менее 100 целесообразно использовать вариант 3.2. При большем количестве точек применялся вариант 3.1, в его усовершенствованной модификации (с дополнительным условием по углу).

С разрезанными контурами производились аналогичные операции по поиску среди них кандидатов в искомые прямоугольники. Все они, имеющие ошибку аппроксимации ниже некоторого заданного порога, добавлялись к уже найденному ранее множеству прямоугольников – для использования на последующих этапах решения общей задачи распознавания и точного определения местоположения всего составного объекта.

### 3.5 Проблемы «шумов» и погрешностей, возникающих в ходе работы алгоритмов

Как уже отмечалось выше, на эффективность поиска прямоугольников заметно влияют как предобработка (способ построения контуров, их степень сглаживания), так и собственно взаимодействие описываемых алгоритмов.

Так, например, искажения, возникающие в процессе построения контуров в ходе предобработки, и имеющие характер «дрожания» границы контура (Рис. 11), заметно сказываются на точности поиска прямоугольника, наиболее близкого к анализируемому контуру (п. 2.2).



Рис. 11. «Дрожание» границы контура

Степень сглаживания, применяемого как с целью сокращения числа обрабатываемых точек контура (и, следовательно, повышения быстродействия), так и с целью «огрубления» формы контуров и выделения наиболее крупных деталей, требует определенной оптимизации. Так, в случае недостаточного сглаживания затрудняется отрезание выпуклых частей от контура на прямых в среднем участках (п. 3.1). Наоборот, при чрезмерной степени сглаживания происходит заметное искажение формы контура (Рис. 12).

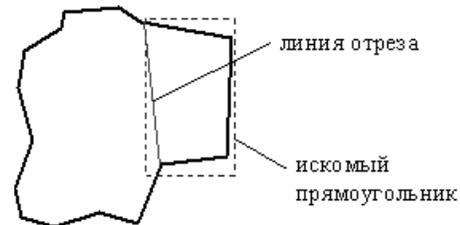


Рис. 12. Искажение формы

Операции разрезания, в свою очередь, вносят искажения в форму деталей, отрезанных от контура. Все это неблагоприятно сказывается на последующем анализе отрезанных от контура многоугольников, среди которых, собственно, и ищутся ожидаемые прямоугольники.

Наибольшую чувствительность к вышеперечисленным факторам имеет, как это было определено в ходе совершенствования алгоритма, операция определения угла поворота найденного прямоугольника по отношению к прямоугольнику шаблона. Для компенсации воздействия этих факторов вычислялась допустимая ошибка по углу – в соответствии с «весом» каждого из факторов (формы прямоугольника, уровня «шумов» и погрешности разрезания):

$$\delta_{\alpha} = W_S \cdot \delta_{square} + W_N \cdot \delta_{noise} + W_C \cdot \delta_{cut}$$

где  $\delta_{\alpha}$  – допустимая ошибка по углу,

$\delta_{square}$  – ошибка, связанная с формой прямоугольника, определяется степенью его близости к квадрату,

$\delta_{noise}$  – ошибка, определяемая соотношением между диагональю прямоугольника и уровнем «шумов»,

$\delta_{cut}$  – ошибка, связанная с разрезанием, нормированная на угол симметрии прямоугольника,

$W_S, W_N, W_C$  – веса ошибок.

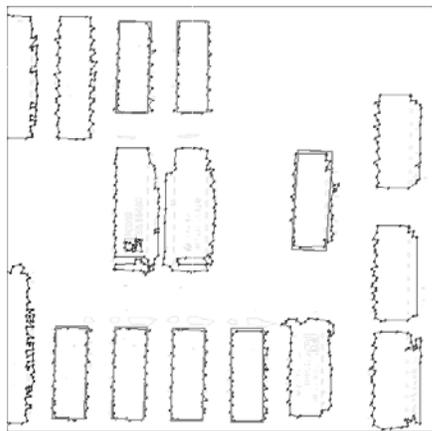
## 4. ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Описанный алгоритм поиска прямоугольников обладает следующими свойствами:

- допускает различные варианты входного изображения (искомые прямоугольники могут соответствовать отдельным контурам или быть частями одного контура);

- успешно работает при относительно низком качестве векторной модели (допускает различные искажения формы контуров типа «пиков», неровностей границы, отклонений сторон от прямоугольности, отсутствия частей прямоугольников);
- нечувствителен как к «белому» шуму, так и к шуму типа «соль и перец»;
- имеет высокое быстродействие;
- легко настраивается на любые прямоугольники (с помощью шаблона).

Приложение, в составе которого был реализован данный алгоритм, использовалось для распознавания объектов, заданных шаблонами, состоящими из прямоугольников различной формы и размера. База данных, на которой проводилось тестирование, состояла приблизительно из 1000 изображений электронных компонент различного качества (около половины из них составляли изображения с частично видимыми объектами). Размер исходных изображений составлял, в среднем, около 500x500 пикселей.



**Рис. 13.** Результаты поиска  прямоугольника

Проведенные тесты полностью подтвердили свойства алгоритма. Среднее время поиска объекта составило 120-150

мс на PC Pentium III с тактовой частотой 500 МГц, при этом надежность результатов на полностью видимых объектах составила около 93%. Также были получены хорошие результаты при поиске частично видимых объектов, т.е. когда часть объекта отсутствует на изображении. Надежность результатов при этом составила 83%. Пример результата поиска прямоугольников приведен на Рис. 13.

## 5. ЛИТЕРАТУРА

- [1] Бовырин А.В., Губанов А.В., Колесов А.И., Курякин В.Ф., Родюшкин К.В., Чудинович Б.М. *Библиотека функций векторной обработки видео изображений в реальном времени*. // IX конференция “Математические методы распознавания образов”, тез. док. М. 1999, С. 145-147.
- [2] Toussaint G. *Solving geometric problems using “rotation calipers”*. Proc. IEEE MELECON 83, 1983.
- [3] Illingworth, Kittler, Fogleyn, Paler, Phillips *A Computer Vision Solution to a Packaging Problem*. Digital Systems for Industrial Automation, 1987.
- [4] Ф. Препарата и М. Шеймос. *Вычислительная геометрия. Введение*. М., «Мир», 1989.
- [5] De Berg... *Computational Geometry Algorithms and applications*. М., 1997.

## Авторы

Алексей В. Губанов, Нижегородский филиал фирмы “TESIS-Software BVBA”.

E-mail: [gubanov@tesis.nnov.ru](mailto:gubanov@tesis.nnov.ru)

Алексей В. Куликов, Нижегородский филиал фирмы “TESIS-Software BVBA”.

E-mail: [kulikov@tesis.nnov.ru](mailto:kulikov@tesis.nnov.ru)

Авторы выражают благодарность Олегу Александровичу Башкирову за полезные обсуждения и помощь в рецензировании работы.