

Научные труды Дальрыбвтуза. 2024. Т. 69, № 3. С. 183–195.

Scientific Journal of the Far Eastern State Technical Fisheries University. 2024. Vol. 69, no 3. P. 183–195.

РЫБНОЕ ХОЗЯЙСТВО, АКВАКУЛЬТУРА И ПРОМЫШЛЕННОЕ РЫБОЛОВСТВО

Научная статья

УДК 639.3; 532.3; 004.92; 004.94

DOI: doi.org/10.48612/dalrybvtuz/2024-69-18

EDN: QRFAGU

Деформация трехмерных моделей на ГПУ для задач виртуальной реальности на примере рыбы в УЗВ

Алексей Олегович Ражев¹, Александр Алексеевич Недоступ²,
Леонид Алексеевич Кондрашов³

^{1,3} Лаборатория цифровых технологий, Калининград, Россия

^{1,2} Калининградский государственный технический университет, Калининград, Россия

¹ root@digitechlab.ru, ORCID: 0000-0002-0983-834X

² nedostup@klgtu.ru, ORCID: 0000-0002-3851-0984

³ kla@digitechlab.ru

Аннотация. В задачах виртуальной реальности необходимо в реальном времени и с приемлемым качеством визуализировать моделируемые натурные процессы. В статье рассмотрены постановка и решение задачи разработки математической модели и программного кода на графическом процессоре, способных в режиме реального времени производить деформацию статической трехмерной модели объекта визуализации по заданным характеристикам, изменяющимся во времени, на примере визуализации рыб в виртуальной установке замкнутого водоснабжения, представляющей техническую систему для выращивания рыб. Предложены метод, математическая база и вычислительный алгоритм деформации статических трехмерных моделей мягких тел, полученных с применением устройств пространственного сканирования, способные видоизменять (деформировать) тело в процессе визуализации на графическом процессоре. Деформация проводится с применением текущих поверхностей, описываемых математическими зависимостями, связанными как с параметрами деформации, так и с модельным временем. Вычисляются координаты всех вершин поверхностной сетки тела и всех нормалей к поверхности, определенных в трехмерной модели. В статье приведена реализация алгоритма на вершинном шейдере, написанном на языке программирования HLSL. Приведен результат визуализации рыбы при изгибе.

Ключевые слова: математическая модель, виртуальная реальность, ГПУ, HLSL, рыба

Финансирование: Исследование выполнено в ООО «Лаборатория цифровых технологий» за счет гранта Российского научного фонда № 23-21-00010, <https://rscf.ru/project/23-21-00010/>.

Для цитирования: Ражев А. О., Недоступ А. А., Кондрашов Л. А. Деформация трехмерных моделей на ГПУ для задач виртуальной реальности на примере рыбы в УЗВ // Научные труды Дальрыбвтуза, 2024. Т. 69, № 3. С. 183–195.

FISHERIES, AQUACULTURE AND INDUSTRIAL FISHERIES

Original article

Deformation of three-dimensional models on the GPU for virtual reality applications on the example of fish RAS

Alexey O. Razhev¹, Aleksandr A. Nedostup², Leonid A. Kondrashov³

^{1,3} DigiTech Laboratory, Kaliningrad, Russia

^{1,2} Kaliningrad State Technical University, Kaliningrad, Russia

¹ root@digitechlab.ru, ORCID: 0000-0002-0983-834X

² nedostup@klgtu.ru, ORCID: 0000-0002-3851-0984

³ kla@digitechlab.ru

Abstract. In virtual reality applications, it is necessary to visualize simulated natural processes in real time and with acceptable quality. The article discusses the formulation and solution of the problem of developing a mathematical model and program code on a graphics processor unit capable of deforming a static three-dimensional model of a visualization object in real time in accordance with the specified time-dependent characteristics. For example, the visualization of fish in a virtual installation of an aquaculture recirculating system, which is a technical system for growing fish, is considered. A method, a mathematical base and a computational algorithm for deformation of static three-dimensional models of soft bodies obtained using spatial scanning devices capable of modifying (deforming) the body during visualization on a graphics processor are proposed. Deformation is carried out using secant surfaces described by mathematical dependencies related to both deformation parameters and model time. The coordinates of all vertices of the surface mesh of the body and all normal to the surface defined in the three-dimensional model are calculated. The article presents the implementation of the algorithm on a vertex shader created on HLSL programming language. The result of visualization of fish during bending is given.

Keywords: multiphysical similarity, RAS, recirculating aquaculture system, hydrobiont, carp

Funding: The research was carried out in DigiTech Laboratory at the expense of a grant from the Russian Science Foundation (project No. 23-21-00010), <https://rscf.ru/project/23-21-00010/>.

For citation: Razhev A. O., Nedostup A. A., Kondrashov L. A. Deformation of three-dimensional models on the GPU for virtual reality applications on the example of fish RAS. *Scientific Journal of the Far Eastern State Technical Fisheries University*. 2024; 69(3): 183–195. (in Russ.).

Введение

В задачах виртуальной реальности необходимо в реальном времени и с приемлемым качеством визуализировать моделируемые натурные процессы. Для эффективного решения указанных задач требуются как мощное вычислительное оборудование, так и оптимизированные под используемое оборудование математические модели и алгоритмы.

При компьютерной имитации физических процессов можно выделить следующие основные этапы:

1. Определение характеристик математической модели моделируемой системы, объектов системы.
2. Определение входных параметров внешних воздействий на систему и ее объекты.

3. Предсказание поведения объектов системы (предсказательная модель) с учетом законов физики (динамики движения и т.д.), взаимодействия объектов внутри системы, возможных их столкновений (коллизий), влияния на систему внешних факторов.

4. Вычисление положения и ориентации объектов в пространстве и их формы, выходных параметров системы и ее объектов.

5. Визуализация, систематизация и накопление результатов моделирования для последующего анализа.

6. Визуализация пространственной сцены с применением различных технических средств, в том числе и стереоскопических [1].

7. Обработка результатов с целью последующего проецирования на натурную систему, в том числе с использованием технологий дополненной реальности [2], систем принятия решений и управления натурной системой.

В современных персональных ЭВМ расчет и визуализация трехмерных сцен происходит с использованием ресурсов как центрального, так и графического процессорного устройства (ГПУ) [3]. Для визуализации пространственной сцены требуется информация о форме, положении и ориентации в пространстве визуализируемых объектов. Информация о положении и об ориентации в пространстве формируется в процессе имитации (этап 4). Информация о форме твердых тел извлекается из их заранее подготовленных цифровых 3D-моделей. Для визуализации мягких (формоизменяемых) тел информация о форме должна генерироваться в процессе имитации (этап 4). В качестве примера рассмотрим задачу визуализации поведения рыб в виртуальной установке замкнутого водоснабжения (УЗВ). УЗВ представляет собой техническую систему для выращивания рыб [4], применяемую в рыбоводных хозяйствах. В качестве объектов визуализации рассматривается рыба, ограниченная рамками бассейна. Движение рыбы обеспечивается изменением ее формы, в результате которого возникают гидродинамические силы и вращательные моменты, придающие рыбе требуемые линейное и угловое ускорения. В предсказательной модели движения рыбы [5, 6] вычисляются требуемые ускорения, на основе которых определяются ее скорость, перемещение и ориентация в пространстве.

Таким образом, имея информацию о положении и ориентации в пространстве в текущий момент времени, можно визуализировать рыбу как твердое тело по заранее подготовленной ее трехмерной модели. Для этой цели можно использовать 3D-сканер либо другое оборудование, в том числе с лидаром [7] либо ToF-камерой [8], например, мобильный смартфон моделей iPhone или Samsung.

Визуализация статической модели сокращает вычислительные затраты, одновременно ухудшая качество восприятия реальности. Для приближения качества визуализации к реальности необходимо одновременно с движением рыбы повторять ее форму, т.е. производить деформацию трехмерной модели.

Объекты и методы исследований

Задачей исследования является разработка математической модели и программного кода на графическом процессоре, способных в режиме реального времени производить деформацию статической трехмерной модели объекта визуализации по заданным характеристикам, изменяющимся во времени, и отображение деформированного объекта (гибкого тела) [9]. При этом визуализируемая сцена может содержать множество как однотипных, так и разного типа деформируемых объектов, характеристики деформации которых вычисляются индивидуально и в реальном времени.

При визуализации на графическом процессоре объекты трехмерной сцены (тела) описываются поверхностями с дискретизацией треугольными примитивами (гранями). Вершины и ребра примитивов создают поверхностную сетку тела (рис. 1).

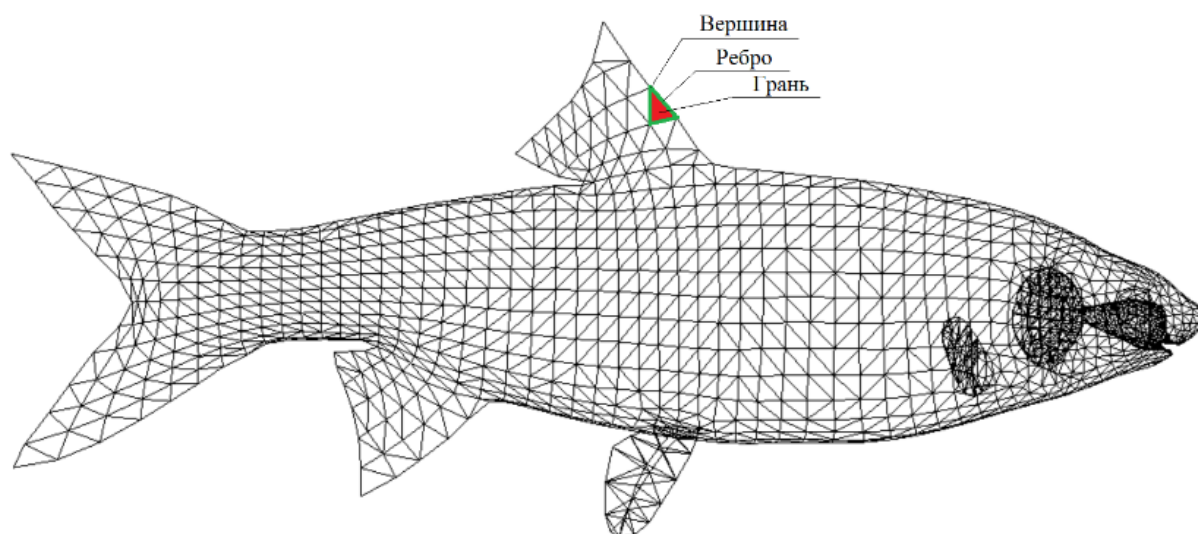


Рис. 1. Поверхностная сетка тела при визуализации на ГПУ. Составлено авторами
 Fig. 1. The surface mesh of the body when visualizing on the GPU. Compiled by the authors

Процесс визуализации пространственной сцены на графическом процессоре называется рендерингом. Рендеринг осуществляется поэтапно на программно-аппаратном графическом конвейере [3]. Конфигурация графического конвейера задается программным способом в зависимости от решаемой задачи. На рис. 2 приведена конфигурация, применяемая для решения поставленной задачи. Аппаратные этапы (стадии) конвейера отображены в виде белых прямоугольников, программные (шейдеры) – в виде темных прямоугольников.

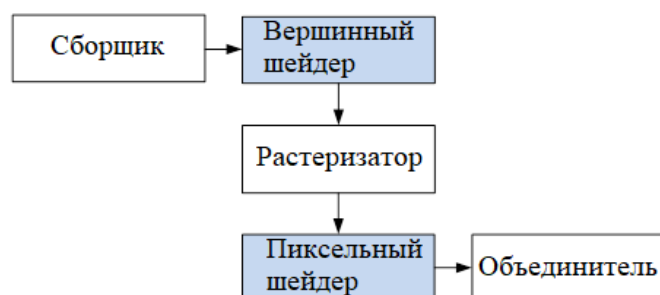


Рис. 2. Графический конвейер. Составлено авторами
 Fig. 2. Graphic pipeline. Compiled by the authors

На входе графический конвейер принимает массивы данных с информацией о вершинах и связях между вершинами. На первом этапе конвейера (этап сборки) происходит подготовка данных для вершинного шейдера. Вершинный шейдер выполняет расчет итоговых координат вершин на плоскости устройства визуализации (поверхности визуализации) и их удаления от поверхности визуализации (глубины), производя их трансформацию и проекцию на плоскость (ортогональную или перспективную), а также расчет цветов вершин согласно используемой цветовой модели с учетом источников освещения и ориентации к ним граней примитивов (поверхности тела). На следующем этапе осуществляется группировка вершин по примитивам и их растеризация согласно настройкам аппаратного растеризатора и информации о координатах на плоскости визуализации, текстурных координатах, глубине и цвете вершин. Растеризатор преобразует каждый примитив в набор пикселей с привязкой информации о глубине и

цвете, выполняя расчет цвета и глубины пикселя методом интерполяции глубин и цветов вершин.

Пиксельный шейдер выполняет пересчет цвета пикселя с учетом цвета текстуры по текстурным координатам по заданному алгоритму.

На последнем этапе происходит объединение наборов пикселей от всех примитивов с учетом глубины. Для этой цели используется двухмерный буфер глубины, в каждой ячейке которого хранится значение текущей (по мере поступления пикселей от пиксельного шейдера) глубины наименее удаленного (с наименьшей глубиной) от поверхности визуализации пикселя. Глубина каждого поступившего пикселя сопоставляется с информацией о минимальной глубине в буфере, на основании чего принимается решение об отображении пикселя на поверхности визуализации с обновлением значения глубины в буфере глубины, либо об исключении пикселя из дальнейшей обработки.

Задача деформации статической трехмерной модели тела заключается в пересчете координат ее вершин и нормалей к поверхности в соответствии с заданными характеристиками деформации. Таким образом, расчет деформации должен быть реализован на вершинном шейдере.

Рассмотрим решение поставленной задачи методом иерархического разбиения мягкого (гибкого, формоизменяемого) тела на вложенные друг в друга компоненты и последующей их деформацией цепочкой последовательных трансформаций. Каждому компоненту поставим в соответствие определенную цепочку трансформаций. На вершине иерархии будет расположен компонент, представляющий собой целиком мягкое тело. На следующем уровне – компоненты, представляющие собой части тела, которые, в свою очередь, могут также состоять из частей. Количество уровней иерархии (вложенностей компонентов) в общем случае не ограничено. Таким образом, компоненты нижних уровней будут претерпевать последовательные трансформации, начиная с заданных для компонента и далее вверх по иерархии, и заканчивая трансформациями, заданными для всего тела.

Обозначим множество вершин поверхностной сетки тела (множество вершин верхнего уровня иерархии) (см. рис. 1) $\Omega_{1,1}$, а $\Omega_{2,j}$ и $\Omega_{2,k}$ – множества вершин компонентов j и k соответственно. Здесь $j \in [1 \dots n_2]$ и $k \in [1 \dots n_2]$ – порядковые номера вложенных компонентов; $j \neq k$; n_2 – количество вложенных в тело компонентов. В приведенном обозначении первый индекс указывает уровень иерархии (начиная с 1 для верхнего уровня), а второй индекс – порядковый номер компонента. Тогда получаем

$$\Omega_{2,j} \subset \Omega_{1,1}, \quad \Omega_{2,k} \subset \Omega_{1,1}, \quad \Omega_{2,j} \cap \Omega_{2,k} = \emptyset.$$

Считаем вышесказанное справедливым как для множества компонентов тела, так и для множеств вложенных в компонент компонентов.

В процессе визуализации трехмерной сцены для корректного отображения тел в пространстве необходимо учитывать источник (источники) освещения. Для учета освещенности необходимо рассчитать углы падения световых лучей на поверхность тела. С этой целью в каждой вершине на поверхности тела хранится информация о нормали к этой поверхности. При растеризации графических примитивов (частей поверхности тела) освещенность в каждом пикселе примитива рассчитывается аппаратным растеризатором интерполяцией значений освещенности его вершин.

Любое линейное пространственное преобразование можно разложить на последовательность преобразований сдвига, поворота и масштабирования. В общем случае количество и последовательность преобразований каждого типа не ограничено. При преобразовании сдвига выполняется поступательное перемещение множества вершин в пространстве, т.е. координаты всех вершин множества изменяются на один и тот же вектор смещения в пространстве t :

$$\mathbf{v}' = \mathbf{v} + \mathbf{t}.$$

При преобразовании поворота множество вершин поворачивается относительно текущего (с учетом предыдущих преобразований) начала координат на заданный пространственный угол (меняет ориентацию в пространстве). При преобразовании масштабирования с сохранением пропорций координаты каждой вершины множества умножаются на масштабный коэффициент s (объект, описываемый множеством вершин, изменяет размер):

$$\mathbf{v}' = s\mathbf{v}.$$

В ГПУ линейные пространственные преобразования математически выполняются с использованием аппарата линейной алгебры: путем последовательного умножения матриц на вектор (реализация умножения заложена в аппаратную часть ГПУ, что обеспечивает ускорение процесса преобразований):

$$\begin{pmatrix} \mathbf{v}' \\ v_4 \end{pmatrix} = \left(\prod_i \mathbf{M}_i \right) \begin{pmatrix} \mathbf{v} \\ v_4 \end{pmatrix},$$

где \mathbf{v} и \mathbf{v}' – вектор до и после преобразования соответственно; v_4 и v_4' – компоненты, повышающие размерность векторов до 4, для координат вершин $v_4 = 1$; \mathbf{M}_i – матрица размерностью 4×4 , описывающая i -е преобразование.

При выполнении пространственных преобразований только при помощи операций умножения, тем самым унифицировав все типы преобразований, мы переходим к размерности всех векторов – 4, а размерности всех матриц – 4×4 (преобразование сдвига невозможно осуществить с использованием трехмерных векторов и матриц 3×3).

Для замены операции векторного сложения при преобразовании сдвига вершин ($v_4 = 1$) на операцию умножения матрицы на вектор произведем замену:

$$(\mathbf{v}; 1)^T + (\mathbf{t}; 0)^T = \mathbf{T}(\mathbf{v}; 1)^T,$$

где T – операция транспонирования; \mathbf{T} – матрица сдвига:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Описанный выше подход является стандартным для задач машинной графики.

Кроме пространственных преобразований координат вершин необходимо также выполнять преобразования ориентации векторов нормалей к поверхностям в пространстве (преобразование поворота). При этом преобразования сдвига к нормальям применяться не должны, а в случае применения преобразования масштабирования в цепочке последовательных преобразований по окончании преобразований вектор нормали необходимо нормализовать (привести его длину к 1). С целью возможности использования для преобразований координат вершин и нормалей к поверхности тела одних и тех же матриц для векторов нормалей устанавливается $v_4 = 0$. При этом получаем

$$(\mathbf{n}', 0)^T = \mathbf{T}(\mathbf{n}, 0)^T = (\mathbf{n}, 0)^T.$$

Каждую вершину \mathbf{V}_b поверхностной сетки тела до деформации представим парой векторов

$$\mathbf{V}_b = \{ \mathbf{p}_b; \mathbf{n}_b \},$$

где

$$\mathbf{p}_b = (x_v; y_v; z_v)^T; \quad \mathbf{n}_b = (x_n; y_n; z_n)^T;$$

\mathbf{p}_b – положение вершины в пространстве; x_v, y_v, z_v – пространственные координаты вершины в системе координат с ортонормированным базисом $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$, привязанной к телу; $\mathbf{e}_1=\mathbf{e}_2=\mathbf{e}_3=1$; \mathbf{n}_b – орт нормали к поверхности тела в точке \mathbf{p}_b ; x_n, y_n, z_n – компоненты орта нормали.

Зададим для каждой трансформации в цепочке трансформаций компонента (или тела) локальную ортогональную систему координат с базисом $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$, нормированной длиной s_x компонента (расстоянием между наиболее удаленными вершинами поверхностной сетки компонента в проекции на \mathbf{e}_x) с сохранением его пропорций (рис. 3):

$$\mathbf{e}_x = \mathbf{e}_y = \mathbf{e}_z = s_x. \quad (1)$$

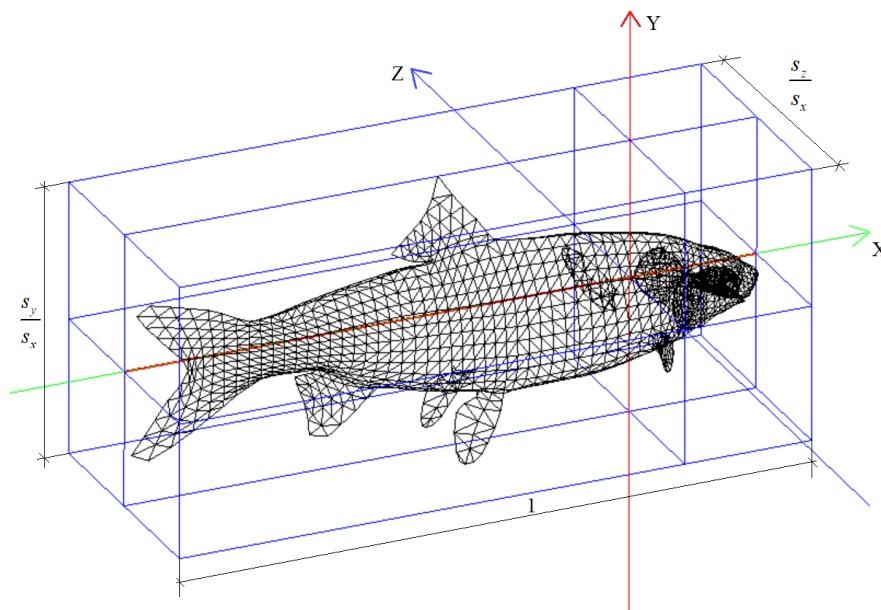


Рис. 3. Локальная система координат. Составлено авторами
Fig. 3. Local coordinate system. Compiled by the authors

Нормирование длины компонента (тела) перед трансформацией необходимо для возможности применения секущих поверхностей, описываемых одними и теми же параметрическими зависимостями, для однотипных тел различного размера (например, большая и маленькая рыба одного вида).

Для преобразования координат положения вершин и нормалей к поверхности из базиса $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ в базис $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ и обратно применим преобразования

$$\begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix} = \mathbf{M} \begin{pmatrix} \mathbf{p}_b \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{p}'_b \\ 1 \end{pmatrix} = \mathbf{M}^{-1} \begin{pmatrix} \mathbf{p}' \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{n}_v \\ 0 \end{pmatrix} = \mathbf{M} \begin{pmatrix} \mathbf{n}_b \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{n}'_b \\ 0 \end{pmatrix} = \mathbf{M}^{-1} \begin{pmatrix} \mathbf{n}'_v \\ 0 \end{pmatrix}, \quad (2)$$

где \mathbf{p} и \mathbf{p}' – положение вершины в базисе $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ до и после трансформации соответственно;

$$\mathbf{p} = (x; y; z)^T; \mathbf{p}' = (x'; y'; z')^T;$$

\mathbf{p}_b и \mathbf{p}'_b – положение вершины в базисе $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ до и после трансформации соответственно; \mathbf{n}_v – орт нормали к поверхности тела в точке \mathbf{p} в базисе $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ до трансформации; \mathbf{n}_b – орт нормали к поверхности тела в точке \mathbf{p}_b в базисе $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ до трансформации; \mathbf{n}'_b – орт нормали к поверхности тела в точке \mathbf{p}'_b в базисе $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ после трансформации; \mathbf{n}'_v – орт нормали к поверхности тела в точке \mathbf{p}' в базисе $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ после трансформации; \mathbf{M} – матрица преобразования; \mathbf{M}^{-1} – обратная матрица преобразования.

Матрица преобразования является произведением трех матриц

$$\mathbf{M} = \mathbf{T}\mathbf{S}\mathbf{R}_b, \mathbf{S} = \begin{bmatrix} 1/s_x & 0 & 0 & 0 \\ 0 & 1/s_x & 0 & 0 \\ 0 & 0 & 1/s_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

где \mathbf{T} – матрица сдвига; \mathbf{S} – матрица нормирования (масштабирования); \mathbf{R}_b – матрица поворота.

Рассмотрим подробнее процесс одной трансформации в цепочке трансформаций. Проведем через компонент тела секущую плоскость xOy . Определим процесс трансформации секущей плоскости, заданной уравнением в векторной форме $F(\mathbf{p}_0) = 0$, в секущую поверхность $F'(\mathbf{p}_0') = 0$ как операцию преобразования координат множества принадлежащих ей точек, результатом которой будет являться множество этих же точек с измененными координатами, принадлежащих секущей поверхности. Запишем зависимость между координатами произвольной точки до и после трансформации в векторной форме:

$$\mathbf{p}_0' = \mathbf{f}(\mathbf{p}_0), \quad (3)$$

где \mathbf{p}_0 – точка до трансформации; \mathbf{p}_0' – точка после трансформации;

$$\mathbf{p}_0 = (x_0; y_0; 0)^T; \quad \mathbf{p}_0' = (x_0'; y_0'; z_0')^T;$$

\mathbf{f} – функция трансформации.

Проведем на секущей плоскости xOy одну или несколько кривых. На операцию трансформации наложим условие неизменности длины любой дуги заданных кривых при трансформации секущей плоскости в секущую поверхность. Запишем уравнение кривой i на секущей плоскости xOy в параметрическом виде:

$$\mathbf{L}_0(\lambda) = (x_i(\lambda); y_i(\lambda); 0)^T, \quad (4)$$

где $x_i(\lambda)$, $y_i(\lambda)$ – функции параметра λ . Тогда для той же кривой i на секущей поверхности после трансформации получаем

$$\mathbf{L}_0'(\lambda) = \mathbf{f}(\mathbf{L}_0(\lambda)). \quad (5)$$

С учетом условия неизменности длины любой дуги кривой до (4) и после (5) трансформации получаем

$$|d\mathbf{L}_0(\lambda)/d\lambda| = |d\mathbf{f}(\mathbf{L}_0(\lambda))/d\lambda|. \quad (6)$$

Таким образом, задача сводится к поиску функции \mathbf{f} , удовлетворяющей условию (6) для всех заданных кривых.

Координаты точек, не лежащих на секущей поверхности (находящихся по обе стороны от секущей поверхности), вычислим по формуле

$$\mathbf{p}' = \mathbf{p}_0' + z\mathbf{n}, \quad (7)$$

где \mathbf{n} – орт нормали к секущей поверхности после трансформации в точке \mathbf{p}_0' .

Поскольку z – координата \mathbf{p}_0 всегда нулевая, получаем

$$\mathbf{n} = \left(\frac{\partial \mathbf{f}(\mathbf{p}_0)}{\partial x} \times \frac{\partial \mathbf{f}(\mathbf{p}_0)}{\partial y} \right) / \left| \frac{\partial \mathbf{f}(\mathbf{p}_0)}{\partial x} \times \frac{\partial \mathbf{f}(\mathbf{p}_0)}{\partial y} \right|,$$

где \times – операция векторного произведения.

Таким образом, применив для всех вершин мягкого тела в недеформированном состоянии выражение (7), мы можем найти координаты всех вершин после деформации тела.

Кроме трансформации координат вершин для корректной визуализации деформированного тела с учетом его освещенности необходимо трансформировать и нормали к его поверхности. То есть нужно найти

$$\mathbf{n}_v' = \mathbf{R}\mathbf{n}_v, \quad (8)$$

где \mathbf{R} – матрица поворота.

Вычислим матрицу поворота \mathbf{R} . Рассмотрим частный случай, когда нормаль к секущей поверхности $\mathbf{n} = \mathbf{e}_z$. Тогда поворот не осуществляется, а матрица поворота

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

В остальных случаях матрица поворота вокруг единичного вектора \mathbf{e}_r на угол θ по часовой стрелке при обзоре по направлению \mathbf{e}_r , $\theta \in [0 \dots \pi]$ [10]

$$\mathbf{R} = \begin{bmatrix} \cos \theta + (1 - \cos \theta)e_{rx}^2 & (1 - \cos \theta)e_{rx}e_{ry} - (\sin \theta)e_{rz} & (1 - \cos \theta)e_{rx}e_{rz} + (\sin \theta)e_{ry} \\ (1 - \cos \theta)e_{ry}e_{rx} + (\sin \theta)e_{rz} & \cos \theta + (1 - \cos \theta)e_{ry}^2 & (1 - \cos \theta)e_{ry}e_{rz} - (\sin \theta)e_{rx} \\ (1 - \cos \theta)e_{rz}e_{rx} - (\sin \theta)e_{ry} & (1 - \cos \theta)e_{rz}e_{ry} + (\sin \theta)e_{rx} & \cos \theta + (1 - \cos \theta)e_{rz}^2 \end{bmatrix},$$

где e_{rx} , e_{ry} , e_{rz} – проекции вектора \mathbf{e}_r на оси координат.

Косинус и синус угла между $\mathbf{n} = (n_x, n_y, n_z)^T$ и \mathbf{e}_z находятся по формулам:

$$\cos \theta = \mathbf{n} \mathbf{e}_z = n_z, \quad \sin \theta = \sqrt{1 - n_z^2},$$

а ось вращения задается единичным вектором

$$\mathbf{e}_r = \mathbf{r} / r, \quad \mathbf{r} = \mathbf{n} \times \mathbf{e}_z = (n_y; -n_x; 0)^T,$$

где r – модуль вектора \mathbf{r} .

Тогда

$$\mathbf{R} = \begin{bmatrix} n_z + (1 - n_z)n_y^2 / r^2 & -(1 - n_z)n_y n_x / r^2 & -n_x \sqrt{1 - n_z^2} / r \\ -(1 - n_z)n_y n_x / r^2 & n_z + (1 - n_z)n_x^2 / r^2 & -n_y \sqrt{1 - n_z^2} / r \\ n_x \sqrt{1 - n_z^2} / r & n_y \sqrt{1 - n_z^2} / r & n_z \end{bmatrix}. \quad (9)$$

Применим (7) и (8) для каждой вершины пространственной сетки компонента (тела), найдем новые значения положений всех вершин и всех нормалей в результате трансформации.

В случае нескольких последовательных трансформаций (цепочки трансформаций) необходимо выполнить данный расчет для каждой трансформации. При этом выходные значения положения и нормалей вершин предыдущей трансформации являются входными для следующей и промежуточными – для всей цепочки трансформаций. Для сохранения промежуточных значений вершин в ГПУ предусмотрен механизм организации потоков. При помощи потоков можно перенаправлять выходные данные, формируемые вершинным шейдером, на вход того же (или другого) вершинного шейдера для дальнейшей их обработки, изменяя параметры обработки.

Результаты и их обсуждение

Предложенный метод можно применять при визуализации рыбы в процессе движения в установке замкнутого водоснабжения (УЗВ). При реализации большинства деформаций тела рыбы можно ограничиться двумя уровнями иерархии. На верхнем уровне деформируется (изгибается) все тело. На нижнем – по отдельности плавники, жаберные крышки, нижняя челюсть и др. Далее рассмотрим реализацию только верхнего уровня иерархии – изгиба тела рыбы. Нижний уровень реализуются аналогично, но по другим параметрическим уравнениям плоскости деформации.

Произведем преобразование (2), после которого в локальной системе координат с базисом \mathbf{e}_x , \mathbf{e}_y , \mathbf{e}_z тело рыбы будет симметрично пересекаться секущей плоскостью xOy , линия

хребта совпадать с осью Ox , голова направлена по направлению оси Ox , а неподвижная при изгибе точка на хребте находится в начале координат (см. рис. 3).

Нормы базисных векторов определим согласно (1). В результате длина рыбы будет приведена к 1 с сохранением пропорций (см. размерные линии на рис. 3).

Зададим функции параметра λ для уравнения кривой на секущей поверхности до трансформации (в данном примере будем использовать одну кривую, $i = 1$):

$$x_1(\lambda) = \lambda, \quad y_1(\lambda) = 0. \quad (10)$$

Подставив (10) в (4) при $\lambda = x$, получаем

$$\mathbf{L}_0(\lambda) = (x; 0; 0)^T. \quad (11)$$

Вычислим левую часть уравнения (6):

$$|d\mathbf{L}_0(\lambda) / d\lambda| = |d\mathbf{L}_0(x) / dx| = 1.$$

Зададим функцию трансформации следующим образом:

$$\mathbf{f}(\mathbf{p}_0) = \mathbf{f}\left((x_0; y_0; 0)^T\right) = \left(x_0'; y; mx^2\right)^T, \quad (12)$$

где m – параметр, определяющий степень и направление прогиба рыбы, при имитационном моделировании являющийся функцией времени. В (12) неизвестным компонентом векторной функции \mathbf{f} является x_0' . Для его нахождения решим уравнение (6). Левая часть уравнения посчитана выше и равна 1. Для нахождения правой части (6) подставим (11) в (12):

$$|d\mathbf{f}(\mathbf{L}_0(\lambda)) / d\lambda| = |d\mathbf{f}(\mathbf{L}_0(x)) / dx| = \sqrt{\left(dx_0' / dx\right)^2 + 4m^2x^2}.$$

Возведя в квадрат левую и правую части (6) и выразив дифференциал, получим

$$dx_0' / dx = \pm\sqrt{1 - 4m^2x^2} \quad \text{при } m \in [-0.5 / x_{max} \dots 0.5 / x_{max}], \quad (13)$$

где x_{max} – максимальное по модулю значение x .

Для нахождения неизвестной x_0' проинтегрируем левую и правую части (13) по dx . При точном интегрировании правой части получаем выражение, содержащее тригонометрическую функцию \arcsin , многократное вычисление которой на графическом процессоре достаточно ресурсоемко. Поскольку при визуализации в задачах имитации рыбы в УЗВ не требуется высокой точности, можно проводить приближенное интегрирование, предварительно разложив правую часть (13) в ряд Маклорена. Для ряда 4-го порядка получаем

$$\sqrt{1 - 4m^2x^2} \approx 1 - 2m^2x^2 - 2m^4x^4.$$

Тогда

$$x_0' \approx \pm \int \left(1 - 2m^2x^2 - 2m^4x^4\right) dx = \pm x \left(1 - \frac{2}{3}m^2x^2 - \frac{2}{5}m^4x^4\right) + c(x), \quad (14)$$

где $c(x)$ – интеграл от остаточного члена разложения.

Для нахождения неизвестной $c(x)$ подставим в (14) значения $x = 0$, $x_0' = 0$ в неподвижной точке, совпадающей с началом координат. Получаем $c(x) = 0$. При условии, что $x_0'x \geq 0$ (до и после изгибной деформации точка находится по одну и ту же сторону от плоскости Oyz), получаем

$$x_0' \approx x \left(1 - \frac{2}{3}m^2x^2 - \frac{2}{5}m^4x^4\right). \quad (15)$$

Подставив (15) в (12), получаем итоговую функцию трансформации

$$\mathbf{f}(\mathbf{p}_0) \approx \left(x \left(1 - \frac{2}{3}m^2x^2 - \frac{2}{5}m^4x^4\right); y; mx^2\right)^T. \quad (16)$$

Для вычисления нормали к секущей поверхности вычислим частные производные

$$\frac{\partial \mathbf{f}(\mathbf{p}_0)}{\partial x} = (1 - 2m^2x^2 - 2m^4x^4; 0; 2mx)^T, \quad \frac{\partial \mathbf{f}(\mathbf{p}_0)}{\partial y} = (0; 1; 0)^T.$$

Тогда векторное произведение

$$\frac{\partial \mathbf{f}(\mathbf{p}_0)}{\partial x} \times \frac{\partial \mathbf{f}(\mathbf{p}_0)}{\partial y} = (-2mx; 0; 1 - 2m^2x^2 - 2m^4x^4)^T,$$

а нормаль к секущей поверхности

$$\mathbf{n} = (-2mx; 0; 1 - 2m^2x^2 - 2m^4x^4)^T / \sqrt{4m^2x^2 + (1 - 2m^2x^2 - 2m^4x^4)^2}. \quad (17)$$

По формулам (7), (16) и (17) находим координаты положений всех вершин, а по формулам (8), (17) и (9) – нормали в вершинах к поверхности тела рыбы.

На рис. 4 показана поверхностная сетка тела рыбы до и после деформации. Красным цветом выделена кривая на секущей поверхности при $m = 0,5$, а зеленым цветом изображено направление нормалей к секущей поверхности в различных ее точках (в начале координат и на границах объекта по оси Ox).

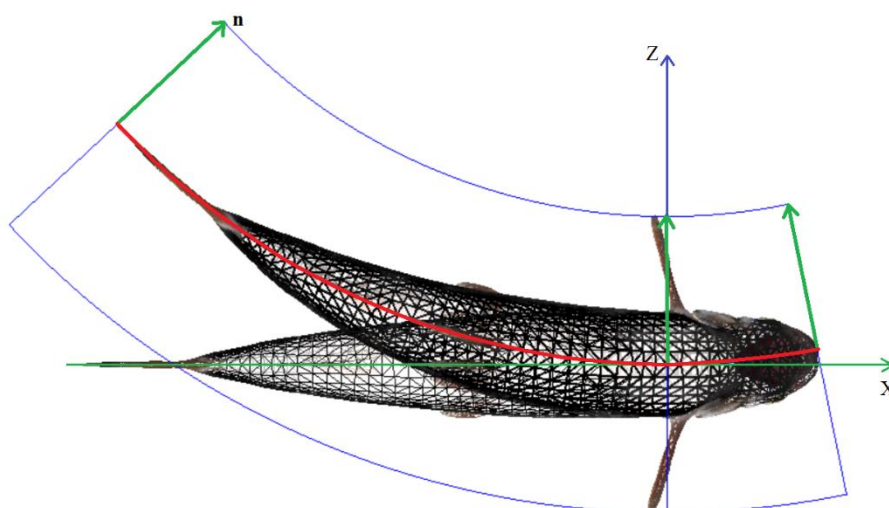


Рис. 4. Деформация рыбы при $m = 0,5$. Составлено авторами
Fig. 4. Fish deformation at $m = 0,5$. Compiled by the authors

На рис. 5 (а, б) показан результат работы алгоритма при различных m , определяющих степень и направление прогиба m .



Рис. 5. Результат визуализации рыбы. Составлено авторами
Fig. 5. Fish visualization result. Compiled by the authors

При визуализации динамических процессов, которыми является движение рыбы, параметр t является функцией от времени.

Заключение

Предложенные математическая база и вычислительный алгоритм деформации статических трехмерных моделей мягких тел, полученных с применением устройств пространственного сканирования, способны видоизменять (деформировать) тело в процессе визуализации на графическом процессоре в режиме реального времени. Деформация проводится с применением секущих поверхностей, описываемых математическими зависимостями, связанными как с параметрами деформации, так и с модельным временем. Количество секущих поверхностей определяется как числом зависимых от времени деформаций его частей, так и степенью требуемой реалистичности визуализации. Незначительные для решаемой задачи деформации можно не проводить, тем самым сократив их количество и уменьшить время расчета одной временной итерации. Для нескольких секущих поверхностей вершины поверхностной сетки визуализируемого тела разбивается на иерархически вложенные подмножества (компоненты), каждый компонент трансформируется с применением отдельной секущей поверхности. При сложных деформациях возможно последовательное применение к одному компоненту нескольких последовательных трансформаций.

Вычислительный алгоритм работает на графическом процессоре, являясь частью конвейера рендеринга. В статье приведен пример вершинного шейдера, написанного на языке программирования шейдеров HLSL для DirectX. Для OpenGL шейдер пишется аналогично.

Предложенный метод можно применять для широкого круга задач виртуальной и дополненной реальности. В частности, данный метод будет применен для визуализации поведения рыб в виртуальной установке замкнутого водоснабжения (УЗВ) в рамках выполнения научного проекта «Разработка математических моделей, технологий дополненной реальности и машинного зрения для задач автоматизации сбора данных в рыбоводных хозяйствах».

Представленные математическая модель и алгоритм были реализованы в виде программного модуля для разрабатываемого в ООО «ЛЦТ» с участием авторов статьи программного обеспечения «Система автоматизированного проектирования орудий рыболовства и технических средств аквакультуры» (САПР-ОР). Программное обеспечение разрабатывается с применением инструментария Embarcadero RAD Studio [11] на языках программирования C++ и HLSL.

Список источников

1. Рожков С. Н., Овсянникова Н. А. Стереоскопия в кино-, фото-, видеотехнике. Терминологический словарь. М. : Парадиз, 2003. 136 с. ISBN 5-98547-003-2.
2. Котельева А. В., Барсов В. В. Проблемы и перспективы дополненной реальности // Информационные системы и технологии: фундаментальные и прикладные исследования. 2017. Т. 1, № 1. С. 454–457.
3. D. Luebke, Mark J. Harris. (2004) GPGPU: general purpose computation on graphics hardware. SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes. doi:10.1145/1103900.1103933.
4. Michael B. Timmons and James B. Ebeling (2013). Recirculating Aquaculture (3rd ed.). Ithaca Publishing Company Publishers. ISBN 978-0971264656.
5. Недоступ А. А., Альтшуль Б. А., Ражев А. О., Дятченко С. В., Бедарева О. М., Багрова А. А. Математическое моделирование поведенческих характеристик стаи рыб при облове разнотлубинным тралом // Морские интеллектуальные технологии. 2019. Т. 4, № 4(46). С. 181–185.

6. Gun-Ho LEE, Chun Woo LEE, Young-Bong KIM, Pingguo He. (2007) Simulating the behavior of Atlantic mackerel and their capture by pelagic trawls / 8th International workshop - Methods for the Development and Evaluation of Maritime Technologies DEMAT 2007. Vol. 5. Germany. Rostock. P. 285–295.
7. Lindner P., Wanielik G. (2009). 3D LIDAR Processing for Vehicle Safety and Environment Recognition. IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems. ISBN 978-1-4244-2770-3. DOI:10.1109/CIVVS.2009.4938725.
8. Hansard Miles, Lee Seungkyu, Choi Ouk, Horaud Radu (2012). Time-of-flight cameras: Principles, Methods and Applications. SpringerBriefs in Computer Science. ISBN 978-1-4471-4657-5. doi:10.1007/978-1-4471-4658-2.
9. Бухтяк М. С., Никульчиков А. В. Моделирование деформации сотовой панели // Вестник Томского государственного университета. Математика и механика. 2013. № 2(22). С. 5–16.
10. Климчик А. С., Гомолицкий Р. И., Фурман Ф. В., Сёмкин К. И. Разработка управляющих программ промышленных роботов : курс лекций. Минск : Белорусский государственный университет информатики и радиоэлектроники, 2008. 131 с.
11. Копылов А. З., Лычагин Ю. В., Осипов В. И. Среда разработки ПО Embarcadero RAD Studio 10.3 Rio: практ. пособие. СПб. : Балт. гос. техн. ун-т, 2019. 27 с.

Сведения об авторах

А. О. Ражев – кандидат технических наук, генеральный директор, главный инженер Лаборатории цифровых технологий, ведущий научный сотрудник.

А. А. Недоступ – кандидат технических наук, доцент, заведующий кафедрой промышленного рыболовства.

Л. А. Кондрашов – инженер Лаборатории цифровых технологий.

Information about the authors

A. O. Razhev – PhD, General Director, Chief Engineer of DigiTech Laboratory; Leading Researcher.

A. A. Nedostup – PhD, Associate Professor, Head of the Department of Commercial Fisheries.

L. A. Kondrashov – Engineer of DigiTech Laboratory.

Статья поступила в редакцию 04.03.2024; одобрена после рецензирования 10.09.2024; принята к публикации 18.09.2024.

The article was submitted 04.03.2024; approved after reviewing 10.09.2024; accepted for publication 18.09.2024.