

# Рациональный метод описания движения в сети Интернет

В.В. Тютин

Нижегородский Государственный Университет им. Лобачевского,  
Нижний Новгород, Россия

## Аннотация

В этой статье рассматриваются основные подходы к проектированию движений анимированного персонажа. Также рассматриваются особенности использования анимации персонажа в сети Интернет.

Предлагается метод позволяющий сократить размер хранимой и передаваемой по сети информации о движении. Описывается подход к проектированию анимации персонажа в сети Интернет и в мобильных телефонах.

*Ключевые слова:* Анимация, 3D, персонаж, интернет, движение.

## 1. ВВЕДЕНИЕ

В наши дни анимация персонажа занимает одну из ключевых позиций в программной индустрии. Область ее применения довольно широка: начиная с мультипликации и кинематографии, и заканчивая сложными авиасимуляторами и симуляторами космических станций.

Задача создания анимации персонажа наиболее адекватно отображающей реальную жизнь довольно сложна, поэтому решение этой задачи не под силу отдельной группе разработчиков. Проектированием систем анимации занимаются крупные корпорации, такие как Discreet [1], Alias [2], ParallelGraphics [3] и т.д.

Но такие системы, как правило, ориентированы на то, чтобы поддержать наиболее широкий круг возможностей системы, не заботясь об оптимизации отдельных методов.

Стоит отметить, что анимация персонажа не ограничивается рамками только одного программного устройства. Сегодня область ее применения простирается шире, благодаря сетевым технологиям. Но Интернет сайты и сетевые игры это не предел для анимации. Трехмерная графика уже распространилось на мобильные телефоны, а это значит, что в скором будущем там появиться и анимация персонажа.

В этой статье рассматриваются методы анимации человекоподобного персонажа, где модель персонажа представлена кинематической схемой. Приводятся основные методы проектирования движений персонажа, и показывается их неэффективность при использовании анимации в сетевых приложениях. Предлагается новый метод представления движений анимированного персонажа.

## 2. ПРЕДСТАВЛЕНИЕ ПРОЦЕССА ДВИЖЕНИЯ

Наверное, наиболее важной задачей в анимации персонажа является задача представления движения. От того, как будет представлен процесс движения, зависит то, как будет храниться и использоваться информация о движении. Двумя противоречивыми требованиями, которые могут быть

применены к способу представления движения, являются простота и компактность. К сожалению, в действительности: чем проще способ представления, тем больше информации требуется для хранения. И наоборот: чем компактнее представлена информация, тем больше вычислительных усилий требуется для ее использования.

В качестве модели персонажа используется кинематическая цепь, положение которой описывается набором обобщенных координат [7], [8].

Рассмотрим три основных подхода к представлению информации о движении.

В первом подходе процесс движения представляется в виде последовательности элементов по количеству кадров движения, где каждый член последовательности представляет собой  $n$ -мерный вектор, содержащий в себе значения контрольных параметров для фигуры персонажа. Такой  $n$ -мерный вектор описывает состояние фигуры персонажа в данном кадре движения. Большинство современных систем используют именно такой способ хранения информации о движении. Преимущества такого подхода очевидны: во-первых, не требуется никаких вычислительных действий для получения состояния персонажа в каждом кадре движения (информация просто считывается из файла и подставляется в модель), а во-вторых, процесс движения может начинаться с произвольного кадра. Существенным недостатком такого подхода является избыточный размер хранимой информации. Но если нас больше интересует скорость воспроизведения движения и мало беспокоит размер информации (как, например, в кинематографии или в мультипликации), такой подход является более предпочтительным.

Ко второму подходу можно отнести, так называемую, процедурную анимацию. В этом случае процесс движения представляет собой набор функций, описывающих изменения, каждого из контрольных параметров. Конечно размер хранимой информации меньше чем в предыдущем подходе, но здесь есть ряд существенных недостатков. Во-первых, такое движение может быть использовано только в той системе, где оно было создано. Во-вторых, для получения состояния фигуры персонажа необходимо вычислить значения функций от времени для каждого контрольного параметра в каждом кадре движения. И, наконец, для того чтобы начать движение с произвольного кадра, необходимо вычислить значения во всех предыдущих.

Третий подход – это так называемая анимация по ключевым кадрам (key-frame animation) [9], [10]. Движение описывается, как и в первом методе, последовательностью  $n$ -мерных векторов. Отличие заключается в том, что значения хранятся не для каждого кадра, а только для некоторых (ключевых) кадров. Значения в остальных интерполируются исходя из информации представленной для ключевых кадров

движения. Преимуществом такого метода является его компактность. Недостаток - вычислительная сложность, поскольку для каждого интервала движения необходимо построить интерполяционную функцию, прежде чем вычислять значения контрольных параметров. Следовательно, время затрачивается не только на вычисление значений функции (как в процедурной анимации), но и на построение самой функции.

### 3. АНИМАЦИЯ ПЕРСОНАЖА В ИНТЕРНЕТ

К сожалению, описанные выше методы представления процесса движения не приемлемы при использовании анимации в сети Интернет или других компьютерных сетях, поскольку они не обладают в должной степени сочетанием компактности и простоты вычислений.

При использовании анимации персонажа в сетевых технологиях необходимо ответить на вопрос: что же более критично для метода представления движений (компактность или быстрота вычислений)? Ответ прост: скорость вычислений современных компьютеров во много раз превосходит скорость передачи информации. Поэтому чем компактнее будет представлена информация, тем быстрее она передастся по сети и, следовательно, пользователь не увидит нежелательных задержек в движении. Но с другой стороны быстрота вычислений столь же важна. Какой смысл быстро передать информацию о движении, если прежде чем отобразить движение, принимающий компьютер будет вынужден произвести вычисления, занимающие больше времени, чем передача полной информации?

Итак, для успешного использования анимации персонажа в компьютерных сетях необходим метод описания движения, сочетающий в себе два противоречивых свойства: компактность представления и быстрота вычислений.

#### 3.1 Новый подход к представлению движения

Разработанный новый метод описания движения основывается на методе анимации по ключевым кадрам. Так же как и в исходном методе выбирается последовательность ключевых кадров, для которых задаются значения контрольных параметров. Первым отличием является то, что в новом методе контрольные параметры могут рассматриваться отдельно, т.е. в некотором ключевом кадре задаются значения тех контрольных параметров, значение которых являются ключевым в данном кадре. Т.е. весь процесс движения представляется не как последовательность  $n$ -мерных векторов, а как  $n$  последовательностей для каждого контрольного параметра (или группы параметров). Конечно, при таком подходе выбор контрольных параметров очень важен. Например, при использовании элементов кватерниона в качестве контрольных параметров, нам необходимо задавать четверку контрольных параметров (один кватернион) [4], а при использовании exponential map [5] – тройку. Это зависит от возможностей параметризации, например Эйлеровы углы могут быть параметризованы независимо, значит, в данном случае, каждый контрольный параметр может рассматриваться отдельно.

Далее после задания значений в ключевых кадрах применяется процесс интерполяции, результатом является набор интерполяционных полиномов. Стандартные методы

анимации по ключевым кадрам могут хранить информацию о движении следующими способами:

1. значения контрольных параметров в ключевых кадрах. Это требует вычисления интерполяционных полиномов в процессе отображения движения, что требует значительных временных затрат.
2. значения интерполяционных параметров. Это позволяет быстро восстановить интерполяционный полином в процессе движения. Но остается та же проблема вычисления значений полинома для каждого кадра.

Большинство методов интерполяции контрольных параметров получают в качестве результата полином третьей степени. Этого вполне достаточно для обеспечения непрерывности второго порядка. Т.е. полученные полиномы могут быть дважды продифференцируемы. Т.к. полином был третьей степени, то его вторая производная будет представлена линейной функцией. Эта функция описывает ускорение изменения значений данного контрольного параметра.

Итак, предлагается хранить только два параметра описывающих линейную функцию ускорения (начальное значение  $p_0$  и наклон  $k$ ) изменения значения контрольного параметра. Т.о. на каждом участке между ключевыми кадрами процесс изменения контрольного параметра описывается всего двумя параметрами.

Кроме того, часто для процесса движения достаточно интерполяция полиномом второго порядка, тогда функция ускорения будет линейной функцией параллельной оси времени. В этом случае для описания процесса движения достаточно всего одного параметра (начального значения функции ускорения).

Как видно предложенный подход позволяет существенно сократить количество информации о движении (максимум 2 значения для каждого контрольного параметра на каждый участок движения). Такое сокращение объема позволяет наиболее быстро передавать информацию о движении по сети. Осталось выяснить насколько быстро данный подход позволяет воспроизводить исходное движение.

Процесс изменения некоторого контрольного параметра представлен начальным значением  $p_0$ , скоростью  $v_0$  и ускорением  $w_0$  и последовательностью пар значений ( $w_i, k_i$ ) для описания функции ускорения. Значение во втором кадре движения может быть вычислено по формуле:  $p_1 = p_0 + v_0$ ; значение в третьем кадре представляется следующим образом:  $p_2 = p_1 + v_1$ , где  $v_1 = v_0 + w_0$ ;

Общая формула вычисления значения контрольного параметра в  $i$  кадре движения следующая:

$$p_i = p_{i-1} + v_{i-1};$$

$$v_{i-1} = v_{i-2} + w_{i-2};$$

$$w_{i-2} = w_{i-3} + k_{i-3};$$

Т.е. значение контрольного параметра для каждого кадра движения строится исходя из информации о предыдущем значении и значениях скорости и ускорения в предыдущем кадре. Это требует временного хранения информации о скорости и ускорении изменения значения параметра. Зато весь процесс вычисления очередного значения сводится к трем операциям суммирования. Для сравнения, вычисление

значения кубического полинома требует четырех операций умножения и трех операций сложения.

Теперь рассмотрим ситуацию, когда движение уже задано. Например, мы имеем движение созданное в других системах анимации и представленное одним из способов описанных в главе “Представление процесса движения” [6]. Прежде чем передавать информацию о движении по сети нам необходимо перезаписать ее в формат описанный выше. Процесс конвертации зависит от способа представления движения.

Если движение представляет собой анимацию по ключевым кадрам и хранит в себе коэффициенты интерполяционных полиномов (и метод интерполяции нам известен), то для конвертации такого движения достаточно просто восстановить интерполяционные полиномы по известным коэффициентам, а затем, дважды продифференцировав их, получить функции ускорений.

Если движение создано процедурной анимацией, то прежде чем конвертировать, нам необходимо привести его к виду стандартной анимации, т.е. рассчитать значения контрольных параметров для каждого кадра движения.

Итак, если мы имеем набор значений контрольных параметров для каждого кадра движения, метод конвертации движения может быть представлен следующими шагами:

1. Получение кривых обобщенных ускорений. Обобщенные ускорения в каждом кадре могут быть вычислены следующим образом:  $w(n) = p(n) + p(n-2) - 2p(n-1)$ , где  $w(n)$  – обобщенное ускорение для кадра  $n$  ( $2 \leq n \leq N$ ),  $p(n)$  – значение обобщенной координаты в кадре  $n$ .
2. Линеаризация полученных кривых. Предлагаются два условия линеаризации:
  - a.  $|w_l(n) - w(n)| \leq \epsilon$  т.е. разница исходного и линеаризованного значений ускорений не превосходит константу  $\epsilon$ .
  - b.  $|p_l(n) - p(n)| \leq \delta$ , т.е. разница исходного и полученного после линеаризации значений обобщенной координаты не превосходит константу  $\delta$ .
3. Для описания полученных обобщенных ускорений вычисляются начальные значения и угол наклона. Данные переписываются в новый формат.

Какими же преимуществами и недостатками обладает данный подход? К преимуществам данного подхода можно отнести следующие:

- Безусловным преимуществом данного подхода является его компактность. Если в стандартном методе анимации требуется хранение  $n$  значений для каждого контрольного параметра ( $n$  – количество кадров движения), в анимации по ключевым кадрам  $(k-1) * m$ , значений ( $k$  – количество ключевых кадров,  $m$  – количество интерполяционных коэффициентов для отрезка движения), то в данном методе хранится максимум  $2(k-1)$  значений.
- Вторым безусловным преимуществом данного подхода является быстрота вычислений исходных

значений контрольного параметра в процессе воспроизведения движения. Как говорилось выше, достаточно выполнить всего три сложения для получения значения параметра в текущем кадре.

К недостаткам подхода можно отнести следующие:

- Как и все методы анимации по ключевым кадрам, данный подход не позволяет начать движение с произвольного кадра. Хотя для большинства анимационных задач такой возможности не требуется.
- В процессе воспроизведения движения необходимо временное выделение памяти для хранения значений скоростей и ускорений изменения контрольного параметра.

Хотя подход и обладает двумя недостатками по сравнению с методом стандартной анимации, не трудно заметить, что эти недостатки могут быть устранены путем перевода движения к формату, используемому в стандартной анимации (т.е. хранение значений контрольных параметров для каждого кадра). Например, информация о движении хранится в сжатом виде и передается по сети в том же компактном формате. Перед процессом воспроизведения движения, если это действительно необходимо, рассчитываются значения для каждого кадра.

#### 4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Для отображения созданной анимации необходимо выбрать язык программирования 3D графики. Наиболее широко используемыми сегодня являются OpenGL и DirectX. Практически все современные системы анимации используют именно эти библиотеки программирования для отображения анимированной сцены на экране монитора. Но если мы говорим об использовании анимации в сети Интернет, такие средства не годятся, поскольку они являются платформо-зависимыми, и мы не можем гарантировать, что пользователь, который будет использовать данную анимацию, имеет такую же платформенную конфигурацию, что и разработчик. Следовательно, язык программирования 3D графики должен быть платформо-независимым и поддерживаться современными Интернет браузерами.

Такой язык существует – это язык VRML. Отличительной чертой VRML является его ориентированность на использование в сети Интернет. Достаточно установить небольшой plug-in для того, чтобы ваш Интернет браузер был способен отображать VRML-сцены. Наиболее популярным и удобным в использовании является plug-in cortona от фирмы ParallelGraphics [3]. Стандарт языка VRML можно найти по адресу [15].

Но недостатком VRML является то, что это не алгоритмический язык, следовательно, невозможно реализовать сложные алгоритмы анимации только возможностями этого языка. Для реализации алгоритмов требуется выбрать дополнительный язык программирования, удовлетворяющий следующим критериям:

- Алгоритмичность (с возможностью вычисления основных арифметических и тригонометрических функций)
- Интеграция с VRML
- Платформо-независимость

- Поддержка современными Интернет браузерами
- Возможность сетевого программирования

Таким языком является Java, который легко интегрируется с VRML через узел Script.

Таким образом, задачу анимации персонажа в сети Интернет можно разделить на две основных подзадачи: задача визуализации сцены (выполняется с использованием VRML) и алгоритмическая реализация или реализация динамики сцены (выполняется с использованием языка Java). Задача передачи информации о движении персонажа так же реализовывается, используя возможности сетевого программирования Java.

Главной отличительной особенностью языка Java является его платформо-независимость, это позволяет использовать этот язык на любых программных платформах с поддержкой данного языка. Значит, система анимации созданная при использовании Java может быть легко переносимой на любые Java-совместимые платформы.

Примером могут служить мобильные телефоны, где широко используются Java технологии. Значит, анимация персонажа доступна и здесь. К сожалению, в мобильных телефонах нет поддержки языка VRML, следовательно, задачу визуализации приходится решать средствами Java. И сегодня эта задача уже решена. Существуют дополнительные Java API, которые позволяют программировать 3D сцены для мобильных телефонов.

## 5. РЕЗУЛЬТАТЫ

Для сравнения эффективности предложенного метода было спроектировано движение “ходьба” для человекоподобного робота на 160 кадров. При проектировании движения использовались 5 ключевых кадров. Данное движение было записано с использованием данного метода и без него (для каждого кадра движения хранились значения обобщенных координат кинематической схемы персонажа). Кроме того, такое же движение было создано в системе Poser 4.0.

После проектирования движения оно было сохранено в файлы. Результаты были получены следующие:

- размер файла Poser = 6700 байт
- размер файла без применения описанного метода = 6500 байт
- размер файла с применением описанного метода = 380 байт

Т.е. использование предложенного метода позволяет сократить размер хранимой информации более чем на 94% при 160 кадрах движения с использованием 5 ключевых кадров.

Данный метод был использован при создании анимации персонажа в мобильном телефоне. Движения “бег”, состоящее из 26 кадров и 4 ключевых кадров было спроектировано с использованием предложенного метода и без него. Результаты следующие:

- размер файла без применения описанного метода = 808 байт
- размер файла с применением описанного метода = 204 байт

Т.е. сжатие информации о движении составляет 75%.

Как видно из результатов, чем больше кадров движения используется, тем больше эффективность предложенного метода.

## 6. ЗАКЛЮЧЕНИЕ

Предлагаемый метод представления движения анимированного персонажа решает проблему уменьшения объема хранимых и передаваемых по сети данных. Применение этого метода позволит более широко использовать анимацию персонажа в сети Internet и компьютерных сетевых играх, виртуальных тренажерах с множественным доступом и в, развивающемся сейчас направлении, 3D-графике для мобильных телефонов.

## 7. БИБЛИОГРАФИЯ

- [1] <http://www.discreet.com/3dsmax> домашняя страница системы компьютерной графике 3D Max Studio.
- [2] [http://www.alias.com/eng/index\\_flash.shtml](http://www.alias.com/eng/index_flash.shtml) официальный сайт компании Alias – производителя системы 3D графики Maya.
- [3] <http://www.parallelgraphics.com> официальный сайт компании ParallelGraphics.
- [4] Ken Shoemake. Animating Rotations with Quaternion Curves. In Brian A. Barsky, editor, Computer Graphics (SIGGRAPH '85 Proceedings), volume 19, pages 245-254, July 1985.
- [5] F. Sebastian Grassia. Practical Parameterization of Rotations Using the Exponential Map. to appear in jgt, Journal of Graphics Tools. Volume 3.3, 1998. A K Peters, Ltd.
- [6] Тютин В. Рациональное описание движения анимированного персонажа в системах 3D - графики. Международная конференция по компьютерной графике Graphicon 2004.
- [7] Qiang Liu and Edmond C. Prakash. The Parameterization of Joint Rotation with the Unit Quaternion. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10-12 Dec. 2003, Sydney
- [8] F. Sebastian Grassia. Practical Parameterization of Rotations Using the Exponential Map. The Journal of Graphics Tools, volume 3.3, 1998.
- [9] A. Bruderlin & T. Calvert, Goal-directed dynamic animation of human walking, Proc.16th Annual Conference on Computer Graphics and Interactive Techniques, Boston, MA, 1989, 233–242.
- [10] A. Bruderlin & T. Calvert, Knowledge-driven, interactive animation of human running, Proc. of the Conference on Graphics Interface, Toronto, Ontario, Canada, 1996, 213–221.

## Об авторах

Тютин Владимир аспирант факультета ВМК Нижегородского Государственного Университета им. Лобачевского.

e-mail: [vtutin@telma.kis.ru](mailto:vtutin@telma.kis.ru)

# **Rational method of motion description in Internet**

## **Abstract**

In this paper there will be reviewed general approaches to represent animated character motion. There will also be indicated particularities of using animation in Internet.

The article suggests the methods to reduce the movement of the animated character data stored and transmitted over network. The article also provides a programming approach to implement animation in Internet.

**Keywords:** Animation, 3D, Internet, animated character, motion.

## **About the author(s)**

Vladimir V. Tyutin, postgraduate student.

Nizhny Novgorod State University.

e-mail: [vtutin@telma.kis.ru](mailto:vtutin@telma.kis.ru)