ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 57591— 2017

АДДИТИВНЫЕ ТЕХНОЛОГИЧЕСКИЕ ПРОЦЕССЫ. БАЗОВЫЕ ПРИНЦИПЫ — часть 4

Обработка данных

Издание официальное

Москва
Стандартинформ
2019

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Федеральным государственным унитарным предприятием «Всероссийский научно-исследовательский институт стандартизации и сертификации в машиностроении» (ВНИИНМАШ), Акционерным обществом «Наука и инновации» (АО «Наука и инновации»)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 182 «Аддитивные технологии»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 9 августа 2017 г. № 847-ст

4 ВВЕДЕН ВПЕРВЫЕ

5 ПЕРЕИЗДАНИЕ. Апрель 2019 г.

*Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Стандартинформ, оформление, 2017, 2019

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Термины, определения и сокращения 1
4 Общие положения 3
4.1 Принципы форматирования 3
4.2 Рекомендации для включения новых элементов 3
5 Структура данной спецификации 3
5.1 Формат 3
5.2 Гибкость 4
5.3 Точность 4
5.4 Будущие изменения и дополнения 4
6 Общая структура 4
7 Характеристики геометрии 5
7.1 Общие 5
7.2 Сглаженная геометрия 6
7.3 Ограничения в геометрии 6
8 Характеристики материала 6
8.1 Общие 6
8.2 Смешанные и градиентные материалы и субконструкции 8
8.3 Пористые материалы 8
8.4 Стохастические материалы 8
9 Характеристики цвета 8
9.1 Общие 8
9.2 Градации цвета и отображение текстур 9
9.3 Прозрачность 10
10 Характеристики текстур 10
11 Элементы группирования (созвездия) 10
12 Метаданные 11
13 Архивирование 12
14 Минимальные требования для реализации 12
Приложение А (справочное) Руководство по внедрению схемы AMF XML 13
Приложение Б (справочное) Данные о производительности и будущие возможности 19
Библиография 21

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

АДДИТИВНЫЕ ТЕХНОЛОГИЧЕСКИЕ ПРОЦЕССЫ. БАЗОВЫЕ ПРИНЦИПЫ — часть 4

Обработка данных

Additive processes. Basic principles. Part 4. Dataprocessing

Дата введения — 2017—12—01

1 Область применения

Настоящий стандарт содержит требования к формату файлов для аддитивного производства (AMF), которые используются для коммуникационной (для обмена данных) электронной геометрической модели аддитивного производства, включающей в себя пространственное описание геометрии поверхности, со встроенной поддержкой для цвета, материалов, сеток координат, групп элементов и метаданных.

Для поддержки совместимости со стандартами аддитивного производства файлы AMF могут быть использованы только при условии соблюдения настоящего стандарта и при условии использования расширяемого языка разметки (XML) (см. [1]).

Объединение XML и AMF возможно в соответствии со стандартами ISO и ASTM. Руководство по осуществлению такого объединения представлено в приложении А.

Настоящий стандарт не устанавливает каких-либо четких механизмов обеспечения целостности данных, электронных подписей и шифрования.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ 2.511 Единая система конструкторской документации (ЕСКД). Правила передачи электронных конструкторских документов. Общие положения

ГОСТ 20886 Организация данных в системах обработки данных. Термины и определения

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

3 Термины, определения и сокращения

3.1 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями:

3.1.1 **потребитель AMF:** Программное обеспечение для чтения файлов аддитивного производства (AMF), используемых для изготовления, визуализации или анализа.

Издание официальное

Примечание — AMF-файлы также используются для просмотра, анализа и верификации программного обеспечения.

3.1.2 **редактор AMF:** Программное обеспечение для считывания и перезаписи файлов аддитивного производства (AMF) при их преобразовании.

Примечание — Редактор AMF используется для преобразования AMF из одной формы в другую (например, когда необходимо преобразовать все изогнутые треугольники в плоские треугольники).

3.1.3 **производитель AMF:** Файл программного обеспечения для записи (генерации) файла аддитивного производства (AMF) по исходным геометрическим данным.

Примечание — AMF-файлы обычно экспортируются автоматически из программного обеспечения для проектирования (САПР), сканирования или непосредственно из алгоритмов вычислительной геометрии.

3.1.4 **атрибут:** Характеристика данных, представляющих собой один или более аспектов или дескрипторов данных в элементе.

3.1.5 **редактор:** Все элементы текста, связанные с файлами аддитивного производства (AMF).

Примечание — AMF-файлы обычно экспортируются автоматически из программного обеспечения для проектирования (САПР), сканирования или непосредственно из алгоритмов вычислительной геометрии.

3.1.6 **элемент:** Информация в XML-документе, состоящая из открывающего тега, конца тега, содержимого между тегами и атрибутами;

3.1.7 **расширяемый язык разметки, XML;** eXtensible Markup Language. Рекомендован Консорциумом Всемирной паутины (W3C): Стандарт Консорциума Всемирной паутины, разработанный для пометки информации, содержащейся в документах, предлагающий средства для предоставления содержимого в одинаково хорошо удобном формате для чтения человеком и компьютерными программами.

Примечание — Благодаря использованию настраиваемого стиля таблиц и схем, информация может быть представлена унифицированным способом, позволяющим обмен как информацией (данными), так и форматом (метаданными).

3.1.8 **набор данных:** Идентифицированная совокупность физических записей, организованная одним из установленных в системе обработки данных способов и представляющая файлы или части файлов в среде хранения.

[ГОСТ 20886—85, статья 8]

3.1.9 **информационный объект:** Идентифицированная (наименованная) совокупность данных в информационной системе, обладающая набором атрибутов (характеристик).

[ГОСТ 2.511—2011, статья 3.1.13]

3.2 Сокращения

AMF — формат компьютерных данных, альтернативный формату Stl., применяемому с 1987 г. в качестве базового для передачи данных компьютерной 3D-модели в аддитивную машину для построения физической модели. AMF базируется на языке EML (Extensible Markup Language — расширяемый язык разметки) и поддерживает размерность, цвет, текстуру, триангуляцию, сетчатые структуры, которые Stl-формат не поддерживает; более компактный, чем Stl-файл;

XML — текстовый файл, содержащий список элементов и атрибутов;

Бит — Единица измерения информации в двоичной системе счисления;

ID — уникальный признак объекта, позволяющий отличать его от других объектов;

RGBA — изображение цвета материала (красный/зеленый/синий) с техникой создания эффекта полупрозрачности альфа-композитинг (англ. Alpha compositing);

SRGB — стандарт представления цветового спектра с использованием модели RGB;

RGB — аддитивная цветовая модель, описывающая способ кодирования цвета для цветовоспроизведения;

2D — двухмерное проектирование;

3D — трехмерное проектирование;

ZIP — формат архивации файлов и сжатия данных без потерь;

PRSM — код для псевдослучайной пространственной карты;

STL — формат файлов, применяемый с 1987 г. в качестве базового для передачи данных компьютерной 3D-модели в аддитивную машину для построения физической модели.

4 Общие положения

4.1 Принципы форматирования

4.1.1 Существует естественный компромисс между форматом файла и его полезностью для определенных целей. Функции, предназначенные для удовлетворения одних целей, могут помешать достижению других. Требования к файлам AMF, описанные в настоящем стандарте, предназначены для решения задач, указанных в 4.1.2—4.1.7.

4.1.2 Технология независимости

AMF описывает объект таким образом, чтобы любая машина могла построить его в соответствии со своими характеристиками. Разрешение и толщина слоя построения независимы и не содержат информацию, относящуюся к какому-либо конкретному производственному процессу или методу. Это не отменяет функций, которые поддерживаются оборудованием (например, цвет, материалы и т. п.). Эти функции определены таким образом, чтобы избежать исключительности.

4.1.3 Доступность

AMF легко распознать и понять. Файлы можно читать и отлаживать в простом текстовом редакторе. Идентичная информация не хранится в нескольких местах.

4.1.4 Масштабируемость

Размер файла и время обработки хорошо масштабируются с увеличением сложности конфигурации детали и улучшением разрешающей способности, а также точности производственного оборудования. На случай, когда необходимо производство с очень высоким разрешением, предусмотрена возможность обработки больших массивов однотипных объектов (например, сеток и решеток) и криволинейных поверхностей, в том случае, когда необходимо производство с очень высоким разрешением.

4.1.5 Производительность

AMF обеспечивает разумный срок для чтения и записи, а также разумный размер файла для крупных объектов. Характеристики представлены в приложении Б.

4.1.6 Обратная совместимость

Любой существующий файл STL может быть преобразован непосредственно в файл AMF без потери информации и необходимости добавления какой-либо дополнительной информации. AMF-файлы можно легко конвертировать обратно в формат STL для использования на устаревших системах, при этом дополнительные функции будут потеряны. Этот формат поддерживает треугольную сетку геометрии поверхности.

4.1.7 Перспективы совместимости

Чтобы оставаться актуальным в быстро меняющейся отрасли, этот формат является легко расширяемым, при этом сохраняется совместимость с предыдущими версиями.

4.2 Рекомендации для включения новых элементов

4.2.1 Любой новый предложенный элемент должен применяться во всех аппаратных платформах и технологиях, которые могут быть использованы для достижения желаемого результата.

4.2.2 Новые элементы, предложенные в настоящем стандарте, описываются объектами, а не способами их построения. Например, в будущем элемент <hollow> может быть использован в виде пустого объема, если это возможно. Однако элемент <objectLayerFillPath>, в котором описывается создание полого объема, не предполагает особый процесс изготовления.

5 Структура данной спецификации

5.1 Формат

Информация, указанная в спецификации, сохраняется в формате XML 1.0. XML — это текстовый файл, содержащий список элементов и атрибутов. Применяя этот широко принятый формат данных, возможно использовать множество инструментов для создания, просмотра, манипулирования, обработки и хранения AMF-файлов. XML является удобочитаемым для человека. XML может быть сжат или зашифрован с использованием стандартных процедур.

5.2 Гибкость

Еще одним существенным преимуществом XML является его гибкость. Отсутствие какого-либо параметра или дополнительный параметр не являются проблемой для синтаксического анализатора при условии, что документ соответствует стандарту XML. Использование имен XML позволяет добавлять — например, в устаревшее программное обеспечение — новые возможности без разрушения старой версии синтаксического анализатора.

5.3 Точность

Этот формат файла не учитывает точность представления числовых значений. Он несет ответственность за программу, генерирующую написание такого количества цифр, которое необходимо для правильного представления целевого объекта. Однако потребитель AMF должен прочитать и обработать числа двойной точности (64 бит).

5.4 Будущие изменения и дополнения

В то время как дополнительные элементы XML могут быть предварительно добавлены к любому файлу AMF для внутренних целей, такие добавки не должны рассматриваться как часть данной спецификации. Неофициальный элемент AMF может быть проигнорирован любым потребителем AMF и не может быть сохранен или воспроизведен приложением редактора. Элемент становится официальным только тогда, когда он принят в данной спецификации.

6 Общая структура

6.1 Файл AMF должен начинаться с объявления XML с указанием версии XML и кодировки, например:

<?xmlversion=“1.0” encoding=“UTF-8”?>

По умолчанию XML-версия должна быть 1.0. Необходимо указывать только utf-8 и utf-16. Нераспознанная кодировка должна вызывать сбой файла загрузки.

6.2 Пробелы и XML-стандарты могут быть внедрены в файл и должны игнорироваться любым интерпретатором:

<!-- ignorethiscomment -->

6.3 Оставшаяся часть файла должна быть заключена между началом </amf> и концом </ amf> тегов. Элемент указывает на тип файла и выполняет требование о том, чтобы все XML-файлы имели один корневой элемент. Атрибут версии файла должен быть совместим с версией стандарта AMF. Также можно использовать стандартные атрибуты XML. Размеры изделия определяют в миллиметрах, дюймах, футах, метрах и микронах. По умолчанию размеры изделия определяют в миллиметрах, например:

<amfunit=“millimeter” version=“1.0” xml:lang=“en”

xmins:amf=“gost.e-ecolog.ru >

6.4 Стартовый элемент и закрывающие теги заключены в <amf/>. Есть пять элементов верхнего уровня:

- элемент <object> определяет объем или объемы материала;
- элемент <material> определяет один материал для изготовления. Этот материал должен быть идентифицирован. Идентификатор должен быть уникальным и не равняться 0;
- элемент <texture> — необязательный элемент текстуры. Определяет одно изображение или текстуру для цвета. Текстура имеет свой идентификатор. Идентификатор текстуры должен быть уникальным;
- элемент <constellation> — необязательный элемент созвездия. Иерархически объединяет объекты и другие созвездия в шаблон для печати. Элемент созвездия может содержать идентификатор объекта, который должен быть уникальным. Если не указан элемент созвездия, каждый элемент объекта должен быть импортирован. Потребительское программное обеспечение может определять относительное расположение объектов, если в файле указано более одного объекта;
- элемент <metadata> — необязательный элемент метаданных. Указывает дополнительные сведения об объектах и элементах, содержащихся в файле.

6.5 Для полнофункционального файла AMFнеобходим только один элемент.

7 Характеристики геометрии

7.1 Общие

7.1.1 Элемент верхнего уровня <object> объявляет уникальный идентификатор и должен содержать один дочерний элемент <mesh>. Элемент <mesh> содержит два дочерних элемента: <vertices> и <volume>. Элемент <object> необязательно может ссылаться на материал.

7.1.2 Элемент <vertices> описывает все вершины треугольников, которые используют при построении объекта. Каждой вершине треугольника присваивается идентификационный номер в виде целого числа в том порядке, в котором он объявлен, начиная с нуля. Требуемый дочерний элемент <coordinates> дает позицию в трехмерном (3D) пространстве, используя дочерние элементы <x>, <y> и <z>.

7.1.3 По крайней мере один элемент <volume> должен быть включен после получения информации о вершине треугольника. Каждый <volume> содержится в замкнутом объеме объекта. Несколько элементов <volume> могут быть включены в один объект. Не должно быть перекрытых элементов.

7.1.4 В процессе тесселяции поверхности объема в пределах каждого элемента <volume> несколько элементов <triangle> будут использованы для определения треугольников. Каждый элемент <triangle> должен иметь три вершины. Индексы трех вершин треугольника должны быть указаны с использованием элементов <v1>, <v2> и <v3>. Вершины должны быть упорядочены в соответствии с правилом правой руки (вершины перечисляются в порядке обхода против часовой стрелки). Каждому треугольнику присваивается идентификационный номер в виде целого числа в том порядке, в котором он был объявлен, начиная с нуля (см. рисунок 1).

7.1.5 Геометрия не может быть использована для описания структуры поддержки. Должна быть описана только конечная целевая структура.

```
<?xml version="1.0" encoding="UTF-8"?>
<amf unit="millimeter">
  <object id="0">
    <mesh>
      <vertices>
        <vertex>
          <coordinates>
            <x>0</x>
            <y>1.32</y>
            <z>3.715</z>
          </coordinates>
        </vertex>
        <vertex>
          <coordinates>
            <x>0</x>
            <y>1.269</y>
            <z>2.45354</z>
          </coordinates>
        </vertex>
        ...
      </vertices>

      <volume>
        <triangle>
          <v1>0</v1>
          <v2>1</v2>
          <v3>3</v3>
        </triangle>
        <triangle>
          <v1>1</v1>
          <v2>0</v2>
          <v3>4</v3>
        </triangle>
        ...
      </volume>
    </mesh>
  </object>
</amf>
```

Рисунок 1 — Базовый AMF-файл

Примечание — На рисунке 1 показан базовый AMF-файл, содержащий только список вершин треугольников. Эта структура совместима со стандартом STL.

7.2 Сглаженная геометрия

7.2.1 По умолчанию все треугольники должны быть плоскими. В случае наличия изогнутых треугольников необходимо уменьшить количество сетчатых элементов, необходимых для описания криволинейной поверхности. Некоторое программное обеспечение (см. раздел 14) может проигнорировать информацию о кривизне треугольников.

7.2.2 В процессе преобразования изогнутый треугольник должен быть рекурсивно разделен на четыре треугольника, чтобы сформировать окончательный временный набор плоских треугольников. Глубина рекурсии должна равняться пяти.

7.2.3 Создаваемое программное обеспечение, которое генерирует геометрию объектов, должно автоматически определять число криволинейных треугольников. Создаваемое программное обеспечение должно генерировать целевую геометрию объекта, заменяя изогнутые треугольники на плоские.

7.2.4 Вершина изогнутого треугольника может содержать дочерний элемент <normal>, который указывает нормаль к поверхности в этой вершине. Нормаль должна быть единичной длины, направлена наружу и расположена перпендикулярно к вершинам треугольника.

7.2.5 Если вершина треугольника ссылается на два элемента <volume>, нормаль является тождественной для каждого элемента <volume>.

7.2.6 Элемент <edge> используют для определения кривизны одного нелинейного края, соединяющего две вершины треугольников. Кривизна задается касательными векторами, направленными от начала к концу кромки.

7.2.7 Нормали не определяют для вершин плоских треугольников. Боковые элементы не определяются для кромок плоских треугольников.

7.2.8 При интерпретации нормалей и касательных необходимо использовать Эрмитову интерполяцию второй степени. Формулы для проведения интерполяции указаны в пункте А.3 (приложения А).

7.3 Ограничения в геометрии

При построении геометрии объекта следует соблюдать следующие ограничения:

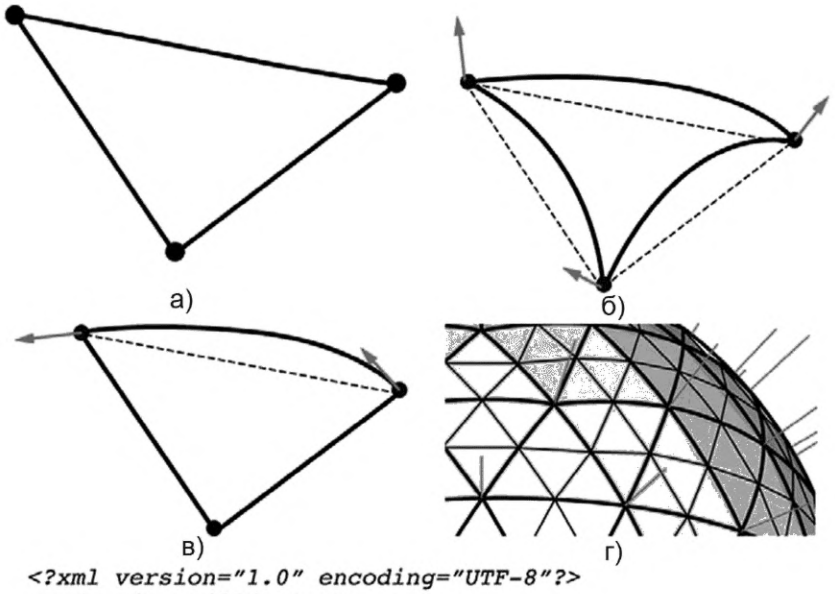
- каждый треугольник должен иметь ровно три различные неколлинеарные вершины;
- треугольники не должны пересекаться. Треугольники могут соприкасаться только в их общих ребрах или общих вершинах;
- объемы должны быть описаны в виде непрерывного пространства;
- объемы не должны перекрываться;
- каждая вершина должна иметь ссылку не менее чем на три треугольника;
- каждая пара вершин должна ссылаться либо на ноль, либо на два треугольника в одном элементе <volume>;
- любые две вершины не должны иметь одинаковые координаты. Может быть использован допуск 8—10 единиц;
- внешнее направление треугольников с общим ребром в объеме должно быть последовательным. Внешнее направление определяют порядком вершин.

8 Характеристики материала

8.1 Общие

8.1.1 Материалы вводятся с помощью дополнительного элемента <material>. Каждый материал может быть определен только с помощью одного элемента <material>. Каждому материалу присваивают уникальный идентификатор. Материал может указываться через значение атрибута ID в элементе <volume>. ID “0” материала зарезервирован для представления пустот в объеме изделия (рисунок 2).

Характеристики материала содержатся внутри каждого элемента <material>. Дочерний элемент <colour> используют для указания цвета материала (красный/зеленый/синий/альфа (RGBA)) (пункт 9). Дополнительные свойства материала могут быть определены с помощью элемента <metadata> (см. рисунок 3). Описание элементов AMF содержится в пункте А.1 (приложения А).




```
<?xml version="1.0" encoding="UTF-8"?>
<amf unit="millimeter">
  <object id="0">
    <mesh>
      <vertices>
        <vertex>
          <coordinates >
            ...
          </coordinates >
          <normal>
            <nx>0</nx>
            <ny>0.707</ny>
            <nz>0.707</nz>
          </normal>
        </vertex>
        ...
        <edge>
          <v1>0</v1>
          <dx1>0.577</dx1>
          <dy1>0.577</dy1>
          <dz1>0.577</dz1>
          <v2>1</v2>
          <dx2>0.707</dx2>
          <dy2>0</dy2>
          <dz2>0.707</dz2>
        </edge>
        ...
      </vertices>

      <volume materialid="0">
        <triangle>
          ...
        </triangle>
        ...
      </volume>
    </mesh>
  </object>
</amf>
```

д)

а) — плоский треугольник; б) — изогнутый треугольник с использованием нормалей; в) — треугольник, использующий касательные; г) — разбиение изогнутого треугольника на четыре треугольника; д) — файл AMF, содержащий криволинейную геометрию

Рисунок 2 — Типы треугольников, используемых в сетке

Примечание — На рисунке показан AMF-файл, содержащий пять материалов. Материал 3 — это однородная смесь первых двух материалов в соотношении 40/60 %. Материал 4 — вертикально-градиентный материал. Материал 5 — каркас со структурой шахматной доски.

8.2 Смешанные и градиентные материалы и субконструкции

8.2.1 Новые материалы можно определять как композиции из других материалов. Элемент <composite> используют для указания пропорций композиции как константу или формулу в зависимости от координат *x*, *y* и *z*. Постоянная пропорция смешивания приведет к получению однородного материала. Изменение состава в зависимости от координат приводит к градиентному составу материала. Более сложные координатно-зависимые пропорции могут приводить к получению нелинейных градиентных материалов. Формула удельного веса может также обращаться к функции текстур tex (*textureid,x,y,z*), описание приведено в А.1 (приложение А).

8.2.2 В композите может быть использовано любое количество материалов.

8.2.3 Любое отрицательное значение доли материала должно быть интерпретировано как нулевая пропорция. Сумма долей различных материалов должна быть равна 1.

8.3 Пористые материалы

8.3.1 Ссылка на функцию materialid “0” (пусто) используется для определения пористой структуры. Доля пустоты должна быть или 0 или 1. Любое другое число будет толковаться как 1.

8.3.2 Элемент <composite> теоретически может быть использован для полного описания геометрии объекта, как отдельная функция или как текстура со ссылкой на пустоту. Целевое использование элемента <composite> со ссылкой на пустоту необходимо для описания клеточных мезоструктур.

8.4 Стохастические материалы

8.4.1 Ссылка на rand-функцию (*x*, *y*, *z*) может быть использована для определения псевдослучайных материалов. Например, композитный материал может объединить два базовых материала в случайных пропорциях, в которых точная пропорция может зависеть от координат различными способами. Rand-функция (*x*, *y*, *z*) генерирует случайную точку в диапазоне [0,1], которая сохраняется на протяжении всех вызовов функции, см. А.4 (приложения А).

9 Характеристики цвета

9.1 Общие

9.1.1 Цвета могут быть введены с использованием элемента <colour> и указанием формата RGBA. По умолчанию цветовое пространство должно быть SRGB (см. [2]), но альтернативные профили могут быть установлены с помощью тега метаданных в корневой <amf> элемент, см. А.1 (приложение А). Элемент <colour> может быть связан с элементом <material>. Чтобы связать цвет и материал, используется элемент <object>; чтобы покрасить весь объект, используется элемент <volume>. Элемент <triangle> используется для покраски треугольника. Элемент <vertex> служит для связи цвета с определенной вершиной треугольника (см. рисунок 3).

9.1.2 Если цвет не указан, то по умолчанию используют белый цвет.

9.1.3 Цвет объекта может быть определен цветом материала спецификации. Цвет перекрывает объем объекта.

```
<?xml version="1.0" encoding="UTF-8"?>
<amf unit="millimeter">

  <material id="1">
    <metadata type="Name">StiffMaterial</metadata>
      <color>
        <r>0</r>
        <g>z</g>
        <b>1-z</b>
      </color>
  </material>

  <texture id="1" width="10" height="26" type="grayscale">
    TWFuIGlzIGRpc3RpbmdlaXNoZWQsIG5vdCB
    vbmx5IGJ5IGhpcyByZWFzb24sIGJ1dCBieS
    B0aGlzIHNpbmdlbGFyIHBhc3Npb24gZnJvb
    SBvdGhlcIBhbmltYWxzLCB3aGljaCBpcyBh
    ...
  </texture>

  <object id="0">
    <mesh>
      <vertices>
        ...
      </vertices>
      <volume materialid="1">
        <color>
          <r>0.9</r>
          <g>0.9</g>
          <b>0.2</b>
          <a>0.8</a>
        </color>
        ...
        <triangle>
          <v1>0</v1>
          <v2>1</v2>
          <v3>3</v3>
          <texmap rtexid="1" gtexid="2" btexid="3">
            <utex1>0.1</utex1>
            <utex2>0.21</utex2>
            <utex3>0.15</utex3>
            <vtex1>0.65</vtex1>
            <vtex2>0.72</vtex2>
            <vtex3>0.91</vtex3>
          </texmap>
        </triangle>
      </volume>
    </mesh>
  </object>
</amf>
```

Рисунок 3 — Характеристики цвета

Примечание — Сплошной цвет может быть связан с материалом томаса или вершины. Вершина может быть также связана с координатой в файле цвета текстуры.

9.2 Градации цвета и отображение текстур

9.2.1 Цвет может также быть определен со ссылкой на формулы, которые могут использовать различные функции, включая функцию карты текстуры.

9.2.2 При обращении к формуле элемент <colour> указывает цвет, который зависит от координаты. В файлах могут быть использованы любые математические выражения, которые сочетают в себе функции, описанные в А.2 (приложения А). Например, использование rand-функции (функции случайных значений) допускается для псевдослучайных цветовых схем. Tex-функция допускает зависимость цвета от карты текстуры или изображения. Чтобы задать полноцветное изображение, как правило, требуется три текстуры — по одной на каждый цветовой канал. Чтобы создать монохромное изображение, обычно достаточно одной текстуры.

9.2.3 Когда вершины одного треугольника имеют разные цвета, внутренний цвет треугольника будет линейной интерполяцией между этими цветами, если цвет треугольника не был указан явно (цвет треугольника имеет приоритет над цветом вершины). Если все три вершины треугольника содержат

отображение одной и той же ID-текстуры для любого канала (r, g, b или a), то цвет этого канала треугольника должен быть извлечен из текстурной карты, отменяя при этом цвет треугольника.

9.3 Прозрачность

Создание видимости частичной прозрачности может быть выполнено через канал прозрачности <a> альфа-композитингом путем объединения цвета переднего плана с цветом фона. Значение “0” указывает нулевую прозрачность, то есть ситуацию, когда используется только цвет переднего плана. Значение “1” указывает полную прозрачность, то есть ситуацию, когда используется только цвет фона. Промежуточные значения линейно интерполируются между цветом фона и цветом переднего плана. Отрицательные значения округлены до 0, а значения больше единицы — до 1. Цвет фона треугольника должен соответствовать цвету вершины. Цвет фона вершины должен соответствовать цвету объема, затем — цвету объекта и затем — цвету материала (в порядке убывания приоритета).

10 Характеристики текстур

10.1 Элемент <texture> используется для связывания textureid (текстурирования) с определенными текстурными данными. Размер текстурной карты должен быть указан. Поддерживаются двумерные (2D) и трехмерные (3D) текстуры. Данные должны быть представлены в виде ряда значений градаций серого в диапазоне [0—255]. Каждое значение хранится в одном байте и кодируется в Base64.

10.2 Упорядочение данных должно начинаться с левого верхнего угла и идти слева направо и сверху вниз. 3D текстура будет указывать сначала на первоначальный слой, а затем повторяться для всех последующих слоев в соответствии с правилом правой руки. Данные будут оптимизированы или добавлены с нулевыми значениями в соответствии с указанным размером текстуры.

10.3 Для наложения текстуры на треугольник используется элемент <texmap>, определяющий U-, V- (опционально) и W-координаты для каждой вершины этого треугольника. Если текстура имеет свойство “черепица” и соответствует значению True, то любые значения U, V , W, не попадающие в диапазон [0,1], будут определяться по модулю 1. Если текстура имеет свойство “черепица” и соответствует значению false, то значения, которые не попадают в пределы диапазона [0,1], будут соответствовать 0 (прозрачный). Текстуры должны быть линейно интерполированы для каждого треугольника. Треугольник должен включать только один элемент <texmap>. Перекрывающиеся текстуры должны быть объединены в единую текстуру, прежде чем накладываться на сетку.

11 Элементы группирования (созвездия)

11.1 Несколько объектов могут быть размещены вместе с помощью элемента <constellation> (см. рисунок 4). Элемент <constellation> может определить положение и ориентацию объектов для повышения эффективности упаковки и описать большие массивы одинаковых предметов. Элемент <instance> определяет перемещение и вращение объекта в процессе преобразования. Смещение и вращение должны быть определены по отношению к исходной позиции и ориентации объекта, определенным первоначально. Углы поворота должны быть указаны в градусах. Сначала применяются вращения вокруг оси *x* и оси *y*, а затем — вокруг оси *z*.

11.2 Элемент <constellation> может включать в себя другие элементы <constellation> с несколькими уровнями иерархии. Однако циклические определения элементов <constellation> использоваться не должны.

11.3 При наличии нескольких объектов и элементов <constellation>, определенных в одном файле, доступны для печати только объекты и элементы <constellation> верхнего уровня.

11.4 В элементе <constellation> ориентация печати будет указана по умолчанию.

```
<?xml version="1.0" encoding="UTF-8"?>
<amf unit="millimeter">

  <material id="1">
    <metadata type="Name">StiffMaterial</metadata>
      <color>
        <r>0</r>
        <g>z</g>
        <b>1-z</b>
      </color>
  </material>

  <texture id="1" width="10" height="26" type="grayscale">
    TWFuIGlzIGRpc3RpbmdlaXNoZWQsIG5vdCB
    vbmx5IGJ5IGhpcyByZWFzb24sIGJldCBieS
    B0aGlzIHNpbmdlbGFyIHBhc3Npb24gZnJvb
    SBvdGhlcIBhbmltYWxzLCB3aGljaCBpcyBh
    ...
  </texture>

  <object id="0">
    <mesh>
      <vertices>
        ...
      </vertices>
      <volume materialid="1">
        <color>
          <r>0.9</r>
          <g>0.9</g>
          <b>0.2</b>
          <a>0.8</a>
        </color>
        ...
        <triangle>
          <v1>0</v1>
          <v2>1</v2>
          <v3>3</v3>
          <texmap rtexid="1" gtexid="2" btexid="3">
            <utex1>0.1</utex1>
            <utex2>0.21</utex2>
            <utex3>0.15</utex3>
            <vtex1>0.65</vtex1>
            <vtex2>0.72</vtex2>
            <vtex3>0.91<1vtex3>
          </texmap>
        </triangle>
      </volume>
    </mesh>
  </object>
</amf>
```

Рисунок 4 — Элементы <constellation>

Примечание — Элементы <constellation> могут включать в себя несколько объектов вместе.

12 Метаданные

Элемент <metadata> необязательно может использоваться для указания дополнительной информации об определенных объектах, геометрии и материалах (см. рисунок 5). С помощью этого элемента могут указываться имя, текстовое описание, авторство, информация об авторских правах и специальные инструкции. Элемент <metadata> может быть включен в качестве дочернего элемента в <amf> с целью указания атрибутов всего файла или в качестве дочернего элемента объектов, объемов и материалов. Зарезервированные типы метаданных и их значения перечислены в А.1 (приложение А).

```
<?xml version="1.0" encoding="UTF-8"?>
<amf unit="millimeter">
  <metadata type="description">Product 123</metadata>
  <metadata type="author">John Smith</metadata>
  <metadata type="cad">SolidX 2.2</metadata>
  <metadata type="name">Part 1</metadata>
  <metadata type="revision">1.3A</metadata>
      ...
  <object ObjectID="0">
     <metadata type="name">Component 1</metadata>
      ...
  </object>
</amf>
```

Рисунок 5 — Метаданные

Примечание — Дополнительные сведения об объекте могут храниться с помощью элемента метаданных.

Модифицированные типы метаданных должны использовать другое имя, например:

<xamf:metadata type="myattribute">mydata</xamf:metadata>

13 Архивирование

13.1 AMF должен храниться либо в виде простого текста, либо в сжатом виде. Сжатие должно быть выполнено в формате ZIP-архива (см. [3]) и может быть сделано пользователем во время записи с помощью приложения и нескольких открытых библиотек сжатия, таких как Reference (см. [4]).

13.2 Обе версии файла — сжатая и несжатая — будут иметь расширение AMF. Используемое программное обеспечение должно определить, является ли файл архивным или нет. Если файл заархивирован, то необходимо выполнить его декомпрессию во время чтения. Любой файл, который не начинается с тега <?xml>, должен быть интерпретирован как сжатый файл xml.

13.3 Кроме того, другие файлы — например, электронные подписи — могут быть включены в архив ZIP. Тем не менее только AMF-файл с тем же именем, что и файл архива, будет проанализирован программным обеспечением. Отсутствие файла с таким именем вызовет ошибку.

14 Минимальные требования для реализации

14.1 Минимальный AMF производителя должен генерировать сжатый файл с одного объекта. Один объект будет содержать сетку, включающую только один том.

14.2 Минимальный AMF потребителя должен разархивировать файл с одного объекта и игнорировать любые цвет, материал, текстуру, элемент <constellation> или метаданные.

Приложение А
(справочное)

Руководство по внедрению схемы AMF XML

А.1 Элементы AMF

Перечень элементов AMF представлен в таблице А.1.

Таблица А.1 — Перечень элементов AMF

Элемент	Родительский элемент (элементы)	Атрибут	Разрешено ли несколько элементов?	Описание
<nx>, <ny>, <nz>	<normal>		Нет	Нормализованные *x*-, *y*- или *z*-компоненты нормали к поверхности в вершине.
<volume>	<mesh>		Да	Определяет объем от установленного списка вершин.
		materialid		Идентификатор материала, применяемый к этому объему.
<triangle>	<volume>		Да	Определяет треугольник из трех вершин в 3D, в соответствии с правилом правой руки (против часовой стрелки).
<v1>, <v2>, <v3>	<triangle> <edge>			Индекс вершин, определяющих треугольник или грань.
<texture>			Да	Указывает данные текстуры, которая будет использоваться, с помощью tex-функции в формуле. Содержит последовательность Base64 с закодированными значениями и с указанием значения пикселей текстуры.
		id		Назначает для новой текстуры уникальный ID.
		width		Ширина (размер по горизонтали, *x*) текстуры в пикселях.
		height		Высота (боковой размер, *y*) текстуры в пикселях.
		depth		Глубина (вертикальный размер, *z*) текстуры в пикселях.
				Определяет, может ли текстура быть плиточной. Значение true означает “плиточная”. Любое другое значение не подразумевает плитку. Любая текстура, отображенная за пределами заданного диапазона, должна возвращать к нулевому значению.
		type		Кодирование данных в текстуре. В настоящее время должны быть “оттенки серого”. В режиме градаций серого каждый пиксель представлен одним байтом в диапазоне [0—255]. Когда текстура задается с помощью tex-функции, эти значения преобразуются в одну точку в диапазоне [0—1] (см А.2). Полноцветное изображение, как правило, требует три текстуры — по одному для каждого из цветовых каналов. Графическое отображение прозрачности может потребовать наличия четвертого канала.

Продолжение таблицы А.1

Элемент	Родительский элемент (элементы)	Атрибут	Разрешено ли несколько элементов?	Описание
<texmapl>	<triangle>		Нет	Карта вершин этого треугольника (*u,v,w*) с координатами заданной текстуры. Если параметр не указан, координата w должна быть равна нулю.
		rtexid		ID текстуры красного канала.
		gtexid		ID текстуры зеленого канала.
		btexid		ID текстуры голубого канала.
		atexid		ID текстуры альфа-канала.
<utext1>, <utex2>, <utex3>	<texmap>		Нет	Описание: координаты *u*, *v* и *w* (опционально) вершин треугольника 1, 2 и 3.
<ytext1>, <ytex2>, <ytex3>				
<wtext1>, <wtex2>, <wtex3>,				
<material>	<amf>		Да	Доступный материал.
		id		Уникальный ID материала. Значение “0” ID материала зарезервировано для обозначения пустот или расходуемого мат-ла.
<composite>	<material>		Да	Композитный материал. Значение числовой константы или математические функции координат x, y, z, определяющее соотношение материалов (materialid). Если значение отрицательное, оно считается равным нулю. Дробное значение числа должно быть округлено до 1. Если оно равно нулю, то материал отсутствует (пустота). Пустоты материалов не должны быть смешаны. См. в п. А.2 список допустимых математических функций.
		materialid		Ссылка на существующий материал. Ссылка не должна быть рекурсивной или циклической.
<constellation>	<amf>		Да	Совокупность объектов с конкретным местоположением.
		id		Определяет ID нового созвездия.
<instance>	<constellation>		Да	Совокупность объектов для печати.
		objectid		Ссылка на ID существующего или создаваемого объекта или созвездия. Рекурсивные или циклические ссылки не должны использоваться.
<deltax>, <deltay>, <deltaz>	<instance>		Нет	Расстояние переноса в x-, y- или z-направлении, соответственно, в системе координат объекта для создания экземпляра объекта в текущем созвездии.
<rx>, <ry>, <rz>	<instance>		Нет	Поворот объекта в градусах вокруг его x-, y- и z-осей, соответственно, для создания экземпляра объекта в текущем созвездии. Повороты должны быть выполнены в порядке x, а затем y и z.

Окончание таблицы А.1

Элемент	Родительский элемент (элементы)	Атрибут	Разрешено ли несколько элементов?	Описание
<metadata>	<amf> , <object>, <volume>, <material>, <vertex>		Да	Указывает дополнительную информацию об объекте.
		type		Тип атрибута. Значения должны быть одним из следующих: name — буквенно-цифровая метка объекта, которая может быть использована интерпретатором при взаимодействии с пользователем; description — описание сущности; url — ссылка на внешний источник, относящийся к объекту; author — задает имя(имена) автора(ов); company — определяет компанию, генерирующий объект; producer — задает имя исходного программного обеспечения и версии; revision — определяет пересмотр сущности; tolerance — задает необходимый производственный допуск объекта в системе предприятия/подразделения; volume — определяет общий объем объекта в единице системы субъекта, который будет использоваться для проверки (применяется только к объекту и объему); elasticmodulus — задает модуль упругости объекта, в единицах СИ (только для материала); poissonratio — определяет коэффициент Пуассона материала, в единицах СИ (только для материала); colourprofile — цвет ICC пространства, используемый для интерпретации трех цветовых каналов <r>, <g> и <b>. Может быть один из 9sRGB9, 9AdobeRGB9, 9WIDE-Gamut-RGB9, 9CIERGB9, 9CIELAB9 или 9CIEXYZ9 (верхний уровень только <AML>)

А.2 Математические операции и функции

Перечень математических операций и функций представлен в таблице А.2.

Формулы, содержащие символы, которые ограничены в XML, такие как “<” и “>”, должны содержаться в пункте CDATA (т.е. начинаться с “[CDATA]” и заканчиваться “]]>”).

Таблица А.2 — Перечень математических операций и функций

Старшинство	Оператор	Определение
1	()	Скобки блока
2	^	Степень
3	*	Умножение
3	/	Деление
Логические операторы возвращают логическое значение 1 или 0, представляющий TRUE и FALSE, соответственно. Нулевое значение представляет FALSE и ненулевое значение представляет TRUE.		

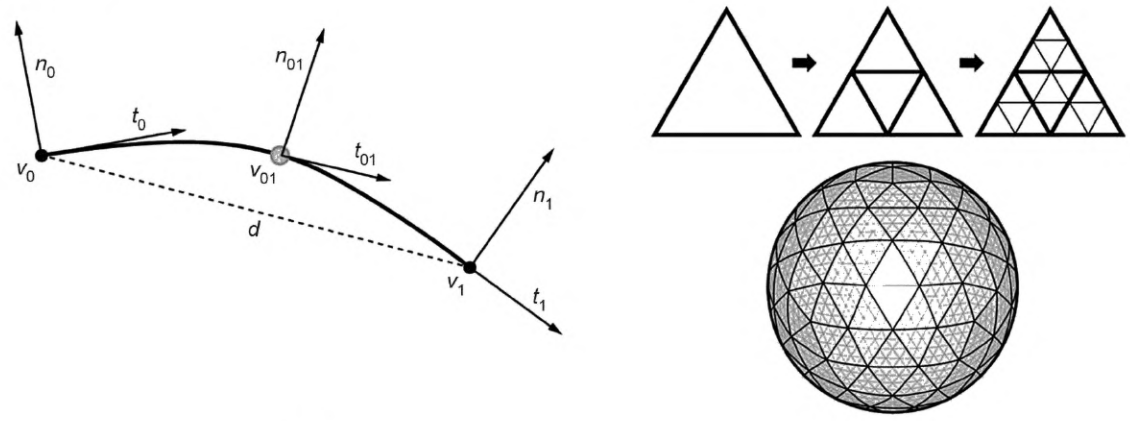
Окончание таблицы А.2

Старшинство	Оператор	Определение
4	+	Добавить
4	–	Вычитание
5	=	Равно
5	<, < =	Меньше, чем(или равно)
5	>, >=	Больше, чем(или равно)
6	and	Точка пересечения (логическое И)
6	or	Союз (логическое ИЛИ)
6	xor	Разница (исключающее ИЛИ)
6	1	Отрицание (логическое НЕ)
6	mod(a,b)	Модуль, в том числе частичный. Возвращает остаток от деления a на b.
6	sin(x)	Синус, радианы
6	cos(x)	Косинусы, радианы
6	tan(x)	Тангенс, радианы
6	asin(x)	Арксинус, радианы
6	acos(x)	Арккосинус, радианы
6	atan(x)	Арктангенс, радианы
6	floor(x)	Округлить до ближайшего целого
6	ceil(x)	Округлить до ближайшего целого
6	sqrt(x)	Квадратный корень
6	ln(x)	Натуральный логарифм
6	log10(x)	Базовый десятичный логарифм
6	exp(x)	Экспонента
6	abs(x)	Абсолютное значение
6	max(x,y)	Максимальное значение
6	min(x,y)	Минимальное значение
6	rand(x,y)	2D или 3D координаты интервале [0,1] (без учета 1). Возвращаемое число должно быть стойким (тот же номер должен всегда быть возвращен на те же координаты). См. пример реализации в пункте А.4.
6	tex (textured,u,v,w) tex(textureid,u,v)	Возвращает скалярное значение с плавающей запятой в диапазоне [0,1], что интерполирует текстуру с textureid в начало координат (u, v, w) для 3D-текстур и (u, v) для 2D-текстур. Если текстура имеет тип «градации серого», диапазон [0,1] соответствует диапазону [0—255] в данных текстурах. Целые значения координат относятся к центру текстуры с первого пикселя, имеющего индекс 1. Если значения дробное, необходимо использовать метод линейной интерполяции. Если координаты выходят за пределы не плиточной текстуры, нулевое значение должно быть возвращено. Если текстура 2D и z координат задана, координата z должна быть проигнорирована.
Логические операторы возвращают логическое значение 1 или 0, представляющий FALSE и TRUE, соответственно. Нулевое значение представляет FALSE и ненулевое значение представляет TRUE.		

А.3 Формулы для перехода от криволинейных треугольников к прямолинейным

А.3.1 Нелинейные треугольные участки с установленными нормалями поверхности или краями касательных должны быть интерполированы из их трех вершин и шести касательных векторов и/или трех нормалей поверхности с использованием кривых интерполяции Эрмита второго порядка, как описано в пунктах А.3.2—А.3.4.

А.3.2 Для каждого из трех ребер треугольника (см. рисунок А.1 (а)) выполните шаги, описанные в пунктах А.3.2.1—А.3.2.6.



а) б)

а) — обозначения, используемые для разбиения кривой;
б) — треугольники разделены рекурсивно в глубину до пяти

Рисунок А.1 — Типы треугольников, используемых в сетке

Примечание — На рисунке показан пример сферической поверхности, содержащей 320 треугольников, каждый разделен на 16 субтреугольников, используя процедуру, описанную в пункте А.3.

А.3.2.1 Если нормаль n_0 в точке v_0 не была установлена явно с помощью элемента <normal>, вычислить нормаль n_0 можно путем вычисления декартова произведения между двумя краевыми касательными, сходящимися в этой точке. Для этого вычисления используют краевые касательные, указанные элементом <edge>, если таковые имеются, от А.3.2.6, выполненные в предшествующем уровне рекурсии или, если ни один из этих вариантов не доступен, используют прямые линии, соединяющие конечные точки.

А.3.2.2 Повторяют действия пункта А.3.2.1 для нормали n_i в точке v_i.

А.3.2.3 Если тангенс t_0 не указан явно в элементе <edge> или предыдущей рекурсии, вычисляют касательный вектор t_0 таким образом, чтобы он был перпендикулярен к нормали n_0 и находился в плоскости, определяемой нормалью и вектором, соединяющим две вершины v_0 и v_1. Для расчета t_0 используется формула (А.1), при этом $d = v_1 - v_0$:

$$t_0 = |d| \frac{-(n_0 \cdot d)n_0}{\|(n_0 \cdot d)n_0\|}. \quad (А.1)$$

А.3.2.4 Повторяют действия пункта А.3.2.1 для тангенса t_1 в точке v_1.

А.3.2.5 Центральную точку $v_{01} = h_{(0,5)}$ рассчитывают с помощью интерполяционной кривой Эрмита второго порядка, в соответствии с формулой (А.2):

$$h(s) = (2s^3 - 3s^2 + 1)v_0 + (s^3 - 2s^2 + s)t_0 + (-2s^3 + 3s^2)v_1 + (s^3 - s^2)t_1. \quad (А.2)$$

А.3.2.6 Центральную касательную $t_{01} = t_{(0,5)}$ рассчитывают с помощью интерполяции кривой Эрмита второго порядка, в соответствии с формулой (А.3):

$$t(s) = (6s^2 - 6s)v_0 + (3s^3 - 4s^2 + 1)t_0 + (-6s^2 + 6s)v_1 + (3s^2 - 2s)t_1. \quad (А.3)$$

А.3.3 Используя полученные три новых вершины и нормали, делят треугольник на четыре подгруппы треугольников.

А.3.4 Повторяют рекурсивно действия пункта А.3.3 для каждого треугольника на глубину до пяти (см. рисунок А.1 (б)).

А.4 Код для псевдослучайной пространственной карты (PRSM)

Цель функции случайных значений (rand) — построить псевдослучайные текстуры, которые будут использоваться в построении объекта. Случайные (*x, y*), случайные (*x, y, z*) и случайные (*x, y, z, k*) возвращают постоянное случайное число в зависимости от установленной координаты. Эти функции позволяют получать 2D- или 3D-координаты в интервале [0,1] (без учета 1). Возвращаемое число должно быть все время постоянным. Если $k = 1$, то второй номер может быть возвращен для этой координаты. Третий номер может быть возвращен для координаты с $k = 2$ и так далее. Пример реализации кода C ++ для rsm (*x, y, z, k*) приведен на рисунке А.2.

```
/* prsm.h
 *  Spatial random number generator based on the
maximally
 *  equidistributed combined Tausworthe-88 generator.
 *  By Daniel Ly and Hod Lipson (2010)
*/

#ifndef __SPATIAL_TAUS88_H__
#define __SPATIAL_TAUS88_H__

#include <climits>

/*************************************
 * Declarations
 *************************************/

typedef struct
{
    unsigned long int s1, s2, s3;
}
taus_state;

unsigned long int rand_seed(unsigned long int x);
unsigned long int taus_get(taus_state* state);
double prsm(double x, double y, double z=0, int k=0);

/*************************************
 * Definitions
 *************************************/
unsigned long int rand_seed(unsigned long int x)
{
    return (1664525*x+1013904223) & 0x7fffffffUL;
}

unsigned long int taus_get(taus_state* state)
{
    unsigned long b;

    b = (((state->s1 << 13UL) & 0xffffffffUL) ^ state-
>s1) >> 19UL;
    state->s1 = (((state->s1 & 0xfffffffeUL) << 12UL) &
        0xffffffffUL) ^ b;

    b = (((state->s2 << 2UL) & 0xffffffffUL) ^ state-
>s2) >> 25UL;
    state->s2 = (((state->s2 & 0xfffffff8UL) << 4UL) &
        0xffffffffUL) ^ b;

    b = (((state->s3 << 3UL) & 0xffffffffUL) ^ state-
>s3) >> 11UL;
    state->s3 = (((state->s3 & 0xfffffff0UL) << 17UL) &
        0xffffffffUL) ^ b;

    return (state->s1 ^ state->s2 ^ state->s3);
}

double prsm(double x, double y, double z,
int k)
{
    taus_state state;

    float tx, ty, tz,
    tx = (float) x;
    ty = (float) y;
    tz = (float) z;

    /* Convert floating point numbers to
ints*/
    unsigned long int ts1 = *(unsigned
int*)&tx;
    unsigned long int ts2 = *(unsigned
int*)&ty;
    unsigned long int ts3 = *(unsigned
int*)&tz;

    /* Convert coordinates to random seeds
*/
    state.s1 = rand_seed(ts1);
    state.s2 = rand_seed(ts2);
    state.s3 = rand_seed(ts3);

    state.s1 = rand_seed(state.s1 ^
state.s3);
    state.s2 = rand_seed(state.s2 ^
state.s1);
    state.s3 = rand_seed(state.s3 ^
state.s2);

    state.s1 = rand_seed(state.s1 ^
state.s3);
    state.s2 = rand_seed(state.s2 ^
state.s1);
    state.s3 = rand_seed(state.s3 ^
state.s2);

    /* "warm up" generator and generate k-
th number */

    for (int i=0; i<k+9; i++) {
        taus_get(&state);
    }

    return
((double)taus_get(&state)/UINT_MAX);
}

#endif /* __SPATIAL_TAUS88_H__ */
```

Рисунок А.2 — Пример реализации кода C++ для PRSM-функции

Приложение Б
(справочное)

Данные о производительности и будущие возможности

Целью данной спецификации является определение требований к AMF в части оптимального времени чтения и анализа файлов, а также его размеров в зависимости от объема содержащихся данных. В таблице Б.1 приведены статистические данные по эффективному размеру файлов в зависимости от количества треугольников, описывающих объект. Время обработки файла определяется как время, которое необходимо на чтение и анализ данных, содержащихся в файле.

Примечание — Время, необходимое на чтение и анализ файла, невелико по сравнению с общим временем, необходимым для обработки файла в процессе изготовления объекта.

Таблица Б.1 — Размер файлов

Количество треугольников	Двоичный Формат STL (несжатый)	Двоичный Формат STL (сжатый)	AMF (несжатый)	AMF (сжатый)
1016388	49,6Mb	25,3Mb	205,9Mb	12,2Mb
100536	4,9Mb	2,3 Mb	20,1Mb	1,2Mb
10592	518 K	249K	2,1Mb	129K
1036	51 K	20K	203K	12K

Таблица Б.2 — Время записи (в секундах)

Количество треугольников	Двоичный Формат STL (несжатый)	Двоичный Формат STL (сжатый)	AMF (несжатый)	AMF (сжатый)
1016388	0,372	~3,4	6,8	15,5
100536	0,038	0,038	0,79	1,78
10592	0,005	0,005	0,11	0,21
1036	0,001	0,001	0,06	0,06

Таблица Б.3 — Время чтения и анализа (в секундах)

Количество треугольников	Двоичный Формат STL (несжатый)	Двоичный Формат STL (сжатый)	AMF (несжатый)	AMF (сжатый)
1016388	0,384	~1,3	6,447	6,447
100536	0,043	0,043	0,669	0,687
10592	0,005	0,005	0,107	0,107
1036	0,001	0,001	0,056	0,056

Таблица Б.4 — Точность (погрешность вычисляется на единичной сфере)

Количество треугольников	STL	AMF (с нормалями)
20	0,102673	0,006777
80	0,032914	0,000788

Окончание таблицы Б.4

Количество треугольников	STL	AMF (с нормалями)
320	0,008877	$8,28E^{-05}$
1280	0,001893	$1,01E^{-05}$
5120	0,000455	$1,95E^{-06}$
20480	$1,13E^{-04}$	$4,51E^{-07}$
81920	$2,81E^{-05}$	$1,11E^{-07}$
327680	$7,03E^{-06}$	$2,75E^{-08}$
1310720	$1,76E^{-06}$	$6,87E^{-09}$

Библиография

[1] W3C — Extensible Markup Language (XML)1.0 (Fifth Edition). W3C,2008,
gost.e-ecolog.ru

[2] IEC 61966-2-1 — Multimedia systems and equipment — Colour measurement and management — Part 2-1: Colour management — Default RGB colour space — sRGB

[3] File Format Specification Z.I.P. — Version: 6.3.4. PKWARE Inc., 2014,
gost.e-ecolog.ru

[4] Info-ZIP
gost.e-ecolog.ru

УДК 774:002:006.354

ОКС 71.020
71.100.01
77.160

Ключевые слова: аддитивные технологии, аддитивные технологические процессы, базовые принципы

Редактор *Н.Е. Рагузина*
Технический редактор *И.Е. Черепкова*
Корректор *О.В. Лазарева*
Компьютерная верстка *Л.А. Круговой*

Сдано в набор 03.04.2019. Подписано в печать 23.04.2019. Формат 60×84 1/8. Гарнитура Ариал.
Усл. печ. л. 3,26. Уч.-изд. л. 2,95.
Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении ФГУП «СТАНДАРТИНФОРМ» для комплектования Федерального информационного фонда стандартов, 117418 Москва, Нахимовский пр-т, д. 31, к. 2.

gost.e-ecolog.ru gost.e-ecolog.ru