

УДК 681.3.06

К. С. Бобровский, А. В. Романовский

(Новосибирск)

**МОДЕЛИРОВАНИЕ ДЕФОРМАЦИИ
ТРЕХМЕРНЫХ ПОЛИГОНАЛЬНЫХ ПОВЕРХНОСТЕЙ
В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ**

Деформирование трехмерных объектов – одно из весьма перспективных направлений компьютерной графики, которое в последнее время бурно развивается. В данной работе предлагается алгоритм деформации полигональных поверхностей с линейной в зависимости от числа граней (вершин) производительностью. Производительность известных алгоритмов деформации, как правило, выражается полиномиальной оценкой. Достижение линейной производительности оказалось возможным, потому что целью работы не являлось точное физическое моделирование процесса деформации. Однако разработанный алгоритм обеспечивает эстетически приемлемое и интуитивно ожидаемое качество моделирования реакции тела (поверхности) на деформирующее воздействие. Доказаны изотропность и устойчивость алгоритма.

Введение. Деформация (или морфинг) трехмерных объектов – это большая область компьютерной графики, в которой решаются задачи преобразования геометрических характеристик объектов. Примеры задач из этой области:

- «превращение» трехмерных объектов друг в друга [1, 2];
- анимация – построение промежуточных модификаций объекта (морф-последовательности) по двум крайним, причем каждая модификация может отличаться как пространственным положением, так и формой [3–5];
- произвольная деформация объекта с целью получения новых геометрических форм [6–8];
- моделирование неупругого взаимодействия физических тел [9, 10];
- моделирование роста биологических объектов и т. д.

Конкретный алгоритм, по которому осуществляется деформация, сильно зависит от способа задания объекта. Наиболее употребительными являются:

- а) объемное задание – объект представляется в виде набора элементарных объемов [11];
- б) геометрическое задание – объект представляется в виде набора плоских граней и их вершин;
- в) аналитическое задание – объект задается аналитически в виде поверхностей либо с помощью функций трех координат $F(x, y, z)$ и граничных условий.

К настоящему времени разработано большое количество методов трехмерной деформации, отличающихся по способу задания деформируемых объектов, степени физической достоверности моделируемого изменения формы (здесь имеется в виду соответствие наблюдаемой картины физическим законам) и, конечно, по производительности или по скорости работы. Как правило, деформация тел на основе физических законов требует больших вычислительных затрат, чем свободная деформация, т. е. рассчитываемая без учета физических свойств объекта и влияния других тел. Это объясняется тем, что физика деформации твердых тел по своей природе нелинейна и, кроме того, для достижения реалистичности требуется обработка столкновений разных частей одного объекта, что тоже весьма трудоемкая задача, когда объект имеет более или менее сложную форму.

Анализ предметной области. В работе [6] изложен метод деформации тела на основе задаваемого другим телом «искривления» пространства. В деформации участвуют два объекта – деформируемый (A) и объект-инструмент (B). Внутри B выбирается некоторая точка C – его «центр», и между каждой из вершин M_i объекта A и точкой C проводятся отрезки. Далее находятся их пересечения Π_i с поверхностью объекта B и выполняется следующее преобразование:

$$M_i \rightarrow C + U_i \sqrt[3]{|CM_i|^3 + |C\Pi_i|^3}, \quad (1)$$

где CM_i – вектор, проведенный из C в M_i ; $C\Pi_i$ – вектор, проведенный из C в Π_i ; U_i – единичный вектор в направлении CM_i .

Наблюдаемая деформация – вмятина в A (если C лежит вне A) или выпуклость (если C внутри A) в форме объекта-инструмента B . Изменением положения C в пространстве либо геометрической формы объекта B достигаются разные визуальные эффекты.

В работах по классической деформации решается проблема построения морфпоследовательности по двум крайним представлениям трехмерного объекта. В [1] эта задача ставится в более абстрактном виде: для данного множества S пар трехмерных векторов (P_i, Q_i) , $i = 1, \dots, |S|$, найти такое преобразование G , что $G(P_i) = Q_i$ для любого i . Авторы [1] взяли за основу уже существующее решение:

$$G(P) = \frac{\sum_{i=1}^n \frac{Q_i}{|Q_i - P|^2}}{\sum_{i=1}^n |Q_i - P|^{-2}}, \quad (2)$$

на основе которого они построили новую функцию $G(P)$, обладающую более приемлемыми характеристиками: инвариантностью относительно аффинных преобразований, устойчивостью по отношению к пространственному распределению P_i и Q_i , гладкостью (в математическом смысле).

При деформации сложных объектов [3] часто используют подобие исходного и конечного объемов: каждый из них разбивается на простые составляющие (элементы) и между ними устанавливается соответствие. Тогда процесс построения морфпоследовательности разделяется на несколько параллельных процессов для каждой из пар соответствующих элементов. Это, в частности, делается для упрощения управления такой деформацией.

В [9] получил свое дальнейшее развитие метод деформации на основе использования так называемого неявного задания поверхности объекта. Основа метода – моделирование физического тела как набора связанных частиц и ассоциированной с ними неявной поверхности. На частицы могут действовать силы притяжения/отталкивания F_{int} , а также силы жидкого трения F_{fr} :

$$F_{int}(P_1 \rightarrow P_2) = \frac{k(P_1 - P_2)}{r^2} \left(\left(\frac{r_0}{r} \right)^8 - \left(\frac{r_0}{r} \right)^4 \right), \quad (3)$$

$$F_{fr}(P_1 \rightarrow P_2) = m(r) \left| \frac{dP_1}{dt} - \frac{dP_2}{dt} \right| \left(\frac{dP_1}{dt} - \frac{dP_2}{dt} \right), \quad (4)$$

где P_1 и P_2 – координаты частиц; k – параметр, зависящий от жесткости; $r = |P_1 - P_2|$, $m(r)$ – убывающая функция.

Каждая частица является источником поля f_i , напряженность которого убывает с расстоянием до нуля. Поверхность тела определяется множеством точек P , для которых

$$f_1(P) + \dots + f_n(P) = \text{const.} \quad (5)$$

Вместе с техникой контроля изменения объема при столкновении этот метод позволяет моделировать сложнейшие эффекты взаимодействия аморфных тел с поверхностями (протекание сквозь отверстие, растекание по поверхности) и между собой, но вычислительные затраты при его применении чрезвычайно высоки.

В случае моделирования деформации плоских протяженных объектов (одежды, поверхности человеческого лица) часто используется классический способ представления трехмерной поверхности в виде связанных треугольных граней, как, например, в [10]. В этой работе авторы также вводят в рассмотрение такие физические характеристики тела, как модуль Юнга, коэффициент Пуассона, плотность и толщина материала. Деформация происходит под действием внешних сил и внутренних напряжений в материале.

В данном методе элементарный треугольник рассматривается как жесткий элемент, на который действуют силы кручения и поступательные силы. На их основе вычисляются векторы сил, действующих на все вершины объекта, затем уравнения движения каждой из них интегрируются и вычисляются их новые координаты. При этом деформация под воздействием силы F , меньшей, чем некоторый порог F_0 , считается упругой и подчиняющейся закону Гука:

$$F(t) = k(x(t) - x(0)), \quad (6)$$

если $F > F_0$, то деформация неупругая. Этот метод вместе с предложенным авторами алгоритмом отслеживания столкновений дает впечатляющие по своей реалистичности результаты при анимации одежды на движущемся человеке, флага на ветру, ряби на поверхности воды, но весьма трудоемок.

Достоинства и недостатки описанных методов. В большей части приведенных здесь методов деформации используются весьма сложные пред-

ставления трехмерных объектов и алгоритмы изменения их формы. Эта усложненность имеет целью либо достижение высокой физической достоверности процесса деформации [9, 10], либо получение эффектно и реалистично выглядящих морфоследовательностей. В работах [1, 3, 6] отсутствует рассмотрение каких-либо внешних воздействий или взаимодействия объектов друг с другом, в них изложены методы изменения формы объекта по каким-либо заранее заданным правилам генерации новых геометрических форм.

Постановка задачи. Целью данной работы является создание алгоритма деформации трехмерного тела (в общем случае полигональной поверхности), который бы совмещал в себе высокую скорость работы и приемлемое качество моделирования реакции тела (поверхности) на внешнее воздействие. При разработке алгоритма деформации авторы исходили из того, что модель поверхности и алгоритм должны обеспечивать легко предсказуемую деформацию как реакцию тела на конкретное воздействие извне; скорость работы алгоритма должна позволять его использование в реальном или близком к реальному времени.

Модель трехмерной поверхности и алгоритм распространения деформации. Поверхность представляет собой набор плоских граней, грань – набор вершин, соединенных ребрами, причем между любыми двумя вершинами должен существовать путь по ребрам, т. е. поверхность должна быть «связной». С каждой вершиной ассоциируется ее «масса», а с каждым ребром – некий «коэффициент жесткости». Шаги алгоритма распространения деформации:

1. Выделенная вершина тела, называемая «вершиной поколения 0», смещается в трехмерном пространстве на некоторый вектор. Эта вершина помечается как обработанная.

2. Для вершины поколения 0 находятся все смежные по какому-либо из ребер вершины – это вершины поколения 1. Для каждой из них вычисляется вектор смещения, зависящий только от вектора смещения вершины поколения 0. Эти вершины также помечаются как обработанные.

3. Далее для каждой из вершин текущего поколения находятся все ее потомки – смежные и еще не помеченные вершины. Для текущего потомка в его список «родителей» помещается текущая вершина-родитель. Все потомки помечаются и становятся вершинами следующего поколения, смещение каждой из которых вычисляется на основе смещений вершин из списка родителей.

Распространение деформации продолжается до тех пор, пока не останутся непомятые вершин. Схема распространения показана на рис. 1.

Доказательство сходимости и изотропности алгоритма. Из описания алгоритма распространения деформации очевидно, что он является модификацией алгоритма «поиска в ширину» в графе. Однако поскольку этот граф связный, то теория гарантирует обход всех вершин за время, линейно зависящее от их количества. Этот алгоритм также изотропен, т. е. смещение любой из вершин любого из поколений (кроме поколения 0) не зависит от порядка обхода вершин поколений, предшествующих данному. Докажем это индукцией по числу поколений. Смещение вершин первого поколения зависит только от смещения вершины нулевого поколения, поэтому порядок вычисления смещений не влияет на их векторные значения. Пусть теперь это верно для вершин n -го поколения. Докажем, что это также верно для вершин

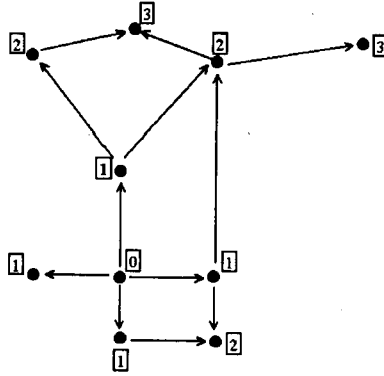


Рис. 1. Схема распространения деформации по ребрам

$n + 1$ -го поколения. Ясно, что множество родителей вершин поколения $n + 1$ совпадает с множеством вершин поколения n (по построению). Однако поскольку на каждом шаге алгоритма смещения вычисляются сначала для всех родителей, а уже затем находятся для каждого из потомков на основе лишь списка его родителей, то смещение потомка не зависит от порядка вычисления смещений родителей. Итак, для любого n , принадлежащего множеству номеров поколений N (за исключением нулевого), смещения вершин поколения n не зависят от порядка вычисления смещений (ПВС) вершин поколения $n + 1$, что и требовалось доказать.

Вычисление новых координат вершин. Рассмотрим зависимость вектора смещения вершины от векторов смещений ее вершин-родителей. Пусть у вершины v имеется N вершин-родителей v_1, \dots, v_N со смещениями d_1, \dots, d_N , начальными длинами ребер, соединяющих вершину v и ее родителей v_1, \dots, v_N , равными l_1, \dots, l_N , и коэффициентами жесткости этих ребер k_1, \dots, k_N . Предлагается следующая формула для вычисления смещения вершины v :

$$\begin{aligned} \overline{dV}(v_1, \dots, v_N, d_1, \dots, d_N, m, l_1, \dots, l_N, k_1, \dots, k_N) = \\ = \frac{1}{(m + 1, 0)N} \sum_{i=1}^N k_i (|v_i + d_i - v| - l_i) e_i, \end{aligned} \quad (7)$$

где m – масса вершины v , $m \geq 0$; v_1, \dots, v_N – радиусы-векторы соответствующих вершин; e_1, \dots, e_k – единичные векторы в направлении разности векторов $v_i + d_i$ и v (рис. 2), $0 \leq k_i \leq 1$.

При выведении формулы для вычисления смещения вершины v авторы рассмотрели случай двух вершин (родителя и потомка) и руководствовались следующим:

- модуль смещения вершины должен быть убывающей функцией ее массы (по аналогии с инерционностью тела в физике);
- модуль смещения вершины также должен быть возрастающей функцией жесткости соединяющего вершины ребра: условно говоря, оно может «растягиваться» и «сжиматься»;

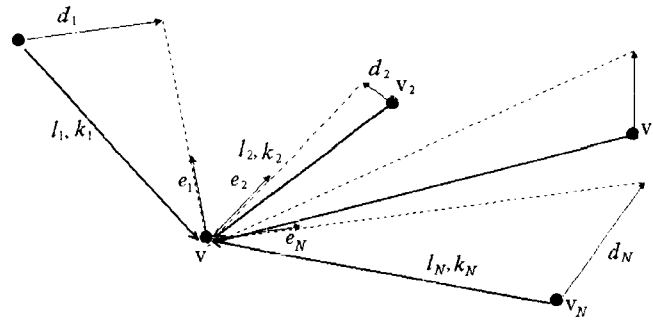


Рис. 2. Вычисление смещения вершины

- для каждой вершины при вычислении ее смещения необходимо учитывать только вклад вершин из списка ее родителей;
- формула должна быть достаточно простой для использования ее при вычислениях в реальном времени.

На основе этого и был выбран конкретный вид упомянутых функций. Зависимость от массы, как видно из формулы, – обратная пропорциональность (слагаемое l введено для избежания деления на нуль); коэффициенты жесткости входят в члены, соответствующие вершинам-родителям, как множители.

Рассмотрим теперь остальную часть формулы. Каждое слагаемое суммы есть произведение следующих величин: $|v_i + d_i - v| - l_i$ – знаковое изменение длины ребра при допущении, что вершина v остается на месте; e_i – единичный вектор в направлении, в котором бы двигалась вершина v в отсутствие всех родителей, за исключением i -го. Линия смещения есть линия, проведенная через вершину v (до смещения) и конечное положение вершины v_i . Конечный же вектор смещения вершины v есть результат суперпозиции (т. е. суммы) влияний всех родителей. Множитель l/k введен для усреднения этого влияния, так как если радиусы-векторы всех родителей, их векторы смещений и жесткости ребер соответственно равны, то смещение вершины v интуитивно должно быть таким же, как если бы у v был всего один родитель с таким же смещением, радиусом-вектором и жесткостью соединяющего ребра.

Оценка производительности. На основе приведенной выше формулы можно легко оценить вычислительную трудоемкость данного алгоритма: для этого необходимо лишь знать среднее число родителей у каждой вершины трехмерного объекта и общее количество вершин. Из рис. 2 видно, что для объектов, поверхность которых аппроксимируется треугольниками и четырехугольниками соответственно, первый параметр равен двум, а именно эти случаи аппроксимации поверхности являются наиболее употребительными в трехмерной графике. (Здесь следует отметить, что если полигоны, составляющие поверхность объекта, – не треугольники, то после деформации ребра такого полигона потребуются его разбиение на треугольники, так как он перестанет быть плоским.) Ясно, что затраты на вычисление смещения одной вершины, как это видно из формулы, таковы:

сложений – 10, умножений – 9, делений – 1.

С ростом числа вершин среднее количество родителей у вершины не меняется. Можно считать это эмпирическим фактом, поэтому зависимость количества вычислений от числа вершин в объекте линейная. Отсюда получаем следующую оценку для времени полного вычисления деформации тела:

$$T = N(11mul + 10add + div + 2sqrt), \quad (8)$$

где N – общее количество вершин; mul , add , div и $sqrt$ – время, затрачиваемое процессором на выполнение соответственно одного умножения, сложения, деления и вычисления квадратного корня.

Предлагаемый алгоритм деформирования полигональных поверхностей имеет линейную в зависимости от числа граней (вершин) производительность. Производительность известных авторам алгоритмов деформирования твердых тел (поверхностей) выражается полиномиальной оценкой.

Эффективность и логическая ясность алгоритма делают его привлекательной основой для последующих экспериментов. Достоинства алгоритма: хорошая предсказуемость результата в зависимости от воздействия; параметризуемость (пользователь может получать из объекта-оригинала отличные друг от друга деформированные объекты); масштабируемость (скорость работы легко повысить введением ограничения на глубину обрабатываемых поколений).

К недостаткам алгоритма можно отнести не очень высокую степень физической достоверности моделирования деформации.

Область применения алгоритма, его возможные модификации. По классификации, приведенной во введении, данный алгоритм может быть использован для решения следующих задач:

1) произвольная деформация объекта с целью получения новых геометрических форм;

2) моделирование неупругого взаимодействия физических тел.

Отсюда следует, что область возможного применения алгоритма весьма широка – трехмерные игры, компьютерная мультипликация, компьютерное искусство и т. д.

Описанный алгоритм можно также рассматривать как базовый, на основе которого легко далее строить более сложные модели. Рассмотрим некоторые возможные его модификации и дополнения, которые придадут процессу деформации большую зрелищность, а самому алгоритму – большую универсальность.

1. Введение порога растяжения/сжатия отдельного ребра. Под растяжением ребра здесь понимается отношение его длин до и после смещения вершины-потомка. Если это отношение больше/меньше определенного порогового значения, то это ребро можно разбить на два (или больше) и считать новую вершину родителем для старого потомка. Для реализации такого дополнения, конечно, потребуются генерация новых граней и ребер, смежных с вновь созданной вершиной. Пример подобной деформации куба показан на рис. 3.

2. «Закрепление» некоторых выбранных вершин, т. е. такие вершины не будут менять свои координаты при смещениях остальных «свободных вершин». В данном случае даже не потребуется менять сам алгоритм: этого можно достичь путем присваивания вершинам такого рода бесконечно большой массы (реально – максимального числа вещественного типа).

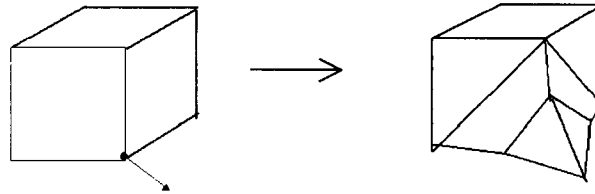


Рис. 3. Эффект увеличения числа граней

3. Деформирующее воздействие сразу на несколько вершин. Изменения в алгоритме в этом случае незначительны. Меняются лишь начальные условия: он стартует не с нулевого, а с первого поколения, как будто бы у необработываемой вершины нулевого поколения есть N потомков, где N – число первоначально смещаемых вершин.

4. «Квантование» внешнего воздействия, т. е. смещение первоначальных вершин под его влиянием, должно осуществляться в несколько проходов алгоритма: на каждом проходе вершины смещаются на векторы некоторой характерной длины, в сумме дающие полное смещение. Для получения «гладкости» наблюдаемой деформации эта длина должна быть достаточно мала по сравнению со средней длиной ребра деформируемого объекта или с какой-либо другой величиной, характеризующей его пространственный масштаб.

СПИСОК ЛИТЕРАТУРЫ

1. **Fang S., Raghavan R., Richtsmeier J. T.** Volume morphing methods for landmark based 3D image deformation // SPIE. 1996. **2710**: SPIE Internat. Symp. on Medical Imaging. Newport Beach, CA, 1996.
2. **Lazarus F., Verroust A.** Feature-based shape transformation for polyhedral objects // 5th Eurographics Workshop on Animation and Simulation. Oslo, Norway, 1994.
3. **Lerios A., Garfinkle C. D., Levoy M.** Feature-based volume metamorphosis // Proc. SIGGRAPH, 1995.
4. **Erikson C.** Morphing three dimensional polyhedral objects // Oberlin College Honors Paper, 1994.
5. **Chen M., Jones M. W., Townsend P.** Methods for volume metamorphosis // Image Processing for Broadcast and Video Production /Eds. Y. Paker, S. Wilbur. London: Springer-Verlag, 1995.
6. **Decaudin P.** Geometric deformation by merging a 3D-object with a simple shape // Proc. Graphics Interface. Toronto, Canada, 1996. P. 55.
7. **Kent J. R., Carlson W. E., Parent R. E.** Shape transformation for polyhedral objects // Computer Graphics (Proc. SIGGRAPH). 1992. **26(2)**, P. 47.
8. **Sederberg T. W., Parry S. R.** Free-form deformations of solid geometric models // Computer Graphics (Proc. SIGGRAPH). 1986. **20(4)**, P. 151.
9. **Desbrun M., Gascuel M.-P.** Animating soft substances with implicit surfaces // First Internat. Eurographics Workshop on Implicit Surfaces, 1995.
10. **Volino P., Courchesne M., Thalmann N. M.** Versatile and efficient techniques for simulating cloth and other deformable objects // Proc. SIGGRAPH. 1995. P. 137.
11. **He T., Wang S., Kaufman A.** Wavelet-based volume morphing // Proc. Visualization. Los Alamitos, CA (IEEE Computer Society and ACM SIGGRAPH). 1994. P. 85.

Поступила в редакцию 30 марта 1998 г.