DOI: 10.15514/ISPRAS–2020–32(5)–13

Модификация алгоритма Marching Cubes для получения трехмерного представления плоского изображения

[1] *Д.И. Хернандес Фарсас, ORCID: 0000-0002-7435-7753 <di.hernandez@ugto.mx>*
[1] *Р. Гусман Кабрера, ORCID: 0000-0002-9320-7021 <guzmanc@ugto.mx>*
[1] *Т. Кордова Фрага, ORCID: 0000-0002-6486-7530 <theo@_sica.ugto.mx>*
[1] *Х.З. Уамани Луна, ORCID: 0000-0003-1776-9820 <zajorize@gmail.com>*
[2] *Х.Ф. Гомез Агилар, ORCID: 0000-0001-9403-3767 <jfranciscogoma@hotmail.com>*

[1] *Университет Гуанахуато,*
Мексика, 36000, Гуанахуато, ул. Ласкурен де Ретана, 5
[2] *Национальный совет по науке и технологиям,*
Мексика, 03940, Мехико, район Бенито Хуарес, Аллея Повстанцев, 1582

Аннотация. Совмещение трехмерной модели с изображением можно рассматривать как установку визуальных соответствий, извлекаемых из данных, описывающих эти изображения. Эта непростая задача еще более усложняется, если при построении изображений используются разные методы визуализации. В статье представлен подход, позволяющий сопоставлять характеристики, обнаруживаемые в двух различных видах изображений – фотографиях и 3D-моделях – с использованием общего 2D-представления. Наш подход основан на модификации алгоритма Marching Cubes, позволяющей избежать неоднозначных решений, не добавляя вычислений при обработке каждого куба. Мы разделяем идею о решающей важности разделения случаев эквивалентности на два класса. С учетом всех возможных состояний внутри и снаружи в четырех углах одной грани куба имеются только четыре нетривиальных варианта после исключения эквивалентных вариантов путем вращения. Полученные результаты демонстрируют применимость предлагаемой методики.

Ключевые слова: Marching Cubes; трехмерное представление; изоповерхности

Для цитирования: Эрнандес Фариас Д.И., Гусман Кабрера Р., Кордова Фрага Т., Уамани Луна Х.З., Гомез Агилар М. Модификация алгоритма Marching Cubes для получения трехмерного представления плоского изображения. Труды ИСП РАН, том 32, вып. 5, 2020 г., стр. 167-180. DOI: 10.15514/ISPRAS–2020–32(5)–13

Modification of the Marching Cubes Algorithm to Obtain a 3D Representation of a Planar Image

[1] *D.I. Hernández Farías, ORCID: 0000-0002-7435-7753 <di.hernandez@ugto.mx>*
[1] *R. Guzmán Cabrera, ORCID: 0000-0002-9320-7021 <guzmanc@ugto.mx>*
[1] *T. Cordova Fraga, ORCID: 0000-0002-6486-7530 <theo@_sica.ugto.mx>*
[1] *J.Z. Huamaní Luna, ORCID: 0000-0003-1776-9820 <zajorize@gmail.com>*
[2] *J.F. Gomez Aguilar, ORCID: 0000-0001-9403-3767 <jfranciscogoma@hotmail.com>*

[1] *Universidad de Guanajuato,*
Lascuráin de Retana No. 5, Col. Centro C.P. 36000, Guanajuato, Gto., México
[2] *Consejo Nacional de Ciencia y Tecnología,*
Av. Insurgentes Sur 1582, Alcaldía Benito Juárez, C.P. 03940, Ciudad de México

Abstract. The registration of a 3D model over an image can be seen as the alignment of visual correspondences extracted from these two data. This is a challenging task and it is even more complex when the two images have a different modality. This paper introduces an approach that allows matching features detected in two different modalities: photographs and 3D models, by using a common 2D representation. Our approach is based on a modifcation of the Marching Cubes algorithm aiming to remove ambiguous cases without adding further calculations in each cube. We share the idea about the crucial importance of splitting the equivalence cases into two classes. Considering all the possible states inside/outside in the four corners of a cube side, indeed, there are only four non-trivial cases after eliminating those equivalences through the rotation. The obtained results allow us to validate the feasibility of the proposed methodology.

Keywords: marching cubes; 3D representation; Iso-surfaces

For citation: Hernández Farías D.I., Guzmán Cabrera R., Cordova Fraga T., Huamaní Luna J.Z., Gomez Aguilar J.F. Modification of the marching cubes algorithm to obtain a 3D representation of a planar image. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 5, 2020. pp. 167-180 (in Russian). DOI: 10.15514/ISPRAS–2020–32(5)–13

1. Введение

Важность объемной визуализации как инструмента анализа данных привела к разработке нескольких методов, позволяющих создавать проекции на основе двухмерных изображений. Методы визуализации подразделяются на три основные категории: мультипланарный рендеринг (multi-planar rendering), объемный рендеринг (volume rendering) и поверхностный рендеринг (surface rendering) [1]. В каждая из этих категорий имеется своя техника реконструкции изображения.

Мультипланарный рендеринг позволяет визуализировать оттенки серого в произвольных поперечных сечениях на основе пространственных данных. *Объемный рендеринг* используется при генерации изображения с использованием рейкастинга (ray casting, бросание лучей). *Поверхностный рендеринг* обеспечивает визуализацию заданного объекта на основе данных, представленных в виде набора базовых элементов (вокселей, voxel), которые определяют границы заданных структур.

Существуют различные алгоритмы поверхностного рендеринга, в число которых входит алгоритм *Marching Cubes* (MC, алгоритм *шагающих кубиков*), широко распространенный в таких областях, как биология, биохимия, биомедицина, экология и т.д. [2, 3]. MC может успешно применяться для генерации трехмерных медицинских изображений на основе нескольких источников, таких как компьютерная томография (КТ) или магнитно-резонансная томография (МРТ).

Реконструкция медицинских изображений может способствовать развитию межплатформенных систем, которые можно использовать для улучшения клинической

диагностики, а также для выполнения точных хирургических операций с низким уровнем риска [1, 4].

В данной работе мы представляем модифицированную версию алгоритма MC, поддерживающей генерации предельно детализированных изображений. Работа затрагивает следующие аспекты визуализации.

- Редактирование небольших участков поверхности без потребности в реконструкции всей поверхности, содержащейся на изображении.
- Генерация сетки с как можно меньшим числом искажений между треугольниками или областями.
- Оптимизация использования имеющихся ресурсов хранения данных путем совершенствования процесса генерации поверхности. Это достигается за счет обеспечения наличия у кубов общей.

Статья имеет следующую структуру. В разд. 2 представлен краткий обзор литературы по данной тематике. В разд. 3 описан наш подход, основанный на модифицированном алгоритме MC. В разд. 4 обсуждаются проведенные эксперименты, а также полученные результаты. В заключение, в разд. 5 представлены выводы и указаны направления будущих исследования.

2. Родственные работы

Алгоритм MC был впервые предложен Уильямом Э. Лоренсеном (William E. Lorensen) и Харви Э. Клайном (Harvey E. Cline) [5]. Это один из наиболее широко используемых методов поверхностного рендеринга, который считается надежным и простым. Более подробно с данным алгоритмом можно ознакомиться в [3]. В *алгоритме MC* используется подход «разделяй и властвуй» для последовательной обработки пространственных данных с использованием *вокселей*, эквивалентных *кубам*. С каждым кубом ассоциируется изоповерхность с изозначением h (задаваемым в качестве входного аргумента алгоритма), на которой генерируется триангуляционная сетка путем определения того, как куб пересекается с этой поверхностью, после чего алгоритм переходит к следующему кубу. Пересечение поверхностей с кубами находится путем сравнения значений их вершин с h. Если значение вершины больше или равно h, то вершина считается внутренней. В противном случае она считается внешней. Имеется 256 возможных способов пересечения поверхности с кубом. Это число можно сократить из-за того, что некоторые пары вариантов пересечения являются инверсными или симметричными, оставив лишь 15 вариантов, показанных на рис. 1. После определения типа пересечения выполняется построение триангуляционной сетки.

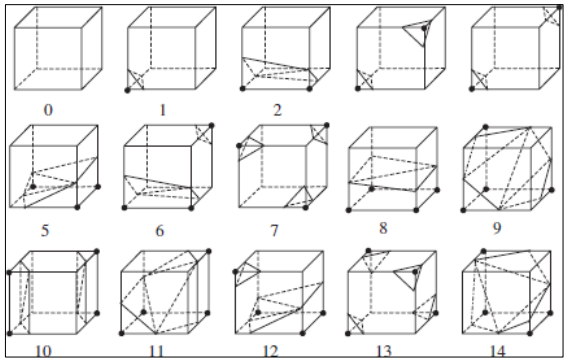


Рис. 1. Пятнадцать вариантов пересечения, учитываемых в традиционном алгоритме Marching Cubes

Fig. 1 Fifteen intersection cases considered in the traditional Marching Cubes algorithm

Алгоритм MC применялся для решения широкого спектра задач, и при этом были выявлены различные возможности для его усовершенствования. Кроме того, важно отметить, что у алгоритма шагающих кубиков есть некоторые недостатки, такие как вероятность появления видимых артефактов огранки, сложность выбора изозначения для обеспечения корректной аппроксимации поверхности, потеря информации при генерации небольших структур и т.д.

В некоторых описанных в литературе исследованиях [3-5] в основном изучались следующие аспекты:

(i) топология;

(ii) избыточность и неоднозначность при генерации сетки;

(iii) тип данных и скорость.

В [6] Дарст (Durst) описывает проблему наличия нескольких способов триангуляции для вариантов пересечения в алгоритме MC. Неоднозначность выбора интерполяции рассматривается в работе [7]. В [8] впервые было предложено расширить набор вариантов алгоритма MC. Позднее для устранения внутренних неоднозначностей авторы [9] предложили расширить набор вариантов триангуляции до 33 (этот вариант MC принято называть *Marching Cubes 33 – MC33*). Расширенная версия *MC33* представлена в [10]. Предлагаемый метод предусматривает наличие дополнительных меток узлов, позволяющих избежать вырождения треугольников. Одним из важнейших усовершенствований алгоритма является возможность сохранения топологии трилинейной интерполяции. Авторы экспериментируют как с реальными, так и со случайно сгенерированными наборами данных. Сравнение производительности предложенной версии с другими алгоритмами, основанными на MC, позволяет заострить внимание на их различиях, относящихся к топологическим проблемам в генерируемой сетке.

Одной из областей, нуждающихся в развитии технологий 3D-реконструкции, является медицина. В [11] представлен подход для 3D-реконструкции в медицинской области, сочетающий в себе алгоритм MC с алгоритмом начального заполнения (seed-occupying algorithm). Для генерации изоповерхности вместо треугольников используются многоугольники. Затем уменьшается число граней, а также смягчается проблема неоднозначности. Результаты экспериментов на реальных медицинских КТ-снимках человеческого черепа свидетельствуют о преимуществах предложенного подхода по сравнению с традиционным алгоритмом с точки зрения требуемых объема памяти и времени обработки. Лонг (Long) и Нагамуне (Nagamune) в [4] описывают алгоритм MC для реконструкции эндоскопических изображений. В их подходе используется метод выбора средней точки для получения позиции пересечения поверхности, и авторы демонстрирует перспективные результаты. В работе [12] авторы выполняют трехмерную реконструкцию поверхности на основе данных КТ брюшной полости (тканей печени), применяя алгоритм наращивания областей (region growing algorithm) для сегментации изображения, а также методы математической морфологии. В предлагаемом подходе построение контура поверхности производится аналогично стандартному алгоритму MC, однако пересечение выполняется только по пикселям со значением 1 бинаризованных изображений. Кроме того, предлагается усовершенствование алгоритма, позволяющее уменьшить число избыточных операций, выполняемых в ходе построения. Наконец, в работе [1] представлен обзор различных методов трехмерной реконструкции на основе алгоритма MC, применяемых в области черепно-лицевой хирургии. В этих методах применяется параллельный обход, а также сокращается число фрагментов изоповерхности (iso-surface patches).

Как и в других областях, связанных с вычислениями, улучшению методов объемной визуализации способствовали продвижения в разработке более мощного графического вычислительного оборудования (такого как графические процессоры, GPU). В [2] описан подход к ускорению алгоритма MC на основе использования GPU. Кроме того, авторы также

обсуждают различные способы повышения производительности MC с помощью применения вспомогательных пространственных структурах данных. Некоторые эксперименты, проведенные над различными наборами пространственных данных, подтвердили эффективность предложенной модели. Предложенный метод демонстрирует существенную разницу в скорости рендеринга – в 18 раз – по сравнению подходом, основанном на использовании только обычных процессоров.

3. Наш подход

3.1 Алгоритм Marching Cubes

Предлагаемые модификации алгоритма MC направлены на устранение неоднозначных вариантов без потребности в дополнительных вычислениях в каждом кубе. Мы разделяем идею о решающем значении разделения эквивалентных вариантов на два класса [13].

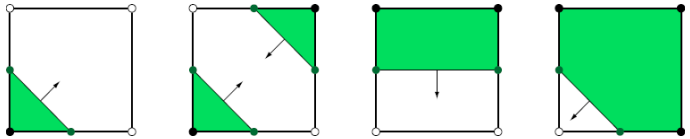


Рис. 2. Нетривиальные конфигурации, допустимые в предложенных модификациях алгоритма MC. Зеленая область обозначает сплошное пространство. Стрелки представляют собой направление нормали поверхности по отношению к внешней части ребер

Fig. 2. Non-trivial configurations allowed by the proposed modifications to the MC algorithm. The green region represents the solid space. The arrows represent the surface normal direction towards outside in the edges

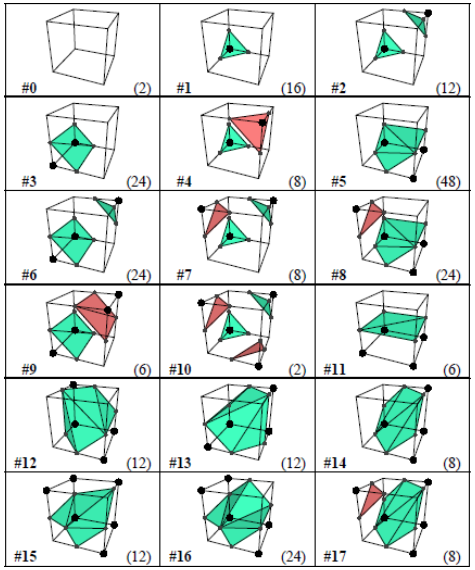


Рис. 3. Предлагаемые классы эквивалентности

Fig. 3. Equivalence classes proposed

С учетом всех возможных расположений углах одной стороны куба внутри и снаружи изоповерхности можно установить, что в действительности существуют только четыре нетривиальных варианта после исключения эквивалентных вариантов путем вращения.

Требование *пространственного соответствия ребер* выполняется, когда все вершины соединены на смежных гранях, разделяющих нижний угол.

Далее, в соответствии со стратегией *предпочтительной полярности* (preferred polarity), четыре варианта, показанные на рис. 2, являются нетривиальными конфигурациями ребер, допустимыми в нашем подходе [14]. Черные точки обозначают внутренние вершины, белые – внешние [13].

Мы предлагаем разрешать эту проблемную ситуацию, устраняя инверсии отношения эквивалентности для классов, порождающих неоднозначность, и оставляя только вращение. Для вариантов с четырьмя внутренними углами на поверхности изображения мы образуем три новых класса. Образуется набор из 18 классов эквивалентности, показанных на рис. 3. Классы 15, 16 и 17 являются новыми классами, состоящими из инверсного представления классов 2, 6 и 7 и их соответствующих вращений [15].

В каждой ячейке таблицы число в левом нижнем углу означает номер класса, а число в правом нижнем углу – количество вариантов, относящихся к данному классу эквивалентности, из 256 всех вариантов. Черная точка в углу обозначает, что он находится в пределах объема изображения, а угол без точки обозначает, что он находится снаружи. Зеленые треугольники находятся перед гранью куба, а красные треугольники – позади материала куба [16].

В идеале, число вариантов триангуляции при переходе от одного куба к другому должно быть одного и того же порядка. Для достижения этого можно разделить каждый переходной блок на два блока меньшего размера, как показано на рис. 4. Таким образом, переходный куб делится на части максимального и среднего разрешения плоскостью, параллельной грани. Левый блок (максимального разрешения) триангулируется с помощью выборочных значений с его граней. Значения углов *A*, *B*, *C* и *D* дублируются на противоположной грани куба. Затем производится переход внутри куба, что позволяет использовать более контролируемые данные. С другой стороны, в правом блоке триангуляция производится обычным способом с использованием данных среднего разрешения [17].

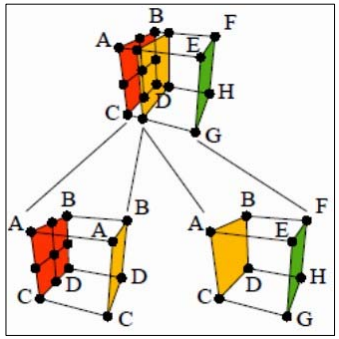


Рис. 4. Переходный куб разделяется на два блока меньшего размера. Левый куб триангулируется на основе выборочных значений с грани максимального разрешениея, а правый – по традиционному алгоритму MC

Fig. 4. A transition cube is divided into two smaller blocks. The left side cube is triangulated by exploiting sample values coming from the maximum resolution face, while the one in the right is done with the traditional MC algorithm

3.2 Предлагаемая методика

На рис. 5 схематично показана методика, предлагаемая в нашей работе. Как можно видеть, задается двухмерное изображение, из которого нужно получить трехмерное представление. Это изображение может быть, например, медицинским снимком. Представленные в нашей

статье результаты получены при использовании в качестве исходного изображения снимка МРТ. Ниже приводится описание каждого из этапов методики.

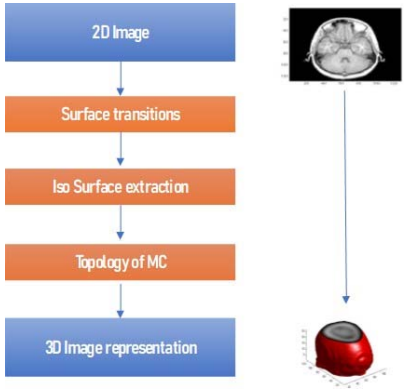


Рис. 5 Предлагаемая методика
Fig. 5. Proposed methodology

3.2.1 Переходы кубов на поверхности

Сетки разделяются блоками кубов, соединенных со следующим уровнем восьмью кубами. Позиции вторичных вершин вычисляются линейным преобразованием внутренних позиций граничного куба таким образом, что полные кубы масштабируются до меньшего размера в зависимости от их расположения [18]. Для этого для координат (x, y, z) вычисляются $(\Delta x, \Delta y, \Delta z)$ с использованием формулы (1):

$$\Delta x = \begin{cases} (1-2^{-k}x)w(k) & if \quad x < 2^k \\ 0 & if \quad 2^k \le x \le 2^k(s-1) \\ (s-1-2^{-k}x)w(k) & if \quad x \le 2^k(s-1) \end{cases}. \tag{1}$$

Здесь k – значение уровня детализации (Level of Detail – LOD), а s – размер блока. Функция $w(k)$ задает ширину переходов куба с LOD равным k. В нашем подходе она определяется как $w(k) = 2^{k-2}$. Δy и Δz вычисляются аналогично. Вершины на гранях полного разрешения переходных кубов остаются неизменными. Добавление смещений приводит к тому, что в области перехода, как показано на рис. 6, поверхность становится настолько плоской, что могут возникать деформации.

Эти проблемы могут быть устранены путем проецирования вектора смещения вершины на касательную плоскость, проходящую через исходное положение вершины, с использованием формулы:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 1-N_x^2 & -N_xN_y & -N_xN_z \\ -N_xN_y & 1-N_y^2 & -N_yN_z \\ -N_xN_z & -N_yN_z & 1-N_z^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}, \tag{2}$$

где N – вершина блока нормальной длины.

На рис. 6-(a) к вершинам низкого разрешения применяется линейное преобразование в соответствии с (1), чтобы освободить место для переходного куба. Это может привести к нежелательным последствиям при создании плоской области и вогнутости. Для решения этих проблем применяется формула (2) для проецирования разности вершин на касательную плоскости относительно нормальной вершины N.

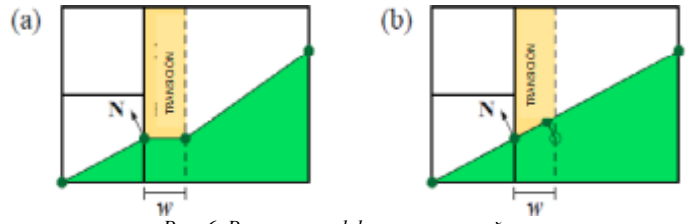


Рис. 6. Различные эффекты проекций
Fig. 6. Different effects of projections

При выборе первичной и вторичной вершин принято использовать основную вершину в качестве позиции углов, общих для блоков разного уровня детализации.

3.2.2 Извлечение изоповерхностей

Основной целью является построение точного контура заданного трилинейного скалярного поля по следующему параметрическому описанию:

$$\begin{aligned} s(x,y,z) = (1-x)(1-y)(1-z)c_{000} + (1-x)(1-y)zc_{001} \\ +(1-x)y(1-z)c_{010} + (1-x)yzc_{011} + x(1-y)(1-z)c_{100} \\ +x(1-y)zc_{101} + xy(1-z)c_{110} + xyzc_{111}, \end{aligned} \tag{3}$$

где c_{ijk} $(i, j, k) \in [0, 1]$ – скалярные значения в вершине заданного куба.

M\ Контур задается фиксированным пространством τ, состоящим из всех точек в $(x, y, z)^T$, преобразованных в соответствии с

$$s(x, y, z) = \tau. \tag{4}$$

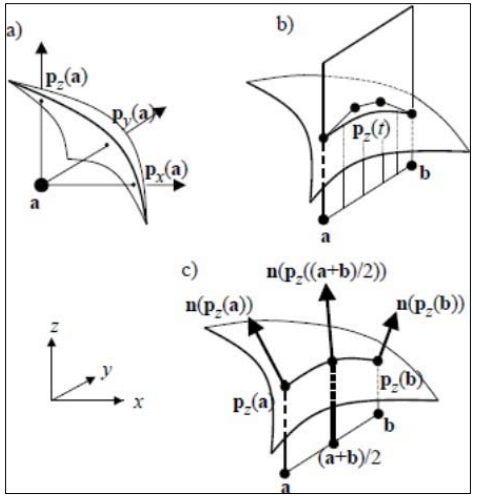


Рис. 7. Проекции a на контур в направлениях x −, y − и z −
Fig. 7 Projections of a over the contour in x −, y −, and z − directions

При рассмотрении точки $a = (x_a, y_a, z_a)^T$ нам было бы интересно найти пересечение заданной линии с контуром, определенном формулами (3) и (4), поскольку требуется решить кубическое уравнение [19].

В частном случае, когда имеется линия, параллельная одной из осей координат, задачу можно упростить до решения линейного уравнения. Пусть $p_x(a)$ является пересечением линии a +

$\lambda(1,0,0)^T$ с контуром, определяемым соотношениями (3) и (4), $p_y(a)$ – пересечением линии $a + \lambda(0,1,0)^T$ с этим контуром, и $p_z(a)$ – пересечением линии $a + \lambda(0,0,1)^T$. Тогда $p_x(a)$, $p_y(a)$ и $p_z(a)$ – это проекции a на контур по направлениям $x-$, $y-$ и $z-$. Это означает, что можно построить три точки на контуре для заданной точки простым образом. На рис. 7 приводится пример. На рис. 7a представлены проекции $p_x(a)$, $p_y(a)$ и $p_z(a)$ для заданной точки в направлении контура $x-$, $y-$, $z-$. Как видно из рис. 7b, проекция $p_z(t)$ отрезка линии $(1-t)a + tb$ на контуре является рациональной кубической кривой. Наконец, конфигурация для расчета точки привязки $p_z(t)$ показана на рис. 7c.

Имея точку a, которая служит для построения контура, определяемого формулой (3), можно вычислить его нормаль – вектор $n(a)$ по формуле:

$$n(a) = \left(s_x(x_a, y_a, z_a), s_y(x_a, y_a, z_a), s_z(x_a, y_a, z_a)\right)^T, \tag{5}$$

где s_x, s_y и s_z являются частичными производными s, определяемого формулой (3). Кривые контура строятся по проекциям отрезков линий.

Расширяя понятие контурных кривых до параметрических поверхностей в контуре, определенном с формулами (3) и (4), можно получить треугольник:

$$x(\mu, \nu, \omega) = \mu a + \nu b + \omega c. \tag{6}$$

Координатами барицентра являются вершины a, b и c. Положим $\mu + \nu + \omega = 1$. Тогда $p_x(x(\mu,\nu,\omega))$, $p_y(x(\mu,\nu,\omega))$ и $p_z(x(\mu,\nu,\omega))$ являются проекциями x на контур, определенный с помощью формул (3) и (4). См. рис. 8.

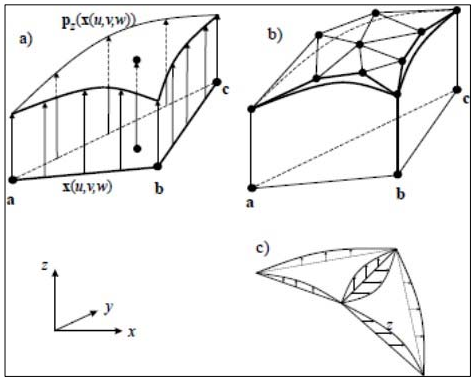
1. 

Рис. 8. Проекция $p_z(x(\mu,\nu,\omega))$. (a) значение $p_z(x(\mu,\nu,\omega))$ получается путем проецирования каждой из формулы (6) в направлении $z-$ в контуре, определенном формулами (3) и (4). (b) $p_z(x(\mu,\nu,\omega))$ является рациональной кубической поверхностью. (c) два смежных треугольника, образованных алгоритмом MC в разных направлениях

Fig. 8. Projections of $p_z(x(\mu,\nu,\omega))$. (a) the value of $p_z(x(\mu,\nu,\omega))$ is obtained by means of the projection of each point of Eq. (6) in the $z-$ direction in the contour defined by Eq. (3) and 4. (b) $p_z(x(\mu,\nu,\omega))$ is a rational cubic surface. (c) two adjacent triangles generated by MC in different directions

Поверхности $p_x(x(\mu,\nu,\omega))$, $p_y(x(\mu,\nu,\omega))$ и $p_z(x(\mu,\nu,\omega))$ являются рациональными кубическими поверхностями. Например, $p_z(x(\mu,\nu,\omega))$ можно описать как треугольную рациональную поверхность Безье:

$$p_z(x(\mu,\nu,\omega)) = \frac{\sum_{i+j+k=3} \omega_{ijk} b_{ijk} B^3_{ijk}(\mu,\nu,\omega)}{\sum_{i+j+k=3} \omega_{ijk} B^3_{ijk}(\mu,\nu,\omega)}, \tag{7}$$

где B^3_{ijk} — многочлены [20] и

$$\begin{aligned}
w_0 &= z_{n(p_z(a))},\\
w_1 &= \tfrac{4}{3} z_{n(p_z(\frac{a+b}{2}))} - \tfrac{1}{3} z_{n(p_z(b))},\\
w_2 &= \frac{4}{3} z_{n\left(p_z\left(\frac{a+b}{2}\right)\right)} - \frac{1}{3} z_{n(p_z(a))},\\
w_3 &= z_{n(p_z(b))},\\
b_0 &= p_z(a),\\
b_3 &= p_z(b).
\end{aligned} \tag{8}$$

Значения $w_0, \ldots, w_3$ определяются координатами $z-$ некоторых нормальных векторов контура. Например, w_0 определяется компонентом $z-$ нормального вектора $p_z(a)$. Кривые относительно $p_x(t)$ и $p_y(t)$ можно получить аналогичным образом.

В базовом случае для получения контура (определенного с помощью формул (3) и (4)), нежно воспользоваться алгоритмом MC вместе с проекциями каждого исходящего треугольника в контуре. К сожалению, это может привести к проблемам в смежных треугольниках из-за направлений проекции. Чтобы избежать этого, все поверхности срезов быть треугольными рациональными кубическими, а не треугольными [19].

Треугольник (a, b, c), полученный с помощью алгоритма MC представляет собой контур с учетом того, что требуется:

1. определить проекции в направлениях $q_{ab}, q_{bc}, q_{ca} \in x, y, z$ на краевых кривых;
2. определить проекцию в направлении q_{abc} всего треугольника.
3. спроецировать границы треугольника (a, b, c) по контуру в соответствии с (1); границами треугольника (a, b, c) являются кривые x_{ab}, x_{bc} и x_{ca}:
$$\begin{aligned}
x_{ab}(t) &= p_{q_{ab}}((1-t)a + tb)\\
x_{bc}(t) &= p_{q_{bc}}((1-t)b + tc)\\
x_{ca}(t) &= p_{q_{ca}}((1-t)c + ta)
\end{aligned}$$
4. спроецировать x_{ab}, x_{bc} и x_{ca} в направлении q_{abc} на плоскость, определенную a, b и c; для получения кривых y_{ab}, y_{bc} и y_{ca} на плоскости (a, b, c) используются краевые кривые области в окончательном варианте.
5. рассчитать обрезанную поверхность в проекции $p_{q_{ab}}(\mu a, \nu b, \omega c)$ с $\mu + \nu + \omega = 1$; область обрезанной поверхности задается границами y_{ab}, y_{bc} и y_{ca} [21].

3.2.3 Топология Marching Cubes

Мы стремимся определить набор внутренних точек, которые служат для получения топологически точной триангуляции для каждого варианта. Вершины многоугольника помечаются метками v_1, v_2, v_3, v_4, v_5 и v_6. Триангуляция этого многоугольника производится в ребрах (v_2, v_6) и (v_3, v_5); их не следует использовать, так как у контура нет ребер между этими вершинами в верхней грани ячейки [19].

Предположим, что все точки контура имеют нормальное направление в (x, y, z). Для контура, полученного с помощью соотношений (3) и (4), существует не менее двух точек x_0 и x_1 с нормалью в направлении x.

Определим d:

$$d = ar^2 + br + c \tag{9}$$

Теперь можно описать алгоритм генерации *топологически точных* шагающих кубиков:

1. создать многоугольники, вмещающие полное сглаженное изображение; эти многоугольники называются *наружными кольцами*;
2. рассчитать *внутренние кольца* по формуле (9);

3. если внутреннее кольцо не является реальным или находится за пределами изображения, то внешние кольца триангулируются независимо; в противном случае продолжить;
4. проверить связность между *внутренним кольцом* и каждым из *наружных колец*; если внутреннее кольцо и одно из внешних колец принадлежат к одному и тому же сегменту контура, то зона между ними триангулируется.

4. Эксперименты

Нашей основной целью является создание трехмерной модели заданного объекта на основе 2D-изображения, полученного с помощью сканера. Для оценки эффективности предложенного подхода был использован реальный снимок МРТ. Далее будет описано, как генерировалась 3D-изображение с использованием различных этапов, описанных ранее.

Сначала для определения границ на поверхности исходного изображения мы используем среднюю интенсивность пикселей и обозначаем эти границы разными цветами (рис. 9).

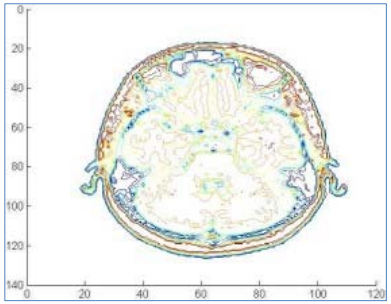


Рис. 9. Определение границ на поверхности изображения на плоскости
Fig. 9. Defining the borders on the surface of an image in the plane

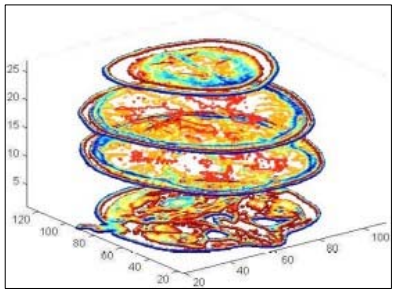


Рис. 10. Создание изображения в виде срезов на плоскости
Fig. 10. Generate a sliced image, by defining the edges in a plane

Следующий шаг заключается в генерации изображения, представленного в виде срезов. Это делалось путем определения ребер на плоскости, как описывалось в предыдущем разделе. На рис. 10 приводится изображение в виде срезов.

Как можно заметить, удалось определить плоскости различной интенсивности, в которых показаны текстурные пластины с четырьмя изображениями в матрице размером 1024x1024, спроецированные из единственной текстуры карты расположения двухмерных подслоев (матрицы 3x3) в поднаборе данных. В одной восьмой части текстуры каждого среза копируются противоположные данные для частичного воспроизведения виртуальных стыков этих плоскостей, что дает конечный результат в виде трехмерного изображения. Затем нужно заполнить изображение в виде срезов, чтобы получить трехмерную версию, см. рис. 11. Это

достигается путем предварительного извлечения из поверхностного слоя отфильтрованного образца, так как в поверхностной сетке отдельные текстуры часто повторяются.

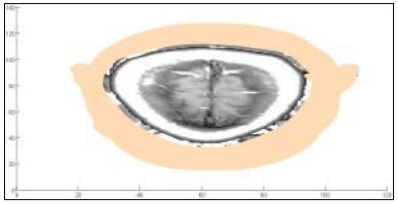


Рис. 11 Изображение в виде срезов для получения 3D-версии
Fig. 11 Sliced image in order to produce a 3D version

Для этого требуется применять методы сглаживания и заполнения пустот. Мы использовали метод, описанный в 3.2.2.

На рис. 12 изображен один срез изображения при увеличении интенсивности пикселей. Как можно заметить, что под воздействием этого процесса уменьшаются контрастность.

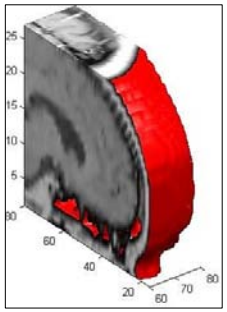


Рис. 12 Срез изображения при увеличении интесивности пикселей
Fig. 12 Slide of the image when its pixels are elevated

Как видно, увеличение яркости всех пикселей достигается за счет фильтрации краев. На рис. 12 показано увеличение интенсивности 10 элементов, что позволяет наблюдать внутреннюю часть изображения на определенном уровне яркости и, таким образом, иметь возможность оценивать эффект при различных вариантах увеличения пикселей.

5. Заключение

В этой статье описана методика получения 3D-представления двухмерного изображения. Предложенная методика заключается в модификации алгоритма Marching Cubes и включает три этапа, первый из которых помогает нам получить выборку первичных и вторичных вершин, где в качестве главной вершины используются углы, в которых соединяются блоки различных уровней детализации. На втором этапе полученный алгоритмом MC треугольник представляется контуром и, наконец, идентифицируется топология, которая определяет внутренние точки, необходимые для получения топологически точного треугольника для каждого варианта. Полученные результаты показывают практичность реализованной методики. В рамках дальнейшей работы планируется разработать метрики, которые позволили бы измерить производительность предложенного алгоритма.

Список литературы / References

[1] T. Senthil Kumar and Anupa Vijai. 3D Reconstruction of Face from 2D CT Scan Images. Procedia Engineering, vol. 30, 2012, pp. 970-977.

[2] Marcos Cirne and Helio Pedrini. Marching Cubes Technique for Volumetric Visualization Accelerated with Graphics Processing Units. Journal of the Brazilian Computer Society, vol. 19, 2013, pp. 223-233.

[3] Timothy Newman and Hong Yi. A survey of the Marching Cubes algorithm. Computers Graphics, vol. 30, issue 5, 2006, pp. 854-879.

[4] Zhongjie Long and Kouki Nagamune. A Marching Cubes Algorithm: Application for Three-dimensional Surface Reconstruction Based on Endoscope and Optical Fiber. Information, International Information Institute, vol. 18, issue 4, 2015, pp.1425-1437.

[5] William E. Lorensen and Harvey E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. ACM Computer Graphics, vol. 21, issue 4, 1987, pp. 163-169.

[6] M.J. Durst. Letters: Additional Reference to Marching Cubes. ACM Computer Graphics, vol. 22, issue 2, 1988, pp, 72-73.

[7] Gregory Nielson and Bernd Hamann. The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes. In Proc. of the 26th IEEE Conference on Visualization, 1991, pp. 83-91.

[8] B. Natarajan. On Generating Topologically Consistent Isosurfaces from Uniform Samples. The Visual Computer, vol. 11, 1994, pp. 52–62.

[9] Evgeni V. Chernyaev. Marching Cubes 33: Construction of Topologically Correct Isosurfaces. Technical report, Institute for High Energy Physics, 1995.

[10] Lis Custodio, Sinesio Pesco, and Cláudio Silva. An Extended Triangulation to the Marching Cubes 33 Algorithm. Journal of the Brazilian Computer Society, vol. 25, 2019, article no. 6.

[11] F. Gong and X. Zhao. Three-Dimensional Reconstruction of Medical Image Based on Improved Marching Cubes Algorithm. In Proc. of the International Conference on Machine Vision and Human-machine Interface, 2010, pp. 608-611.

[12] S. Liu and J. Peng. Optimization of Reconstruction of 2D Medical Images based on Computer 3D Reconstruction Technology. Journal of Digital Information Management, vol. 13, issue 3, 2015, pp. 142-146.

[13] Maxim A. Olshanskii, Arnold Reusken, and Jörg Grande. A Finite Element Method for Elliptic Equations on Surfaces. SIAM Journal on Numerical Analysis, vol. 47, issue 5, 2009, pp. 3339-3358.

[14] Alexey Y. Chernyshenko and Maxim A. Olshanskii. An Adaptive Octree Finite Element Method for PDEs posed on Surfaces. Computer Methods in Applied Mechanics and Engineering, vol. 291, 2015, pp. 146-172.

[15] A. Bonito, R.H. Nochetto, and M.S. Pauletti. Dynamics of Biomembranes: Effect of the Bulk Fluid. Mathematical Modelling of Natural Phenomena, vol. 6, issue 5, 2011, pp. 25-43.

[16] Matteo Cacciari and Gavin P. Salam. Dispelling the N 3 myth for the kt jet-finder. Physics Letters B, vol. 641 issue 1, 2006, pp. 57-61.

[17] Thatcher Ulrich. Rendering Massive Terrains Using Chunked Level of Detail Control. In Proc. of the 29th Annual Conference on Computer Graphics and Interactive Techniques, 2002.

[18] Gregory M. Nielson, Liyan Zhang, Kun Lee, and Adam Huang. Parameterizing Marching Cubes Isosurfaces with Natural Neighbor Coordinates. Lecture Notes in Computer Science book series, vol. 4975, 2008, pp. 315-328.

[19] Allen Van Gelder and Jane Wilhelms. Topological Considerations in Isosurface Generation. ACM Transactions on Graphics, vol. 13, issue 4, 1994, pp. 337-375.

[20] Gerald E. Farin. Curves and Surfaces for Computer-Aided Geometric Design: A Practical Code. 4th edition. Academic Press, 1996, 429 p.

[21] Claudio Montani, Riccardo Scateni, and Roberto Scopigno. A modified look-up table for implicit disambiguation of Marching Cubes. The Visual Computer, vol. 10, 1994, pp. 353-355.

Информация об авторах / Information about authors

Делия Иразу ХЕРНАНДЕС ФАРСАС, PhD, доцент, факультет науки и техники. Научные интересы включают искусственный интеллект, машинное обучение, обработку естественного языка.

Delia Irazu HERNÁNDEZ FARÍAS, PhD, Associated Professor, Division of Sciences and Engineering. Research interests include Artificial Intelligence, Machine Learning, Natural Language Processing.

Рафаэль ГУСМАН КАБРЕРА – PhD, профессор кафедры электротехники, заведующий лабораторией. Область научных интересов: искусственный интеллект, машинное обучение, обработка изображений, интеллектуальный анализ текстов, обработка естественного языка.

Rafael GUZMÁN CABRERA – PhD, titular professor at the Electrical Engineering Department, Head of Laboratory. Research interests include Artificial Intelligence, Machine Learning, Image Processing, Text Mining, Natural Language Processing.

Теодоро КОРДОВА ФРАГА – PhD, профессор кафедры инженерной физики. Область научных интересов: физика, биомагнетизм, биомедицинские инструменты, обработка цифровых сигналов и изображений.

Teodoro CORDOVA FRAGA – PhD, Professor at the Department of Engineering Physics. Research interests: Physics, Biomagnetism, Biomedical Instrumentations, Digital Signal and Imaging Processing.

Хосе Сакариас УАМАНИ ЛУНА – магистр наук в области физики. Область научных интересов: физика, цифровая обработка изображений, медицинские приложения.

Jose Zacarías HUAMANÍ LUNA – Master of Sciences, Physics. Research interests: Physics, Digital Image Processing, Medical Applications.

Хосе Франсиско ГОМЕС АГИЛАР – PhD, профессор, заведующий лабораторией. Область прикладная математика, медицинская физика, дробное исчисление.

Jose Francisco GOMEZ AGUILAR – PhD, Professor, Head of Laboratory. Research interests include Applied Mathematics, Medical Physics, Mathematical Physics, Fractional Calculus.