

## КОМПЬЮТЕРНАЯ ГРАФИКА И ВИЗУАЛИЗАЦИЯ

УДК 004.92

### УПРОЩЕНИЕ CAD-МОДЕЛЕЙ ПУТЕМ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ И ПОДАВЛЕНИЯ ЦЕПОЧЕК СКРУГЛЕНИЙ

© 2020 г. С. Е. Сляднев<sup>а,\*</sup>, В. Е. Турлапов<sup>а,\*\*</sup>

<sup>а</sup> Нижегородский государственный университет им. Н.И. Лобачевского  
603950 Нижний Новгород, пр. Гагарина, д. 23, Россия

\*E-mail: sergey.slyadnev@gmail.com

\*\*E-mail: vadim.turlapov@itmm.unn.com

Поступила в редакцию 20.12.2019 г.

После доработки 09.01.2020 г.

Принята к публикации 19.01.2020 г.

Описана процедура упрощения CAD-моделей путем распознавания и подавления некоторых типов скруглений и их цепочек. Предлагаемый метод основан на эйлеровых операторах KEV, KEF и KFMV, реализованных на базе геометрического ядра с открытыми исходными кодами. Упрощение задействует два этапа, а именно, распознавание скруглений и их подавление с гарантией топологической и геометрической целостности результата. Описанный подход ориентирован на использование в автоматическом режиме, предъявляющем высокие требования к надежности алгоритма. Ключевыми свойствами разработанного подхода являются надежность, предсказуемость результата и расширяемая архитектура, допускающая добавление новых топологических случаев без изменения основной процедуры упрощения. Распознавание состоит в построении графа смежности граней с атрибутами, содержащими информацию о типах ребер, их свойствах и предполагаемых видах скруглений. На этапе подавления, алгоритм итеративно проходит граф смежности граней, формируя цепочки скруглений. Для каждой грани в цепочке осуществляется распознавание локальной топологической ситуации, определяющей способ подавления в терминах эйлеровых операторов. Алгоритм может быть расширен путем добавления дескрипторов новых топологических ситуаций. После применения эйлеровых операторов затронутые ребра перестраиваются для получения геометрически корректного граничного представления.

DOI: 10.31857/S0132347420030085

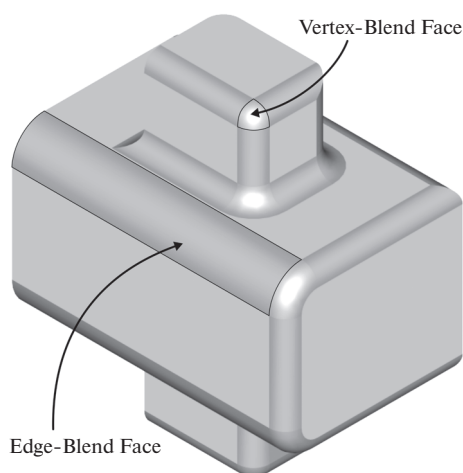
#### 1. ВВЕДЕНИЕ

Упрощение CAD-моделей и сборок выполняется для решения разнообразных инженерных задач. Среди прочих назовем подготовку геометрии к расчетам, сжатие данных для эффективной визуализации, защиту интеллектуальной собственности при обмене информацией, повышение производительности работы в системах проектирования, устранение вторичных конструктивных элементов для обеспечения распознавания первичных и т.п. В системах параметрического проектирования, основанных на истории построения, упрощение CAD-модели может быть выполнено путем исключения соответствующего набора конструктивных элементов из дерева операций. В случае же, если история построения отсутствует или не содержит искомым конструктивных элементов, такой подход неприменим. Возникает необходимость использования операторов прямого редактирования, которые на сегодняшний день отсут-

ствуют в открытых библиотеках геометрического моделирования.

В данной работе мы описываем оператор подавления цепочек скруглений, основанный на распознавании конструктивных элементов с последующим применением эйлеровых операторов. Вкладом настоящей работы являются: а) реализация эйлеровых операторов KEV, KEF и KFMV на основе открытого ядра геометрического моделирования; б) разработка расширяемой архитектуры распознавания и подавления цепочек скруглений; в) реализация алгоритма подавления целого ряда конструктивно-значимых типов цепочек скруглений.

Предлагаемый алгоритм допускает работу в глобальном и локальном режимах. В первом случае осуществляется инкрементное подавление, т.е. цепочки удаляются одна за другой с промежуточным перестроением вспомогательных структур данных. Во втором — локальном — режиме, распознавание цепочки начинается из грани, вы-



**Рис. 1.** Распознаваемые типы скруглений: EBF (edge-blend face) и VBF (vertex-blend-face).

бранной пользователем в качестве кандидата, притом остальные цепочки оказываются нетронуты. Алгоритм подавления накапливает непрерывную историю модификации модели, позволяющую соотнести результирующее граничное представление с исходным, обеспечивая, тем самым, возможность сохранения ассоциированных данных (цветов, имен, аннотаций, допусков и т.п.).

Далее в разделе 2 дан обзор наиболее значимых работ, посвященных распознаванию и подавлению цепочек скруглений. В разделе 3 описан наш подход к распознаванию скруглений на основе атрибутированного графа смежности граней. В разделе 4 представлен алгоритм подавления скруглений и инкрементная процедура для его применения в пакетном режиме. Раздел 5 содержит примеры, демонстрирующие использование алгоритма на CAD-моделях из повседневной практики авторов. Там же дается сравнение разработанного метода с более общим алгоритмом удаления граней. Вопросы, оставленные для дальнейшего исследования, приведены в разделе 6.

## 2. СОСТОЯНИЕ ПРОБЛЕМЫ

Публикации, посвященные проблеме упрощения CAD-моделей и сборок, были рассмотрены нами в работе [1]. Там же дано описание соответствующего программного комплекса. Остановимся на результатах, имеющих отношение преимущественно к подавлению цепочек скруглений.

Упрощение CAD-модели состоит в геометрическом преобразовании формы сообразно функциональным и технологическим требованиям, предъявляемым к детали. Поскольку инженерная семантика модели находит отражение в ее конструктивных элементах (КЭ), для упрощения гео-

метрии необходимо выделить КЭ в явном виде. Наличие параметрической модели с историей построения иногда позволяет обойтись без вычислительно сложных процедур, таких как распознавание КЭ или морфологический анализ модели [2]. К сожалению, параметрические модели доступны не всегда, и даже в случае их наличия, история построения может не содержать искоемых конструктивных элементов. Так, глухое отверстие, заданное в параметрической модели, может преобразоваться в сквозное при удалении из модели нового элемента объема. Аналогичным образом, скругления, заданные на плоском эскизе, при вытягивании призматического тела не будут автоматически преобразованы в конструктивные элементы скруглений на ребрах. Таким образом, для упрощения геометрии возникает необходимость в привлечении средств прямого редактирования даже в тех случаях, когда параметрическая модель доступна.

Авторы работ [3, 4] описывают процедуру подавления скруглений, реализованную в популярной САПР Rhinoceros. Алгоритм выполняет предварительную классификацию граней по типам EBF (edge-blend face) и VBF (vertex-blend face) (рис. 1). Затем алгоритм рассчитывает новые геометрические носители граничных элементов (ребро для скруглений типа EBF и вершина для VBF). Топология модели изменяется на завершающем этапе процедуры и не предугадывается изначально.

Авторы работ [5, 6] описывают “топологическую” процедуру подавления скруглений, основанную на применении эйлеровых операторов с последующим “стягиванием” геометрии. В отличие от эвристических “геометрических” подходов, таких как [4], алгоритмы [5, 6] заранее готовят программу преобразования топологии модели в виде последовательности эйлеровых операторов, гарантирующих целостность граничного представления. Предсказуемость результата, достижимая благодаря локальному анализу топологии скруглений, делает, на наш взгляд, алгоритмы [5, 6] предпочтительными для автоматического упрощения CAD-моделей в пакетном режиме. Так, сложные конфигурации скруглений, которые не могут быть обработаны в силу ограничений алгоритма, окажутся нераспознанными еще на начальной стадии и будут пропущены. Таким образом, алгоритм примет в обработку только те топологические случаи, для которых достоверно известна соответствующая программа эйлеровых операторов. С другой стороны, реализация подобных алгоритмов требует наличия геометрического ядра, содержащего эйлеровы операторы (например, Parasolid или ACIS). Поскольку единственное открытое геометрическое ядро OpenCascade данных операторов не предоставляет, одной из

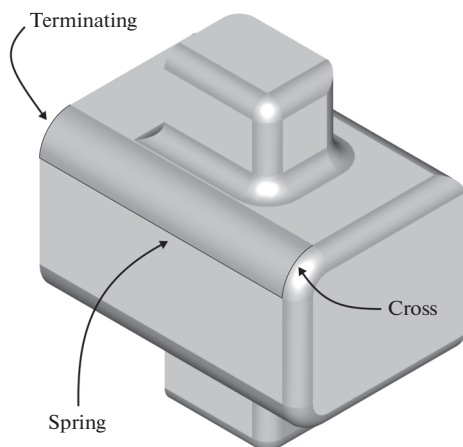


Рис. 2. Типы ребер, ограничивающих грани скруглений.

наших задач в контексте настоящего исследования являлась их реализация.

Распознавание элементов скруглений на граничном представлении считается относительно простой задачей (в отличие от задачи их подавления). Действительно, всякая грань задается точным уравнением параметрической поверхности  $s(u, v)$ , имеющей, как правило, явный спецификатор собственного типа на уровне структур данных (плоскость, цилиндр, сфера, тор, сплайн и т.п.). На основании данного спецификатора и дифференциальных свойств поверхности, можно заключить, является ли некоторая грань элементом скругления.

Распознавание цепочек скруглений требует локального анализа топологии в окрестности нескольких граней-кандидатов. Достаточно общий (с практической точки зрения) алгоритм распознавания цепочек, дающий также порядок их построения, был предложен в работе [7]. Работа [6] дополняет результат, оформленный в [7], алгоритмом подавления, где в качестве одного из этапов задействован оператор удаления грани [8] тех же авторов.

Комплекс работ [6–8] дает, на наш взгляд, принципиальное решение проблемы распознавания и подавления скруглений. Как указано выше, основная идея этого “топологического” подхода состоит в угадывании структуры результирующей модели с последующим преобразованием исходного граничного представления к заранее известному результату. Наша работа, таким образом, состоит в формализации и реализации данного принципа с использованием открытых средств геометрического моделирования.

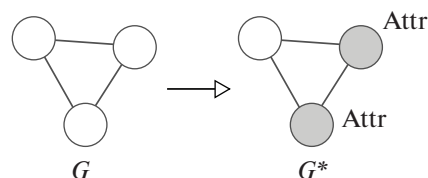


Рис. 3. Обогащение графа смежности граней  $G$  атрибутами как результат процесса распознавания скруглений.

### 3. РАСПОЗНАВАНИЕ СКРУГЛЕНИЙ

Для подавления цепочек скруглений их следует автоматически распознать в исходной модели. На данном этапе задача состоит не только в поиске собственно граней, но также в классификации принадлежащих им ребер (рис. 2). Классификация осуществляется по следующим типам: а) опорные ребра (spring edges); б) поперечные ребра (cross edges); в) терминальные ребра (terminating edges). Результат распознавания помещается в качестве атрибутов графа смежности граней. На данном этапе цепочки не формируются. Техника распознавания, преобразующая граф смежности граней с целью определения последовательности эйлеровых операторов, ранее не публиковалась.

Кратко, процесс распознавания состоит в следующем:

1. Построение атрибутированного графа смежности граней (attributed adjacency graph, AAG).
2. Распознавание граней типа EBF (edge-blend faces).
3. Распознавание граней типа VBF (vertex-blend faces).
4. Уточнение терминальных ребер (терминальное ребро не может содержаться в двух гранях, распознанных как элементы скруглений).
5. Извлечение результата.

Для распознавания граней типа EBF и VBF используется набор геометрических и топологических эвристик, рассматриваемых ниже.

В результате процедуры распознавания, исходный граф смежности граней  $G$  преобразуется в эквивалентный граф  $G^*$ , снабженный дополнительными атрибутами, ассоциированными с его вершинами (рис. 3). Основными являются два типа атрибутов: а) элемент скругления (значение атрибута  $A(\cdot) = 1$ ); б) несущая грань (значение атрибута  $A(\cdot) = 2$ ).

#### 3.1. Граф смежности граней

Ключевым аппаратом предлагаемой архитектуры распознавания является атрибутированный граф смежности граней (attributed adjacency graph,

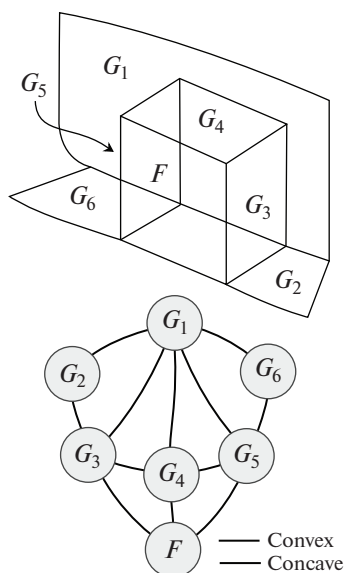


Рис. 4. Атрибутированный граф смежности граней.

ААГ). Эта структура данных может применяться для решения широкого круга задач распознавания. Так, благодаря графу смежности, содержащему данные о выпуклости и вогнутости ребер (рис. 4), удастся различить некоторые базовые конструктивные элементы, например, цилиндрические отверстия, карманы и выступы [14]. ААГ является удобной структурой хранения данных, позволяющей акцентировать отношение смежности, извлекаемое путем анализа топологии граничного представления модели. Перечислим основные достоинства графа смежности граней в решении задач распознавания конструктивных элементов.

Во-первых, ААГ позволяет осуществить переход от геометрического описания модели к соответствующему формализму теории графов. В процедурах типизации моделей (распознавание листовых тел, труб, 2.5D-моделей и проч.) применение находят такие операции на графах, как выделение компонент связности, поиск подграфа, нахождение кратчайшего пути, анализ степеней вершин и т.д.

Во-вторых, граф смежности играет роль кратковременного хранилища (кэша) для идентификаторов ребер, вершин и граней модели. Граничное представление модели в библиотеке OpenCascade не содержит уникальных идентификаторов граничных элементов в явном виде. Эти идентификаторы должны быть извлечены в процессе анализа топологических структур, что может негативно сказаться на производительности соответствующего программного обеспечения. Наличие кэша идентификаторов в составе ААГ позволяет использовать его в качестве «ускоряющей структуры данных».

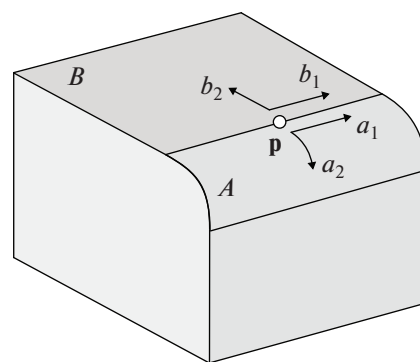


Рис. 5. Анализ кривизны для определения грани-скругления.

В-третьих, ААГ является носителем семантики граничного представления, дополняя его информацией об извлеченных в процессе распознавания конструктивных элементах.

### 3.2. Распознавание граней EBF

Распознавание граней типа EBF опирается на классификацию всех ребер модели по следующим типам (рис. 2).

1. Опорное ребро (spring edge). Опорные ребра — это гладкие ребра-направляющие для скруглений типа EBF.

2. Поперечное ребро (cross edge). Поперечные ребра могут быть гладкими и негладкими. Они характеризуются тем, что в них реализуется смежность двух граней скруглений (EBF или VBF).

3. Терминальное ребро (terminating edge). Эти ребра завершают цепочки скруглений. В случае замкнутой цепочки терминальные ребра отсутствуют.

На первом этапе выполняется поиск всех гладких ребер. Поскольку классификация двугранных углов осуществляется еще на этапе построения ААГ, данный этап сводится к выборке соответствующих ребер по атрибутам из графа смежности.

Из набора гладких ребер выделяются опорные ребра граней типа EBF. С этой целью осуществляется анализ главных кривизн на ребре-кандидате. Соотношения между абсолютными величинами кривизн позволяют сделать предположение о том, является ли грань  $A$  с кривизнами  $a_1$ ,  $a_2$  гранью скругления (рис. 5). Так, должны выполняться следующие соотношения:

1. Абсолютная величина  $a_2$  превосходит абсолютную величину  $a_1$ , то есть кривизна в поперечном направлении предполагаемого скругления доминирует.

2. Абсолютная величина  $a_2$  превосходит абсолютную величину  $b_2$  не менее, чем в  $c$  раз, где  $c > 1$ .

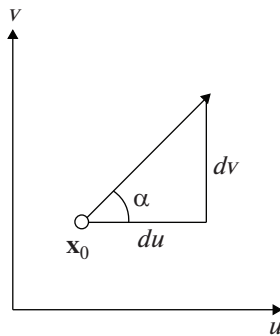


Рис. 6. К измерению кривизны вдоль ребра.

Эта эвристика выражает то наблюдение, что искомая грань EBF опирается на менее искривленную поверхность, например, на плоскость.

Радиус предполагаемого скругления полагается равным  $1/a_2$ .

Поперечные ребра можно найти из условия, что кривизна грани EBF вдоль такого ребра совпадает с главной кривизной  $a_2$ , полученной ранее. Вычисление кривизны поверхности вдоль кривой осуществляется по известной формуле дифференциальной геометрии, привлекающей коэффициенты первой и второй квадратичных форм поверхности:

$$k(\lambda) = \frac{L + 2M\lambda + N\lambda^2}{E + 2F\lambda + G\lambda^2}$$

Здесь  $\lambda = dv/du = \tan\alpha$ ,  $(u, v)$  – параметры несущей поверхности (рис. 6).

Терминальное ребро не является гладким и не содержится во множествах опорных и поперечных ребер. Замкнутая цепочка не имеет терминальных ребер. Незамкнутая цепочка ограничена терминальными ребрами. Некоторые терминальные ребра доопределяются до негладких поперечных (см. раздел 3.4).

### 3.3. Распознавание граней VBF

Следующей стадией распознавания цепочек является выделение граней типа VBF (vertex-blend face). На данном этапе привлекается набор топологических эвристик, подсказанных типичными конфигурациями скруглений на реальных моделях. Во-первых, грань VBF не может соседствовать с другой гранью того же типа (эта эвристика отражает ограничение алгоритма). Во-вторых, для грани типа VBF должно найтись не менее трех смежных граней типа EBF, где смежность реализуется через поперечные (нетерминальные) ребра. Распознанные грани типа VBF получают соответствующий атрибут в графе смежности.

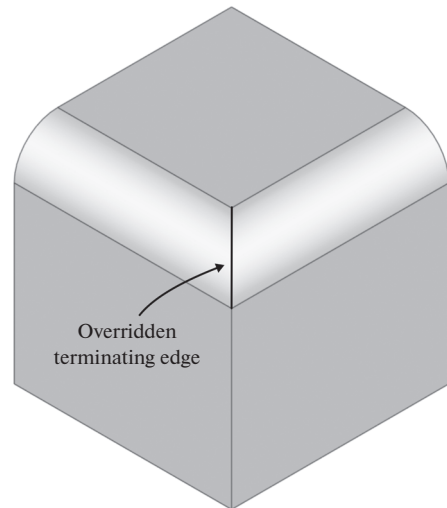


Рис. 7. Типичный пример терминального ребра, доопределенного до поперечного ребра.

### 3.4. Уточнение терминальных ребер

Наконец, когда все грани скруглений промаркированы, происходит уточнение терминальных ребер. Для каждого терминального ребра  $e$  разыскивается пара граней с индексами  $(f_e, g_e)$  таких, что  $A(f_e) = A(g_e) = 1$ , то есть обе грани являются элементами цепочки скруглений. Поскольку терминальное ребро не должно встречаться внутри цепочки, его тип доопределяется до поперечного ребра. Особенностью доопределенных поперечных ребер является тот факт, что они не реализуют гладких сопряжений между элементами скруглений (рис. 7).

## 4. ПОДАВЛЕНИЕ СКРУГЛЕНИЙ

Подавление скруглений и их цепочек заключается в удалении соответствующих граней и стягивании соседних для получения “водонепроницаемой” оболочки. Собственно удаление может быть реализовано несколькими способами: а) исключением граней из топологической структуры модели с последующим затягиванием разрывов; б) применением последовательности эйлеровых операторов. В первом случае удаление выполняется в известной степени “вслепую”, так как нет гарантии, что топологическая конфигурация в окрестности цепочки допускает корректное стягивание разрыва. Так, для управления процессом стягивания, авторы работы [10] осуществляют доплатительный анализ, сопоставляя локальную топологическую конфигурацию цепочки гомеоморфную ей абстрактную область. В случае, если для удаления используются эйлеровы операторы [11], топологические разрывы оказываются невозможными по определению. С другой стороны,

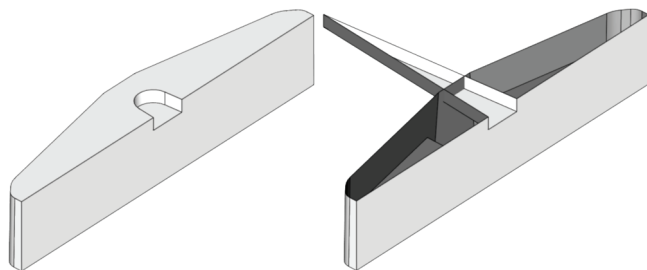


Рис. 8. Нарушение топологической целостности модели при удалении внутреннего скругления.

эйлеров оператор выполняет лишь синтаксическое преобразование модели, оставляя ее семантику (геометрию) нетронутой. Следовательно, после применения эйлеровых операторов необходимо локальное перестроение геометрии. Данный этап мы называем процессом “нормализации геометрии”.

Формально, процесс удаления скруглений состоит в следующей последовательности действий:

1. Обходом графа  $G^*$  формируются цепочки скруглений в виде множеств граней модели  $F_k = \{f_j : j = N_k, M_k\}$ . Здесь  $k = 1, K$ , где  $K$  – количество цепочек;  $N_k, M_k$  – начальный и конечный индексы граней в цепочке с номером  $k$ . Извлекаются только те вершины графа AAG, которым отвечает подграф  $C_k \subset G^*$ , где  $A(V_i(C_k)) = 1$ . Функция  $V_i(\cdot)$  возвращает вершину графа  $f_i$  с номером  $i = N_k, M_k$ . Функция  $A(\cdot)$  возвращает значение атрибута для данной вершины (1 для элементов скруглений, 2 для опорных граней и 0 для остальных).

2. Грани цепочки  $F_k$  удаляются из модели. Граф  $G^*$  играет вспомогательную роль в процессе удаления граней, поэтому его перестроение (или модификация) необходимо лишь в том случае, если подавление будет продолжено для следующей цепочки, т.е.  $k \neq K$ .

3. Затронутые ребра модели перестраиваются для “стягивания” опорных граней.

#### 4.1. Базовый алгоритм

В базовом алгоритме подавления рассматриваются следующие топологические случаи для единичной грани:

1. Данная грань является изолированным скруглением.
2. Данная грань входит в замкнутую или незамкнутую цепочку скруглений, состоящую из граней типа EBF и VBF.

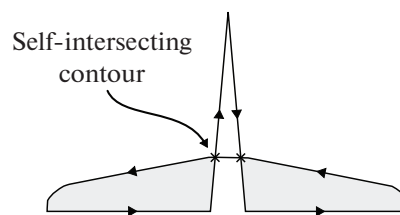


Рис. 9. Самопересечения контура базовой грани.

На предварительном этапе выполняется проверка возможности подавления цепочки в целом. С этой целью используются дополнительные топологические эвристики. В частности, любое поперечное ребро цепочки должно содержаться ровно в двух гранях той же цепочки. В противном случае цепочка оказывается распознанной не полностью и считается неподдаваемой.

Последующие этапы алгоритма подавления состоят в следующем:

1. Применение операторов Эйлера для топологической подготовки результата.

2. Выполнение операции “стягивания” ребер для адаптации геометрии к новой топологической структуре (нормализация).

3. Апостериорная проверка корректности затронутых граней.

Шаг 1 состоит в последовательном применении операторов KEV (kill-edge-vertex), KEF (kill-edge-face), KFMV (kill-face-make-vertex) в порядке, определяемом типом той или иной топологической конфигурации.

Шаг 2 выполняет чисто геометрические построения на predetermined топологической структуре модели. На данном этапе реконструируются затронутые топологической операцией ребра, вершины и грани. Так, для получения новой несущей кривой ребра, осуществляется пересечение соответствующих опорных поверхностей.

Шаг 3 предохраняет алгоритм от возможных ошибок геометрических построений (пересечение поверхностей, проецирование кривой на поверхность и т.п.), связанных с небезупречной надежностью геометрического ядра. Кроме того, на данном этапе выполняется проверка самопересечений границы области определения  $D$  для каждой затронутой грани. Последнее необходимо, поскольку подавление скруглений может, вообще говоря, привести к изменению количества компонент связности области  $D$  (рис. 8–9). Предполагается, что количество компонент связности данной области есть инвариант оператора подавления скруглений. Рассмотрение альтернативных инвариантов, таких как ограничивающий прямоугольник области  $D$ , позволит ввести новые режимы работы алгоритма подавления.

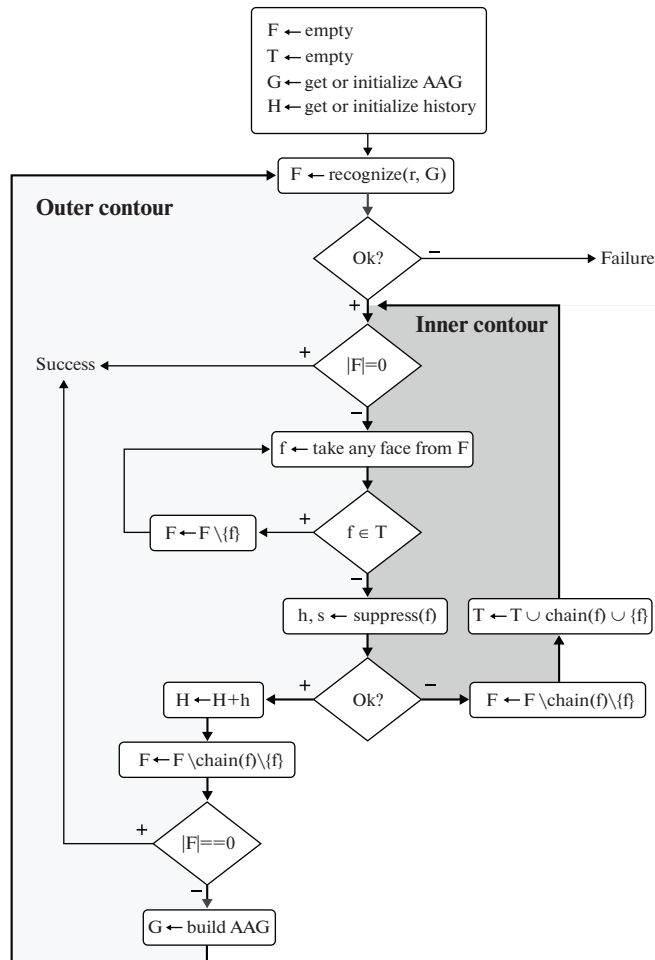


Рис. 10. Блок-схема инкрементной процедуры подавления.

#### 4.2. Инкрементная процедура

Базовый алгоритм, описанный выше, позволяет упростить CAD-модель путем подавления единственной выбранной цепочки скруглений. Для упрощения моделей в пакетном режиме (без привлечения пользователя) этого недостаточно. Для удаления всех цепочек скруглений, реализована вспомогательная процедура так называемого инкрементного подавления.

Инкрементная процедура начинается с конструирования атрибутированного графа смежности граней  $G$ . Граф  $G$  должен быть перестроен всякий раз, когда изменяется топология модели, поскольку вершины графа адресуют граничные элементы модели через их целочисленные индексы, неустойчивые к операторам модификации [12].

Процедура реализует два вложенных цикла обработки данных (рис. 10). На каждой итерации внешнего цикла, происходит распознавание цепочек скруглений  $F$  всей модели. После выбора

произвольной грани  $f \in F$ , выполняется попытка подавления соответствующей ей цепочки  $chain(f)$ . В случае успеха, граф  $G$  перестраивается для обновленной модели  $s$  и осуществляется конкатенация локальной истории модификаций  $h$  в глобальную историю  $H$ . В случае же, если цепочка  $chain(f)$  не может быть удалена (внутренний цикл), соответствующие грани изымаются из очереди  $F$  на подавление, и выбирается новая цепочка без перестроения графа смежности. Кроме того, алгоритм сохраняет множество адресов неподавляемых граней  $T$ , которые, как правило, остаются актуальными даже при изменении топологии (это т.н. “транзиентные индексы” граничных элементов модели). Во внутреннем контуре алгоритм перебирает цепочки до тех пор, пока подавление не будет осуществлено, либо не опустеет очередь  $F$ .

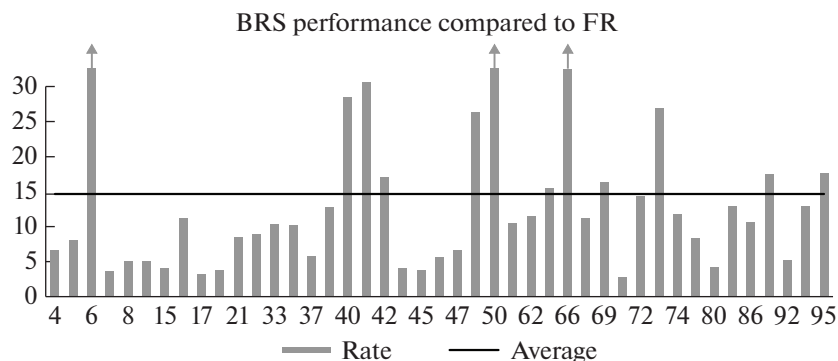


Рис. 11. Сравнение производительности алгоритма BRS с алгоритмом FR.

#### 4.3. История модификаций

Как и любой оператор редактирования формы, оператор подавления должен сохранять ассоциативность между граничными элементами результата и исходной модели. С этой целью в алгоритм подавления введена структура данных, представляющая историю модификации топологических элементов в виде графа. Поскольку описанная выше инкрементная процедура состоит в неоднократном вызове оператора подавления, соответствующие структуры данных подлежат конкатенации (“склейке”) для получения непрерывной истории каждого граничного элемента.

### 5. СРАВНЕНИЕ

Для подавления скруглений и их цепочек можно задействовать оператор удаления граней общего назначения (face removal, FR). Принцип алгоритма FR состоит в изъятии целевой грани или набора граней из состава модели с последующим стягиванием соседних граней и восстановлением смежности [8]. В библиотеке OpenCascade данный подход реализован в пакете BRepAlgoAPI. Алгоритм распознавания и подавления скруглений (blends recognition and suppression, BRS), предложенный в настоящей работе, является альтернативой алгоритму FR.

В Таблице 1 приведено сравнение двух алгоритмов по трем критериям: время работы в секундах ( $T_{FR}$  и  $T_{BRS}$ ), корректность полученного результата ( $V_{FR}, V_{BRS} \in \{0, 1\}$ ), эффективность в смысле количества удаленных граней ( $N_{FR}$  и  $N_{BRS}$ ). Эксперимент проводился на аппаратном обеспечении Intel(R) Core(TM) i5-9500 CPU @ 3.00GHz, 32GB RAM.

Величина  $r_{max}$  задает ограничение на максимальный радиус распознаваемых скруглений. Символом “i” (“interactive”) отмечены замеры времени для случаев, где оказалось необходимым привлечение пользователя. Алгоритмы запуска-

лись на заранее отобранных тестовых данных, доступных публично [16]. Для запуска алгоритма FR использовался предварительный этап распознавания цепочек скруглений, описанный в разделе 3. Замеры времени для алгоритма FR не включают этап распознавания и не учитывают пользовательских действий. Замеры времени для алгоритма BRS включают как распознавание, так и подавление, поскольку данные этапы в алгоритме BRS неотделимы. Символом “d” (“design intent”) отмечены результаты упрощения, являющиеся формально корректными, но нарушающие замысел проектировщика на уровне конструктивных элементов модели. Символ “?” означает, что соответствующий результат получить не удалось (алгоритм не завершился за приемлемое время, составлявшее 5–10 минут).

Ключевым свойством алгоритма автоматического упрощения CAD-моделей является его надежность. Ячейки, выделенные в Таблице 1 серым цветом, отвечают результатам, демонстрирующим недостаточную надежность соответствующего алгоритма.

Алгоритм BRS работает в среднем в 14.6 раза быстрее (рис. 11), чем алгоритм FR для случаев, когда оба алгоритма результативны, т.е.  $N_{FR} \neq 0$  и  $N_{BRS} \neq 0$ . Более высокая производительность инкрементной процедуры BRS объясняется, прежде всего, тем, что предложенный алгоритм является локальной операцией (стр. 287 в [17], см. также [18]), тогда как задействованный алгоритм FR основан на аппарате глобальных булевых операций. Локальность алгоритма BRS гарантирует также сохранение конструктивного замысла CAD-модели, поскольку граничные элементы вне окрестности подавляемой цепочки не оказываются затронутыми.

При меньшей общей результативности ( $N_{BRS} < N_{FR}$  для большинства рассмотренных случаев) алгоритм BRS оказывается более надежным, демонстрируя некорректное поведение лишь в

**Таблица 1.** Сравнение алгоритмов FR и BRS (инкрементной процедуры) на тестовых данных [16]. Выделены ячейки, отвечающие неудовлетворительным результатам работы того или иного алгоритма

| Case | $r_{\max}$ | $T_{FR}$  | $V_{FR}$ | $N_{FR}$ | $T_{BRS}$ | $V_{BRS}$ | $N_{BRS}$ |
|------|------------|-----------|----------|----------|-----------|-----------|-----------|
| 1    | $\infty$   | 0.05      | 1        | 1        | 0.01      | 1         | 0         |
| 2    | $\infty$   | 0.025     | 1        | 1        | 0.003     | 1         | 0         |
| 3    | $\infty$   | 0.07      | 1        | 2        | 0.008     | 1         | 0         |
| 4    | $\infty$   | 0.08      | 1        | 3        | 0.012     | 1         | 3         |
| 5    | $\infty$   | 0.13      | 1        | 12       | 0.016     | 1         | 4         |
| 6    | $\infty$   | 4.98      | 0        | 9        | 0.08      | 1         | 4         |
| 7    | $\infty$   | 0.018     | 1        | 1        | 0.005     | 1         | 1         |
| 8    | $\infty$   | 0.02      | 1        | 2        | 0.004     | 1         | 2         |
| 9    | $\infty$   | 0.02      | 1        | 2        | 0.003     | 1         | 0         |
| 10   | $\infty$   | 0.025     | 1        | 4        | 0.005     | 1         | 4         |
| 11   | $\infty$   | 0.035     | 1        | 7        | 0.006     | 1         | 0         |
| 12   | $\infty$   | 0.0005    | 1        | 0        | 0.003     | 1         | 0         |
| 13   | $\infty$   | 0.033     | 1        | 4        | 0.006     | 1         | 0         |
| 14   | $\infty$   | 0.01      | 1        | 4        | 0.007     | 1         | 0         |
| 15   | $\infty$   | 0.08      | 1        | 18       | 0.02      | 1         | 18        |
| 16   | $\infty$   | 0.067     | 1        | 4        | 0.006     | 1         | 1         |
| 17   | $\infty$   | 0.026     | 1        | 4        | 0.008     | 1         | 4         |
| 18   | $\infty$   | 0.075     | 1        | 5        | 0.02      | 1         | 5         |
| 19   | $\infty$   | 0.025 (i) | 1        | 4        | 0.004     | 1         | 0         |
| 20   | $\infty$   | 0.02 (i)  | 1        | 20       | 0.017     | 1         | 0         |
| 21   | $\infty$   | 0.085     | 1        | 8        | 0.01      | 1         | 8         |
| 22   | $\infty$   | 0.045     | 1        | 2        | 0.005     | 1         | 2         |
| 23   | $\infty$   | 0.028     | 1        | 2        | 0.003     | 1         | 0         |
| 24   | $\infty$   | 0.96      | 1 (d)    | 26       | 0.005     | 1         | 0         |
| 25   | $\infty$   | 0.0004    | 1        | 0        | 0.004     | 1         | 0         |
| 26   | $\infty$   | 0.085     | 1 (d)    | 20       | 0.01      | 1         | 0         |
| 27   | $\infty$   | 0.15      | 1 (d)    | 34       | 0.04      | 1         | 0         |
| 28   | 0.45       | 16.7 (i)  | 1        | 44       | 0.64      | 1         | 0         |
| 29   | $\infty$   | 1.28 (i)  | 1        | 13       | 0.004     | 1         | 0         |
| 30   | $\infty$   | 0.0004    | 1        | 0        | 0.012     | 1         | 6         |
| 31   | 4          | 11.1      | 0        | 111      | 0.25      | 1         | 0         |
| 32   | $\infty$   | 0.0005    | 1        | 0        | 0.01      | 1         | 3         |
| 33   | $\infty$   | 2.07      | 1        | 6        | 0.2       | 1         | 5         |
| 34   | $\infty$   | 0.51      | 1        | 11       | 0.05      | 1         | 2         |
| 35   | $\infty$   | 0.0004    | 1        | 0        | 0.015     | 1         | 3         |
| 36   | $\infty$   | 44.5      | 1        | 0        | 8.92      | 1         | 80        |
| 37   | $\infty$   | 111.7     | 1        | 210      | 19.5      | 0         | 105       |
| 38   | $\infty$   | 8.14      | 1        | 47       | 0.64      | 1         | 27        |
| 39   | $\infty$   | 4.26      | 1        | 16       | 0.06      | 1         | 0         |
| 40   | $\infty$   | 13.67     | 1 (d)    | 39       | 0.48      | 1         | 13        |
| 41   | $\infty$   | 10.73     | 1 (d)    | 39       | 0.35      | 1         | 13        |

Таблица 1. Продолжение

| Case | $r_{\max}$ | $T_{FR}$ | $V_{FR}$ | $N_{FR}$ | $T_{BRS}$ | $V_{BRS}$ | $N_{BRS}$ |
|------|------------|----------|----------|----------|-----------|-----------|-----------|
| 42   | $\infty$   | 1.7      | 1        | 0        | 0.1       | 1         | 8         |
| 43   | $\infty$   | ?        | ?        | ?        | 2.58      | 1         | 4         |
| 44   | $\infty$   | 0.08     | 1        | 18       | 0.02      | 1         | 18        |
| 45   | $\infty$   | 0.15     | 1        | 35       | 0.04      | 1         | 35        |
| 46   | $\infty$   | 0.017    | 1        | 1        | 0.003     | 1         | 1         |
| 47   | $\infty$   | 0.04     | 1        | 2        | 0.006     | 1         | 2         |
| 48   | $\infty$   | 11.8     | 1        | 97       | 0.447     | 1         | 75        |
| 49   | $\infty$   | 2.22 (i) | 1        | 3        | 0.096     | 1         | 0         |
| 50   | $\infty$   | 8.11     | 1        | 42       | 0.19      | 1         | 8         |
| 51   | $\infty$   | ?        | ?        | ?        | 0.036     | 1         | 0         |
| 52   | 41         | ?        | ?        | ?        | 0.17      | 1         | 1         |
| 53   | $\infty$   | 3.46     | 1 (d)    | 80       | 0.062     | 1         | 0         |
| 54   | $\infty$   | 0.0005   | 1        | 0        | 0.01      | 1         | 3         |
| 55   | $\infty$   | ?        | ?        | ?        | 2.56      | 1         | 0         |
| 56   | $\infty$   | 66       | 1        | 4        | 0.17      | 1         | 0         |
| 57   | $\infty$   | 1.58     | 1 (d)    | 15       | 0.15      | 1         | 6         |
| 58   | $\infty$   | 0.5      | 1        | 11       | 0.003     | 1         | 0         |
| 59   | $\infty$   | ?        | ?        | ?        | 1.76      | 1         | 11        |
| 60   | $\infty$   | 0.0004   | 1        | 0        | 1.22      | 1         | 11        |
| 61   | $\infty$   | ?        | ?        | ?        | 49.62     | 1         | 90        |
| 62   | $\infty$   | 0.15     | 1        | 5        | 0.013     | 1         | 3         |
| 63   | $\infty$   | 0.14     | 1        | 5        | 0.009     | 1         | 3         |
| 64   | $\infty$   | 0.09     | 1        | 3        | 0.004     | 1         | 0         |
| 65   | $\infty$   | 1.01     | 1        | 0        | 0.07      | 1         | 8         |
| 66   | $\infty$   | 143.5    | 1        | 39       | 1.69      | 1         | 4         |
| 67   | $\infty$   | 1.48     | 1        | 27       | 0.133     | 1         | 6         |
| 68   | $\infty$   | 0.0004   | 1        | 0        | 0.14      | 1         | 12        |
| 69   | $\infty$   | 0.327    | 1        | 4        | 0.02      | 1         | 4         |
| 70   | $\infty$   | 11.25    | 1        | 32       | 3.98      | 1         | 24        |
| 71   | $\infty$   | 1.08 (i) | 1        | 6        | 0.017     | 1         | 0         |
| 72   | $\infty$   | 0.591    | 1 (d)    | 12       | 0.041     | 1         | 4         |
| 73   | $\infty$   | 21.2     | 1 (d)    | 42 +     | 0.789     | 1         | 6         |
| 74   | $\infty$   | 0.82     | 1        | 4        | 0.07      | 1         | 4         |
| 75   | $\infty$   | ?        | ?        | ?        | 9.48      | 1         | 23        |
| 76   | $\infty$   | 0.15     | 1        | 4        | 0.018     | 1         | 4         |
| 77   | $\infty$   | 0.39     | 1        | 2        | 0.44      | 1         | 0         |
| 78   | $\infty$   | ?        | ?        | ?        | 0.21      | 1         | 0         |
| 79   | 5          | 3.1      | 1 (d)    | 32       | 0.022     | 1         | 0         |
| 80   | $\infty$   | 0.29     | 1        | 46       | 0.068     | 1         | 46        |
| 81   | $\infty$   | 28.52    | 1        | 51       | 0.14      | 1         | 0         |
| 82   | 5          | ?        | ?        | ?        | 3.39      | 1         | 21        |
| 83   | $\infty$   | 8.29     | 1        | 47       | 0.64      | 1         | 27        |

Таблица 1. Окончание

| Case | $r_{\max}$ | $T_{FR}$ | $V_{FR}$ | $N_{FR}$ | $T_{BRS}$ | $V_{BRS}$ | $N_{BRS}$ |
|------|------------|----------|----------|----------|-----------|-----------|-----------|
| 84   | 6.5        | 2.08     | 1        | 30       | 0.071     | 1         | 0         |
| 85   | $\infty$   | 0.38     | 1        | 9        | 0.011     | 1         | 0         |
| 86   | $\infty$   | 0.16     | 1        | 3        | 0.015     | 1         | 3         |
| 87   | 6.5        | 2.14     | 1        | 34       | 0.004     | 1         | 0         |
| 88   | $\infty$   | 4.55     | 1 (d)    | 87       | 0.94      | 1         | 0         |
| 89   | $\infty$   | 9.77     | 1        | 47       | 0.56      | 1         | 6         |
| 90   | $\infty$   | 79.26    | 1        | 0        | 0.66      | 1         | 0         |
| 91   | $\infty$   | 1.25 (i) | 1        | 0        | 0.053     | 1         | 0         |
| 92   | $\infty$   | 13.32    | 1        | 232      | 2.55      | 1         | 161       |
| 93   | $\infty$   | 0.194    | 1        | 3        | 0.015     | 1         | 3         |
| 94   | $\infty$   | 0.0004   | 1        | 0        | 0.01      | 1         | 3         |
| 95   | $\infty$   | 87.67    | 1        | 38       | 4.95      | 1         | 61        |

1.05% случаев против 23.15% для алгоритма FR. Отметим также тот неочевидный факт, что алгоритм FR не гарантирует упрощения модели, о чем свидетельствует результат на тестовом наборе #73 (см. таблицу 1). Действительно, в алгоритме FR топология модели меняется дважды: при удалении грани и при стягивании соседних граней до пересечения. Второе топологическое изменение может привести к образованию новых граней в некоторых случаях. Такие случаи возникают, как правило, при недостаточно тщательном выборе граней на удаление (рис. 12).

Некоторые результаты применения инкрементной процедуры BRS показаны на рис. 13.

## 6. ЗАКЛЮЧЕНИЕ

Настоящая работа заканчивает формирование нового подхода к упрощению CAD-моделей путем распознавания и подавления цепочек скруг-

лений. В рамках данного исследования впервые реализованы эйлеровы операторы KEV, KEF, KFMV на базе открытого геометрического ядра OpenCascade и предложена архитектура распознавания конструктивных элементов на графе смежности граней  $G$ . Результатом распознавания является граф  $G^*$ , изоморфный графу  $G$  и содержащий дополнительные атрибуты, используемые на этапе подавления конструктивных элементов.

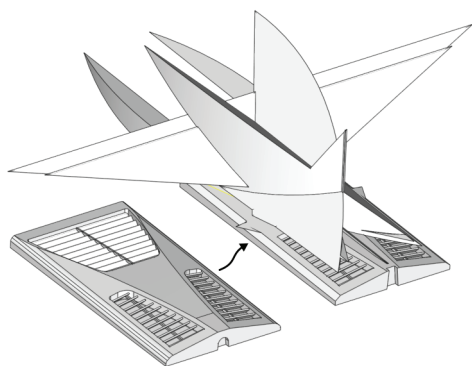


Рис. 12. Аномальный результат алгоритма FR в тестовом наборе #73 (таблица 1) при  $r_{\max} = \infty$ .

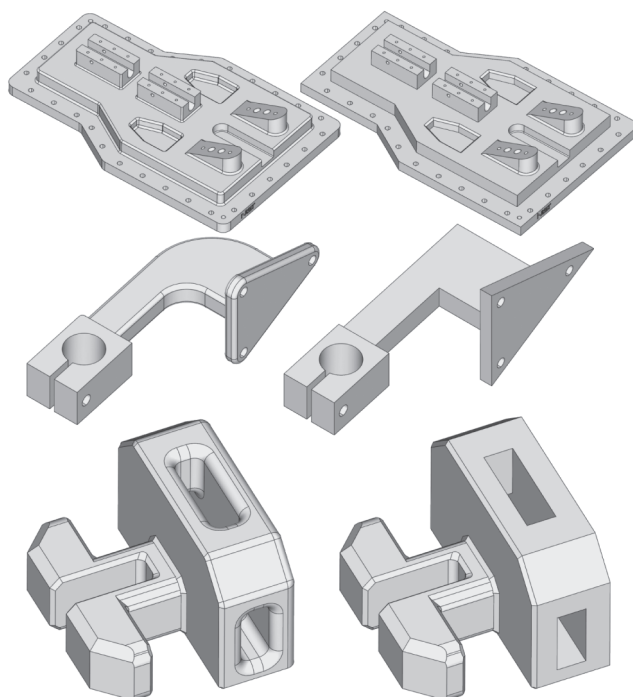


Рис. 13. Некоторые результаты применения алгоритма BRS. Сверху-вниз: тестовые наборы #92, #80, #48.

Алгоритм подавления снабжен инкрементной процедурой, позволяющей удалять обнаруженные цепочки скруглений в пакетном (batch) режиме. Алгоритм может быть расширен путем добавления дескрипторов новых топологических ситуаций. Благодаря накоплению истории модификации всех затронутых граничных элементов, конструктивно-значимая информация (цвета, имена, допуски), ассоциированная с исходной моделью, может быть перенесена на упрощенную модель без потерь.

Упрощение CAD-моделей, заданных без истории построения, остается актуальной инженерной проблемой, о чем свидетельствуют современные работы [19, 20] и другие. В рамках настоящего исследования предложена экспериментальная программная среда [13], доступная на некоммерческой основе с открытыми исходными кодами. В ее состав входят реализация атрибутированного графа смежности граней, эйлеровы операторы, алгоритм распознавания и подавления скруглений, а также соответствующая инкрементная процедура. Входными и выходными данными являются файлы формата STEP (ISO 10303). Построенный граф смежности граней может быть передан в сторонние системы, такие как MATLAB, для последующего анализа. В работе [20] передача данных AAG осуществлялась посредством формата JSON.

Предложенный подход к распознаванию и подавлению цепочек скруглений реализован на языке C++ с использованием открытого геометрического ядра OpenCascade и платформы Analysis Situs [13], где осуществлялось прототипирование. Последняя версия алгоритма и набор тестовых данных находятся в открытом доступе [9]. Алгоритм также интегрирован в коммерческое приложение CAD Processor [15].

По результатам эксперимента намечено дальнейшее развитие предложенного алгоритма. Во-первых, “каталог” обрабатываемых типов скруглений на сегодняшний день не является полным и может быть расширен. Во-вторых, представляется возможной и целесообразной оптимизация производительности инкрементной процедуры, которая выполняет глобальное перестроение графа смежности граней, несмотря на то что каждая итерация изменяет топологию локально. Перестроение графа смежности может быть заменено его модификацией с изъятием на каждой итерации граней удаленной цепочки и копированием существующих атрибутов.

Сравнение с алгоритмом FR показывает, что в некоторых случаях алгоритм BRS не справился с цепочками скруглений, на которых алгоритм FR отработал корректно. Для таких случаев алгоритм BRS должен быть доработан. Должен быть также

расширен и протестирован список типов топологических конфигураций доступных алгоритму BRS.

Очевидной перспективой развития реализованного подхода является его адаптация для подавления фасок, так как это потребует лишь адаптации алгоритма распознавания, в то время как набор эйлеровых операторов остается неизменным, в силу топологической эквивалентности цепочек скруглений и фасок.

## СПИСОК ЛИТЕРАТУРЫ

1. Сляднев С.Е., Малышев А.С., Турлапов В.Е. Автоматизированное упрощение машиностроительных CAD-моделей и сборок без использования истории построения // Труды международной конференции Графикон. 2018. С. 488–494.
2. Belaziz M., Bouras A., Brun J.M. Morphological analysis for product design // Computer-Aided Design. 2000. V. 32. № 5–6. P. 377–388.
3. Lai J.Y., You Z.W., Chiu Y.K., et al. On the Development of Blend Faces and Holes Recognition Algorithm for CAE Applications // Key Engineering Materials. 2015. № 656–657. P. 789–794.
4. Lai J.-Y., Wong C., Huynh T.T. et al. Small blend suppression from B-rep models in computer-aided engineering analysis // Journal of the Chinese Institute of Engineers. 2016. V. 39. № 6. P. 735–745.
5. Cui X., Gao S., Zhou G. An Efficient Algorithm for Recognizing and Suppressing Blend Features // Computer-Aided Design and Applications. 2004. V. 1. № 1–4. P. 421–428.
6. Venkataraman S., Sohoni M., Rajadhyaksha R. Removal of blends from boundary representation models // Proceedings of the seventh ACM symposium on Solid modeling and applications – SMA '02, ACM Press. 2002. P. 83.
7. Venkataraman S., Sohoni M., Elber G. Blend recognition algorithm and applications // Proceedings of the sixth ACM symposium on Solid modeling and applications – SMA '01, 2001, ACM Press. P. 99–108.
8. Venkataraman S., Sohoni M. Reconstruction of feature volumes and feature suppression // Proceedings of the seventh ACM symposium on Solid modeling and applications. 2002. P. 60–71.
9. Analysis Situs: suppress blend. [http://analysis-situs.org/features/features\\_suppress-blends.html](http://analysis-situs.org/features/features_suppress-blends.html).
10. Zhu H., Menq C. B-Rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition // Computer-Aided Design. 2002. № 34. P. 109–123.
11. Mantyla M., Sulonen R. GWB: A Solid Modeler with Euler Operators // IEEE Computer Graphics and Applications. 1982. V. 2. № 7. P. 17–31.
12. Kripac J. 1997. A mechanism for persistently naming topological entities in history-based parametric solid models // Computer-Aided Design. 1997. V. 29. P. 113–122.

13. *Slyadnev S., Malyshev A., Turlapov V.* CAD model inspection utility and prototyping framework based on OpenCascade // GraphiCon. 2017. P. 323–327.
14. *Malyshev A., Slyadnev S., Turlapov V.* Graph-based feature recognition and suppression on the solid models // GraphiCon. 2017. P. 319–322.
15. OPEN CASCADE. CAD Processor software. <https://www.opencascade.com/content/cad-processor>
16. Analysis Situs: test data for BRS algorithm. [http://analysis situs.org/features/suppress-blends/features\\_suppress-blends\\_data.html](http://analysis situs.org/features/suppress-blends/features_suppress-blends_data.html)
17. *Corney J.R., Lim T.* 3D Modelling with ACIS, Saxe-Coburg Publications; 2nd ed. Edition, 2001. 388 p.
18. *Stroud I., Nagy H.* Solid Modelling and CAD Systems: How to Survive a CAD System. Springer, 2011. 689 p.
19. *Song P.-P., Lai J.-Y., Tsai Y.-C., Hsu C.-H.* Automatic recognition and suppression of holes on mold bases for finite element applications. Engineering with Computers. 2019. V. 35. № 3. P. 925–944.
20. *Zhu F.* Application of analytical and AI-based feature detecting methods for an Energy optimized industrial process. Bachelor's thesis. 2019.