

УДК 004;519;621

Особенности визуализации клеточных автоматов в области нанoeлектроники

Г. Я. Красников, Н. А. Зайцев, И. В. Матюшкин^а, С. В. Коробов

¹ОАО «НИИ молекулярной электроники»

Россия, 124460, г. Москва, Зеленоград, 1-ый Западный проезд, д. 12/1

E-mail: ^аimatyushkin@sitronics.com

Получено 13 февраля 2012 г.

Представлена формализация моделей визуализации клеточных автоматов (КА), рассмотрена их классификация. Также описаны возможные подходы к генерации звуковых рядов. Приведены частные случаи вариантов визуализации для КА различной размерности. На примере простого 3D КА указаны особенности визуализации наноразмерных систем.

Ключевые слова: клеточные автоматы, визуализация, нанотехнология, моделирование, мультимедиа

The peculiarities of cellular automata visualization in nanoelectronics

G. Ya. Krasnikov, N. A. Zaitsev, I. V. Matyushkin, S. V. Korobov

Moscow JSC “Research Institute of Molecular Electronics”, 12/1 1-y Zapadny Proezd, Zelenograd, Moscow, 124460 Russia

Abstract. — The general formalization of visualization models in cellular automata (CA) scope is presented, their classification is examined. It also describes possible approaches to the sound scales generation. We consider special cases of visualization manners for CA of various dimensions. By a simple 3D CA example the features of nanoscale systems imaging are indicated.

Keywords: cellular automata, visualization, nanotechnology, modeling, multimedia

Citation: *Computer Research and Modeling*, 2012, vol. 4, no. 4, pp. 735–756 (Russian).

Введение

Теория клеточных автоматов, связанная с именами фон Неймана и Конрада Цузе, имеет фундаментальное значение для всей науки и многообразное прикладное применение. Начиная с работ Т. Тоффли и Н. Марголуса 80-х гг. [Тоффли и др. 1991], клеточные автоматы (КА) стали использоваться в моделях физико-химических процессов. К середине 90-х гг. клеточно-автоматное моделирование проникло [Pinto et al. 2007] в гуманитарные науки при изучении мультиагентных систем в урбанистике (толпа, транспортная пробка). Обзорная статья [Ванаг, 1999] В. Ванага по вероятностным КА еще раз легитимизировала для отечественных исследователей клеточные автоматы как метод математического моделирования. Последнее десятилетие ознаменовалось бумом публикаций в самых разных разделах науки, связанных с КА-моделями [Лобанов, 2010]; одновременно с этим продолжает развиваться и математическая теория клеточных автоматов (на русском языке см. монографию В. Аладьева [Аладьев, 2009]). Реализация КА-моделей специализированным программным обеспечением, так называемым «машинами клеточных автоматов» (МКА), в той или иной мере неизбежно связана с вопросами визуализации. Анализ существующих МКА дан нами в [Матюшкин, 2011]; не потерял актуальности и обзор Л. Наумова. Вместе с тем авторы КА-моделей, несмотря на ряд стихийных находок, не подходили к вопросам визуализации системно. Восполнить этот пробел и призвана данная статья, продолжающая нашу предыдущую работу [Матюшкин, Хамухин, 2010], посвященную разработке МКА SoftCAM.

Особое внимание мы уделим области 3D КА. Во-первых, именно КА-модели 3D наиболее адекватно и естественно отражают технологию получения и свойства наноструктурированных материалов (в частности пористых [Псахье и др., 1998] и керамических, используемых в нано- и микросистемной технике, а также фотонных кристаллов). В частности, остро стоят вопросы переноса вещества и энергии в междоузлиях СБИС, в межслойном low-K диэлектрике (аэро- или ксерогель SiO_2). Во-вторых, именно здесь наиболее остры вопросы Data Mining (интеллектуального анализа данных). Те причины, которые и породили из потребностей решения задач гидродинамики сравнительно молодую (1987) дисциплину «научная визуализация», продолжают действовать и в отношении КА-моделирования [Клименко, 2000]: «... суперкомпьютеры выдают числа как из пожарного шланга, поэтому нужны какие-то нетривиальные подходы, чтобы все это переваривать».

Кроме того, КА часто используются непосредственно с целью создания образов искусства (арт-визуализация [Hopkins], алгоритмическая музыка). В качестве иллюстрации можно привести пример использования непрерывных КА (Continuous-Valued Cellular Automata) в программе CAPOW, моделирующей [Rucker, 2003] волновое уравнение. С точки зрения потенциальных применений для рынка мобильных мультимедийных устройств, популярных среди подростков, анализ возможных стратегий визуализации также представляет интерес.

Визуализация в контексте архитектуры МКА

Задачи визуализации, с одной стороны, решаются отдельной подсистемой в составе МКА, а с другой — неразрывно связаны со всем вычислительным процессом. Взаимосвязь может быть даже неожиданной: так, параллельное исполнение КА-модели по CUDA-технологии [Калгин, 2011; Алексеенко, 2010] задействует на аппаратном уровне те же ресурсы, что и рендеринг. Низкоуровневые вопросы визуализации на удаленном терминале при облачной/сетевой реализации КА-модели также требуют внимания. Однако сосредоточимся на более высокоуровневых аспектах визуализации КА, предположив ситуацию штатного использования OpenGL или DirectX.

Следует ли разделять задачи визуализации и задачи редактирования поля КА? Во многих программах, таких как Golly, они решаются для пользователя почти одинаковыми интерфейсными средствами. Несомненно, хорошая МКА должна, например, предоставлять возможность пользователю по результатам визуализации в режиме реального времени останавливать расчет в любой момент времени, изменять содержимое ячейки (или ячеек, если требу-

ется загрузить из библиотечного файла нужный паттерн) и вновь наблюдать за ходом продолженного расчета. Мы ясно видим, что задачи визуализации возникают не только на этапе представления выходной, но уже на этапе подготовки входной информации, прежде всего начальной глобальной конфигурации КА, которая с математической точки зрения является неотъемлемой частью КА-модели. В этом проявляется активный характер визуализации [Бондарев и др. 2010]. Правильнее, однако, разделять визуализацию данных непосредственно КА-расчета и визуализацию исходных данных КА-модели (даже если эти данные суть предшествующие результаты), причем параметры каждой из них могут существенно различаться. Прекрасной иллюстрацией этого положения является характерное для многих КА наличие неограниченно и довольно быстро растущих в пространстве конфигураций; чтобы понять сущность процесса, нужно включать автомасштабирование. При редактировании поля КА необходим крупный и средний масштаб с детальным отображением ячеек, а при визуализации КА с неограниченно растущей популяцией — средний и мелкий масштаб, когда экранный пиксел соответствует десяткам ячеек.

Многие симуляторы КА, в первую очередь реализованные как java-апплеты и потому похожие на занимательные игры, заставляют и разработчиков МКА неоправданно много внимания уделять именно режиму реального времени. Поэтому забывается, что, вообще говоря, визуализация данных и их генерация разделены во времени. Некоторым компромиссом выглядит запись видеоролика по скриншотам экрана, генерируемым в процессе КА-моделирования. Возможно, правильнее с точки зрения разделения времени и данных генерировать кадры видеоряда непосредственно по глобальным конфигурациям КА. Если провести параллель с графикой системы MATLAB, нужно предпочесть команду `imwrite` командам `print/saveas`. В вышеуказанном случае для 3D-модели можно было бы записать сразу четыре видеофайла (или даже один видеофайл с удвоенным размером кадра), каждый из которых описывал бы вид сцены при фиксированном положении камеры (без всякого внимания к фактическому рендерингу).

Следующим вопросом к разработчикам МКА является: сохранять ли параметры визуализации отдельно от КА-модели или включать их в ее описание? Такими очевидными параметрами для игры «Жизнь» является цвет пиксела или квадрата, отвечающий живой или мертвой клетке. Для 3D-КА естественно поставить соответствие, но никоим образом не отождествить, между ячейкой КА и вокселем/метаболом геометрической модели. Тогда вместо цветовой идентификации состояния ячейки интересно было бы использование форм и размеров метаболов. В любом случае допустимо говорить о параметрах визуализации абстрактно. Разумеется, эти параметры привязаны к моделям визуализации и имеют смысл только в рамках последних. Симуляторы КА предлагают обычно раскраску по умолчанию и обладают интерфейсными средствами смены цветовой карты (например, в Golly предлагается опция градиентной заливки между двумя цветами). Даже в развитых симуляторах вопросы визуализации и интерфейса визуализатора решаются примитивно. Рассмотрим суть проблемы на примере симулятора Golly и 2D-КА «Поколения», частная фигура «Взрыв» которого задана файлом `burst.mcl` (формата Mirek's Celebration MCell, широко распространенного симулятора, оперирующего данными модифицированного RLE-формата):

```
01 #MCell 4.00
02 #GAME Generations
03 #RULE 0235678/3468/9
04 #BOARD 400x400
05 #SPEED 0
06 #WRAP 1
07 #CCOLORS 9
08 #L 12.A$12.A$12.A$12.A$12.A$12.A$11.3A$11.3A$9.7A$9.7A$7.11A$25A$7.
09 #L 11A$9.7A$9.7A$11.3A$11.3A$12.A$12.A$12.A$12.A$12.A$12.A
```

(строки пронумерованы, строки-комментарии удалены, жирным шрифтом выделены строки, относящиеся к визуализации).

Вторая строка указывает, какому классу принадлежит КА, и одновременно подсказывает Golly, какую цветовую карту загрузить (по умолчанию — градиентную от желтого до красного для живых клеток, т. е. от юных до старых). Третья строка описывает правила КА, в данном случае длительность жизни поколения (9 ходов), а также правила выживания и рождения. Иногда в этой строке ставят текстовое имя функции перехода, ссылающееся на более подробное описание. Четвертая строка задает начальный размер поля КА и одновременно размер поля визуализации. Пятая и шестая строки относятся к особенностям алгоритма расчета. Седьмая строка утверждает, что количество использованных цветов — 9. В других вариантах mcl-файлов её вообще опускают, ведь это следует из строки 03 и принципа «каждому состоянию свой цвет». Строки 08 и 09 компактно задают конфигурацию креста по правилам RLE («.» — мертвая клетка, А — юная клетка, \$ — знак перехода на новую строку матрицы, число перед точкой или буквой — число повторений).

Если строки 02 и 04 относятся главным образом к КА-модели и лишь опосредованно к визуализации, то лишь строка 07 указывает за внимание к вопросам визуализации. Этот код не дает нам понять, какая цветовая карта будет использована, и не содержит ссылок на код, который помог бы нам узнать это. Для серьезных МКА наиболее перспективен путь структурирования данных в XML-коде и обращение к DOM-стандарту, продвигаемому консорциумом W3C. Однако осознание этого возникло лишь в последние годы [Veenhuis et al., 2002; Weimar, 2002]. В разрабатываемой нами МКА SoftCAM приняты за основу следующие принципы:

- Классы ячеек, шаблоны окрестностей, правила перехода (эта триада составляет собственно КА-модель), начальные конфигурации (паттерны), и параметры визуализации распределены по библиотечным файлам.
- Сборка происходит в виде репрезентирующего КА XML-файла, поля структуры которого доступны для редактирования пользователем (в частности, он может вставлять в редакторе SoftCAM собственный C++ код).
- Модель и параметры визуализации, заданные в XML-файле, могут быть изменены интерфейсными средствами визуализатора.

Третий вопрос связан с размерностью. Важно подчеркнуть, что *размерность КА еще не фиксирует размерность модели визуализации*. Последнее понятие определяется нами как размерность множества элементов данных, на основе которых формируется изображение. Если, например, ячейке КА «Поколения» поставить в соответствие цветной столбик высоты, пропорциональной цвету, то модель визуализации не станет от этого трехмерной, хотя явно будет использована 3D-графика. Если же в этом 2D-КА отображать воксели для разных номеров шагов (ось аппликата — время), то модель визуализации станет 3D. Мы сами вправе выбирать нужное нам представление, хотя, конечно, для 1D-КА естественна линейная модель визуализации, для 2D-КА — в виде клеточной доски, а для 3D — куб или сфера.

Четвертый аспект касается необходимости генерации дополнительных математических структур, порожденных потребностью анализа «сырых» данных массива КА и соответственно в первую очередь подлежащих визуализации. Если КА разыгрывается в кубе с ребром 256 клеток в течение 256 ходов, а состояние клетки оценивается байтом, то «сырых» данных 2^{32} байт = 100 Гб. Эти структуры, как правило, носят интегративный характер и аккумулируют в себе научное значение КА-расчета, поскольку результаты КА-моделирования должны быть семантически истолкованы, а это удобно сделать через упомянутые вспомогательные структуры. Для наполнения их числовым содержанием удобно после каждой итерации КА-расчета предусмотреть вызов определенной функции (что легко прописать в соответствующем разделе XML-файла МКА SoftCAM). Необходимость выполнения таких операций над глобальной конфигурацией КА обусловлена и усложняющейся структурой ячейки КА, призванной моделировать физический процесс. Очевидно, что усложнение состояния ячейки ведет к увеличению многообразия вариантов визуализации. В МКА SoftCAM принято, что состояние ячейки типизируется не битом, не целочисленным байтом, а объединением (union/record/structure в разных языках программирования) любых элементарных объектов.

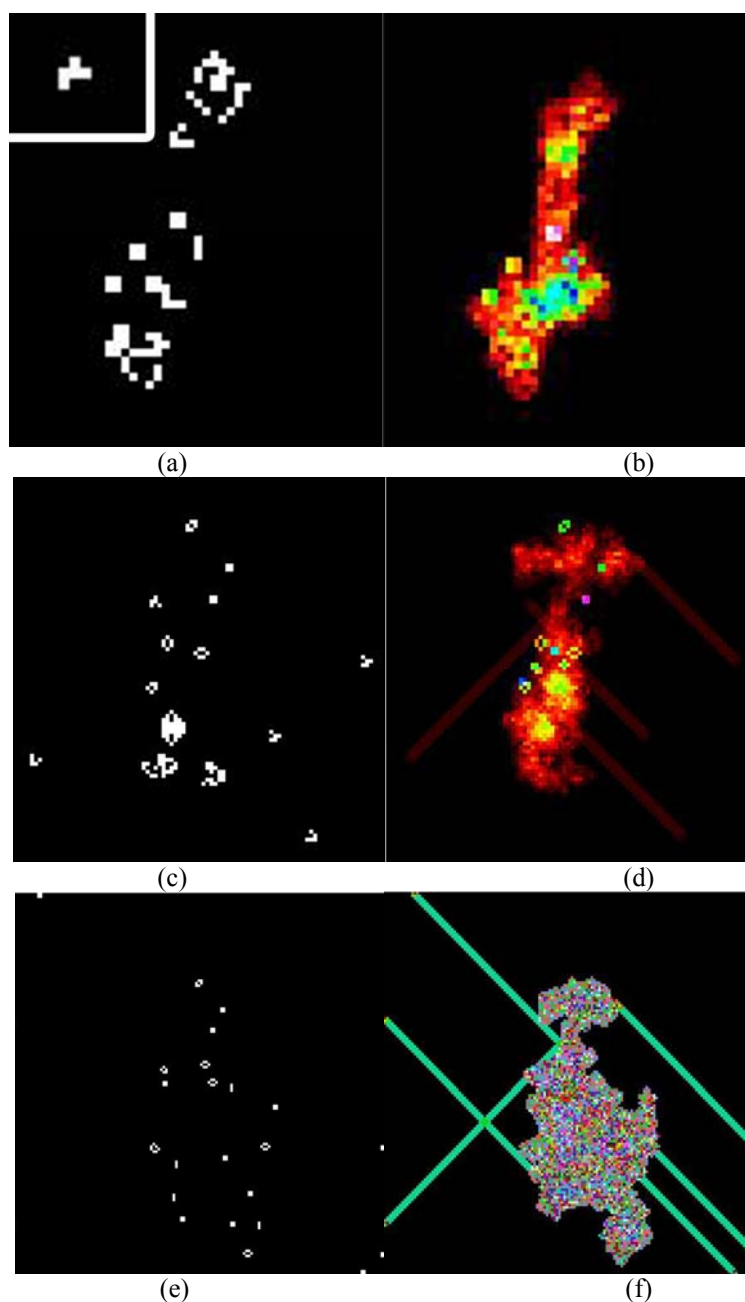


Рис. 1. Эволюция r -пентамино в игре «Жизнь». Левая колонка (a, c, e) — классическая визуализация, для 25-го хода — (a, b), для 175-го хода — (c, d) и для 1103-го хода — (e, f). Правая колонка — визуализация средним арифметическим, причем (b, d) даны для цветовой схемы А, (f) — для цветовой схемы В (см. Приложение). На рис. (a) кадированно показано исходное r -пентамино, занимающее 5 клеток. Диагональный след отвечает траектории полета «планера». Условие «линии смерти» на границах КА приводит к тому, что улетающие планеры порождают белые точки на границах рис. (e, f)

Рассмотрим поучительный пример, реализованный в начале 90-х Денисом Крыловым в симуляторе игры Конвэя [Матюшкин, 2011]. Сама по себе игра «Жизнь» уныло двухцветна, и чтобы раскрасить её анимацию, на каждом ходу запоминалось среднее арифметическое состояния каждой ячейки по времени (или сумма битов, «побывавших» в ячейке), т. е. если $x_{ij}(t)$ — состояние ячейки на t -м шаге, то цвет вокселя определен значением $y_{ij}(t) = \frac{1}{t} \sum_{k=1}^t x_{ij}(k)$.

Выбиралась индексация цветов вещественной переменной (см. Приложение) и вместо привычного черного/белое на месте ячейки отображался цветной воксел. Плавные цветовые переходы позволяли получать художественные образы. Ниже на рис.1 показана реализация такого способа визуализации в МКА SoftCAM для довольно известной фигуры r -пентамино, дающей стохастическую эволюцию на первых 1103-х ходах, позже переходящую в периодическую с периодом 2. Такой способ позволяет понять историю предшествующих ходов эволюции КА. Например, в центре рис. 1d виден зеленый блок, отсутствующий на рис. 1с; это означает, что эта фигура долгое время существовала, но к настоящему ходу уже исчезла.

Гипотетически можно рассмотреть КА-модель «Поколения» в отношении распространения некой эпидемии (цвет — стадия болезни), начавшейся в точечном центре круга. Сетка автомата, допустим, прямоугольная, а с некоторой вероятностью можно оценить возможность появления иммунитета. Эпидемиолога может интересовать в динамике количество умерших и имеющих иммунитет в зависимости от расстояния до очага инфекции. Тогда модель визуализации будет 1D, а «пропажа» измерения обусловлена интегрированием по концентрическим окружностям на каждой итерации. Таким способом получим анимированную столбиковую диаграмму.

Генерируемые на основе «сырых» КА-данных структуры могут верифицировать саму КА-модель. При клеточно-автоматном моделировании диффузионных процессов необходимо, чтобы уравнения локальных переходов удовлетворяли требованиям сохранения вещества. Особенно это важно, поскольку классические КА дискретны. В блочных КА типа автомата Марголуса данное требование выполняется автоматически, но в других случаях проблема по-прежнему актуальна. Справедливо писал В. Ванаг: «Возникает вопрос: как же тогда находить локальные правила, следуя которым множество отдельных элементов системы в сумме будет воспроизводить желаемую динамику? Этот вопрос остается без ответа». Поэтому на каждой итерации КА-расчета желательно интегрировать по всему полю, чтобы распознать «незаконное» изменение количества атомов того или иного сорта.

Общие принципы визуализации КА

В области научной визуализации уже разработана своя «периодическая таблица». Однако она достаточно абстрактна и слишком ориентирована на представление данных из гуманитарных наук; для КА требуется разработать свои, специфичные методы.

Нужно относиться к хранящейся в памяти или на жестком диске матрице КА как к неопределенной метафизической сущности, допускающей многообразие вариантов представления. Этот подход, заимствованный из геометрического моделирования, применим не только для 3D, но и для любых, даже более простых КА, или даже более сложных в отношении топографии (например для треугольных или шестигранных сеток). Опыт геометрического моделирования может быть использован при решении задач визуализации КА, хотя и ограниченно, поскольку: а) первое имеет дело непосредственно с трехмерным твердым телом; б) КА-модель доставляет исчерпывающую информацию об объекте, а геометрическая модель наоборот стремится вывести её из компактным математических формул; в) как одно твердое тело может описываться несколькими представлениями (моделями), так и один КА может быть описан несколькими моделями визуализации; г) способы взаимодействия с графическими средами ЭВМ одни и те же. Полезно заимствовать из компьютерной графики термин «воксел»; пока определим его как то, что на экране ЭВМ соответствует ячейке КА.

Одним из принципов научной визуализации, особенно важным для 3D-КА, является устранение избыточной информации. Еще в 70-х гг. предтеча научной визуализации У. Боумен писал: «В графической фигуре естественные характеристики могут быть показаны непосредственно с точки зрения их физического вида — или они могут быть формализованы с целью подчеркнуть наиболее существенные детали. Структура понимается как физическое строение объекта, которое в обычных условиях восприятию не видно. Структура выявляется, когда в графической структуре показываются те части объекта, которые обычно скрыты от глаза, или те

внутренние элементы объекта, которые можно увидеть, лишь допустив, что его внешние элементы прозрачны».

Введение вспомогательных математических структур в этой связи представляет один из вариантов применения Data Mining к данным клеточно-автоматного моделирования. Методы Data Mining для КА еще ожидают своего применения, но уже, даже исходя из сугубо потребностей математической теории КА, можно очертить предварительно круг задач. Например, в случае массивной конфигурации игры «Жизнь» (допустим, 300×300 ячеек) нас не интересует поведение каждой клетки, а сколько рождалось «планеров», устойчивых конфигураций типа «улья» или «блока», «космических кораблей» и прочих известных паттернов. Тогда при визуализации вместо летящего «планера» можно просто рисовать какой-либо стилизованный знак, метафору — в терминологии В. Авербуха. Если обратится к КА-моделям нанотехнологий, то при описании процессов кристаллизации вещества или образования пор МКА желательно реализовывать интеллектуальные функции определения числа кластеров в системе КА, момента (и условий) перколяционного порога и т.д. В конечном счете КА-модель только тогда имеет научное значение, когда она в состоянии предсказывать макроскопические параметры, а для этого каким-либо способом нужно агрегировать микроскопические и непосредственно доступные параметры КА-данных.

Дадим абстрактное определение модели визуализации. Для этого сначала нужно определить «сырые» данные Y КА-расчета. Введем B_1, B_2, \dots, B_N — элементарные множества, на базе которых строится состояние ячейки X , множество индексов матрицы КА M , которая строится на базе коротжа времен T и множества пространственных индексов M_s размерности d :

$$\begin{aligned} X &= B_1 \times B_2 \times \dots \times B_N, M_s \subset \mathbb{N}^d, T = \langle 0, 1, \dots, T \rangle, M = M_s \times T, \\ f: M &\rightarrow X \Leftrightarrow Y \subset M \times X, \\ (\forall m \in M) &(\exists y \in Y: pr_1 y = m), (\forall y_1, y_2 \in Y) ((pr_1 y_1 = pr_1 y_2) \Rightarrow (y_1 = y_2)). \end{aligned} \quad (1)$$

Здесь pr — операция взятия проекции для множества или его элемента, определенного декартовым произведением (\times). Заметим, что хотя определение (1) похоже на определение КА, тем не менее отсутствует понятие функции перехода и шаблона окрестности, которые в значительной степени придавали практический смысл величине d , т. е. размерности КА. При визуализации размерность данных КА, наследуемая из модели КА, имеет ничтожное значение; в действительности, если исходить из способа хранения данных в памяти ЭВМ как линейной последовательности указателей на блоки фиксированного размера, эту размерность можно считать равной 1.

Введем множество индексов A , принадлежащее D -мерной решетке, и разбиение исходного множества M на несколько частей:

$$M = \bigcup_{\alpha \in A} M_\alpha, \quad A \subset \mathbb{N}^D, \quad D = 1, 2, 3. \quad (2)$$

Если решетка совпадает с исходной для КА-модели, то $D = d + 1$. Данное разбиение, с одной стороны, вводит множество индексов визуализации и соответственно размерность модели визуализации D , а с другой стороны, готовит последующее агрегирование данных (если агрегирование мы не проводим, то $|A| = |M|$). Способ разбиения может быть различным: либо безусловным, заключааясь, например, в фиксации одновременности, т.е. $|A| = |M_s|$, либо условным, зависящим от состояний клеток (тогда в общем случае $|M_\alpha| \neq |M_\beta|$, $\alpha, \beta \in A$). Например, при подсчете числа нанокристаллов в КА-модели кристаллизации (или числа пор в моделях порообразования) можно принять, что ячейки принадлежат одному кластеру, если по одной из компонент состояния (B_1) в окрестности ячейки найдется хотя бы 1 или 2 ячейки с таким же значением компоненты. При этом нужно задать, очевидно, правила определения шаблона окрестно-

сти U , в общем случае отличного от используемого при расчете КА:

$$(\forall \alpha \in A)(\forall m \in M_\alpha)(\exists m' \in M_\alpha)((m' \in U(m)) \wedge (pr_1 f(m) = pr_1 f(m')) \wedge (pr_2 m = pr_2 m')). \quad (3)$$

Поэтому можно говорить об абстрактной функции разбиения $g: 2^{(M \times X)} \rightarrow \{A\} \times \{M_\alpha\}_{\alpha \in A}$, т.е. каждому «сырому» данному Y отвечает множество индексов A и семейство множеств разбиения $\{M_\alpha\}$. Для агрегирования данных введем закон, представляющий собой функцию (в смысле программирования), допускающую список аргументов переменной длины:

$$h: \{n \in \mathbb{N}, X^n\} \rightarrow Z \subset \mathbb{R}^p, \quad 0 < p \leq 5 - D. \quad (4)$$

Для ситуации рис. 1. в роли h выступило правило среднего арифметического, $d = 2$, $D = 3$ (двумерная решетка и время) и $p = 2$ (цвет и время). Если учесть возможность анимации, то отображать мы могли $(p + 1) = 3$ параметра, а именно: динамику цвета и высоты/размера 3D-воксела (в предположении, что цвет отображает только один параметр). Поэтому максимально возможно было бы отобразить 2 параметра, а с учетом когнитивной сложности 3D на 2D-экране получаем только 1 отображаемый параметр. В (4) мы формально прописали \mathbb{R} , хотя возможны случаи агрегирования в целые переменные (да для ЭВМ вещественные типы все равно дискретны) и даже в булевы.

Зададим также обобщенную цветовую карту для агрегированных характеристик, переводящее состояние группы клеток Z в значения цвета и формы воксела Q (при этом даже обычные графики типа $y(x)$ или $z(x, y)$ могут трактоваться как разновидности воксельного представления, а множество Q' соответствует схемам освещения сцены):

$$v: Z \rightarrow Q \times Q'. \quad (5)$$

Желательно, но не обязательно, наложить на функцию v требование взаимной однозначности. **Базовой моделью визуализации мы назовем семерку, состоящую из трех множеств и четырех функций:** $\langle X, M, A, f, g, h, v \rangle$. Визуализироваться могут либо агрегированные характеристики по всему семейству множеств с индексами A , либо, наоборот, часть данных внутри одного множества с фиксированным индексом $\alpha \in A$. В первом случае дополним базовую модель визуализации множеством V_1 :

$$V_1 = \left\{ \forall \alpha \in A \left(\alpha, v(h(|M_\alpha|), f(M_\alpha)) \right) \right\}. \quad (6)$$

Во втором случае дополним её множеством V_2 :

$$V_2 = \left\{ \forall m \in M_\alpha \left(m, v(h(1, f(m))) \right) \right\}_{\alpha \in A}. \quad (7)$$

Тот случай, когда на экране ЭВМ отображается сразу несколько визуализаций (например для 3D-автомата — 3 поперечных сечения и объемная анимация), мы не рассматриваем. Вероятно для этого надо задать математическую структуру на основе базовой модели и некоторого объединения множеств типа V_1 и V_2 . Аналогичным образом можно задать сценарий анимации, трактуемой как последовательность моделей визуализации.

На рис. 2. показана классификация способов (моделей) визуализации КА. Первый критерий состоит в размерности модели визуализации D , введенной по (2). Уяснить случай моделей с особенностями топологии можно, представив КА, заданный на поверхности сферы, на которой, например, проходят химические реакции с учетом влияния адсорбционного слоя молекул и подповерхностного, допустим, заряженного слоя. Каждый слой задается матрицей ячеек, имеющей свой размер. Такая модель, с одной стороны, уже не двумерна, но еще не трехмерна. Нельзя не упомянуть о параллельных (coupled) КА [Кауров, 2011], где тоже возможны особен-

ности топологии. Поскольку речь идет об индексации данных и в программировании известны, допустим, 4-мерные массивы, то возможны 4D- и большей размерности модели; если мы моделируем с помощью КА эндофуллерены, то использование представления «куб в кубе» в этом случае может быть целесообразным [Séquin et al., 2007].

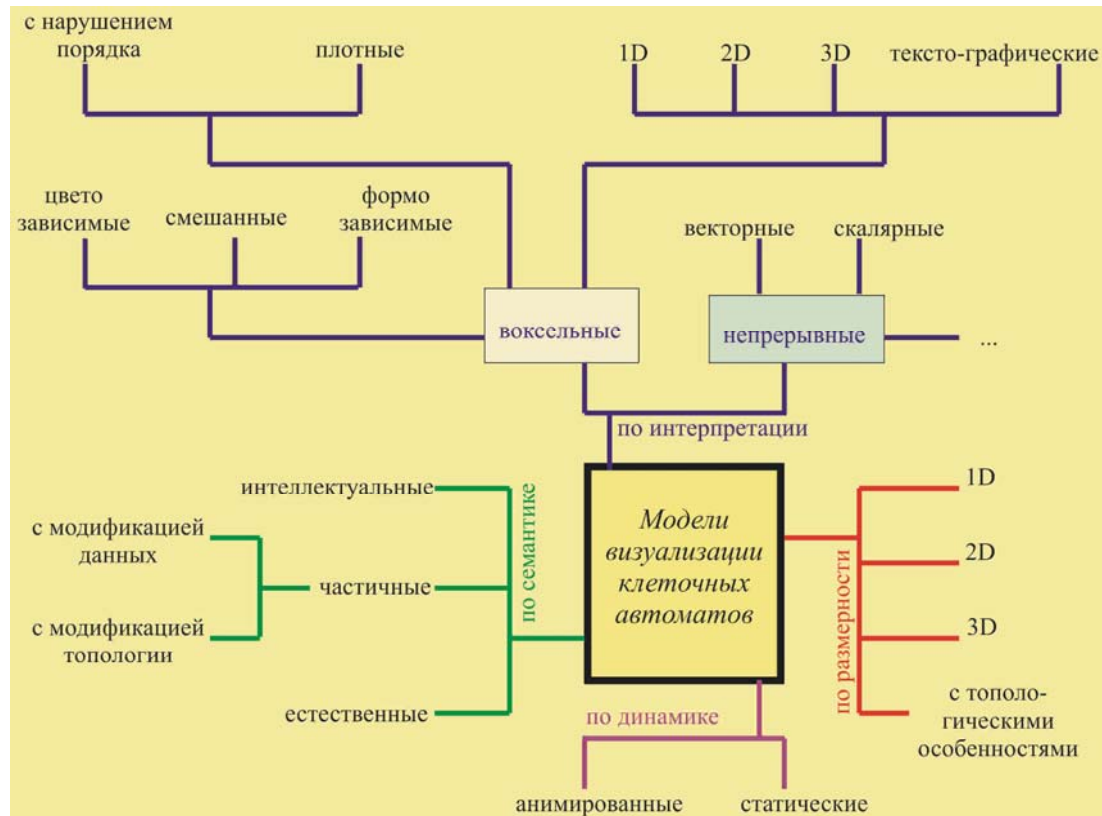


Рис. 2 Классификация способов визуализации данных клеточно-автоматных вычислений по четырем критериям

Второй критерий относительно прост, интересует ли нас статичная или динамичная картинка. Заметим только, что анимация может быть в режиме реального времени и отложенной (пошаговой), а также оставлять или полностью убирать старые данные с экрана. По третьему критерию, семантике, модели разделим на естественные, частичные и интеллектуальные. В естественных моделях реализуется аналог принципа WYSIWYG, т. е. состояние ячейки полностью отражается без какой-либо предварительной обработки («что есть в КА, то и вижу»). Возможно, более точно назвать естественные модели полными, а слово «естественный» зарезервировать для более узкого класса моделей: полных, анимированных в режиме реального времени, воксельных, без нарушения размерности и топологии, т.е. когда $D = d$. В частичных моделях отображается либо некий компонент состояния ячейки (один из B_1, \dots, B_N , а не X , см. (1)), либо некая простая функция от них $h(1, X)$ (4). Другая разновидность частичных моделей основана на том, что визуализируется какой-то срез данных по пространству, что формализуется разбиением по (2) и (7). В интеллектуальных моделях предполагается использование методов Data Mining с неизбежным агрегированием (6), а рис. 1 (правый ряд) является как раз результатом интеллектуальной модели визуализации.

По четвертому критерию модели делятся на непрерывные и воксельные. При написании абстрактной модели визуализации (1)–(7) мы подразумевали прежде всего воксельные и оговорились, что возможные графические зависимости могут быть истолкованы через воксели. Эта оговорка касалась прежде всего аспекта визуализации; при этом мы пользовались тем, например, что часть обычного графика $y = y(x)$, лежащая между двумя вертикальными линиями сет-

ки, рассматривается как криволинейный край трапециевидального воксела с номером, определяемым подробностью шкалы оси абсцисс. Однако при более внимательном анализе эта оговорка лишь затеняет суть дела. Непрерывные модели хороши, когда отображаются результаты Data Mining, лишенные КА-специфичности. Эти результаты вполне могут обработаны средствами общего назначения и не в пределах МКА [Rucker, 2003]. Следуя традиционному делению для научной визуализации, мы выделили скалярные и векторные модели, а многоточие на рис. 2 означает возможность применения к КА уже известных неспецифичных методов классификации. Кроме того, многое зависит от постановки исходной физической задачи: например, если в каждой клетке, соответствующей довольно крупному участку физической поверхности или пространства, КА-расчетом задан вектор электрического поля, то непосредственно исходным вопросом могло быть получение непрерывного распределения потенциала. То есть надо вспомнить об используемых квантах пространства и времени и от отображения индексов вокселей перейти к метрам и секундам, а для этого нужна интерполяция КА-данных, и, возможно, наиболее интересен сам вид интерполянта. Для решения же последних задач более уместно использование таких средств, как пакет MATLAB и т. п.; при этом данные, сбрасываемые в текстовый файл в процессе КА-расчета, могут многократно обрабатываться.

Поэтому основной акцент мы сделали именно на воксельные модели. Термин «воксел(ь)» пришел из компьютерной графики и будучи слиянием двух английских слов «volume» и pixel, и означает элемент объемного изображения, содержащий значение элемента раstra. Применительно к КА дадим чуть иное определение: воксел — это один или группа смежных пикселей, семантически связанных с одной клеткой КА или одним элементом разбиения КА. Соответственно, и размерность, хотя и говорим о такой условно, его разная: 1D — один пиксел, отрезок линии, стрелка (для отображения скорости потока жидкости, например, [Медведев, 2005]) или дуга; 1.5D (тексто-графические модели, характерные для старых симуляторов [Rucker et al.] и использующие техническую концепцию знакоместа) — символ или пиктограмма. Не следует недооценивать последний способ, казалось бы, уместный лишь для КА-моделей лингвистики, см. работу [Зубко и др., 1999] пермских ученых в области кинетики образования дислокаций. Более распространены 2D-воксели (квадраты, круги, гексагоны) и 3D-воксели (кубики, сферы), например, в форме блоков детского конструктора Lego или игральных костей. КА даже могут использоваться для морфинга 3D-объектов [Semwal et al., 2005].

Как правило, цвет и/или прозрачность воксела визуализируют состояние клетки (цветозависимые модели), но формой и размером воксела также можно кодировать дополнительную информацию, и здесь косвенно возникают тексто-графические мотивы. Например, рис. 1f допускает 3D-интерпретацию, если ввести высоту воксела пропорциональной его цвету (точнее среднему арифметическому). Под плотными воксельными моделями мы понимаем случай естественного заполнения экранного пространства (линейное, матричное, кубическое) вокселями без нарушения порядка следования и пустот. Если воксел прозрачный, т. е. невидим, то все равно считаем модель плотной. Однако исходная физическая топология задачи (например нанопористый материал) или просто желание исследователя могут диктовать другую расстановку вокселей; представим, например, что нужно было рассчитать механические напряжения в несущих балках стадиона «Птичье гнездо» в Китае или исследовать поведение крысы в лабиринте.

Методы визуализации одномерных КА. Генерация звукоряда

Говоря об одномерных КА, нельзя пройти мимо работ одного из творцов теории КА Стивен Вольфрама, создателя пакета Mathematica, концепции экспериментальной математики и автора бестселлера «Новый стиль науки», посвященного КА [Wolfram, 2002]. Он ввел классификацию правил перехода 1D-КА следующего типа: а) состояние ячейки — $\{0,1\}$; б) шаблон окрестности — сама ячейка, и по одному соседу слева и справа. Тогда каждой тройке битов отвечает будущее состояние ячейки 0 или 1. Если все сочетания выписать от (0,0,0) до (1,1,1), то

им будет соответствовать восьмиразрядное число, кодирующее в себе всю функцию перехода. Таким способом мы получаем нумерацию 256 правил перехода, некоторые из которых позволяют получать красивые конфигурации: правило 18 дает фрактал «ковер Серпинского». 1D-КА легко визуализируются статической 2D-моделью (рис.3, причем мы намеренно использовали 3D-воксельную модель) в пространстве «длина – время».

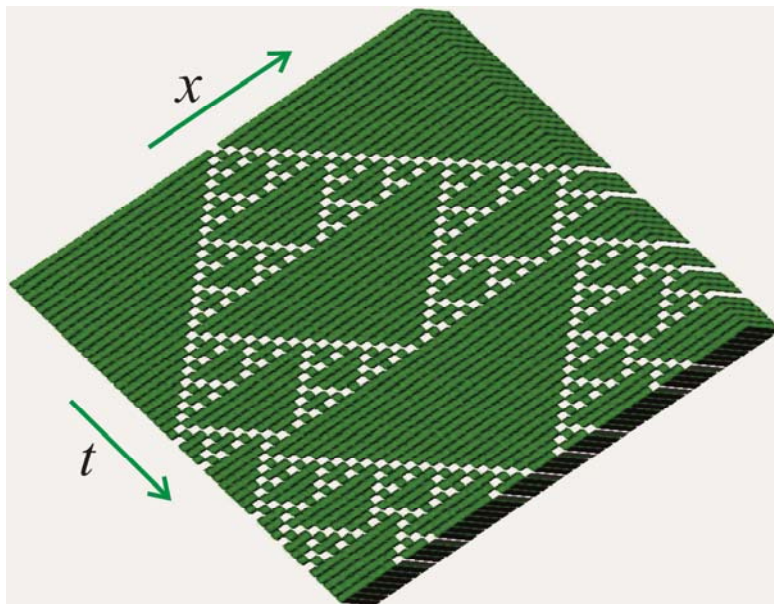


Рис. 3. Фрактал «ковер Серпинского», полученный по правилу 18 в МКА SoftCAM. Размерность КА-модели $d = 1$, размерность модели визуализации $D = 1$, размерность воксела — 3

Очень длинные ряды ячеек КА целесообразно располагать в виде матрицы, причем соответствие нумераций допускает вариации (рис. 4). Тот же способ применим и при свертке 2D-КА к 1D-модели визуализации. Его применение [Zenkin, 1990] к ряду натуральных чисел средствами системы PYTHAGORAS дало результаты в аддитивной теории чисел; идея состояла в том, чтобы выписать числа строчками по 16. Верно и обратное: небольшой по размеру поля 2D-КА можно развернуть в линию и отображать также, как 1D-КА. Однако при способах свертки желательно не нарушать исходные окрестности, как на рис. 4с.

По традиции именно 1D-КА используются для создания музыкальных сэмплов (<http://tones.wolfram.com/>), но, конечно, никаких ограничений по размерности КА нет; наряду с визуализацией можно говорить об *аудизации* КА. Мультимедийные возможности КА уже коммерциализованы. Рынок, в том числе российский, таких мини-игр представляется широким, охватывая подростков, соблазненных не только бегущими огоньками, но и неповторимыми звуками, а также начинающих композиторов, не имеющих музыкального образования.

Недавно фирма Novation выпустила USB-контроллер Launchpad, предназначенный для живой работы с программой Ableton Live, сделав шаг на пути к простому и наглядному управлению музыкальным редактором. Контроллер имеет 8×8 назначаемых клавиш и кнопки выбора режимов, которые позволяют полностью контролировать программное обеспечение. В 2009 г. Джейсон Хотчисс запрограммировал аудизацию игры Конвэя для этого устройства.

КА-аудизация выступает одной из разновидностей алгоритмической музыки, поскольку КА способны генерировать периодические фигуры (а любая музыка содержит повторяющиеся части) и вместе с тем способны к хаотическому поведению. В начале 2000-х гг. Эдуардо Мирандо (Sony Corp.) опубликовал ряд статей на эту тему и создал для КА-аудизации инструмент CAMUS [Miranda, 2001; Burraston et al. 2004] Есть и другие работы. Общие принципы аудизации, по-видимому, не отличаются от принципов визуализации, что не исключает специфики. Здесь мы только кратко рассмотрим вопрос перевода КА-данных в midi-последовательность.

Что касается mp3-формата и более развитых музыкальных форм, включая имитацию человеческого голоса, то мы выразим уверенность, что с помощью КА можно генерировать и такую музыку. Если состояние ячейки КА сложнее, чем битовое, а, например, представляется байтом (соответствует нотам от до до си), то получение midi-последовательностей заметно облегчается.

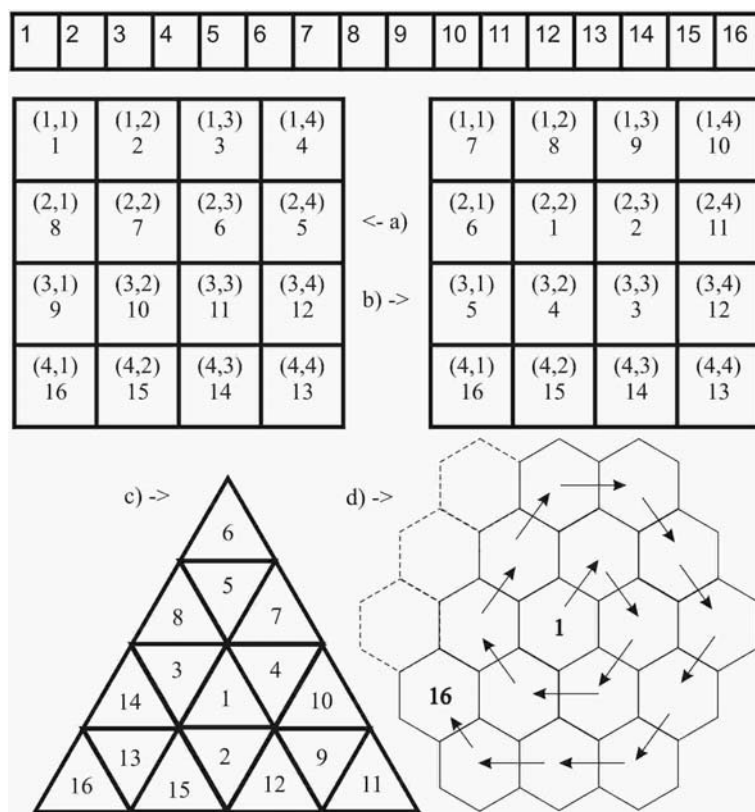


Рис. 4. Визуализация 16-клеточного 1D-КА двумерной моделью. Последовательная нумерация квадрата 4×4 (a); спиралевидная нумерация (b); нумерация треугольниками «4 в 1» (c); спиралевидная нумерация в шестигранных сотах (d)

В самом простом варианте, если допустить L инструментов, то лента булевого 1D КА разбивается на $8L$ блоков; каждый байт кодирует (рис. 5): первыми двумя битами – громкость (4 уровня), следующей тройкой битов – длительность (по отношению к такту, не менее $1/8$ периода такта), последней тройкой – значение ноты (до-ре-ми-...). Как иногда делают [Афанасьев, 2010], можно ограничиться полем 8×8 ячеек КА, подразумевая 8 инструментов в оркестре. При этом, очевидно, 2D-КА или 1D-КА, рассмотренный в 2D-модели визуализации, аудируется с алгоритмической точки зрения не имеет значения.



Рис. 5. Простая схема генерация midi-звукоряда с помощью 1D-КА. $L = 3$

Для генерации wav-файла можно использовать следующую схему: а) общее время звучания делится на одинаковые маленькие отрезки длительности T ; б) на каждом k -м отрезке волна

представляется парами (амплитуда, фаза) = (a_s, φ_s) , $1 \leq s \leq S$, а число S зависит от глубины дискретного преобразования Фурье; в) каждая пара задана двумя смежными ячейками байтового КА.

$$W(t_k \leq t < t_k + T) = \sum_{s=1}^S a_s \cos\left(\frac{2\pi}{T}s + \varphi_s\right), \quad T \sim \frac{S}{440} \text{ сек.} \quad (8)$$

Соответственно делим (рис. 6) ленту 1D-КА на множество фрагментов, если хотим получить приемлемую длительность звучания, длины $2S$.

Шаг КА ↑ период сигнала ↑ ↑ ↑	a_1 1	φ_1 1	a_2 2	φ_2 2	a_3 3	φ_3 3	a_4 4	φ_4 4	a_5 5	φ_5 5	a_6 6	φ_6 6	a_7 7	φ_7 7	a_8 8	φ_8 8
	a_1 1	φ_1 1	a_2 2	φ_2 2	a_3 3	φ_3 3	a_4 4	φ_4 4	a_5 5	φ_5 5	a_6 6	φ_6 6	a_7 7	φ_7 7	a_8 8	φ_8 8
	a_1 1	φ_1 1	a_2 2	φ_2 2	a_3 3	φ_3 3	a_4 4	φ_4 4	a_5 5	φ_5 5	a_6 6	φ_6 6	a_7 7	φ_7 7	a_8 8	φ_8 8

Рис. 6. Простая схема генерации wav-звукоряда с помощью 1D-КА. Три временных отрезка умещаются в одном такте (шаге) КА. $S = 8$, состояние ячейки — 1 байт

Методы визуализации двумерных КА. Изометрическая проекция

На сегодняшний день 2D КА, в том числе используемых в нанoeлектронике, большинство. 2D-КА визуализируются наиболее естественно: 2D-модель визуализации 2D-воксельного типа с анимацией по шагам КА. Разумеется, возможна и 3D-модель визуализации, когда временная координата переносится в третье измерение. Трудно, однако, назвать причину, по которой может понадобиться редукция к 1D (рис. 4). Некоторую сложность может составить треугольность или шестиугольность сетки, но она в основном находится в рамках КА-модели, точнее говоря, шаблона окрестности и слабо затрагивает визуализацию. Тогда возможны модели визуализации с модификацией топологии (рис. 2); на рис. 7. показаны подводные камни, сопряженные с переходами сеток.

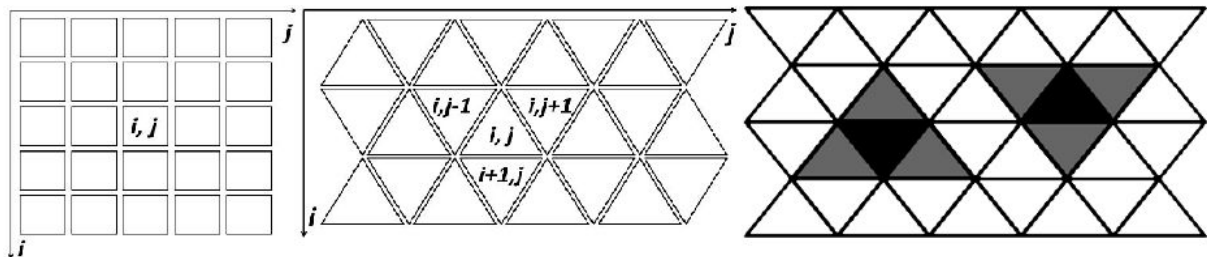


Рис. 7. Клетка КА как множество индексов на треугольной сетке [Афанасьев, 2010]. Блок в 25 клеток транслируется в блок из 24 треугольников. Окрестность Неймана занимает 4 клетки на квадратной сетке и 3 на треугольной. Для задания последней нужно отдельно рассматривать четные и нечетные ряды

Как это предлагалось в одной из работ, при слабости графической системы вокселю треугольной сетки целесообразно придавать прямоугольную форму. Наряду с описанием кристаллизации веществ на треугольной сетке авторы [Афанасьев, 2010] вводят коэффициент активности КА на ходе, т. е. процент клеток, изменивших свое состояние за несколько последних итераций. Процедуры Data Mining по аналогии могут для каждой клетки вычислять на основании последних, допустим, 10 ходов, принадлежит ли она 2-, 3- или 4-периодической конфигурации или неизменно, или меняется хаотически. Тогда, используя сходную с рис. 1. визуализацию, реализуется интеллектуальная модель.

Гексагональные сетки необходимо использовать для процессов, в которых важна изотропность, например, в задачах гидродинамики и диффузии. О. Бандман и соавторы описали [Бандман и др., 2011] аналог гидродинамической FHP-модели, имитирующей движение вольфрамсо-держающих частиц в плотно заполненном гексагонами 2D-пространстве. В каждой клетке одновременно находилось несколько частиц с своим вектором скорости, направленным по одной из 6 прямых к центрам соседних ячеек. Как пишут авторы, «здесь принят самый простой способ отображения гексагональной решетки на квадратную с растяжением оси j . Он состоит в том, что одному гексагону ставится в соответствие пара пикселей (i, j) и $(i, j + 1)$. Эти пары в четных и нечетных строках сдвинуты по отношению друг к другу. Такой способ изображения упрощает визуализацию процесса в клеточном массиве, так как обратное сжатие массива позволяет каждой клетке поставить в соответствие один пиксель на мониторе, интенсивность окраски которого соответствует исследуемой величине». И здесь, мы видим, проблемы перехода сеток решаются различием четных и нечетных индексов.

Работа [Медведев, 2005] показывает, однако, что и прямоугольные сетки не потеряли своего значения в науке. Ценой прямоугольности сетки будет, например, блочность автомата. В работе [Маркова, 2011] рассмотрены блочно-синхронные и асинхронные КА со сложно определяемой окрестностью, быть может, зависимой от состояния ячейки; КА моделируют поверхностную реакцию окисления палладия кислородом и оксидами углерода. Сложность задания блоков ставит задачу для визуализации — интерактивная отрисовка границ блоков/шаблонов окрестностей для выбранного вокселя.

Далее мы изложим способ визуализации «изометрическая проекция», обогащающий наше видение 2D-КА, и который применим и для 3D-КА. Формально он относится к частичным с модификацией топологии, 2D-воксельным. Если рассмотреть декартову систему координат $Oxyz$ в пространстве, то она делит область экрана на три части: xz -срез при $z = 0$, xz -срез при $y = 0$, yz -срез при $x = 0$. На каждую область возможно спроецировать какой-то срез данных из пространства (x, y, t) . Например, на один срез (slice) — текущую глобальную конфигурацию, на другой — конфигурацию прошлого хода, а на третий срез — битовую разность двух срезов, т. е. для ячейки среза задается 0, если состояние ячейки не менялось, и 1 — если состояние ячейки изменилось (аналог коэффициента активности). Метод изометрической проекции применим и для того случая, когда состояние ячейки сложное, допустим, описываемое тремя компонентами (B_1, B_2, B_3) , или если мы имеем дело с разделенным (partitioned) КА — тогда каждая компонента отображается на своем срезе.

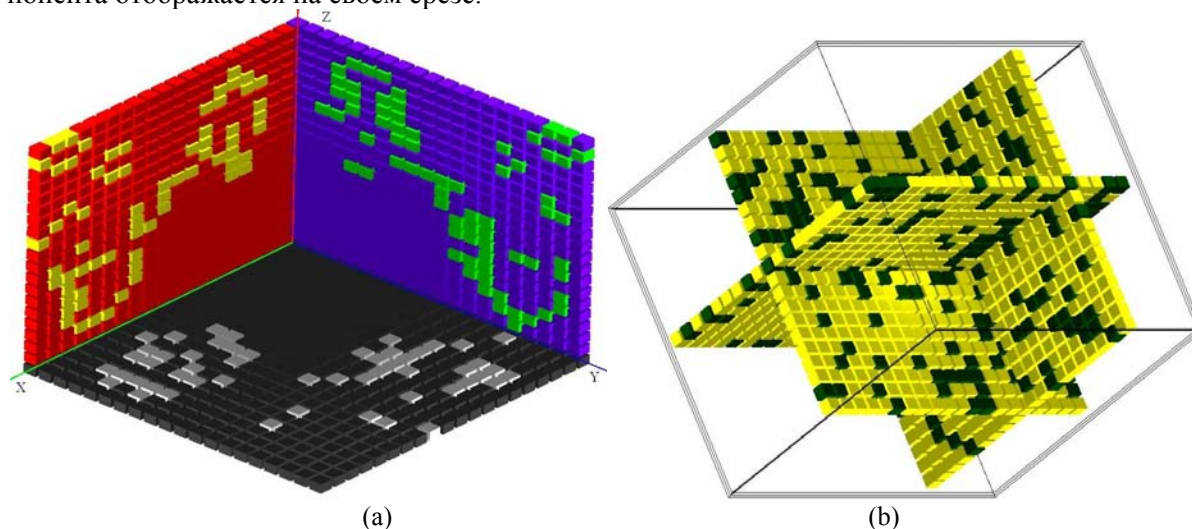


Рис. 8. Метод изометрической проекции в МКА SoftCAM: (a) случайные конфигурации игры «Жизнь» (на стене xz — состояние в момент $t - 1$, на стене zy — состояние в момент t , на полу xy — светлыми вокселями показаны клетки, изменившие состояние на текущем ходе); (b) три среза в произвольной точке (x, y, z) битового 3D-КА

Еще раз подчеркнем, двумерность КА не означает двумерность визуализации. В 2007 г. группа студентов университета Варвика в качестве дипломного проекта сделала 3D симулятор 2D-КА, и при творческом подходе можно получить эстетически значимые образы. На сайте проекта Mathematica (С. Вольфрам) выложена 3D-визуализация игры «Жизнь»: третьим измерением отмечено время; плавный градиент цвета усиливает переходы во времени; ячейки воксельные (причем есть опция отображения шаров вместо кубов), воксель прозрачен, если клетка мертвая.

Методы визуализации трехмерных КА. 3D-КА с фиксацией

Трехмерных КА-моделей мало, но именно они наиболее физически значимы, в том числе для нанoeлектроники. Иногда они получаются путем обобщения 2D-моделей, как, например, в случае КА Ооно–Кохомото [Лобанов, 2010] для описания диффузии [Weimar, 2002] или автомата для симуляции известной реакции Белоусова–Жаботинского (проект CA3D). Впрочем, в медицине достаточно давно известна 3D-КА модель сердца [Siregar et al., 1998]. Для нанотехнологий принципиальное значение имеет моделирование молекулярной сборки как на поверхности твердого тела, так в объеме вещества. Естественным языком описания таких процессов служат КА. В работе [Huw, 2011] представлена сборка наноробота из 55 тыс. ячеек 3D-КА. Лишь с развитием параллельных вычислений 3D КА будут востребованы в научной практике; сейчас же 3D-КА ресурсоемки.

Для визуализации 3D-КА существует очевидная сложность: в плотноупакованном кубе внешние слои заслоняют внутренние. Пожалуй, можно выделить четыре способа решения этой проблемы:

- удаление (невидимость) неинформативных вокселей [Матюшкин и др., 2011];
- создание разреженности, промежутков между вокселями [Graw et al., 2009];
- использование срезов (либо как изометрической проекции, либо привязанных к точке), рис. 8.;
- использование анимированных срезов [Матюшкин, 2011] и вырезов (заимствовано из методов визуализации в биологии).

Ниже мы рассмотрим 3D-КА, изначально созданный для тестирования МКА SoftCAM. Этот автомат при сравнительно простых правилах перехода обладает довольно интересным поведением, напоминающим динамику кристаллизации или травления вещества. При задании более сложных правил, возможно, удастся добиться более точного и адекватного, более физически содержательного поведения КА-модели данного класса. Похожая идея, однако, присутствующая имплицитно, высказывалась авторами [Беланков и др., 2005], которые рассматривали кристаллизацию материала с сеточной структурой (т. е. в смысле визуализации с топологическими особенностями). Объектом моделирования являлся процесс затвердевания двухкомпонентного расплава, находящегося в отливке прямоугольной формы (в начальный момент времени весь металл находится в жидком состоянии). При моделировании процесса кристаллизации каждая клетка характеризовалась состоянием (свободна, растет, затвердела), обладала масштабом и некоторым произвольным параметром выращиваемой фигуры (например, ориентация растущего дендрита или метка, позволяющая определять связные области). Клетка, только что получившая состояние «затвердела», взаимодействовала со своей окрестностью. При взаимодействии с активной (только что затвердевшей) клеткой брались в расчет только незатвердевшие клетки.

Мы будем рассматривать наш 3D-КА отвлеченно, не наделяя его физическим смыслом, и рассматривая его как представителя нового класса КА с фиксацией (locking). Поле КА кубично, размера $50 \times 50 \times 50$. Состояние ячейки задано четверкой $\langle \delta = \{0,1\}, r = \{0, \dots, 255\}, g = \{0, \dots, 255\}, b = \{0, \dots, 255\} \rangle$, занимая 1 бит и 3 байта. Шаблон окрестности определен по Мору, т. е. 26 соседей и сама ячейка. Как только $\delta = 1$, то ячейка «замораживается» и на протяже-

нии дальнейшего расчета не изменяет своего состояния. В противном случае она обладает следующей функцией перехода (i индексирует все соседние ячейки):

$$\begin{cases} r(t+1) := \lceil \max(r_i) + \max(g_i) - \max(b_i) \rceil, \\ g(t+1) := \lceil g + (b - r) \rceil, \\ b(t+1) := \lceil \min(b_i) + \min(g_i) - \min(r_i) \rceil, \end{cases} \begin{cases} r(x, y, z)(t=0) = \lceil c(1-2x)^2 \rceil, & i \leq I = 26, \\ g(x, y, z)(t=0) = \lceil c(1-2y)^2 \rceil, & c = 255, \\ b(x, y, z)(t=0) = \lceil c \sin(\pi z) \rceil, & 0 \leq x, y, z \leq 1, \end{cases} \quad (9)$$

$$\delta := 1 \Leftarrow \begin{cases} \exists i: \frac{r_i}{r} = \frac{g_i}{g} = \frac{b_i}{b}. \end{cases}$$

Величины без указания аргумента берутся в момент времени t . Операция «квадратные скобки» означает округление в сторону ближайшего целого; если аргумент больше 255 или меньше 0, то она возвращает соответственно 255 или 0. Условие локирования выполняется, если в окрестности ячейки найдется хотя бы одна ячейка с такой же пропорцией цветов. Модель визуализации 3D-воксельная цветозависимая, причем цвета выбираются по RGB-схеме (10):

$$R := r, \quad G := g, \quad B := b, \quad \alpha = (1 + \delta) / 2. \quad (10)$$

Нелокированные клетки отображаются полупрозрачно, а локированные (фиксированные) — полностью (рис. 9).

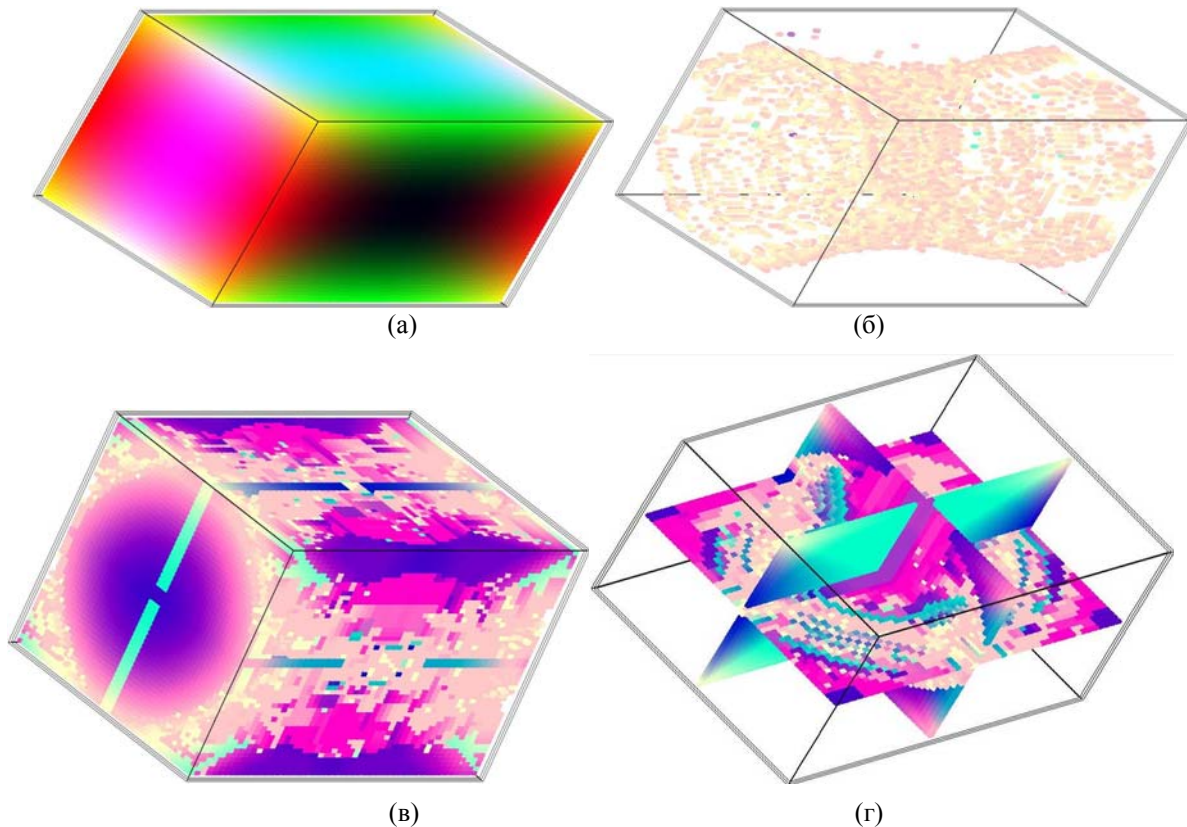
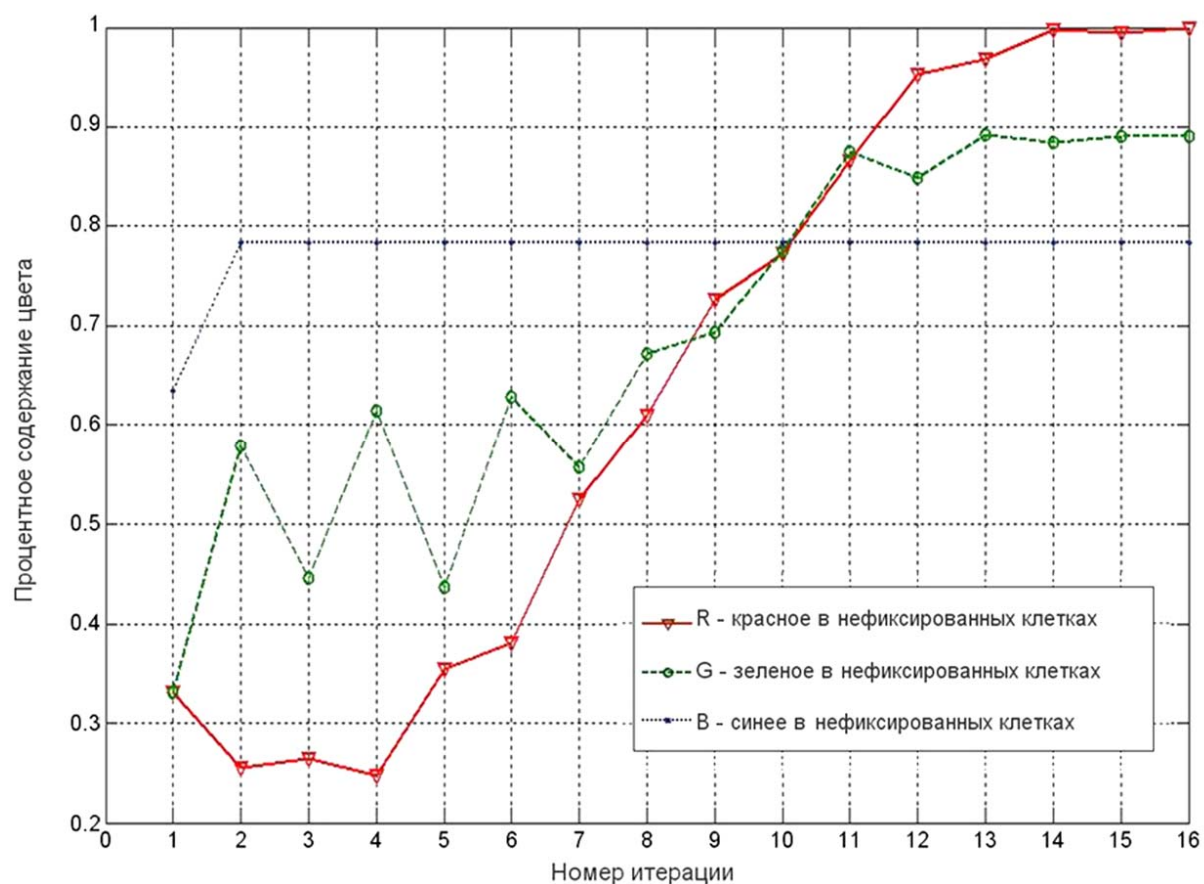
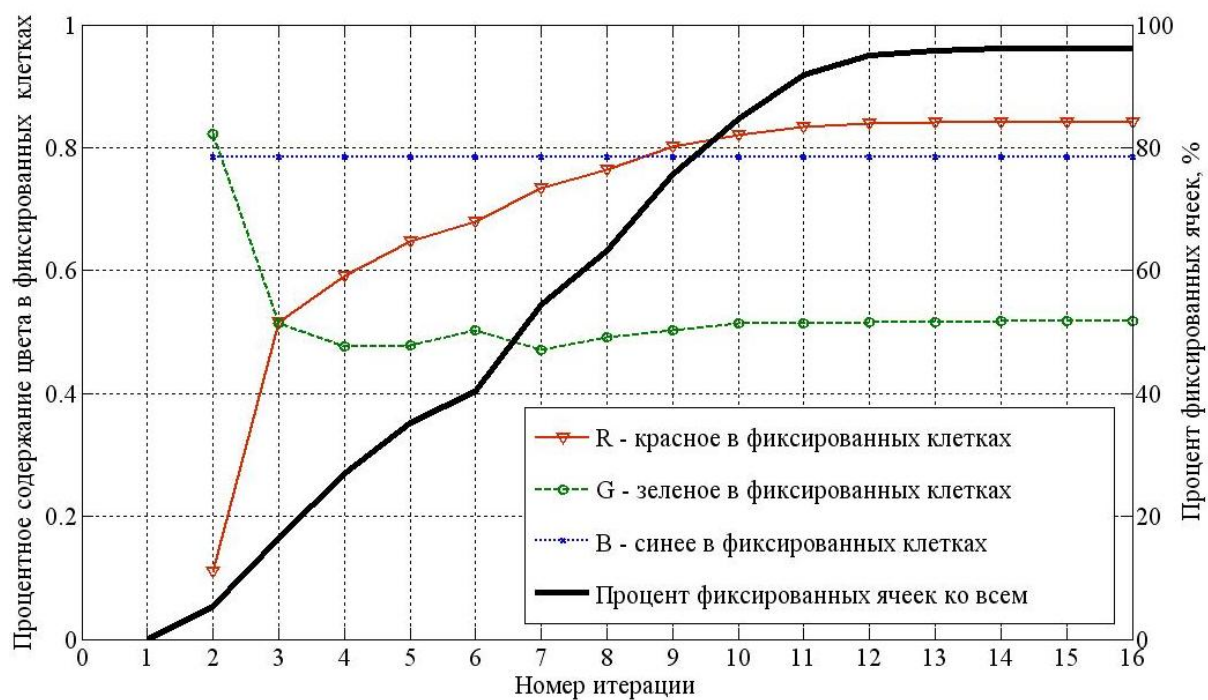


Рис. 9. Эволюция 3D-КА с фиксацией. (а) начальное состояние; (б) стационарное состояние на 25-м ходу (локированные клетки не отображаются); (в) стационарное состояние на 25-м ходу; (г) срезовой вид состояния на 25-м ходу

Будем говорить о физике состояния ячейки в терминах цвета ввиду (10). Начальное состояние, если выбрано случайным, дает двупериодическое колебание по зеленому цвету



(а)



(б)

Рис. 10. Пример интеллектуальной обработки данных 3D КА с фиксацией

при нескольких прогонах среднее число локированных клеток в асимптотике составило $\sim 83\%$. Начальное состояние $t = 0$ КА по каждому цвету определяется кривыми с одним экстремумом; в (10) величинам x, y, z соответствует индексация ребер куба, а величина c носит технический характер (см. Приложение) приведения R, G, B к байтовому виду. Приблизительно с 16-й итерации КА приходит к двупериодическому состоянию с неизменным числом локированных клеток и неизменным, около 4% , числом нелокированных. Это состояние, наиболее удачно отображенное на рис. 9б, обладает круговой симметрией относительно оси Ox , то есть по оси изменения красного цвета в начальной конфигурации, а конфигурация в асимптотике похожа на контур песочных часов или чуть стянутую трубку, что может быть интерпретировано как кристаллизация вещества, например во вращающемся пересыщенном растворе, или как особый случай травления, порождающий упорядоченную систему пор.

На рис. 10 приведены агрегированные характеристики динамики КА: как ведет себя цветовой канал в всех или в локированных клетках (усредненное значение). Мы видим, что пропорции цветов различны для локированных и нелокированных клеток. Рост локированных клеток, очевидно, монотонен и с насыщением, но скорость роста меняется неравномерно.

На примере 3D-КА с фиксацией можно увидеть специфику КА в нанoeлектронике. Большее значение имеет начальное состояние автомата (в том смысле, что если сопоставить влияние правил перехода и начальной конфигурации на эволюцию КА, то субъективно второй фактор можно оценить 50% , в то время как для обычных игровых КА первый фактор превалирует).

Заключение

Мы попытались впервые структурировать способы визуализации клеточных автоматов. Нами рассмотрено отражение задач визуализации в контексте МКА, дана общая формулировка модели визуализации КА, приведены простейшие способы аудизации КА, акцентировано внимание за применение изопериметрической проекции для визуализации 2D-КА, систематизированы методы преодоления препятствий к визуализации 3D-КА.

Специфика визуализации КА в нанoeлектронике заключается:

- в сложности задания состояния ячейки КА;
- как правило, в сложности задания шаблона окрестности КА;
- в отсутствии быстрорастущих, выходящих за рамки фиксированного поля, конфигураций;
- в необходимости интеллектуальной обработки научных данных;
- в большей значимости выбора начальной конфигурации (даже по сравнению с заданием правил перехода).

За рамками статьи остались вопросы низкоуровневого взаимодействия визуализатора и самого процесса КА-вычислений, цены хранения всего объема «сырых» данных КА-моделирования, а также релевантности тех или иных методов визуализации КА по отношению к целям конкретного научного исследования. Творческий подход к визуализации КА-моделей сделает моделирование в конечном итоге более эффективным, и мы надеемся, что данная статья окажется полезной для этого.

Приложение. Задание цветовых карт

КА может порождать математические структуры, содержащими вещественные числа. Очень часто такому вещественному числу X требуется поставить в соответствие тройку вещественных чисел (R, G, B) , каждое в пределах от 0 до 1, задающее разложение палитры на красный-зеленый-синий. Это соответствие называется цветовой картой (colormap). Иногда делают промежуточный шаг и нормируют X его максимальным значением, получая значение индекси-

рованного цвета C на отрезке $[0;1]$ в формате 4-байтового числа с плавающей точкой (float). В свою очередь RGB-цвет кодируется тремя байтами, т.е. величина, допустим, R меняется дискретно в пределах от 0 до 255.

Очень трудно подобрать аналитическую функцию $f: C \rightarrow R \times G \times B$, обладающую некоторыми желательными свойствами. На практике пользуются так называемыми поисковыми таблицами (lookup tables) для задания цветовой карты. В качестве желательных свойств нами выбраны следующие: а) наименьшему значению X отвечает черный цвет, а наибольшему — белый, т.е. $f(0) = (0,0,0)$, $f(1) = (1,1,1)$; б) чистые цвета радуги, т.е. комбинации типа $(0,0,1)$, $(1,0,1)$, должны присутствовать в таблице в естественном порядке; в) таблица должна быть взаимно однозначна. Эти требования облегчают понимание визуализированных данных, например, в том смысле, что синее всегда больше красного. Этим условиям приблизительно удовлетворяет последовательность переходов в RGB $000 \rightarrow 100 \rightarrow 110 \rightarrow 010 \rightarrow 011 \rightarrow 001 \rightarrow 101 \rightarrow 111$ по переменной C в направлении от 0 до 1. Эта последовательность, которую назовем цветовой картой А (см. рис 10), является примером кода Грея по модулю 3.

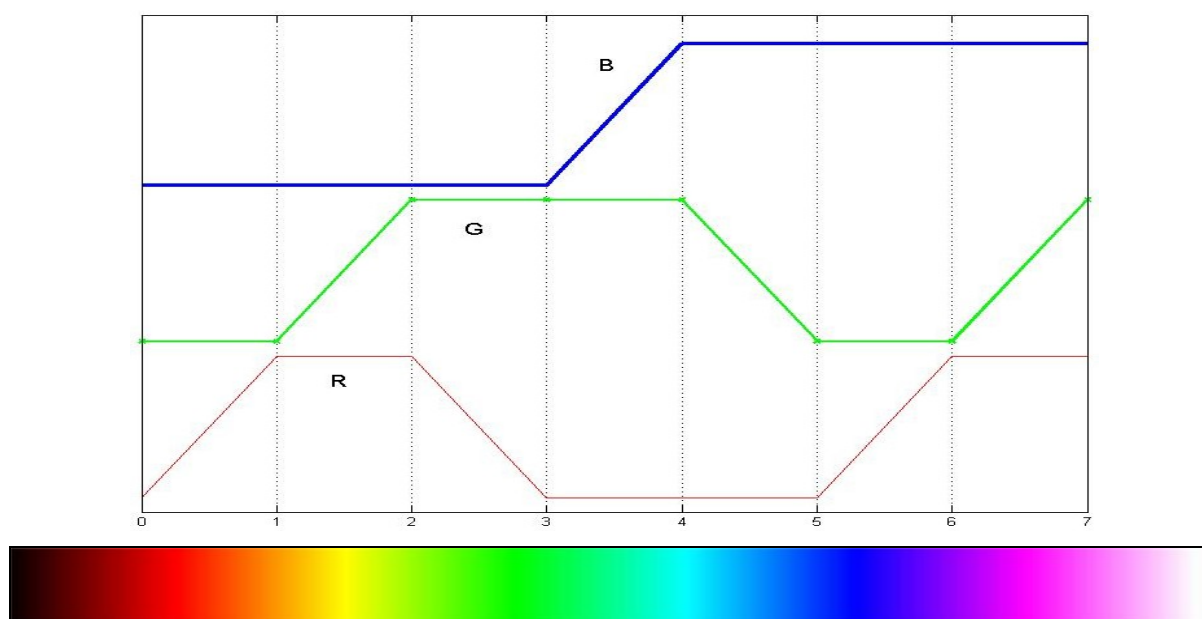


Рис. 10. Цветовая карта (схема) А: график изменения RGB-каналов по индексированному цвету и изображение последнего

Другая цветовая карта В основана на простоте вычислений: первый байт мантиисы C дает R , второй — G , третий — B , а четвертый байт может отвечать за прозрачность вокселя (альфа-канал). Однако из желательных свойств нарушено второе, т.к. отсутствует естественный порядок цветов.

Список литературы

- Авербух В. Л. К теории компьютерной визуализации // Вычислительные технологии, 2005, том 10, № 4 — С. 21–51.
- Аладьев В. З. Классические однородные структуры. Клеточные автоматы. СА: Palo Alto, Fultus Corporation, 2009, 536 с.
- Алексеев А. Е., Казёнов А. М. Реализация клеточных автоматов «игра Жизнь» с применением технологий CUDA и OpenCL // Компьютерные исследования и моделирование, 2010, т. 2, № 3, Ижевск, с. 323–326.

- Афанасьев И. В.* Исследование эволюции клеточных автоматов, моделирующих процесс «разделения фаз» на треугольной сетке // Прикладная дискретная математика, 2010, том 10, вып. 4 — С. 79–90.
- Аудизация КА «Brian's brain». <http://jmge.net/brain.htm>.
- Бандман О. Л., Кинеловский С. А.* Кумулятивный синтез: клеточно-автоматная модель физико-химических процессов на стадии схлопывания порошков облицовки // Прикладная дискретная математика, 2011, том 12, вып. 2. — С. 114–124.
- Беланков А. Б., Столбов В. Ю.* Применение клеточных автоматов для моделирования микро-структуры материала при кристаллизации // Сибирский журнал индустриальной математики, 2005, Том VIII, № 2(22), Апрель–июнь — С. 12–19.
- Бондарев А. Е., Галактионов В. А.* Научная визуализация в задачах вычислительной физики: концепции, методы, перспективы // Новые информационные технологии в автоматизированных системах: материалы 13-го научно-практического семинара. — М.: МГИЭМ, 2010, с. 134–146.
- Боумен У.* Графическое представление информации. — М.: Мир, 1971. — 225 с.
- Ванаг В. К.* Исследование пространственно распределенных динамических систем методами вероятностного клеточного автомата // УФН, 1999, т. 169, № 5. — С. 481–505.
- Демонстрации КА — <http://demonstrations.wolfram.com/GameOfLifeIn3DLayers/>;
<http://webmath.exponenta.ru/ad/aj/dem/020401.html>.
- Зубко И. Ю., Келлер И. Э., Трусов И. В.* Кинетическая модель образования периодических дислокационных структур в кристалле в терминах клеточных автоматов // Физическая мезомеханика, 1999, том 2, № 1–2. — С. 17–26.
- Калгин К. В.* Реализация алгоритмов с мелкозернистым параллелизмом на графических ускорителях // Сиб. журн. вычисл. матем., 2011, 14:1. — С. 59–70.
- Клименко С. В.* Вступительное слово // 10-я Международная конференция Графикон-2000, материалы круглого стола «Применение методов научной визуализации в прикладных задачах», Москва, 28 авг.–2 сент. 2000, с. 4.
- Лобанов А. И.* Модели клеточных автоматов // Компьютерные исследования и моделирование, 2010, т. 2, № 3. — С. 273–293.
- Маркова В. П., Шарифулина А. Е.* Параллельная реализация асинхронного клеточного автомата, моделирующего реакцию окисления СО на палладии // Прикладная дискретная математика, 2011, том 11, вып. 1. — С. 116–126.
- Матюшкин И. В.* Перспективы развития современных средств проектирования клеточных автоматов // Информационные технологии, 2011, № 4. — С. 8–12.
- Матюшкин И. В., Коробов С. В.* Применение машины клеточных автоматов SoftCAM для моделирования формирования нанокристаллов кремния в слое SiO_x // Физико-химические аспекты изучения кластеров, наноструктур и наноматериалов: межвуз. сб. науч. тр. / под общей редакцией В. М. Самсонова, Н. Ю. Сдобнякова. — Тверь: Твер. гос. ун-т, 2011, вып. 3. — с. 129–133.
- Матюшкин И. В., Хамухин А. В.* Применение языка UML при проектировании клеточных автоматов // Известия ВУЗов. Электроника, 2010, № 6. — С. 39–48.
- Медведев Ю. Г.* Метод моделирования трехмерных потоков жидкости клеточными автоматами // Автометрия, 2005, том 41, № 3. — С. 37–48.
- Матюшкин И. В., Коробов С. В.* Визуализация данных клеточно-автоматных моделей на примере задачи роста нанокристаллов Si в матрице SiO_x // Известия высших учебных заведений. Электроника, 2011, М.: МИЭТ, вып. 92. — С. 39–48.

Наумов Л. А. Обзор программного обеспечения для решения задач с использованием клеточных автоматов // Телекоммуникации и информатизация образования, 2006, № 2. — С. 114–125.

Периодическая таблица методов визуализации.

http://www.visual-literacy.org/periodic_table/periodic_table.html.

Псахье С. Г., Моисеенко Д. Д., Дмитриев А. И. и др. О возможности компьютерного конструирования материалов с высокопористой и каркасной структурой на основе метода подвижных клеточных автоматов // Письма в ЖТФ, 1998, Т. 24, №4. — С. 71–76.

Топфолл Т., Марголюс Н. Машины клеточных автоматов. — М.: Мир, 1991 — 280 с.

Automatous Monk: A collection of generative music and art projects. <http://automatous-monk.com/>

Burraston D., Edmonds E., Livingstone D., Miranda E. Cellular Automata in MIDI based Computer Music // Proc. of the 2004 International Computer Music Conference (2004).

CA3D. <http://www.ctechnet.com/ca3d/Gallary.html#6>.

CAMUS: Cellular Automata music generator — <http://tamw.atari-users.net/camus.htm>.

Cellular automata music on Novation Launchpad (видеоролик).

<http://video.ariom.ru/v/rWq4AppMm8A.html>.

Cellulate. <http://cellulate.sourceforge.net>.

Graw F., Regoes R. R. Investigating CTL Mediated Killing with a 3D Cellular Automaton // PLoS Comput. Biol., 2009, 5(8): e1000466.

Hopkins. Fun with Cellular Automata. Статья на сайте <http://www.art.net/~hopkins/Don/art/cell.html>.

Huw D. J., McWilliam R., Purvis A. Design of Self-Assembling, Self-Repairing 3D Irregular Cellular Automata // in book «Cellular Automata — Innovative Modelling for Science and Engineering», Alejandro Salcido (Ed.), Croatia, publisher: InTech, 2011 — 440 p.

Kaurov V. Wolfram Demonstrations Project: Coupled Cellular Automata: Symbiotic Patterns and Synchronization, 2011.

Miranda E. R.. Evolving Cellular Automata Music: From Sound Synthesis to Composition. // Proc. of the Workshop on Artificial Life Models for Musical Applications — ECAL 2001, September 2001.

Otomata: Cellular Automata Music Generator — <http://www.earslap.com/projects/otomata>.

Pinto N. N., Antunes A. P. Cellular automata and urban studies: a literature survey // Architecture, City and Environment, 2007, Vol.1, #3, febrero (Portugal).

Rucker R. Continuous-Valued Cellular Automata in Two Dimensions // in book «New Constructions in Cellular Automata», ed. by D. Griffeath, C. Moore, Oxford University Press, 2003 — 342 p.

Rucker R., Walker J. CelLab: User Guide: Rug and ASCII.

<http://www.fourmilab.ch/cellab/manual/chap4.html#4-TH-rug>.

Semwal S. K., Chandrashekhara K. Cellular Automata for 3D Morphing of Volume Data // The 13-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2005WSCG'2005: 195–202.

Séquin C. H., Hamlin J. F. The Regular 4-Dimensional 57-Cell // SIGGRAPH'2007.

http://www.cs.berkeley.edu/~sequin/PAPERS/2007_SIGGRAPH_57Cell.pdf.

Siregar P., Sinteff J. P., Julien N., Le Beux P. An Interactive 3D Anisotropic Cellular Automata Model of the Heart // Computers and Biomedical Research, Volume 31, issue 5 (October, 1998), p. 323–347.

Veenhuis C., Köppen M. Document Oriented Modelling of Cellular Automata, // Proc. 2nd International Conference on Hybrid Intelligent Systems (HIS02), Santiago, Chile, December 1–4, 2002.

- Weimar J. R.* Translations of Cellular Automata for Efficient Simulation // *Complex Systems*, 2003, 14, pp. 175–199.
- Weimar J. R.*, 2002. Three-dimensional Cellular Automata for Reaction-Diffusion Systems. // *Fundamenta Informaticae*, 2003, Volume 52, Numbers 1–3. pp. 277–284.
- Wolfram S.* A New Kind of Science. Wolfram Media, Champaign, IL, 2002, 1197 p.
- Zenkin A. A.* Waring's problem from the standpoint of the cognitive interactive computer graphics // *Mathematical and Computer Modelling*, 1990, Vol. 13, No. 11, pp. 9–25.