

О визуализации научных данных для высокопроизводительных параллельных приложений

О.В.Джосан

В статье рассматривается проект системы визуализации результатов научных исследований. Данная система предполагает использование высокопроизводительной платформы BlueGene /P в качестве вычислительного сервера. Особенностью предлагаемых в статье методов является поддержка работы новых типов отображающих устройств, а именно стереоскопических дисплеев, 3D дисплеев и мультidisплейных комплексов. Еще одной особенностью является сокращение размера данных, передаваемых на машину пользователя, за счет эффективного алгоритма интерполяции, а также обеспечение динамического, анимационного режима представления результатов научных расчетов, проводимых на параллельных системах. Также система включает в себя новые методы обработки и улучшения визуального качества видео и изображений, а так же подсистему сжатия стерео и трехмерного видео контента, основанную на новом эффективном параллельном методе сжатия изображений.

1. Введение

В настоящее время решение задач обработки данных различной структуры и происхождения имеют большое значение в разных аспектах современной действительности. При этом активно развиваются и становятся более мощными вычислительные системы, что соответственно приводит к возможности обрабатывать данные все большего размера. Таким образом, увеличиваются размеры потоков данных, которые необходимо передавать между вычислительными узлами и локальными машинами пользователей для анализа и выделения полезной информации. Поэтому используются различные способы сокращения размерности. Одним из таких способов является визуализация данных.

При обработке научных данных их визуализация является неотъемлемой частью анализа этих данных. Существенную роль визуализация имеет при проведении крупномасштабного вычислительного эксперимента на современных высокопроизводительных системах, например, на комплексе BlueGene /P, установленном в Московском Государственном Университете [1]. Размер получаемых при таких вычислениях данных может достигать нескольких терабайт. Поэтому возникает необходимость использовать систему визуализации, которая будет осуществлять расчет визуального ряда на узлах вычислителя, т.к. передача такого объема данных на локальную машину пользователя и их визуализация уже на локальной машине весьма затруднительна. Существующие высокопроизводительные системы обладают своими внутренними особенностями коммуникаций между вычислительными узлами, особенностями организации памяти, узлами ввода-вывода, и т.д. Поэтому существующие системы визуализации при установке на высокопроизводительные вычислители требуют существенной, а иногда даже принципиальной, модификации. Одной из особенностей вычислительных систем такого типа, которая может быть эффективно использована в системах визуализации, может быть иерархичная обработка данных, т.е. когда часть операций по визуализации, которые имеют наибольшую вычислительную сложность могут выполняться на вычислительных узлах, а более простые операции могут выполняться на фронтенде.

Очевидная тенденция развития систем визуализации данных большого объема – это модификация работы систем для возможности отображения данных на новых дисплейных устройствах, таких как стереоскопические дисплеи, трехмерные дисплеи и мультidisплейные комплексы. Для этого система визуализации должна быть расширена возможностью производства видео контента в форматах, предназначенных для отображения на этих типах дисплейных устройств. В настоящее время предложено несколько таких форматов данных: анаглиф, стереоизображения, представление «глубина+текстура», объектное представление контента, мультисредовое представление данных [2]. Появление таких форматов усложняет структуру процедуры визуализации: требуются действия по вычислению дополнительной информации.

Требованием пользователей к современным системам визуализации (помимо простоты в использовании) является визуальное качество воспроизводимого контента. Представление данных в так называемом «естественном» для пользователя виде является существенной проблемой, потому как необходимо в реальном времени эффективно обрабатывать данные при наличии ограниченных ресурсов. Поэтому современные системы визуализации должны обладать эффективными методами фильтрации данных, преобразования разрешения изображений и видео, адаптации контента под устройство отображения. Эти условия также требуют внедрения дополнительной функциональности в систему визуализации.

Для функционирования системы возможны два режима – режим постобработки данных и режим работы, совмещенный со счетом. Первый режим более простой в реализации и подразумевает, что данные для визуализации получены заранее и при визуализации проводится считывание этих данных из некоторого файла. Режим совмещения визуализации и счет более сложен, т.к. требует некоторого унифицированного интерфейса в описании вызовов и дополнительных усилий по интеграции визуализации в счетную программу. Однако такой режим позволяет существенно сократить объем данных, которые необходимо сохранять для визуализации.

2. Анализ существующих систем визуализации

Проанализируем тенденции развития современных систем визуализации. Для этого рассмотрим функциональность наиболее распространенных современных систем. Спектр систем, поддерживающих визуализацию научных данных, достаточно широк. Одной из наиболее известных, свободно распространяемых систем, используемых для поддержки параллельных вычислений, является система VTK и ее параллельное расширение ParaView[3]. VTK содержит библиотеку C++ классов и несколько интерфейсов для языков высокого уровня, поддерживает широкий набор алгоритмов визуализации. ParaView включает в себя удобный интерфейс, реализацию распределенных вычислений и параллельный сервер для визуализации. Еще одна часто используемая система – это AVS/Express Parallel Edition [4]. Эта система включает в себя большой набор методов визуализации, большое количество поддерживаемых форматов данных, параллельную реализацию фильтров и сборщиков данных. Так же в последнее время получила распространение система VisIt[5]. Особенность этой системы состоит в том, что она позволяет эффективно визуализировать данные сложной и нестандартной структуры. Система ScientificVR [6] предоставляет поддержку различных режимов визуализации, в том числе в формате анаглифа. Несколько вариантов системы визуализации было предложено компанией IBM. Одна из таких систем – Deep Computing Visualization [7] – предназначена для визуализации научных данных на мульти дисплейных комплексах. Сравнение существующих систем по основным ключевым параметрам приведено в Таблице 1.

Таблица 1. Сравнение систем визуализации

Система визуализации	Открытый код	Параллельные методы	Графический интерфейс	Визуализация динамики	Работа со стерео изображениями
OpenDX	+	+	+	–	–
VTK	+	+	–	+	–
pVTK	+	+	–	–	–
ParaView	+	+	+	–	–
VisIt	+	+	+	+	–
SVR	–	–	+	+	+

Как видно из таблицы, большинство популярных систем визуализации имеют открытый код и теоретически могут быть модифицированы и оптимизированы для нужд конкретной вычислительной системы и задачи. Но на практике зачастую эта задача является сложно осуществимой и затратной по времени, поэтому для обычного пользователя это затруднительно. К тому же, для большинства систем нет подробного описания их функциональности в свободном доступе. Также большинство систем имеет ограниченную функциональность по визуализации динамики процесса, т.е. изменения состояния визуализируемого объекта во времени. Работа со стереоскопическими данными поддерживается только в одной из систем – SVR. Однако в этой

системе возможно только представление данных в формате анаглифа. И, поскольку система имеет закрытый код, то дальнейшее расширение функциональности не возможно.

После проведенного анализа существующих систем были выдвинуты следующие требования для системы визуализации, оптимальной для BlueGene /P. Первым требованием является разработка оптимальной архитектуры для такой системы. Далее необходимо расширить функциональность системы возможностью генерации видео контента для отображения на новых типах дисплейных устройств. Система должна максимально снизить размер данных, передаваемых с вычислительного узла на локальную машину, при этом обеспечить высокое визуальное качество воспроизводимого видео ряда, для этого требуются разработать эффективные методы фильтрации и сжатия видео.

В данной работе предложена архитектура такой системы визуализации. Предложены методы для реализации дополнительной функциональности, например, методы формирования стерео, 3D и мультивидового видео контента. Предложен новый метод фильтрации видео для улучшения его визуального качества, основанный на сглаживании границ, а также метод сжатия видео потока при передаче на машину пользователя, основанный на блочном сжатии видео без визуальных потерь. Еще одной особенностью предлагаемой системы является метод, дающий возможность на локальной машине пользователя в режиме реального времени увеличивать разрешение видеопоследовательности с высоким качеством восстановления деталей. Это позволяет снизить разрешение передаваемого видео потока, что позволяет многократно сократить объем передаваемой информации, а так же оптимизировать видео под разрешение конкретного устройства отображения.

Функционирование предложенной системы возможно в двух режимах – режиме постобработки и режиме совмещения визуализации со счетом.

3. Архитектура и основная функциональность

Высокопроизводительный комплекс BlueGene /P, установленный в МГУ, имеет следующие характеристики. Система включает в себя две стойки с вычислительными узлами и узлами ввода-вывода: 1024 четырехядерных вычислительных узла в каждой из стоек и 16 узлов ввода-вывода в стойке. Вычислительный узел включает в себя четырехядерный процессор, 2 ГБ общей памяти и сетевые интерфейсы. Микропроцессорное ядро - PowerPC 450 с рабочей частотой 850 МГц. Доступ к вычислительным ресурсам осуществляется через фронтенд IBM pSeries 55A.

При проектировании системы визуализации, оптимальной для такой вычислительной системы, необходимо определить, какая основная функциональность системы будет на какой части вычислителя выполняться. В качестве решения была предложена система визуализации, состоящая из трех основных подсистем: подсистемы для высокопроизводительного вычислителя, подсистемы на фронтенде, подсистемы на локальной машине пользователя. Основные блоки предложенной системы визуализации показаны на Рис.1.

Данные для визуализации хранятся на GPFS. На вычислительных узлах BlueGene /P выполняется покадровое построение видео последовательности. При этом особенность предложенной системы состоит в том, что реализована возможность генерации изображений и видео в различных форматах, предназначенных для отображения на различных типах дисплейных устройств: обычный 2D формат данных, 2D анаглифическое представление, многопоточковый формат для многоэкранных дисплеев (multiscreen), генерация стерео видео (stereo video), построение видео в формате 2D+depth, многовидовые видео последовательности (multiview). На фронтенд машине выполняются различные действия по подготовке видео последовательности к отправке на локальную машину пользователя – вычислительное устройство, непосредственно соединенное с устройством визуализации. На фронтенде выполняется параллельный алгоритм предобработки видео для улучшения визуального качества видео данных, основанный на улучшении границ. Также на фронтенде работают параллельные алгоритмы сжатия данных для различных форматов. На машине пользователя осуществляется декодирование полученного видео, а также применяется разработанный эффективный алгоритм изменения размера данных. Далее более подробно рассмотрены методы и алгоритмы для каждой подсистемы.

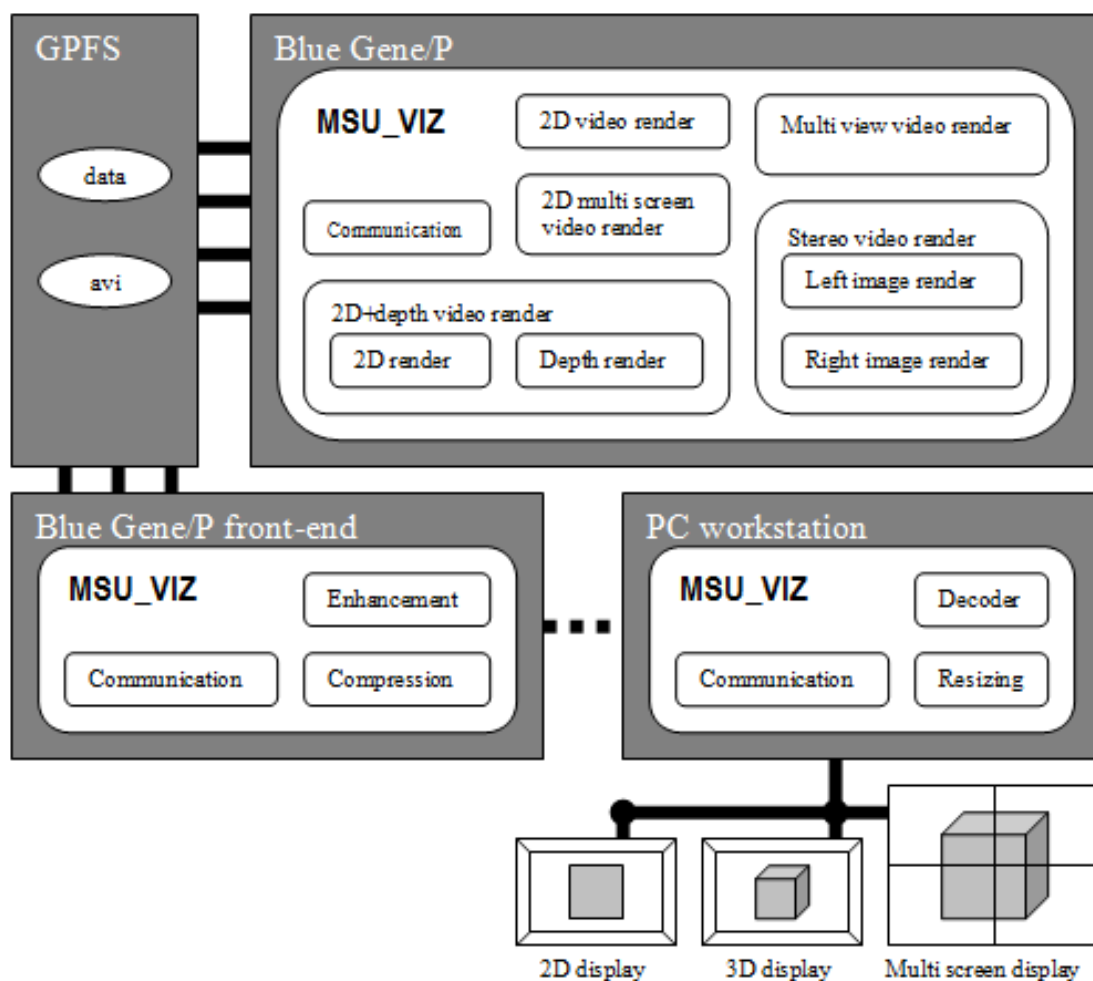


Рис. 1. Схема системы визуализации

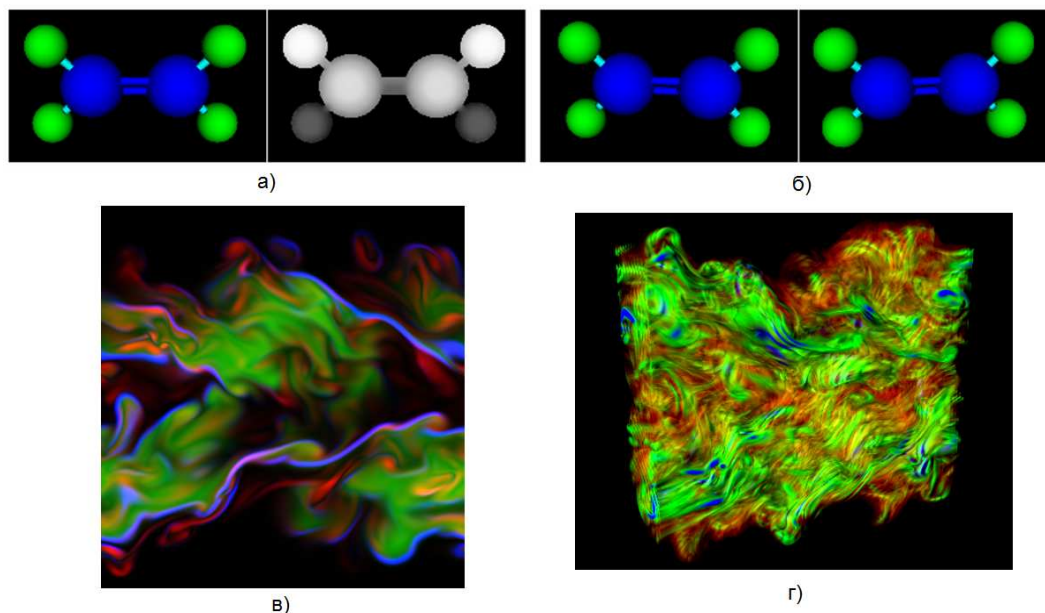


Рис. 2. Примеры работы системы визуализации на BlueGene /P: а) молекула этилена в формате 2D+depth; б) стерео изображение молекулы этилена; в) двумерная визуализации процесса турбулентного горения; г) трехмерная визуализация процесса турбулентного горения¹

¹ Различные примеры работы предложенной системы визуализации, в том числе примеры видео последовательностей в различных форматах, доступны по адресу http://angel.cs.msu.su/~oxana/MSU_VIZ/

3.1 Подсистема для высокопроизводительного вычислителя BlueGene /P

Основная задача подсистемы визуализации на вычислительных узлах – это параллельный рендеринг видео и построение изображения в форматах для различных дисплейных устройств. Построение изображений осуществляется с помощью метода объемной визуализации библиотеки VTK. На основе этого метода реализована параллельная визуализация данных, сохраненных в различных стандартных форматах. Также был реализован специальный метод генерации карты глубины. Идея предложенного метода основана на том, что для качественного воспроизведения видео с научными данными на трехмерном дисплее нет необходимости строить точную карту глубины: достаточно аппроксимировать имеющиеся данные объектами известной формы (в текущей реализации сферы и параллелепипеда) и уже для этих объектов более простой формы строить карту глубины. Также реализован специальный быстрый метод параллельного построения анаглифического изображения и методы генерации многовидовых видеопоследовательностей и мультидисплейных видео последовательностей.

На Рис.2 показаны различные примеры визуализации на системе BlueGene /P.

3.2. Подсистема для фронтэнда

На фронтэнде работают два модуля системы визуализации: модуль обработки и улучшения границ и метод кодирования видео. Первый предназначен для улучшения визуального качества видеопоследовательностей и осуществляет сглаживание ступенчатых границ, которые получаются при рендеринге. Для этого был разработан быстрый алгоритм, который может быть эффективно распараллелен. На первом шаге алгоритм находит ступенчатые границы на изображении. Для областей изображения со ступенчатыми границами предложена математическая модель распределения цветов. В соответствии с этой моделью предложены необходимые и достаточные условия, позволяющие выделить такие области на изображении на изображении.

Необходимое условие включает в себя построение вектора значений абсолютных попарных разностей соседних строк. Далее в этом векторе ищутся области, имеющие колоколообразную форму, т.е. области вокруг точек локального максимума, где выполнено условие (1), где length P и length Q – это длины левой и правой частей области соответственно.

$$0.5 < \frac{\text{length } P}{\text{length } Q} < 2 \quad (1)$$

В качестве достаточного условия используется проверка соответствия выделенных областей предложенной математической модели. При этом выделяются только ступенчатые края, а не все границы на изображении. Далее к этим областям применяется метод изменения выделенных краев, чтобы устранить эффект ступенчатости границ на изображении. Более подробно этот метод описан в работе [8]. Поскольку для работы метода в каждый момент времени необходима только локальная область данных, то нет сильной зависимости по данным, и метод может быть эффективно распараллелен.

Эффективность применения данного метода стоит в том, что он обладает высокой масштабируемостью. Распараллеливание метода проводится следующим образом. Пусть размер обрабатываемого изображения $H \times W$, в видео последовательности имеется T кадров. Тогда для обработки изображение разделяется на горизонтальные полосы размера $L \times D$, в экспериментах полагалось $L=8$. Получается, что число процессов, т.е. число полос, которое получается при разбиении, определяется по формуле(2):

$$N_{pr} = \frac{H \times W \times T}{L} \quad (2)$$

Пусть имеет N процессоров для выполнения счета. Таким образом среднее количество процессов на одна ядро процессора составляет $P=N_{pr}/N$. На системе, где проводился эксперимент на фронтэнд машине установлено 4 процессора по 4 ядра. По экспериментам получено ускорение 12.6 раз.

Блок кодирования видео предназначен для уменьшения размера передаваемого на машину пользователя потока данных. Предложенный метод кодирования основан на блочном кодировании видео последовательности. При этом используется быстрый способ оценки сложности блоков изображения, что позволяет оценить временные затраты на кодирование и осуществить разбиение кадров по процессорам в зависимости от этого времени, чтобы минимизировать общее время кодирования видео. Далее каждый кадр видео кодируется как последовательность блоков, при этом для каждого блока подбирается оптимальная палитра. Палитра определяется путем итеративного перераспределения опорных цветов в гистограмме с целью исключения из рассмотрения длинных участков гистограммы с нулевыми значениями. Процесс поиска палитры проиллюстрирован на Рис.3. После вычисления оптимальной палитры она сжимается с помощью статистического кодирования. Для различных форматов видео последовательностей разработаны различные методы их кодирования, основанные на описанном выше методе блочного кодирования. Для мультимедийного кодирования только первое изображение сжимается в цветовом пространстве, далее сжимаются покadresные разности. При передаче карты глубины используется просто кодирование блока средним значением, т.к. нет необходимости точно передавать карту глубины. Более подробно предложенный метод кодирования описан в работе[9].

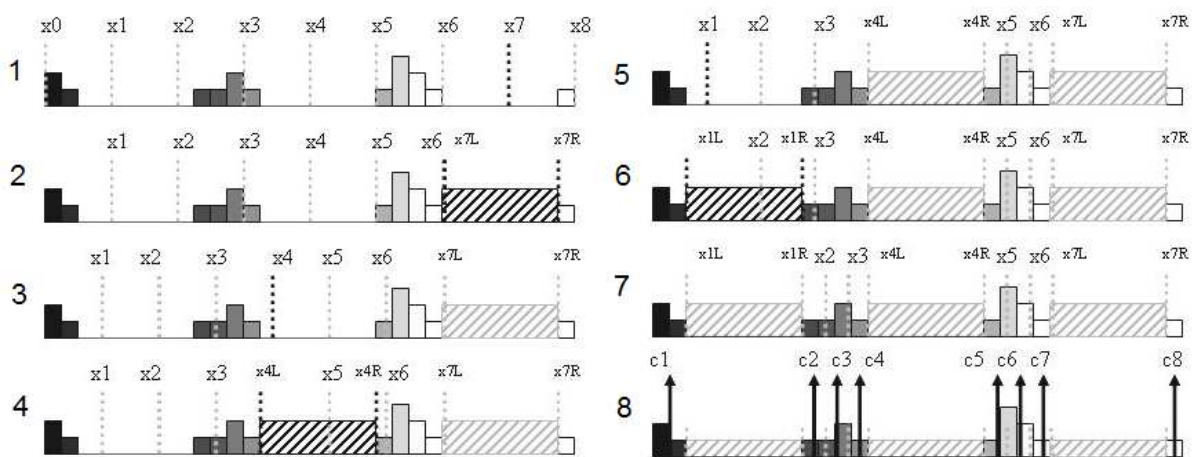


Рис. 3. Пример работы алгоритма поиска оптимальной палитры для блока изображения

3.3. Подсистема для локальной машины пользователя

На локальной машине пользователя происходит декодирование полученных данных и отображение их на дисплее. Для эффективного воспроизведения на дисплеях различного размера и снижения разрешения передаваемого видео был предложен эффективный алгоритм интерполяции, основанный на применении вейвлет-анализа и триангуляции. Алгоритм состоит из двух шагов: с помощью вейвлет-анализа определяется тип интерполируемой области (1-8), далее проводится интерполяция значений пикселей. На Рис. 4 показана схема выбора типа интерполяции в точке.

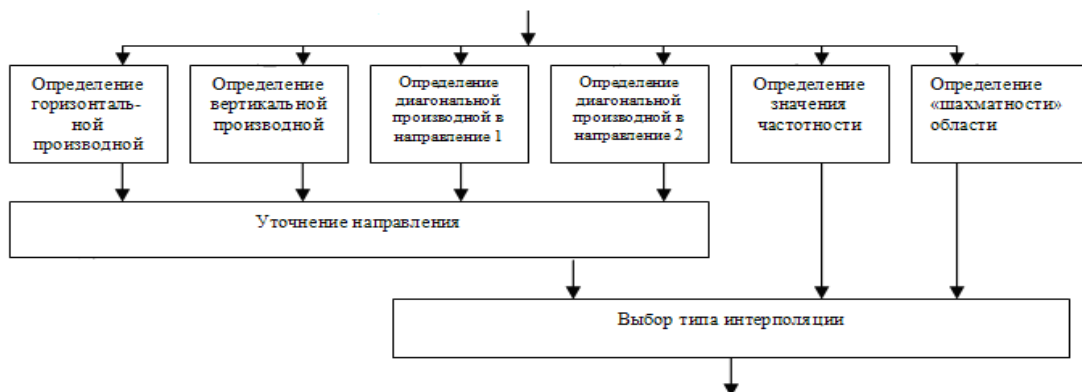


Рис. 4. Схема выбора типа интерполяции в точке

Далее проводится интерполяция значений пикселей по формулам (3), где параметры интерполяции определяются по Таблице 2 в зависимости от типа интерполируемой области [10].

$$C_{s,1} = y_{s,3} - 2y_{s,2} + y_{s,1}, \quad s = 1,2$$

$$C_{s,2} = y_{s,4} - 2y_{s,3} + y_{s,2}, \quad s = 1,2$$

$$P_s = y_{s,2} + \Delta_{s,1} (y_{s,3} - y_{s,2} - \frac{2}{30} (\Delta_{s,1} - 1) (5\Delta_{s,1} (C_{s,1} - C_{s,2}) + 5C_{s,2} - 7(C_{s,1} + C_{s,2}))), \quad s = 1,2 \quad (3)$$

$$Out = \Delta_2 (P_2 - P_1) + P_1$$

Таблица 2. Параметры интерполяции в зависимости от типа области

тип	1	2	3	4	5	6	7	8
$\Delta_{1,1}$	δ_y	δ_x	$ \delta_y - \delta_x $	$\begin{cases} \delta_x + \delta_y, \delta_x + \delta_y \leq 1 \\ \delta_x + \delta_y - 1, \delta_x + \delta_y > 1 \end{cases}$	$\frac{\delta_y - \delta_x}{2}$	$\frac{\delta_y + \delta_x}{2}$	$\frac{\delta_x - \delta_y}{2}$	$\frac{\delta_y + \delta_x}{2}$
$\Delta_{2,1}$	δ_y	δ_x	$1 + \delta_y - \delta_x $	$\begin{cases} \delta_x + \delta_y, \delta_x + \delta_y \leq 1 \\ \delta_x + \delta_y - 1, \delta_x + \delta_y > 1 \end{cases}$	$\frac{1 - \delta_x}{2} + \delta_y$	$\frac{\delta_x - 1}{2} + \delta_y$	$\frac{1 - \delta_y}{2} + \delta_x$	$\frac{\delta_y - 1}{2} + \delta_x$
Δ_2	δ_x	δ_y	$\begin{cases} \frac{\delta_x}{1 + \delta_y - \delta_x}, \delta_x > \delta_y \\ \frac{\delta_y}{1 + \delta_x - \delta_y}, \delta_x \leq \delta_y \end{cases}$	$\begin{cases} \frac{\delta_y}{\delta_y + \delta_x}, \delta_x + \delta_y \leq 1 \\ \frac{1 - \delta_x}{2 - \delta_x - \delta_y}, \delta_x + \delta_y > 1 \end{cases}$	δ_x	δ_x	δ_y	δ_y

Предложенный алгоритм увеличения разрешения видеопоследовательности относится к так называемым алгоритмам интерполяции изображений с малой сложностью, т.е. в быстром времени счета. Сложность метода сравнима со сложностью бикубической интерполяции. Поэтому метод эффективно применим для решения задачи изменения разрешения видео на локальной машине пользователя. Помимо низкой вычислительной сложности особенность предложенного метода состоит в том, что он обеспечивает высокое визуальное качество за счет сохранения границ. Так же алгоритм адаптирован для работы с видеопоследовательностями, содержащими результат визуализации данных, полученных в процессе крупномасштабного вычислительного эксперимента в различных научных областях.

4. Заключение

В настоящее время архитектура универсальных систем визуализации построена таким образом, что система может быть установлена на различных платформах. При этом в большинстве случаев используется клиент-серверная модель взаимодействия. Во многих случаях использование такой модели достаточно эффективно, т.к. для вычислений используется комплекс с небольшим числом процессоров. Однако с ростом количества вычислительных узлов и усложнением архитектуры вычислительной системы возможна организация более эффективного взаимодействия за счет введения в системы визуализации промежуточных звеньев. Такие промежуточные звенья в большинстве случаев позволяют более эффективно организовывать вычисления при визуализации, оптимизировав при этом нагрузку на основной многопроцессорный вычислитель, используя, например, для некоторых этапов визуализации фронтенд-машину. Еще одним преимуществом использования промежуточных звеньев в системе визуализации является тот фактор, что в настоящее время происходит коммерциализация многопроцессорных вычислительных комплексов, т.е. вводятся различные схемы оплаты времени счета задачи на вычислителе. При этом быстрый счет на многопроцессорном вычислителе стоит дороже, чем использование вспомогательных ресурсов. При этом при визуализации посчитанных данных можно выделить специальные блоки, которые не требуют больших вычислительных затрат и при использовании быстрых методов обработки данных могут быть выполнены не на основных счетных узлах, что позволит сократить стоимость вычислений при визуализации. При этом общее время доставки визуализированных данных на машину пользователя увеличивается незначительно, а может даже сокращаться за счет использования эффективных алгоритмов сжатия данных. Таким образом, с учетом перечисленных факторов была разработана специализированная системы визуализации данных с промежуточными звеньями. Такая архи-

тектура системы визуальной поддержки высокопроизводительных вычислений позволит увеличить эффективность и снизить стоимость вычислений при визуализации данных. В работе предложена архитектура системы поддержки визуализации научных данных для высокопроизводительного комплекса BlueGene /P. Система позволяет визуализировать данные в различных видео форматах для отображения на различных типах дисплейных устройств. Также разработаны параллельные методы сжатия и фильтрации видео. Система предназначена для визуализации данных большого объема. В настоящее время проводится апробация и анализ эффективности системы на реальных данных. В качестве примеров рассмотрены результаты решения следующих задач: визуализация турбулентного горения и визуализация построения собственных характеристик и собственных лиц в задаче биометрического распознавания лица *.

Литература

1. IBM Journal of Research and Development, Blue Gene, Vol. 49, No. 2/3
2. Tao Ni et. al. «A Survey of Large High-Resolution Display Technologies, Techniques, and Applications» // Proceedings of the IEEE Virtual Reality Conference, 2006
3. Lisa Sobierajski Avila, Rick Avila, C.Charles Law «Visualization Toolkit», pub. Kitware, Inc. 4th edition
4. Система AVS/Express Parallel Edition http://www.avx.com/software/soft_t/avsxps.html
5. Система VizIt <http://graphics.stanford.edu/~mhouston/VisWorkshop04/VisIt.pdf>
6. Система Scientific VR http://www.cfd.ru/r_index.htm?2
7. IBM DCV <http://www-03.ibm.com/systems/deepcomputing/visualization/>
8. О.В. Джосан, А.Б. Мурынин «Метод улучшения границ на изображении» // журнал Динамика нелинейных систем, том 29(1), № 11, 2007
9. О.В. Джосан, М.Н. Мишуровский, «Анализ методов малой сложности для кодирования изображений без визуальных потерь» // Тезисы конференции «Телевидение: передача и обработка изображений», Россия, Санкт-Петербург, 2008
10. О.В. Джосан, «Способ и система увеличения разрешения видео последовательности, основанные на вейвлетах и объединении декодирования и интерполяции» // Заявка на патент RU 2007132982, 2007

* Данная работа выполняется при поддержке гранта РФФИ № 08-07-00445-а.