

Адаптация систем научной визуализации к сторонним решателям*

К.В. Рябинин, С.И. Чуприна

kostya.ryabinin@gmail.com | chuprinas@inbox.ru

Пермь, Россия, Пермский государственный национальный исследовательский университет

В статье описан метод унифицированного решения задачи адаптации систем научной визуализации к сторонним решателям. Метод обеспечивает автоматизированную конвертацию данных, генерируемых решателем, в пригодный для рендеринга формат, а также автоматическую адаптацию систем визуализации к особенностям программно-аппаратных платформ и высокое качество результирующих изображений. Настройка на сторонний решатель осуществляется при помощи методов онтологического инжиниринга. По онтологическому описанию синтаксиса конструкций ввода-вывода различных языков программирования автоматически генерируется синтаксический анализатор для обработки и визуализации структуры входных и выходных данных из исходного кода решателя. При помощи высокоуровневого графического интерфейса пользователь сопоставляет элементы этой структуры со свойствами визуальных объектов, которые также описаны в виде онтологии и хранятся в базе знаний системы. На основе сделанных настроек автоматически генерируется конвертер данных решателя в описание графической сцены, по которому затем строится итоговое изображение. Адаптация к особенностям различных программно-аппаратных платформ осуществляется посредством автоматической генерации графического интерфейса пользователя под каждую конкретную платформу, а также посредством автоматической предобработки подлежащих визуализации данных, которая основана на расширяемом наборе эвристических правил. Благодаря этому оказывается возможной работа системы визуализации не только на настольных компьютерах, но и на мобильных устройствах. Высокое качество рендеринга обеспечивается поддержкой оригинального адаптивного алгоритма сглаживания ступенчатых границ объектов на изображении. На основе предложенного подхода разработана мультиплатформенная система научной визуализации SciVi, которая успешно применена для решения ряда научно-практических задач из различных предметных областей.

Ключевые слова: научная визуализация, онтологический инжиниринг, мультиплатформенность, мобильные устройства, адаптивное сглаживание границы.

Scientific visualization systems adaptation to third-party solvers*

K.V. Ryabinin, S.I. Chuprina

Perm State University, Perm, Russia

In this paper the new unified method of scientific visualization systems adaptation to the third-party solvers is described. This method ensures automatic conversion of the solvers generated data into the renderable format, automatic adaptation of the visualization systems to the specifics of platforms they run on and high quality of the result images. The adaptation to the third-party solvers is based on the ontological engineering. It is proposed to create knowledge base as a part of scientific visualization systems. This knowledge base contains two ontologies: syntax ontology that describes input and output statements of programming languages and visual objects ontology that describes graphical scenes and objects with their properties. The parser for solvers' source code is created automatically based on the syntax ontology to extract input and output data structures to be visualized. The user maps the elements of these data structures to the visual objects' properties obtained from the visual objects ontology. The system then automatically generates the data converter from the solvers' data format into the graphical scene description based on the settings done. According to the scene description the final image is automatically rendered. The adaptation to the platform specifics consists of the automatic graphical user interface generation as well as the data preprocessing based on heuristics to make the data meet the physical limits of particular platform. Thanks to this it is possible to support both desktop computers and mobile devices. High quality of rendering is ensured by the new adaptive anti-aliasing algorithm proposed. Based on the proposed approach multiplatform scientific visualization system SciVi is created and used to solve real-world tasks.

Keywords: scientific visualization, ontological engineering, multiplatform portability, mobile devices, adaptive anti-aliasing.

Введение

В настоящее время практически любая научная задача сводится к измерению параметров реаль-

ных объектов или моделированию протекающих в них процессов. В обоих случаях имеют место аппаратно-программные системы, называемые решателями, которые выполняют расчёт на основе модели, лежащей в основе научного эксперимента,

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

и в ходе вычислений генерируют данные в некотором формате. Как правило, если эти данные предназначены для последующего анализа людьми, их требуется представлять в наглядном виде (визуализировать). Качественное графическое отображение научных данных позволяет исследователям лучше понять суть описываемых этими данными объектов и явлений, выявить скрытые закономерности и особенности [2]. Кроме того, в случае необходимости становится возможным дальнейшее исследование и усовершенствование лежащей в основе решателя модели, так как системы научной визуализации помогают обнаружить её недочёты, исходя из анализа результатов экспериментов.

Однако форматы выходных данных, также как и интерфейсы, у различных решателей в значительной степени отличаются друг от друга из-за высокого разнообразия объектов исследования в разных научных областях. Зачастую эти различия затрудняют использование одного визуализатора для нескольких решателей даже в рамках одной предметной области. В таких случаях приходится либо разрабатывать визуализаторы для решателей с нуля, либо создавать конвертеры данных из форматов решателей в форматы визуализаторов. Возникает острая необходимость разрабатывать системы визуализации таким образом, чтобы их можно было легко адаптировать к специфике различных решателей, поскольку это способно снизить трудозатраты на разработку промежуточного программного обеспечения и увеличить эффективность работы исследователей [3].

Часто решатели требуют для своей работы большой вычислительной мощности или специализированных устройств ввода/вывода информации, например, магнитно-резонансных томографов или секвенаторов ДНК. В этом случае они выполняются на суперкомпьютерах или на специальных аппаратных установках, и визуализаторы не могут быть их неотъемлемой частью. Чтобы в наглядном виде отображать данные, полученные от решателей такого рода, может быть использована клиент-серверная архитектура построения систем визуализации. Клиент может работать на настольном компьютере или даже на мобильном устройстве пользователя и связываться с решателем по локальной сети или сети Интернет. При этом, однако, система визуализации должна автоматически учитывать особенности инфраструктуры сети и операционных систем, чтобы обеспечить эффективную работу в различных (в частности, динамически меняющихся) условиях [3].

Многие научные задачи требуют трёхмерной, или даже многомерной визуализации [1]. Поскольку визуализация основана на растеризации, являющейся процессом дискретизации, возникает проблема ступенчатости границ объектов на изображении.

Чтобы улучшить визуальное качество и реалистичность изображения, необходимо производить сглаживание границ. Для этого существует большое количество алгоритмов, однако их реализация на мобильных платформах сталкивается с целым рядом нерешённых проблем. Например, проблемы снижения производительности визуализации и нежелательного размытия мелких деталей объектов на изображении до сих пор не имеют своего унифицированного решения.

Целью данной работы является описание принципов построения практически значимых систем научной визуализации, обеспечивающих:

1. Автоматизированную адаптацию к специфике сторонних решателей.
2. Автоматическую адаптацию к инфраструктуре программно-аппаратного обеспечения.
3. Высокое качество рендеринга.

На сегодняшний день отсутствуют какие-либо популярные и широко используемые системы научной визуализации, которые бы удовлетворяли всем вышеуказанным требованиям одновременно [3].

Предлагаемая архитектура систем научной визуализации

Предлагаемая архитектура систем научной визуализации на примере системы SciVi, разработанной авторами совместно с ИТ-компанией ООО «Ньюлана» (г. Пермь), представлена на рис. 1.

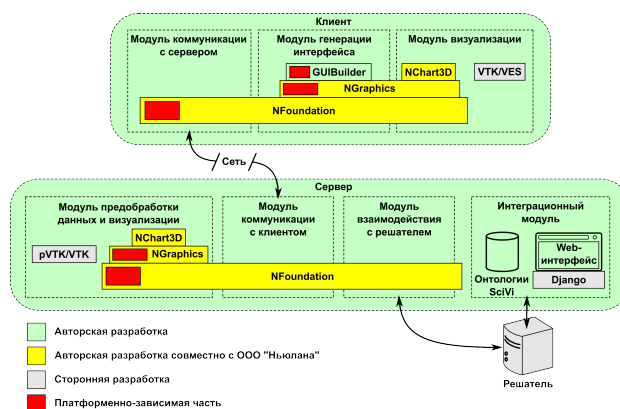


Рис. 1: Архитектура системы научной визуализации SciVi.

Для того чтобы в едином комплексе удовлетворить сформулированным требованиям, предлагается использовать клиент-серверную архитектуру систем научной визуализации и включить в состав этой архитектуры такие важные элементы, как

1. Интеграционный модуль, отвечающий за настройку на сторонние решатели, автоматически генерирующий декларативные описания шаблона графической сцены и графического интерфейса пользователя с решателем.

2. Модуль предобработки данных, отвечающий за преобразование визуализируемых данных к виду, пригодному для отображения на стороне клиента с учётом его программно-аппаратных особенностей.
3. Модуль генерации интерфейса, отвечающий за автоматическое построение графического интерфейса пользователя по его декларативному описанию.
4. Модуль визуализации, отображающий декларативно описанную графическую сцену (в качестве языка для декларативных описаний используется XML).

Онтологический инжиниринг как основа адаптивности систем научной визуализации

Для организации высокоуровневой настройки системы научной визуализации на специфику сторонних решателей предлагается использовать методы онтологического инжиниринга. Онтологический инжиниринг – это методологии обработки и хранения знаний на основе онтологий. Онтология O – это формальная модель предметной области, включающая в себя множество понятий T этой предметной области с их определениями, множество связей R между понятиями и множество аксиом A , $O = \langle T, R, A \rangle$.

В данной работе выдвинута и подтверждена на практике гипотеза о том, что для решения задачи адаптации системы научной визуализации к сторонним решателям достаточно использовать прикладные онтологии без аксиоматики: $A = \emptyset$.

Процесс адаптации осуществляется при помощи двух онтологий: онтологии конструкций ввода-вывода языков программирования L и онтологии визуальных объектов U .

Онтология L описывает синтаксис конструкций объявления переменных и конструкций ввода-вывода различных языков программирования. В настоящее время реализована поддержка языков C/C++, Fortran и Java; трудоёмкость добавления поддержки нового языка сравнима с написанием БНФ для соответствующих синтаксических конструкций этого языка. На основе онтологии L автоматически генерируется синтаксический анализатор, который посредством регулярных выражений разбирает исходный код решателя и вычленяет описание всех переменных, участвующих в операциях ввода-вывода. Пример результата работы анализатора представлен на рис. 2. Если по тем или иным причинам исходный код решателя не может быть предоставлен, пользователю предлагается описать входные и выходные данные вручную, используя для этого высокоуровневый графический интерфейс.

Онтология U описывает внешний вид и визуальные характеристики объектов (например, цвет, положение в пространстве и т.п.), которые могут быть использованы для построения графической сцены с целью визуализации результатов работы решателя. Пользователю предлагается при помощи высокоуровневого графического интерфейса выбрать визуальные объекты, подходящие для его задачи, и связать их характеристики с выходными и, при необходимости, входными, данными решателя. Этот процесс схематично представлен на рис. 3.

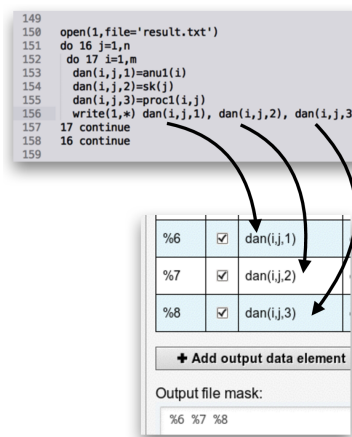


Рис. 2: Результат синтаксического разбора исходного кода решателя в системе SciVi.

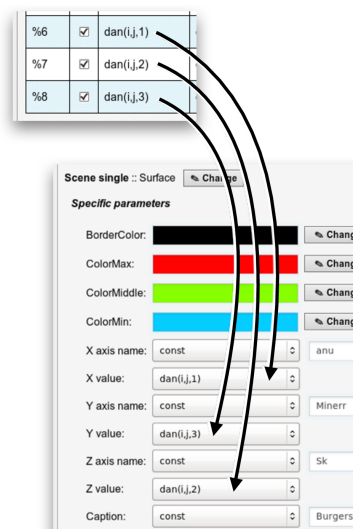


Рис. 3: Результат сопоставления данных решателя с визуальными характеристиками объектов сцены в системе SciVi.

После выполнения указанных настроек автоматически генерируется шаблон графической сцены в формате XML, содержащий описания всех выбранных пользователем объектов. Вместо значений их

характеристик в XML-файле указаны ссылки на соответствующие фрагменты маски разбора входного и выходного файлов решателя. Затем эти значения автоматически замещаются данными, извлекаемыми из указанных файлов. Примеры визуализации различных сцен в системе SciVi представлены на рис. 4.

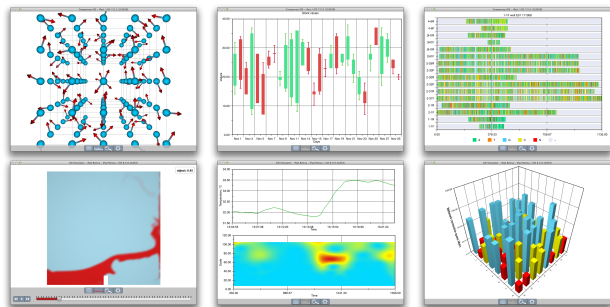


Рис. 4: Результат визуализации в системе SciVi.

На этапе настройки графической сцены пользователь также может указать, какие из входных данных решателя он хотел бы изменять в процессе визуализации. Для таких данных также автоматически генерируется описание соответствующих элементов графического интерфейса, которые встраиваются в итоговый интерфейс визуализатора.

Работа с решателем через сгенерированный интерфейс возможна только при наличии исполняемого файла решателя и возможности запустить его на стороне сервера (при наличии всех необходимых ресурсов, таких, как достаточный объём памяти, динамические библиотеки и пр.). В этом случае данные для визуализации порождаются динамически в процессе работы решателя («режим on-line»). Если исполняемый файл решателя недоступен, остаётся возможность визуализации данных непосредственно из файла, хранящего результаты работы решателя («режим off-line»).

Таким образом, работа с системой визуализации делится на 2 этапа: этап настройки на решатель пользователя и визуализируемые данные (регистрация решателя в системе, выбор параметров визуализации) и этап построения графической сцены (в on-line или off-line режиме).

Автоматический учёт особенностей различных платформ

Важным требованием к системе научной визуализации является эффективная мультиплатформенность, позволяющая автоматически настраиваться на особенности инфраструктуры программно-аппаратного обеспечения для достижения максимальной скорости визуализации и эргономики работы.

Абстрагирование от конкретной платформы достигается использованием библиотек NFoundation и NGraphics, разработанных в соавторстве с компанией ООО «Ньюлана». Библиотека NFoundation обеспечивает уровень абстрагирования от ОС. По функциональности она схожа с библиотекой Boost и включает в себя все необходимые средства для работы с динамической памятью, ресурсами, сетью, потоками, функциями обратного вызова и делегатами, контейнерами, математическими функциями и оптимизационными алгоритмами. Библиотека NGraphics обеспечивает уровень абстрагирования от низкоуровневого графического API и поддерживает работу с OpenGL, OpenGLES, DirectX9 и DirectX11. NGraphics используется для отображения сложных трёхмерных объектов, а также двумерных элементов графического интерфейса пользователя.

Графический интерфейс пользователя автоматически генерируется библиотекой GUIBuilder для каждой конкретной платформы по единому высокоуровневому декларативному описанию на языке XML. Описание интерфейса составлено из высокоуровневых понятий, таких как «кнопка», «ползунок», «текстовая метка» и т. д. Внешний вид и модель взаимодействия пользователя с каждым элементом управления для каждой конкретной ОС извлекаются из встроенной базы данных, за счёт чего нивелируются различия в парадигмах управления на настольных компьютерах и мобильных устройствах. Библиотека NGraphics переиспользуется для отображения графического интерфейса пользователя вне зависимости от программно-аппаратной платформы. Таким образом, отсутствует необходимость переключения графических API и слияния различных изображений при визуализации элементов управления поверх основной сцены, а, следовательно, не снижается производительность рендеринга.

Для осуществления специализированной визуализации используются библиотеки семейства VTK (для отображения сечений и объёмов) и библиотека NChart3D (для построения диаграмм, графиков и поверхностей).

Для поддержки устройств, обладающих малой вычислительной мощностью (таких, как смартфоны и планшетные компьютеры), сервер системы визуализации автоматически планирует распределение нагрузки и, при необходимости, на основе эвристических правил выполняет предобработку подлежащих визуализации данных. При этом учитывается тип клиента (настольный компьютер или мобильное устройство), производительность клиента (быстродействие и объём доступной памяти), скорость сетевого соединения клиента с сервером и загруженность сервера (объём доступной памяти

и количество подключённых клиентов). По результатам планирования выбирается один из режимов работы системы визуализации (от наиболее предпочтительного с точки зрения качества визуализации и скорости отклика на команды пользователя к менее предпочтительному):

1. Визуализация в полном объёме осуществляется на стороне клиента, а сервер лишь передаёт шаблон графической сцены, заполненный подлежащими визуализации данными.
2. Сервер перед отправкой выполняет частичное упрощение данных путём сокращения числа вершин в 3D-моделях, уменьшения разрешения текстур и сокращения числа ключевых кадров анимации.
3. Сервер целиком выполняет визуализацию на своей стороне и передаёт клиенту готовое изображение (как в случае использования технологии VNC).

Благодаря описанным архитектурным решениям удалось обеспечить совместимость системы научной визуализации и с настольными компьютерами, и с мобильными устройствами.

Обеспечение высокого качества визуализации

Для обеспечения высокого качества визуализации используется адаптивный алгоритм сглаживания границ объектов на изображении, являющийся суперпозицией модифицированных алгоритмов сглаживания на основе увеличенного изображения и быстрого ашпроксимированного сглаживания [4].

Суть предложенного алгоритма состоит в том, что сначала сцена визуализируется в текстуру разрешения, в s^2 раз превышающего разрешение экрана (коэффициент s вычисляется динамически на основании максимально допустимого размера текстуры). Затем текстура накладывается на полноэкранный спрайт с фильтрацией, включающей в себя билинейную интерполяцию при уменьшении изображения, а также поиск ступенек по перепаду яркости и их размытие в перпендикулярном к ним направлении. Алгоритм фильтрации реализован во фрагментном шейдере и оптимизирован для выполнения на графических процессорах мобильных устройств.

В отличие от стандартного для многих графических систем алгоритма сглаживания на основе множественной выборки, предложенный алгоритм:

1. Обеспечивает эффективную реализацию как для настольных компьютеров, так и для мобильных устройств, а также автоматически учитывает аппаратные особенности различного графического оборудования.

2. Обеспечивает быстрый отклик системы визуализации, так как сглаживание автоматически отключается на периоды динамического изменения сцены.
3. Обеспечивает визуальное качество результата не ниже, чем у системного алгоритма, при этом позволяет устранить нежелательное размытие объектов.

Заключение

В данной статье предложено использовать подход на основе онтологий для автоматизации процесса настройки систем научной визуализации на специфику сторонних решателей. Реализованная на основе предложенного подхода система научной визуализации SciVi не только легко адаптируема к сторонним решателям, написанным на различных языках программирования, но и позволяет адаптивным образом распределять процесс визуализации между клиентом и сервером. За счёт этого удаётся обеспечить одновременно высокую интерактивность и оптимальную загрузку вычислительной сети с учетом производительности клиента, скорости сетевого соединения и загруженности сервера. Высокое визуальное качество итогового изображения обеспечивается использованием нового алгоритма сглаживания границ, не увеличивающего задержку отклика системы. Мультиплатформенность обеспечивает эффективную работу SciVi как на настольных компьютерах, так и на мобильных устройствах.

SciVi протестирована под управлением операционных систем Windows, GNU/Linux, OS X, iOS и Android в процессе решения практических задач научной визуализации в различных предметных областях: физике, экономике, биологии, медицине, информатике и вычислительной математике.

Литература

- [1] Бондарев А.Е., Галактионов В.А. Анализ многомерных данных в задачах многопараметрической оптимизации с применением методов визуализации // Научная визуализация. – М.: Национальный исследовательский ядерный университет МИФИ, 2012. – К. 2, Т. 4, №2. – С. 1-13.
- [2] Васильев В.Р., Волобой А.Г., Вьюкова Н.И., Галактионов В.А. Контекстная визуализация пространственных данных // Информационные технологии и вычислительные системы. – М., 2004. – №4. – С. 25-34.
- [3] Ryabinin K., Chuprina S. Development of Multiplatform Adaptive Rendering Tools to Visualize Scientific Experiments // Procedia Computer Science. – Elsevier, 2014. – Vol. 29. – P. 1825-1834.
- [4] Рябинин К.В. Адаптивное сглаживание границ объектов на изображении для мобильных устройств // Вестник компьютерных и информационных технологий. – М.: Спектр, 2014. – №8. – С. 23-28.