



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Филиппов С.В.

Метод идентификации
атомов макромолекул,
визуализируемых в 3D
редакторах

Рекомендуемая форма библиографической ссылки: Филиппов С.В. Метод идентификации атомов макромолекул, визуализируемых в 3D редакторах // Препринты ИПМ им. М.В.Келдыша. 2019. № 97. 10 с. doi:[10.20948/prepr-2019-97](https://doi.org/10.20948/prepr-2019-97)
URL: <http://library.keldysh.ru/preprint.asp?id=2019-97>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук**

С.В. Филиппов

**Метод идентификации атомов
макромолекул, визуализируемых в
3D-редакторах**

Москва — 2019

Филиппов С.В.

Метод идентификации атомов макромолекул, визуализируемых в 3D-редакторах

В основу метода идентификации атомов положен рендеринг изображения, «параллельного» по отношению к основному, на котором проекция каждого атома заполняется цветом, соответствующим определённому идентификационному номеру атома. Такие номера формируются в процессе построения 3D-модели молекулярной структуры, сохраняются в индивидуальном для каждого атома шейдере и записываются в текстовый файл – таблицу соответствия цветов PDB-обозначениям атомов. Для наглядной идентификации атомов написана специальная маркировочная Python-программа.

Ключевые слова: визуализация, пространственная структура, макромолекула, идентификация атомов, 3D, Blender, Python

S.V. Filippov

Method for the identification of atoms of macromolecules visualized in 3D editors

The method of identifying atoms is based on rendering an image that is “parallel” to the main one, on which the projection of each atom is filled with a color corresponding to a specific atom identification number. Such numbers are formed in the process of constructing a 3D model of the molecular structure, stored in a shader, individual for each atom, and written to a text file — a color matching table for the PDB designations of atoms. For visual identification of atoms, a special Python-marking program was written.

Key words: visualization, structure, macromolecule, atom identification, 3D, Blender, Python

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 18-07-00354 А).

Оглавление

Введение.....	3
Постановка задачи	4
Метод идентификации атомов на визуализированных в Blender изображениях макромолекулярных структур.....	4
Заключение	9
Библиографический список	9

Введение

Программы 3D моделирования, анимации и рендеринга, такие как Blender [1], Autodesk Maya [2], Maxon Cinema 4D [3] и им подобные, используются для визуализации макромолекулярных структур достаточно часто [4–6]. Благодаря тому, что вышеназванные программы могут выполнять ещё и роль полноценных сред математического моделирования [7, 8], этот класс программ используется всё чаще в качестве единственного инструмента виртуального молекулярного моделирования.

Однако, будучи изначально ориентированными на создание художественной и иллюстративной графики, все вышеназванные программы имеют определённые трудности с отображением служебной или внутренней информации на результирующих изображениях во время их визуализирующего расчёта (рендеринга). Между тем, показ такой служебной информации, как, например, имени объекта (атома) исключительно важен при решении задач математического моделирования и аналитической визуализации биомacroмолекул. А в случае «картографического» представления их структуры, где традиционное обозначение типа атома (химического элемента) с помощью цвета [9] не может быть применено, и вовсе становится критически важным.

Наиболее очевидным способом решения этих проблем является генерация специальных 3D-объектов, идентифицирующих атомы, во время автоматизированного построения молекулярной модели. Например, трёхмерных текстовых меток, располагающихся рядом с идентифицируемыми ими объектами – сферами, соответствующими атомам молекулярной модели. Такие метки будут визуализироваться системой рендеринга 3D-редактора также, как и объекты самой молекулярной структуры.

Однако данному способу присущ ряд существенных недостатков.

Во-первых, это фактическое удвоение и без того большого числа элементов трёхмерной сцены, которое радикально снизит скорость её визуализации. Как при работе исследователя с макромолекулярной моделью в реальном времени, так и во время её финального рендеринга.

Во-вторых, данный способ влечёт за собой необходимость в конструировании особых логических выражений, призванных обеспечить правильную ориентацию текстовой метки как относительно виртуальной камеры, так и относительно других объектов сцены. Это необходимо для того, чтобы метки не закрывали собой важные участки визуализируемой молекулярной структуры, но при этом были всегда хорошо видны независимо от масштаба визуализации. Помимо невозможности обеспечить абсолютно надёжное и однозначное функционирование логических выражений, контролирующих метки, их реализация также будет способствовать значительному снижению интерактивности при работе с такой молекулярной моделью.

В-третьих, объекты-метки, визуализированные вместе с молекулярной структурой, невозможно оперативно корректировать. Любая корректировка потребует как минимум нового рендеринга сцены, как максимум — значительного объема работ по выравниванию и подстройке самих меток.

Постановка задачи

С учётом вышесказанного, нами был сформулирован следующий набор требований для разработки метода идентификации атомов молекулярных моделей на визуализированных 3D-редактором изображениях:

1. Идентифицирующая атомы информация должна храниться в графических файлах вместе с генерируемым системой 3D-рендеринга изображением. При этом формат графических файлов должен быть стандартным.
2. Метод идентификации атомов не должен требовать замены штатной системы рендеринга 3D-редактора, поскольку разработка собственного модуля визуализации 3D-сцен, не уступающего штатным модулям любого из 3D-редакторов, является отдельной и весьма сложной задачей. И, кроме того, потребует значительных временных затрат на поддержку её в актуальном состоянии в ходе непрерывного развития самого 3D-редактора.
3. Расстановка текстовых меток, идентифицирующих атомы, должна выполняться уже на готовых изображениях под непосредственным контролем пользователя. Для этого может быть использован любой доступный графический редактор или же специально созданная для этого внешняя программа.
4. Метод не должен быть специфичным в отношении одного из 3D-редакторов и, соответственно, не может быть основан на модификации исходного кода самого 3D-редактора (например, Blender), поскольку такой подход потребует значительных временных затрат на поддержку модифицированной версии в актуальном состоянии из-за непрерывного развития основной ветви исходного кода 3D-редактора.

Ранее в качестве основного 3D-редактора нами уже был выбран Blender [1]. Этот выбор был обусловлен как свойствами самого программного пакета, так и нашими собственными наработками по построению динамических молекулярных моделей в его среде [6–8]. Соответственно, разработку методов идентификации атомов на визуализированных изображениях макромолекулярных структур было решено вести на этой платформе.

Метод идентификации атомов на визуализированных в Blender изображениях макромолекулярных структур

В основу предлагаемого нами метода идентификации атомов было решено положить принцип «параллельных слоёв» изображения, широко применяющийся в графических 2D-редакторах. Иными словами, метод предполагает генера-

цию визуализирующей подсистемой 3D-редактора, как основного изображения, так и вспомогательного, содержащего идентификационную информацию в закодированном виде. Следовательно, он потребует как минимум двух «проходов рендеринга»: одного для визуализации изображения макромолекулярной структуры, а второго — для вывода в графический файл идентифицирующих атомы кодов, но зато позволит использовать для записи изображений любой стандартный формат графических файлов.

Таким образом, всё вышесказанное определило набор следующих свойств и ограничений предлагаемого нами метода идентификации атомов на визуализированных изображениях макромолекулярных структур:

1. Идентифицированы могут быть лишь видимые на изображении макромолекулярной структуры атомы.
2. Минимальной единицей изображения, идентифицирующей атом, является один пиксель (при одинаковом размере слоёв изображения — основного и идентифицирующего). То есть каждому пикселю, представляющему отображение того или иного атома, соответствует пиксель, хранящий идентифицирующий код, соответствующий этому атому.
3. Для хранения идентифицирующей атом информации могут быть использованы лишь те поля структуры данных, которые согласно формату графического файла определяют свойства одного пикселя изображения. Из соображений компактности мы выбрали 8-битный RGB-формат хранения графической информации, и потому для хранения идентификационных кодов атомов нам доступны три целых числа от 0 до 255.
4. Формат сжатия идентифицирующего изображения не должен использовать алгоритмы, основанные на потере части данных или даже незначительном их искажении. Этим требованиям отвечает, например, формат PNG [10], который и был нами выбран в качестве основного.
5. Для рендеринга вспомогательного изображения, идентифицирующего атомы, не требуется расчёт освещённости и использование сглаживающих растр алгоритмов, поэтому его генерация возможна с минимальными настройками качества, а следовательно — с высокой скоростью.

Поддержка вышеописанного метода была реализована нами в Python-программе¹, строящей динамические молекулярные модели в среде 3D-редактора Blender [6–8]. В основном она состоит в генерации уникального идентификатора для каждого атома и сохранении его в индивидуальном материале каждого атома, а также в записи этого идентификатора в текстовый файл — таблицу соответствия идентификационных кодов PDB-обозначениям атомов.

Поскольку наш метод представляет идентификатор атома в виде RGB-цвета, то нами было принято решение об обеспечении максимально возможного соответствия идентифицирующего изображения схеме цветовых обозначений хими-

¹ Исходный код текущей на момент публикации версии (v.3.6) программы доступен по [ссылке](#).

ческих элементов — СРК [9]. Для этого каждая из 8-битных цветовых RGB-компонент составляется из двух частей: 3-битного значения интенсивности соответствующей цветовой компоненты и 5-битной части идентификатора (рис. 1).

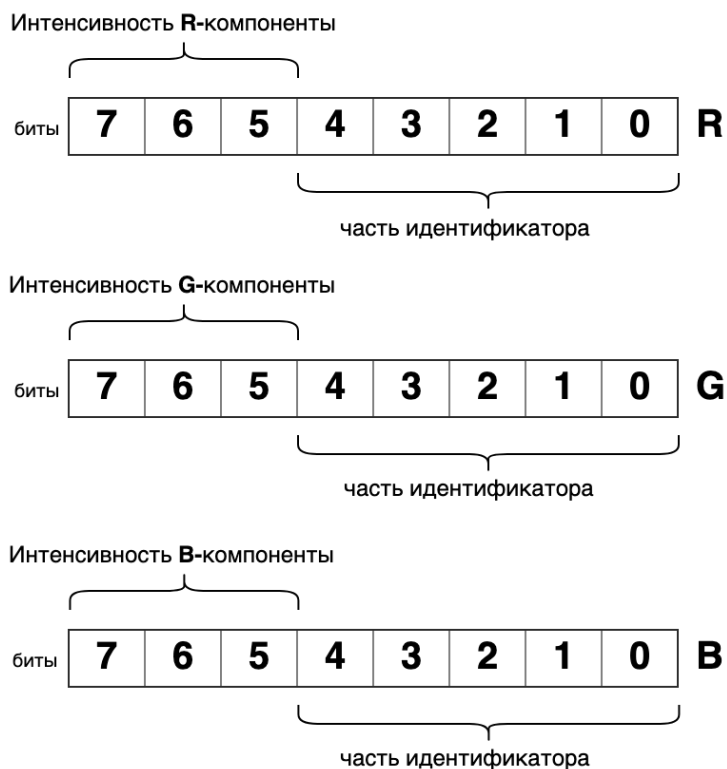


Рис. 1. Схема «упаковки» идентификационных данных атомов в RGB-поля графического файла. Размещение «цветовой» информации в старших частях байтов призвано сделать цветовой тон, назначенный атомам данного типа, доминирующим при любых значениях идентификаторов, что должно способствовать привычному визуальному восприятию идентификационного слоя изображения

Таким образом, идентификатор любого атома представляет собой 24-битное число, в котором для кодирования цветов химических элементов выделено $3 \times 3 = 9$ бит, что позволяет определить $2^9 = 512$ уникальных цветовых оттенков. Для записи же идентификатора остаётся $3 \times 5 = 15$ бит, дающих $2^{15} = 32\,768$ идентификаторов.

Согласно общепринятой логике визуализации молекулярных моделей, атомы одного типа рисуются одним цветом. В соответствии с этим первые три бита каждой из цветовых RGB-компонент одинаковы у атомов, принадлежащих одному и тому же химическому элементу, и не могут повторяться. При этом, будучи частью идентификатора, они образуют своеобразные идентификационные группы, внутри которых 15-битный идентификатор уникален. Однако у атомов, принадлежащих к разным типам (и, соответственно, группам), 15-битные идентификаторы могут совпадать.

В конечном итоге наш метод позволяет идентифицировать атомы макромолекулярных структур, у которых число атомов одного типа не превышает 32 768.

Получившийся «кодовый цвет», помимо записи в текстовый файл (таблицу соответствия идентификаторов PDB-обозначениям атомов), записывается в поле *Color* шейдера *Emission*, входящего в узловую сеть (рис. 2) назначенного каждому атому материала, определяющего цвет его поверхности при рендеринге.

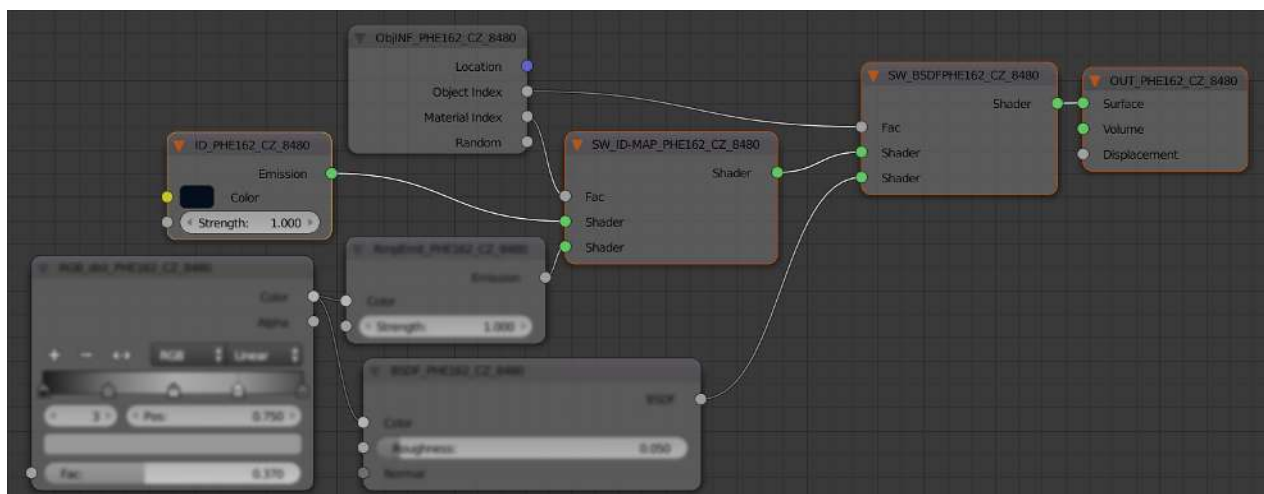


Рис. 2. Пример узловой сети материала одного из атомов. Узел *ID_* — шейдер, равномерно окрашивающий объект — сферу атома — «кодовым цветом», однозначно идентифицирующим его согласно кодовой таблице, сохранённой в текстовом файле. Назначение всех отмеченных оранжевым контуром узлов конвейера — передать «кодирующий цвет» в неизменном виде для окрашивания им поверхности сферы, обозначающей атом. Узлы *ObjINF_* и *SW_* выполняют роль переключателей, позволяющих выбирать способ окрашивания атома: «идентификационным цветом» или цветом, кодирующим расстояние от центра атома до поверхности его проецирования

Тип цветового окрашивания атомов определяется внутри узловой сети материала каждого атома молекулярной модели. Управление выбором происходит при помощи установки атрибутов *Object Index* и *Material Index*. Для автоматического управления этими атрибутами нами написан простой Python-скрипт², перебирающий в цикле все объекты трёхмерной сцены и устанавливающий выше-названные атрибуты в заданные значения у тех из них, у которых есть пользовательское свойство «atom», создаваемое, в свою очередь, при конструировании молекулярной модели нашей Python-программой [6–8]:

```
import bpy
for i in bpy.data.objects: # просматриваем все объекты сцены
    # если объект имеет пользовательский атрибут 'atom',
```

² Исходный код текущей на момент публикации версии программы доступен по [ссылке](#).


```

# то работаем с ним
if "atom" in bpy.data.objects[i.name]:
    # Устанавливаем значение Pass Index в нужное нам значение
    # Object --> Relations --> Pass Index (Object Index)
    # 0 - "ЧИСТЫЙ" ЦВЕТ (ID или DISTANCE)
    # 1 - Diffuse BSDF DISTANCE COLOR
    i.pass_index = 0

    # Material --> Settings --> Pass Index (Material Index)
    # 0 - ID
    # 1 - DISTANCE
    i.material_slots[0].material.pass_index = 0

```

Для наглядной идентификации атомов на изображениях макромолекулярных структур, визуализированных в Blender, нами была написана служебная Python-программа³ (рис. 3), использующая библиотеки PIL [11] и TkInter [12].

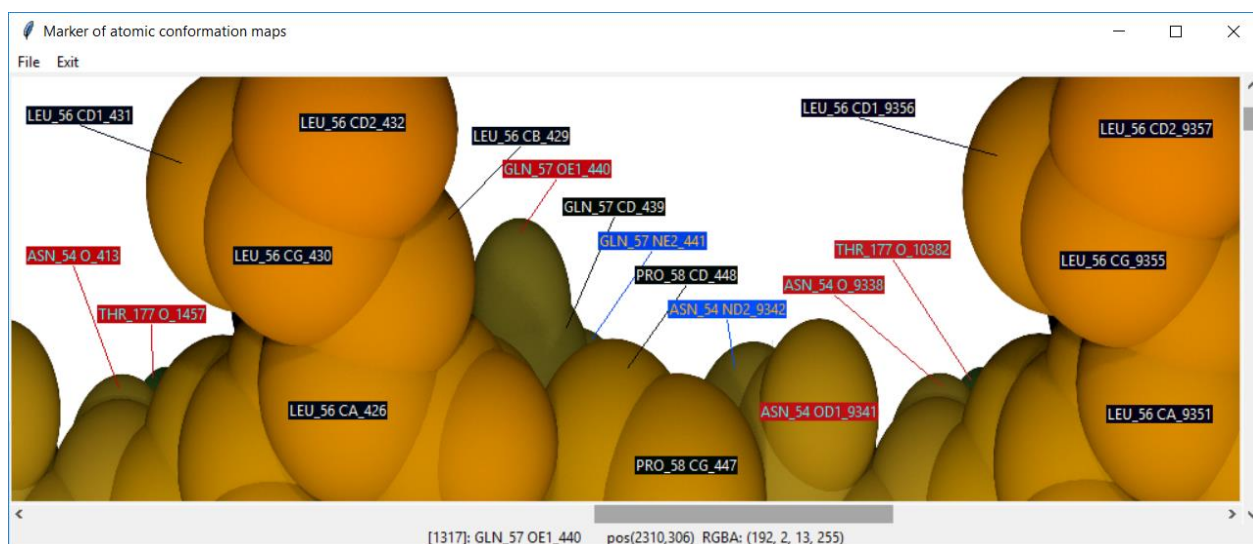


Рис. 3. Интерфейс служебной программы, предназначенной для маркировки атомов

В качестве входных данных программа принимает:

1. Текстовый файл (таблицу соответствия идентификационного кода PDB-обозначению атома), созданный исполняющейся в среде Blender Python-программой построения молекулярных моделей [6–8].
2. Графический файл, содержащий основное изображение визуализированной в Blender макромолекулярной структуры.
3. Графический файл, содержащий «параллельное» основному изображению, где изображения атомов окрашены в «кодовые цвета».

³ Исходный код текущей на момент публикации версии программы доступен по [ссылке](#).

После загрузки всех данных в окне программы отображается основное изображение молекулярной модели. Первый щелчок левой кнопкой мыши на любой области основного изображения трактуется программой как запрос на идентификацию отображаемого в этой точке атома. Далее полученный из «параллельного» изображения «кодовый цвет» ищется в загруженной из текстового файла таблице соответствия «кодовых цветов» PDB-обозначениям атомов. При нахождении совпадения, точке, указанной щелчком мыши, назначается идентифицирующая атом текстовая метка.

Второй щелчок мыши трактуется программой как указание нарисовать в этом месте текстовую метку (рис. 3). При этом в качестве фона для метки выбирается «кодовый цвет», взятый из параллельного идентификационного изображения. Поскольку его цветовой тон соответствует принятому для атомов данного типа цвету, то цветной прямоугольник текстовой метки способствует быстрой и интуитивной идентификации типа атома.

Изображение с нанесёнными на него метками может быть сохранено в виде PostScript-файла и при необходимости конвертировано в растровый формат при помощи графического редактора GIMP [13].

Заключение

Предложенный и реализованный нами метод идентификации атомов на изображениях макромолекулярных структур позволяет преодолеть ограничения 3D-редакторов по отображению внутренней служебной информации на изображениях этих структур.

После маркировки интересующих исследователя групп атомов становится возможным анализ макромолекулярных структур и осмысленная навигация по их изображениям.

В особенности наш метод полезен при визуализации макромолекулярных структур, представленных в виде различных конформационных карт, у которых цвета атомов не могут отображать их тип, потому что они использованы для представления ключевых характеристических параметров.

Библиографический список

1. Сайт программы 3D моделирования, анимации и рендеринга — Blender. [Электрон. ресурс]. URL: <https://www.blender.org> (дата обращения: 15.04.2019).
2. Сайт программы 3D моделирования, анимации и рендеринга — Autodesk Maya. [Электрон. ресурс]. URL: <https://www.autodesk.ru/products/maya/overview> (дата обращения: 17.04.2019).
3. Сайт программы 3D моделирования, анимации и рендеринга — Maxon Cinema 4D. [Электрон. ресурс]. URL: <https://www.maxon.net/en/products/cinema-4d/overview/> (дата обращения: 17.04.2019).

4. Филиппов С.В., Сивожелезов В.С., Ким В.Л., Сычев В.В., Устинин М.Н. Программа трехмерного моделирования и визуализации конформационной динамики биомакромолекул Maya-K-PDB // Математическая биология и биоинформатика, 2015, № 1, Том 10, с.260-282.
5. Сайт встраиваемой программы Molecular Maya (mMaya) для Autodesk Maya. [Электрон. ресурс]. URL: <https://clarafi.com/tools/mmaya/> (дата обращения: 17.04.2019).
6. Филиппов С.В., Сивожелезов В.С. Метод построения динамических молекулярных моделей в среде открытой 3D-платформы Blender на примере β_2 -адренорецептора // Доклады Международной конференции "Математическая биология и биоинформатика". Под ред. В.Д. Лахно. Том 7. Пущино: ИМПБ РАН, 2018. Статья № е45. doi:10.17537/icmbb18.23.
7. Филиппов С.В. Программная платформа Blender как среда моделирования объектов и процессов естественно-научных дисциплин // Препринты ИПМ им. М.В. Келдыша, 2018, № 230, 42 с., doi:10.20948/prepr-2018-230 URL: http://keldysh.ru/papers/2018/prep2018_230.pdf (дата обращения: 28.11.2018).
8. Филиппов С.В. Методы работы с динамическими молекулярными моделями, построенными в среде открытого 3D редактора Blender // Доклады Международной конференции "Математическая биология и биоинформатика". Под ред. В.Д. Лахно. Том 7. Пущино: ИМПБ РАН, 2018. Статья № е43. doi:10.17537/icmbb18.62.
9. Robert B. Corey and Linus Pauling. Molecular Models of Amino Acids, Peptides, and Proteins // Review of Scientific Instruments, Volume 24, Issue 8, pp. 621—627. doi:10.1063/1.1770803.
10. Portable Network Graphics (PNG) Specification (Second Edition). [Электрон. ресурс]. URL: <https://www.w3.org/TR/PNG/> (дата обращения: 18.04.2019).
11. Alex Clark. Pillow (PIL Fork) Documentation. Release 5.4.1. [Электрон. ресурс]. URL: <http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html> (дата обращения: 23.03.2019). doi: 10.5281/zenodo.44297.
12. John W. Shipman. Tkinter 8.5 reference: a GUI for Python. [Электрон. ресурс]. URL: <http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html> (дата обращения: 23.03.2019).
13. Сайт программы GNU Image Manipulation Program - GIMP. [Электрон. ресурс]. URL: <https://www.gimp.org> (дата обращения: 19.04.2019)