

Проблемы и особенности визуализации транспортных потоков

О.П. Бобровская^{1,A,B}, Т.В. Гавриленко^{2,A,B}, В.А. Галкин^{3,A,B}

^A БУ ВО «Сургутский государственный университет»

^B Сургутский филиал НИЦ "Курчатовский институт" – НИИСИ

¹ ORCID: 0000-0001-7045-9085, o-bobrovskaya@mail.ru

² ORCID: 0000-0002-3243-2751, taras.gavrilenko@gmail.com

³ ORCID: 0000-0002-9721-4026, val-gal@yandex.ru

Аннотация

Представлена визуализация движения транспорта на плоскости и в 3-х мерном пространстве. Приведены примеры реализации с использованием фреймворков и библиотек (WF, WPF, OpenVDB+Blender3D, Manim). В качестве траектории движения рассматривалась окружность, состоящая из полос в двумерном варианте и коридоров – в трехмерном. Рассмотрен способ задания произвольной траектории для движения агентов в модели транспортного потока, основанной на потенциале действия. Приведены формулы, позволяющие учитывать смещения, как самих коридоров траектории, так и положения агентов в них. Это позволяет правильно обрабатывать перестроения агентов между траекториями и реализовать визуализацию результатов моделирования.

Ключевые слова: транспорт, визуализация данных, транспортный поток, кватернион, OpenVDB.

1. Введение

Моделирование любого процесса - это только одна составляющая исследования. Ключевой проблемой является процесс представления результатов для их последующей оценки и сопоставления.

Для визуализации можно использовать различные методы от статических до анимации, в зависимости от потребностей. При увеличении в процессе исследований количества рассматриваемых объектов возникают следующие сложности [1]:

- возрастают затраты ресурсов на их обработку и вывод на экран;
- увеличивается объем хранимых объектов;
- снижается информативность визуального представления модели, перегруженного объектами;
- ухудшается качество получаемой визуализации, вмещающей больший рассматриваемый объем;
- увеличивается сложность реализации, обеспечивающей масштабирование и интерактивное взаимодействие.

В данном исследовании рассматривается моделирование транспортных потоков, которое позволяет решать задачи управления потоком, выбора оптимальной конфигурации [2], планирования времени перемещения. Как отмечено в работе [3], компьютерное моделирование может значительно сократить время анализа данных, а визуализация - дать наглядное представление о происходящем процессе. Рассмотрим подходы к решению задачи визуального представления модели.

На плоскости задача визуализации заключается в наглядном представлении параметров и свойств транспортных потоков. При изображении автомобилей отсутствуют

проблемы: объекты не могут пересекаться. Сложности могут возникать при большом количестве агентов, так как изображение становится перегруженным для восприятия и тяжелым для обработки компьютером. В таком случае можно перейти с уровня микро-моделирования на мезо- или макро-моделирование [4], и изображать объектами не отдельные автомобили, а их набор, или же отказаться от этого изображения вовсе, заменив его на числовые характеристики.

Усложнение транспортной сети привело к тому, что транспортная инфраструктура расширяется не только на плоскости, но и в 3-м измерении, что приводит к сложности изображения тоннелей и многоуровневых развязок [5], поскольку части дороги будут перекрывать друг друга. Для решения можно ввести полупрозрачность верхнего уровня или выбор текущего уровня изображения.

По-разному может решаться проблема соотношения реалистичности и производительности расчета модели, которая возникает при большом количестве агентов. В работе [6] для изображения автомобилей на дороге используются графические примитивы – окружности заданного цвета, поскольку этого достаточно в рамках задачи наглядной демонстрации перемещения автомобилей по сети. В свободно распространяемом приложении для моделирования SUMO пользователю позволяет в зависимости от текущей задачи выбирать степень детализации изображения агентов [7]: будут ли это примитивные треугольники, передвигающиеся по схематично обозначенной дороге на белом фоне, более близкие к реальности прямоугольники или практически реалистичные фигурки автомобилей на фоне текстур окружающего мира.

В 3-х мерном пространстве значительно сложнее представить поток, и наглядно изобразить происходящие в нем перемещения отдельных автомобилей, так как значительная часть агентов перекрывается теми, что находятся ближе к положению наблюдателя. Кроме того, как отмечалось в работе [8], при увеличении количества атрибутов/измерений, в данном случае, размерности пространства, в котором происходит движение агентов, уменьшается точность суждений наблюдателя о каждом из имеющихся атрибутов в отдельности.

Даже изображение одного объекта и его движения в 3-х мерном пространстве вызывает трудности. Обычно ограничиваются изображением систем координат (неподвижной и связанной с летательным аппаратом), траектории одного объекта или их проекций на плоскость. Передать взаимодействие в пространстве нескольких объектов сложнее, особенно в виде неподвижного изображения. Проще понять пространственное расположение объектов и их перемещение в нескольких последовательных кадрах.

Относительно небольшое количество агентов можно изобразить детально как есть, и при этом понять, как они расположены, поворачивая изображение вокруг нескольких осей. Однако увеличение количества агентов приведет к увеличению когнитивной нагрузки и усложнит задачу восприятия и анализа. Исследования визуального внимания показывают, что даже для восприятия и оценки количества точек на изображении требуется время, и это время увеличивается с ростом количества точек [9]. Таким образом, если агентов будет хотя бы больше 1000, то воспринимать по-отдельности их самих и их траектории будет проблематично.

Одним из решений проблемы перекрывания агентами друг друга, если их достаточно много, может являться изображение агентов в виде потока (жидкости или облака газа). Если жидкость непрозрачна, тогда можно наблюдать только находящиеся на внешней границе объекты. Но чаще жидкость частично прозрачна и изображается как множество малых примитивных объектов (куб, шар). При сохранении в памяти и последующего изображения непосредственно каждого объекта, для обработки их большого объема можно воспользоваться специализированными средствами, вроде библиотеки OpenVDB, работающей с разреженными объектами (изначально дым и пламя).

Или же можно изображать только макропараметры транспортного потока, то есть использовать трехмерную визуализацию скалярных полей (в нашем случае это поля,

представляющие плотность или модуль скорости объектов). Распространенными являются следующие методы:

- изоповерхности, поверхности уровня – поверхности постоянного значения скалярного поля. Для извлечения поверхности уровня из поля необходимо в ячейках скалярной сетки вычислять точки с заданным постоянным значением поля [10].

- объемный рендеринг [11] – набор методов, представляющих трехмерный набор данных в виде двумерного изображения (например, метод проекции максимальной интенсивности).

. Используемая модель транспортного потока

Имеется модель транспортного потока, основанная на потенциале действия [12], в которой автомобили представлены частицами, притягивающимися или отталкивающимися друг от друга в зависимости от расстояния между ними. Далее будет рассмотрено, как можно визуально представить данные, получаемые в процессе работы этой простой модели, а также ее обобщения на 3-х мерное пространство. Траектория для движения агентов была строго задана окружностью с несколькими коридорами, в которые возможны перестроения.

Целью исследования является обобщение модели путем добавления возможности задавать произвольную траекторию набором точек излома некоторой ломаной.

Решение поставленной задачи позволит расширить возможности модели транспортного потока, что является актуальным ввиду современного состояния транспортной сети и необходимости искать решения по увеличению пропускной способности дорог, и, соответственно, моделировать транспортный поток. Кроме того, развитие беспилотного транспорта ставит перед нами задачу анализа путей внедрения новых типов автомобилей в существующую транспортную систему, что также решается с использованием моделирования.

3. Подходы к визуализации

В зависимости от цели визуализации можно выделить несколько направлений.

Если необходимо визуально представить движение транспортного потока на двумерной плоскости, то его можно изобразить последовательностью сменяющихся с заданной частотой изображений положения всех автомобилей на модели пространства. Для этого подойдет любое программное средство, позволяющее изображать графические примитивы и сохранять их в виде изображения.

В случае изображение потока в 3-х мерном пространстве, предпочтительнее использовать программы, позволяющие работать с объемными объектами, как в статике, так и в динамике.

OpenVDB позволяет создать Grid (сетку) точек с заданной плотностью и сохранить файл с расширением “.vdb”, который открывается в приложении Blender. Еще один Grid тех же точек, позволяет задать затем в блендере цвет для этих точек. Таким образом можно сохранять значение скорости агентов, после чего отображать цветами заторы, как на сайте М. Treiber (<https://traffic-simulation.de/ring.html>). Рендер-движки Blender: 1) Eevee (быстрее); 2) Cycles (больше объектов можно показать).

Manim – удобный инструмент для создания анимации в хорошем качестве с использованием кода Python. Позволяет изображать 3D объекты.

Если у нас нет времени следить за движением многих объектов, то удобно изобразить их обобщенные, агрегированные характеристики на графиках:

- 1) зависимость положения на отрезке дороги от времени (см. рис. 1);
- 2) фундаментальная диаграмма;
- 3) обобщенные характеристики потока (скорость, количество перестроений), зависящие от времени.
- 4) термодинамические характеристики, в частности, температура, как в работе [13].

Еще существуют способы визуализации геопространственных данных для более общего изображения перемещения потоков транспорта, связанных с логистикой.

4. Возможные варианты реализации

Как уже упоминалось, имеется математическая модель транспортного потока, которую требуется визуализировать. Пока траектория движения потока представляет замкнутую линию без пересечений или с одним пересечением (одномерное многообразие). Для транспортных средств заданы правила поведения, согласно которым они взаимодействуют между собой. Поэтому будем рассматривать визуализацию, подходящую для микроскопического моделирования (изображения отдельных агентов).

4.1. Windows Forms (C#).

Инструмент разработки графического интерфейса, который продолжают поддерживать, но перестали развивать в связи с выходом WPF. Имеется возможность выводить в некоторую область графические примитивы, изображать графики, использовать элементы управления, чтобы выполнять команды пользователя в реальном времени. С использованием описанных возможностей было реализовано следующее (рис. 1):

- В поле слева происходит моделирование движения автомобилей, представленных в виде цветных окружностей.
- Были определены параметры модели, которые доступны пользователю для изменения (количество автомобилей, диаметр трека, ограничение скорости, количество полос (одна или две), форма трека (окружность или восьмерка), количество автопилотов среди всего числа автомобилей, коэффициенты отталкивания и притяжения).
- На графики в реальном времени выводятся зависимость координаты на треке от времени и фундаментальная диаграмма.

Для визуализации движения автомобилей использовался таймер. Каждые 0,001с происходит расчет нового положения автомобилей, старое изображение удаляется и на его месте выводится новое.

В реализации рассматриваемой модели большинство описанных ранее проблем не наблюдаются из-за малых масштабов моделирования и количества конфигураций дорог.

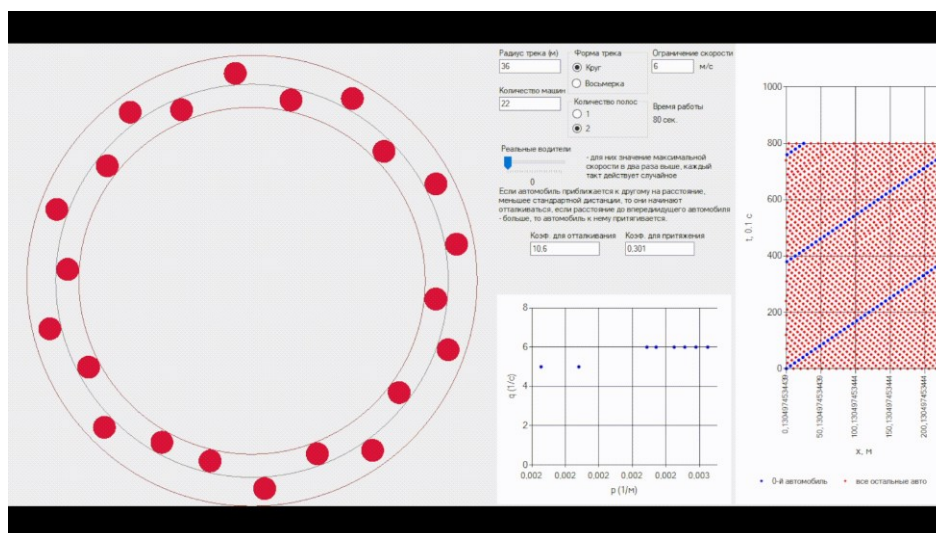


Рис. 1. Интерфейс приложения, созданного в WF

При увеличении диаметра трека (в связи с большим количеством автомобилей) масштаб уменьшается, поэтому начиная с некоторого диаметра автомобили отображаются мелко. Кроме того, время отклика приложения увеличивается при большом ко-

личестве агентов или продолжительном моделировании (поскольку старые значения не удаляются с графиков).

Из недостатков WF стоит отметить отсутствие возможности изображения 3D.

4.2. NET Core WPF.

Более современный и многофункциональный инструмент позволяет реализовать приведенный выше интерфейс приложения. При этом предоставляет возможность изображать трехмерное пространство и его проекцию на плоскость камеры, что является необходимым для дальнейшего расширения модели транспортного потока на 3-е измерение.

Особенности представления объемных объектов делают работу с этим инструментом сложной. На рис. 2 представлены агенты, движущиеся в пространстве по нескольким параллельно расположенным замкнутым окружностям с разных углов обзора.



Рис. 2. Движение агентов (тетраэдров) по окружности в WPF

Тетраэдры – это, в данном случае, графические примитивы в WPF, которые использовались из-за отсутствия сфер. Соответственно, появляется задача ориентации агентов по пути следования, которая не стоит для окружностей и сфер и не была решена ранее, поэтому требует доработок алгоритма.

Визуализацию требуемого движения агентов возможно реализовать только с использованием поворотов относительно оси, проходящей через центры окружностей коридоров, что является ad hoc решением и будет препятствовать будущему расширению модели. Поэтому данный инструмент не будет использоваться в дальнейшем.

4.3. OpenVDB.

Открыто распространяемая библиотека, написанная на C++, позволяет сохранять пространственные положения объектов. Затем сохраненные таким образом файлы можно импортировать в Blender3D. Модель транспортного потока была реализована на языке python, для которого так же доступна библиотека OpenVDB.

Настроить работу OpenVDB в Windows не удалось, поэтому работа осуществлялась в Debian, развернутом на WSL (Windows Subsystem for Linux). Установка согласно [https://github.com/AcademySoftwareFoundation/opensubdiv] и сборка библиотеки для Python3 (OPENVDB_BUILD_PYTHON_MODULE=ON

[https://www.openvdb.org/documentation/doxygen/build.html]). Затем, простой импорт библиотеки pyopenvdb позволяет использовать ее возможности в Python.

```
import pyopenvdb as vdb
```



```

for j in range(1200): # время моделирования 2 мин
    ...
    grid1 = vdb.FloatGrid() # создание сетки
    grid2 = vdb.FloatGrid()
    grid1.name = 'density' # присваивание имени сетке
    grid2.name = 'temperature'
    acc1 = grid1.getAccessor() # получение (аксесора/акцессора) указателя на сетку,
    обеспечивающий возможность изменения вокселей
    acc2 = grid2.getAccessor()
    for i in range(N): # цикл по всем объектам
        ijk = (int(x), int(y), int(z)) # кортеж координат объекта
        acc1.setValueOn(ijk, 0.95) # присвоение вокселю сетки по заданным коор-
        динатам нужного значения
        acc2.setValueOn(ijk, (v/speed_limit))
    vdb.write(f'../../mnt/c/Users/user/project_openvdb/track/mygrids{j}.vdb',
    grids=[grid1, grid2]) # сохранение сетки в файл

```

В процессе расчетов новых положений агентов, осуществлялось сохранение набора их координат в файлы с расширением «.vdb». Вторая, параллельно сохраняемая сетка содержит значения скоростей агентов. После окончания расчетов 2-х минутной симуляции производится импорт файлов (в количестве 1200 шт.) в приложение Blender в качестве нового объекта на сцене. Для отображения скоростей агентов в виде цвета сфер были добавлены ноды (node) в редакторе шейдеров. Затем осуществлен рендеринг видео (возможен рендеринг набора отдельных картинок – рис. 3, с последующей сборкой) и его сохранение.

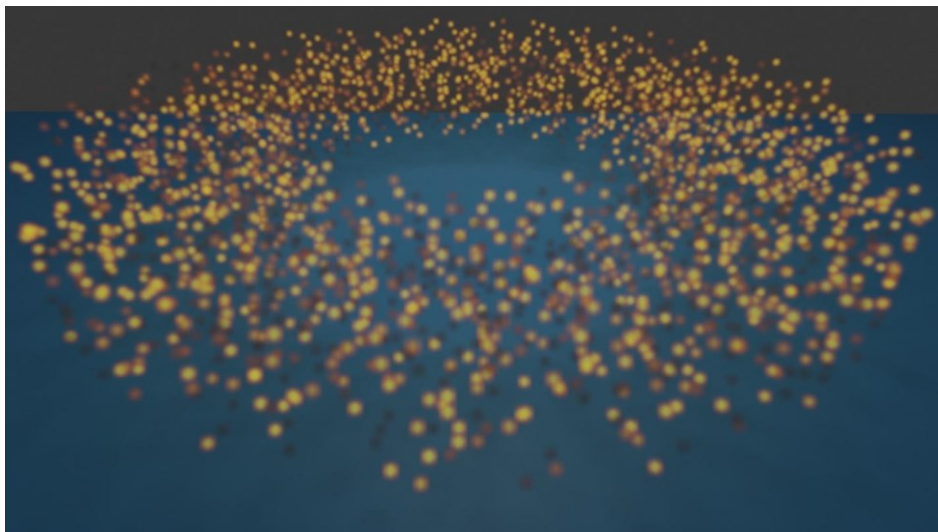


Рис. 3. 4000 агентов на 100 коридорах. Изображение получено с помощью Blender

После знакомства с OpenVDB решение задачи сохранения положения множества частиц не является сложным (требует пары строчек кода). Использование Blender3D для наложения цвета и визуализации движения частиц также не составляет труда (в сети достаточно информации). Затруднения вызывает повышение качества изображения.

4.4. Manim.

Бесплатная открыто распространяемая python библиотека (<https://www.manim.community/>) для создания анимации в области математики предоставляет возможность работать с объектами, имеющими 3 измерения. Для этого имеется специальный класс 3D сцены и объекты, которые можно на ней располагать. После довольно удобного интерактивного перемещения камеры в blender, может оказаться

сложно управлять камерой с помощью двух углов и без возможности задать дистанцию от камеры до начала координат. С другой стороны, в Manim имеется возможность размещать объекты не только в целочисленных координатах, но и на любых действительных.

Программа, написанная с использованием данной библиотеки, позволяет использовать заранее подготовленные данные (в данном случае, координаты агентов, сохраняемые в процессе расчета модели в текстовый файл) для визуализации (рис. 4):

```
from manim import *

class Agents3D(ThreeDScene): # объявление класса, наследуемого от 3-х мерной сцены
    def construct(self):
        self.set_camera_orientation(phi=50*DEGREES, theta=-45*DEGREES) #
        # определение положения камеры
        ...
        for i in range(N): # добавление всех частиц в список и на сцену
            dots.append(Dot3D().shift(UP*x+LEFT*y+OUT*z).set_color(BLUE).scale(0.5))
        for j in range(1199*N): # цикл по каждому объекту в каждый момент времени (2мин)
            temp_coords.append(dots[itr%N].animate.move_to(UP*x+LEFT*y+OUT*z)) # добавление объекту анимации перемещения и сохранение их в список
            if j % N == N-1: # если все объекты за данную временную точку обработаны
                self.play(*temp_coords, run_time=0.05) # вывод на экран всех анимаций перемещения
                temp_coords.clear()
        self.wait(1)
```

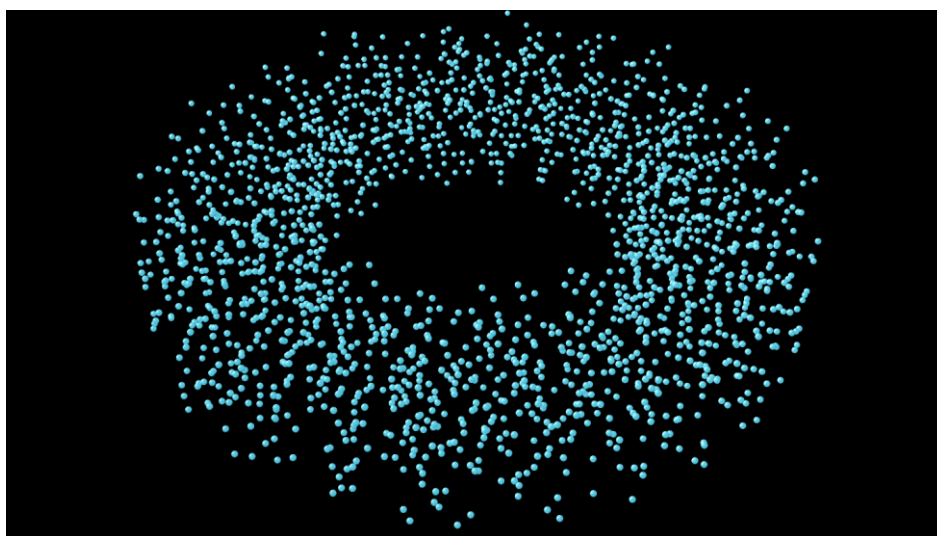


Рис. 4. 2000 агентов на 100 коридорах. Изображение получено с помощью Manim

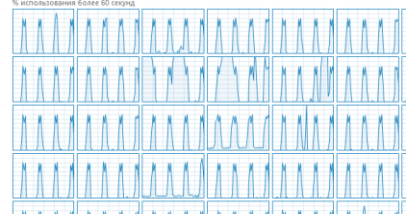
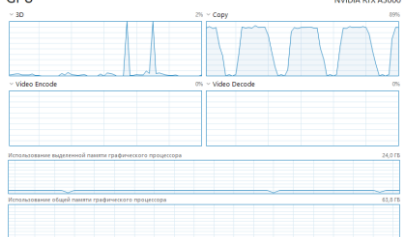
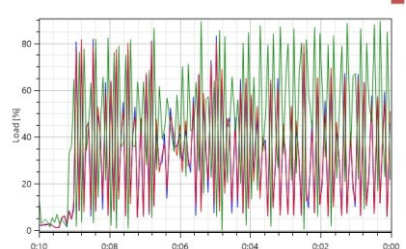
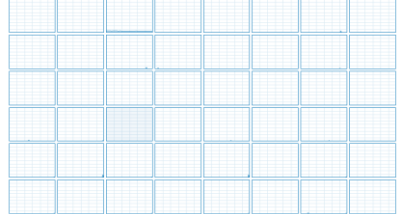
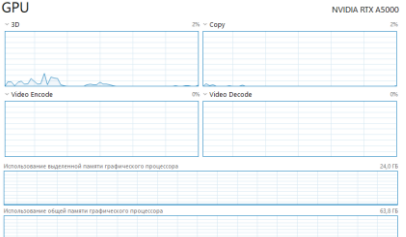
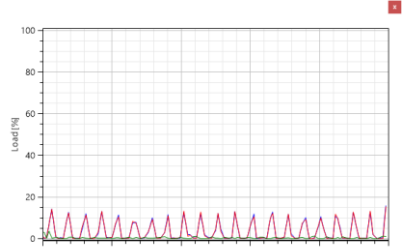
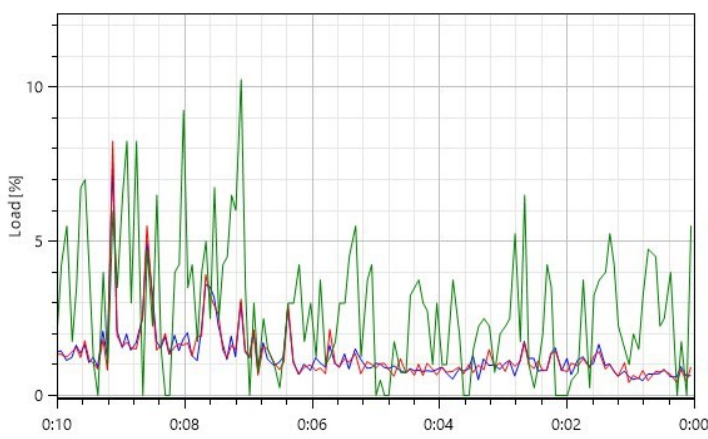
Визуально агенты кажутся более четкими, но и менее реалистичными, чем в Blender.



4.5. Сравнение

Для сравнения двух последних вариантов по времени и ресурсам был проведен запуск следующего эксперимента. 400 объектов в начальный момент времени расположены на одной полосе замкнутой окружности и имеют начальную скорость 8 м/с. Ра-

диус трека 36м. Количество доступных полос 16, расположенных квадратом 4x4. Время моделирования 2 мин. Результаты приведены в таблице 1.

Таблица 1. Сравнение OpenVDB и Manim

	OpenVDB+Blender3D	Manim
Время рендеринга	6,4 ч	9,3 ч
Затраты ресурсов на процесс	<div><p>ЦП Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz</p><p>GPU NVIDIA RTX A5000</p><p>Использование видеопамяти графического процессора 24,0 Гб</p><p>Использование общей памяти графического процессора 63,8 Гб</p></div>	<div><p>ЦП intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz</p><p>GPU NVIDIA RTX A5000</p><p>Использование видеопамяти графического процессора 24,0 Гб</p><p>Использование общей памяти графического процессора 63,8 Гб</p></div>
Выполнение кода в wsl:	<div></div>	
Размер итогового файла	14,9 МБ	16,0 МБ
Размер файла, хранящего реализуемые координаты точек	Папка с 1200 файлами vdb размером от 87 до 101 Кб Итого 119 МБ	txt на 13,9 МБ
Объем кода	Использовались только элементы управления приложения	20 строк для обработки сохраненных в txt данных

Итоговый файл		
---------------	--	---

Manim субъективно демонстрирует более высокое качество при малом количестве агентов из-за удачных настроек по умолчанию. Но при увеличении числа агентов в Manim значительно увеличивается время отрисовки и ухудшается качество, которое можно улучшить только значительно большим временем обработки.

5. Произвольная траектория

Для обеспечения возможности движения агентов по произвольной траектории необходимо задать генератор, приемник и траекторию, их соединяющую. Для генератора (источка) и приемника (стока) задается интенсивность - количество агентов создающихся и удаляющихся соответственно в единицу времени. Траекторию можно задавать как ломаную линию, указав точки изломов. Для обработки перемещений агентов хранится расстояние, пройденное от начала движения (генератора).

Траектории агентов, расположенных в коридорах, кроме нулевого, отличаются от траектории нулевого коридора длиной (что не учитывалось на окружности): на поворотах происходит увеличение или уменьшение проходимого по отрезку траектории расстояния. Чтобы это отразить в модели были реализованы функции, рассчитывающие растяжение и сжатие отдельных коридоров, в зависимости от их удаленности от нулевого коридора.

Поскольку смещения коридоров зависят от смещений на предыдущем отрезке пути, расчет расположения коридоров будет производиться в начале, при создании траектории. Для этого необходимо определить смещения для первого отрезка, а для всех последующих будет производиться поворот, совпадающий с поворотом направляющего вектора отрезка прямой.

Смещения по относительным осям y (y_offset) и z (z_offset) для первого коридора рассчитываются следующим образом (смещение по y мы хотим осуществить в плоскости XOY).

Они ортогональны направляющему вектору прямой коридора, следовательно:

$$\bar{e} \cdot (x - x_0 \ y - y_0 \ z - z_0) = 0$$

где \bar{e} – единичный направляющий вектор прямой, на которой лежит отрезок траектории.

$$y - y_0 = \frac{e_z z_0 - e_x (x - x_0)}{e_y} + y_0$$

$$x - x_0 = 1$$

$$y - y_0 = \frac{e_z z_0 - e_x}{e_y}$$

$$y_{offset} = \frac{y - y_0}{|y - y_0|}$$

$$z_{offset} = e \times y_{offset}$$

Для последующих коридоров, направления смещения зависят от направлений смещений предыдущего коридора и рассчитываются иначе.

Кватернион поворота рассчитывается как:

$$\Lambda = 1 + (\bar{n}_0, \bar{n}_1) + [\bar{n}_0, \bar{n}_1],$$

где \bar{n}_0, \bar{n}_1 – два заданных вектора, между которыми нужно найти угол поворота [14].

$$\Lambda_{\text{норм.}} = \frac{\Lambda}{|\Lambda|}$$

Этот кватернион однозначно задает поворот по кратчайшему пути, который совершил направляющий вектор отрезка траектории, чтобы превратиться в направляющий вектор следующего отрезка траектории.

Особым случаем является поворот на 180 градусов, при котором $\bar{n}_0 = -\bar{n}_1$. При этом значение $\Lambda = 0$, что обрабатывается описанным ниже образом (произвольно выбирается один из множества подходящих векторов).

Кватернион, описывающий поворот, рассчитывается как:

$$\Lambda = \begin{cases} \frac{\Lambda_{\text{ненорм.}}}{|\Lambda_{\text{ненорм.}}|}, & |\Lambda_{\text{ненорм.}}| \neq 0 \\ \frac{(0; -n_{0z}; n_{0y})^T}{\sqrt{n_{0z}^2 + n_{0y}^2}}, & |\Lambda_{\text{ненорм.}}| = 0, \quad |n_{0x}| < 0.58 \\ \frac{(n_{0z}; 0; -n_{0x})^T}{\sqrt{n_{0z}^2 + n_{0x}^2}}, & |\Lambda_{\text{ненорм.}}| = 0, \quad |n_{0x}| \geq 0.58 \end{cases}$$

Далее используется функция `Rotation.from_quat($\Lambda_{\text{норм.}}$)` библиотеки `scipy (python3)` для поворотов векторов смещения коридоров на тот же угол, в формате `scalar-last (x, y, z, w)`.

В результате, вектора смещения по осям y и z поворачиваются на те же углы, что и направляющий вектор прямой при переходе от одного отрезка траектории к другому.

Растяжение (сжатие) коридоров будет производиться следующим образом. Отрезок траектории делится пополам, и каждая половина деформируется в соответствии со стыком соседнего отрезка траектории. Строятся направляющие векторы, находится точка пересечения, определяется смещение для координаты агента.

Местонахождение агента в нулевом коридоре определяется по формуле:

$$p = e(l - L_i) + T_{r_p},$$

где T – последовательный набор вершин всех точек излома траектории; L_i – кумулятивная длина пройденного пути до отрезка траектории для i -го агента; r_p – индекс правой границы отрезка маршрута, найденный с помощью бинарного поиска.

Определим множитель γ , характеризующий растяжение / сжатие коридора в зависимости от точки пересечения с соседним коридором.

$$\bar{p} + \gamma \bar{e} = \bar{p}' + \gamma' \bar{e}'$$

где p – расположение агента, p' – точка, лежащая на прямой соседнего коридора, e – направляющий вектор прямой рассматриваемого коридора, e' – направляющий вектор для соседнего коридора.

$$\gamma = \frac{e'_y p_x - p'_x e'_y + p'_y e'_x - p_y e'_x}{e_y e'_x - e_x e'_y}$$

При этом x и y условные, и могут заменяться на другую пару координат, в случае нулевого значения y одного из направляющих векторов прямых.

То же используется для расчета измененных протяженностей смещенных коридоров. Это позволяет верно учитывать расположение агентов в соседних коридорах при перестроении.

Результат представлен на рис. 5-8.

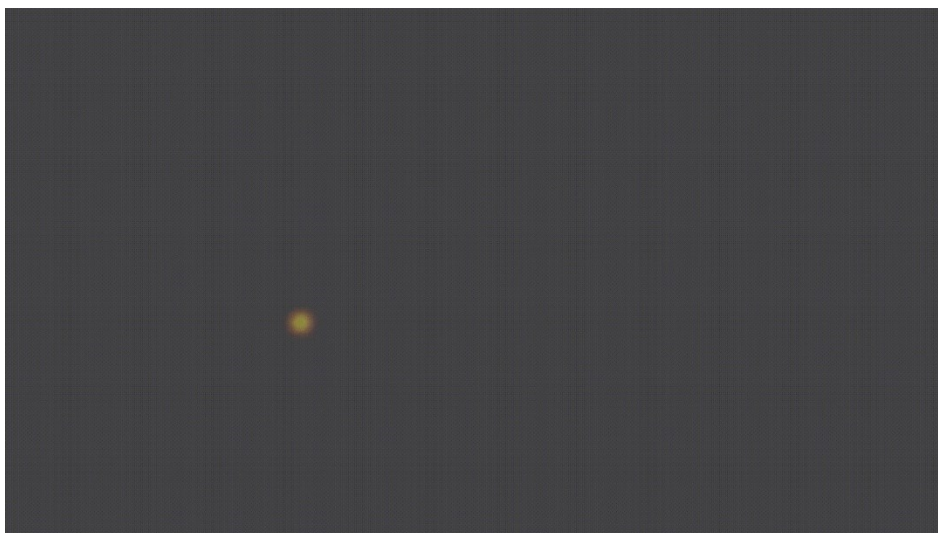


Рис. 5. Пример произвольной траектории 1



Рис. 6. Пример произвольной траектории 2

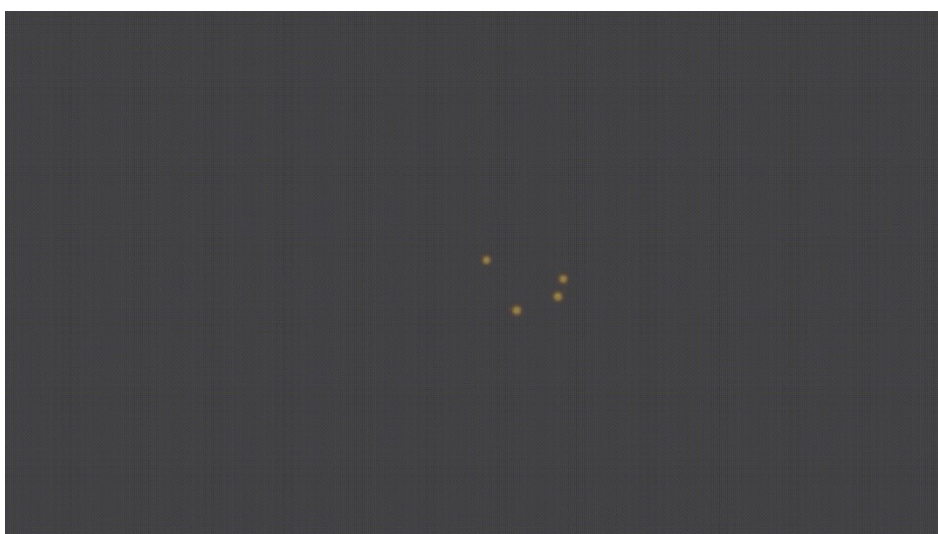


Рис. 7. Пример произвольной траектории 3

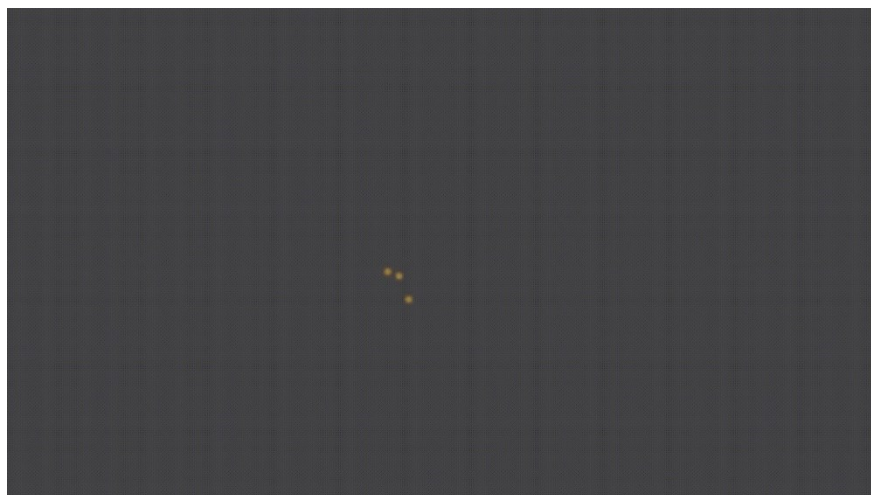


Рис. 8. Пример произвольной траектории 4

На рис. 9 представлена замкнутая произвольная траектория. Принцип работы подобен тому, как это обеспечивалось на окружности: при достижении условного "конца" траектории агент перемещается в начало траектории.

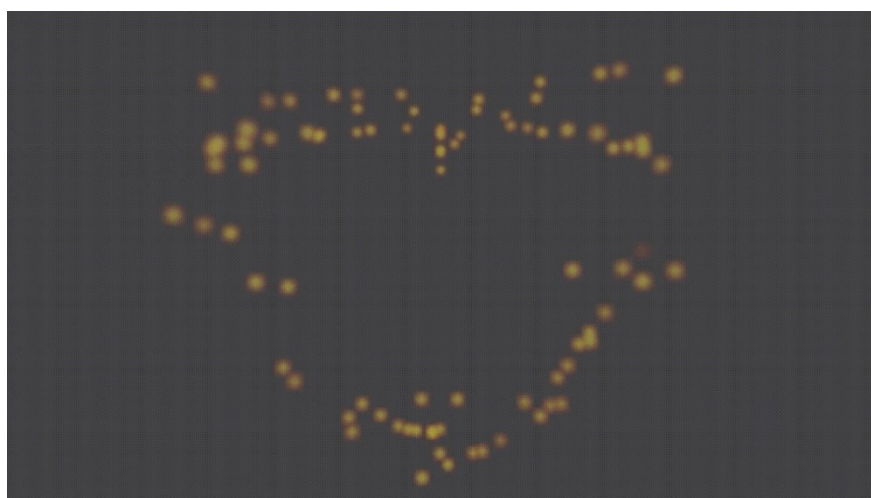


Рис. 9. Замкнутая произвольная траектория

На рис. 10 продемонстрирована возможность выделения отдельного агента в потоке.

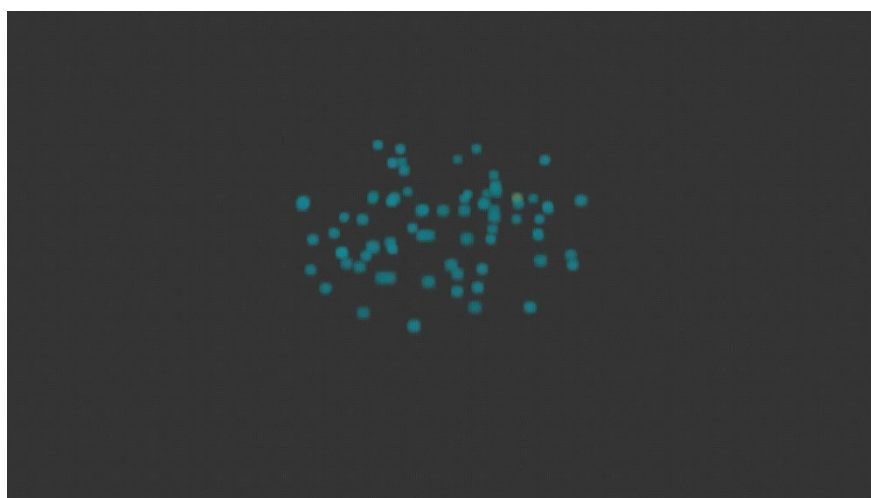


Рис. 10. Выделенный в потоке агент

Это обеспечивается созданием двух дополнительных Grid сеток и настройкой для них другого цвета в Blender.

Вывод

Было рассмотрено несколько способов моделирования транспортных потоков. WinForms и C# могут подойти для небольшой модели двумерного движения. WPF, OpenVDB и Manim подходят для моделирования трехмерных потоков транспорта. Из последних, WPF предназначен больше для статического размещения 3D объектов и их простых трансформаций, нежели для реализации динамики. OpenVDB работает как с C++, так и с Python3, сохраненные с его помощью расположения объектов на сцене можно импортировать в Blender3D для визуализации. Manim позволяет с помощью Python кода осуществить визуализацию движения трехмерных объектов. На небольших моделях выигрывает Manim по объему используемой памяти и затратам ресурсов. Но при увеличении количества моделируемых элементов Manim стремительно увеличивает затраты и проигрывает Blender.

В дальнейшем будет использоваться связка OpenVDB + Blender, позволяющая удобно осуществлять промежуточное редактирование визуального вида и работать с большими моделями.

Рассмотрено моделирование произвольной траектории в пространстве: формулы, обеспечивающие растяжение и сжатие внешних коридоров траектории при поворотах, и кватернионы, позволяющие сохранить относительное расположение агентов в коридорах при повороте.

Имеющаяся модель транспортного потока была расширена возможностью задания произвольной траектории. Были решены вопросы обеспечения правильного относительного расположения коридоров движения агентов в пространстве и расчета их расположения. Значительно расширены возможности модели транспортного потока, обеспечивающие проведение сравнительных экспериментов с большим количеством конфигураций траектории.

Благодарности

Работа выполнена в рамках государственного задания НИЦ «Курчатовский институт» — НИИСИ по теме № FNEF-2024-0001 «Создание и реализация доверенных систем искусственного интеллекта, основанных на новых математических и алгоритмических методах, моделях быстрых вычислений, реализуемых на отечественных вычислительных системах» (1023032100070-3-1.2.1).

Список литературы

1. Ali S.M., Gupta N., Nayak G.K., Lenka R.K. Big data visualization: Tools and challenges // 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), Greater Noida, India, 2016, pp. 656-660. (doi: 10.1109/IC3I.2016.7918044).
2. Ли А.В., Репкин В.С., Семенов Г.Ю., Сермавкин Н.И., Фаерман В.А. Компьютерное моделирование проблемного перекрестка в среде AnyLogic // Известия Томского политехнического университета. Промышленная кибернетика, Т. 1, №2, 2023, С. 1–10, (doi: 10.18799/29495407/2023/2/16).
3. Марухленко П.Г. Трехмерная визуализация движения транспортных потоков на аварийноопасном участке дороги // Горный информационно-аналитический бюллетень (научно-технический журнал), №11, 2016, С. 408-417.
4. Adekunle A. A Review of the Difference Among Macroscopic, Microscopic and Mesoscopic Traffic Models, 2017. (doi: 10.13140/RG.2.2.11508.65929).
5. Котиков Ю.Г., Савченко К.А. 3D-моделирование многоуровневых транспортных развязок на базе платформы ArcGIS // ГИС в меняющемся мире. 2010. №4 (55). URL: <https://arcreview.esri-cis.ru/2010/10/12/3d-modeling/>.

6. Тишкин В.Ф., Трапезникова М.А., Чечина А.А., Чурбанова Н.Г. Моделирование транспортных потоков на основе квазигазодинамического подхода и теории клеточных автоматов с использованием суперкомпьютеров // Компьютерные исследования и моделирование, Т. 16, №1, 2024, С. 175–194 (doi:10.20537/2076-7633-2024-16-1-175-194) (<https://www.mathnet.ru/rus/crm1157>).
7. Panovski D. Simulation, optimization and visualization of transportation data // Networking and Internet Architecture [cs.NI]. Institut Polytechnique de Paris, 2020.
8. Miller G.A. The magical number seven, plus or minus two: Some limits on our capacity for processing information // Psychological Review, Vol. 63, № 2, 1956, pp. 81–97. (doi: 10.1037/h0043158)
9. Kaufman E.L., Lord M.W., Reese T.W., Volkman J. The discrimination of visual number // The American Journal of Psychology, Vol. 62, 1949, pp. 498–525.(doi: 10.2307/1418556)
10. Тимохин П.Ю., Михайлюк М.В. Метод извлечения поверхностей уровня на GPU с помощью программируемой тесселяции // Программирование, № 3, 2020, С. 66-72 (doi: 10.31857/So132347420030103).
11. Xu C., Sun G., Liang R. A survey of volume visualization techniques for feature enhancement // Visual Informatics, Vol. 5, № 3, 2021, pp. 70-81 (doi: 10.1016/j.visinf.2021.08.001).
12. Бобровская О.П., Гавриленко Т.В., Галкин В.А. Модель транспортного потока, основанная на взаимодействии частиц с потенциалом действия // Вестник КРАУНЦ. Физ.-мат. науки, Т. 40, № 3, 2022, С. 72–87. (doi: 10.26117/2079-6641-2022-40-3-72-87).
13. Бобровская О.П., Гавриленко Т.В., Галкин В.А. О применении понятия температуры для описания транспортного потока // Успехи кибернетики, Т. 5, № 2, 2024, С. 26-34. (doi: 10.51790/2712-9942-2024-5-2-03).
14. nabbla1. Статья "Ликбез по кватернионам, часть 6: поворот по кратчайшему пути" // Живой Журнал : платформа для ведения онлайн-дневников. 22 февраля 2018. URL: <https://nabbla1.livejournal.com/187019.html> (дата обращения: 23.01.2025).

Problems and Peculiarities of Visualization of Traffic Flows

O.P. Bobrovskaya^{1,A,B}, T.V. Gavrilenko^{2,A,B}, V.A. Galkin^{3,A,B}

^A Surgut State University

^B Surgut Branch of Federal State Institute “Scientific Research Institute for System Analysis of the Russian Academy of Sciences”

¹ ORCID: 0000-0001-7045-9085, o-bobrovskaya@mail.ru

² ORCID: 0000-0002-3243-2751, taras.gavrilenko@gmail.com

³ ORCID: 0000-0002-9721-4026, val-gal@yandex.ru

Abstract

Visualization of transport movement on the plane and in 3-dimensional space is presented. Examples of implementation using frameworks and libraries (WF, WPF, OpenVDB+Blender3D, Manim). A circle consisting of lanes in two-dimensional version and corridors in three-dimensional version was considered as a motion trajectory. The method of setting an arbitrary trajectory for the movement of agents in the model of transportation flow based on the action potential is considered. The formulas allowing to take into account the displacements of both the corridors of the trajectory and the positions of the agents in them are given. This allows us to correctly handle the rearrangements of agents between trajectories, as well as to visualize the simulation results.

Keywords: transport, data visualization, transport flow, rotation quaternion, OpenVDB.

References

1. Ali S.M., Gupta N., Nayak G.K., Lenka R.K. Big data visualization: Tools and challenges // 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), Greater Noida, India, 2016, pp. 656-660. (doi: 10.1109/IC3I.2016.7918044).
2. Li A.V., Repkin V.S., Semenov G.Yu., Sermavkin N.I., Faerman V.A. Komp'yuternoye modelirovaniye problemnogo perekrestka v srede AnyLogic [Computer simulation of a problem crossroad in the anylogic environment] // Bulletin of the Tomsk Polytechnic University. Industrial cybernetics, Vol. 1, №2, 2023, pp. 1–10, (doi: 10.18799/29495407/2023/2/16) [in Russian].
3. Marukhlenko P.G. Trekhmernaya vizualizatsiya dvizheniya transportnykh potokov na avariynopasnom uchastke dorogi [Three-dimensional visualization of traffic flow on the accident-prone road section] // Mining informational and analytical bulletin, №11, 2016, pp. 408-417.
4. Adekunle A. A Review of the Difference Among Macroscopic, Microscopic and Mesoscopic Traffic Models, 2017. (doi: 10.13140/RG.2.2.11508.65929).
5. Котиков Ю.Г., Савченко К.А. 3D-моделирование многоуровневых транспортных развязок на базе платформы ArcGIS // ГИС в меняющемся мире. 2010. №4 (55). URL: <https://arcreview.esri-cis.ru/2010/10/12/3d-modeling/>.
6. Tishkin V.F., Trapeznikova M.A., Chechina A.A., Churbanova N.G. Modelirovaniye transportnykh potokov na osnove kvazigazodinamicheskogo podkhoda i teorii kletochnykh avtomatov s ispol'zovaniyem superkomp'yuterov [Simulation of traffic flows based on the quasi-gasdynamic approach and the cellular automata theory using supercomputers] // Computer Research and Modeling, Vol. 16, №1, 2024, pp. 175–194 (doi:10.20537/2076-7633-2024-16-1-175-194) (<https://www.mathnet.ru/rus/crm1157>) [in Russian].
7. Panovski D. Simulation, optimization and visualization of transportation data // Networking and Internet Architecture [cs.NI]. Institut Polytechnique de Paris, 2020.

8. Miller G.A. The magical number seven, plus or minus two: Some limits on our capacity for processing information // Psychological Review, Vol. 63, № 2, 1956, pp. 81–97. (doi: 10.1037/h0043158)
9. Kaufman E.L., Lord M.W., Reese T.W., Volkman J. The discrimination of visual number // The American Journal of Psychology, Vol. 62, 1949, pp. 498–525. (doi: 10.2307/1418556)
10. Timokhin P.Y., Mikhaylyuk M.V. Metod izvlecheniya poverkhnostey urovnya na GPU s pomoshch'yu programmiruyemoy tesselyatsii [The Method to Extract Isosurfaces on the GPU by Means of Programmable Tessellation] // Programmirovaniye, № 3, 2020, pp. 66-72 (doi: 10.31857/S0132347420030103).
11. Xu C., Sun G., Liang R. A survey of volume visualization techniques for feature enhancement // Visual Informatics, Vol. 5, № 3, 2021, pp. 70-81. (doi: 10.1016/j.visinf.2021.08.001).
12. Bobrovskaya O.P., Gavrilenko T.V., Galkin V.A. Model' transportnogo potoka, osnovannaya na vzaimodeystvii chastits s potentsialom deystviya [Transport flow model based on interaction of particles with action potential] // Vestnik KRAUNC. Fiz.-mat. nauki, Vol. 40, № 3, 2022, pp. 72–87 (doi: 10.26117/2079-6641-2022-40-3-72-87) [in Russian].
13. Bobrovskaya O.P., Gavrilenko T.V., Galkin V.A. O primeneni ponyatiya temperatury dlya opisaniya transportnogo potoka [The Traffic Flow Temperature Concept] // Russian Journal of Cybernetics, Vol. 5, № 2, 2024, pp. 26-34 (doi: 10.51790/2712-9942-2024-5-2-03) [in Russian].
14. nabbla1. "Likbez po kvaternionam, chast' 6: povorot po kratchajshemu puti" [A tutorial on quaternions, part 6: shortest path turning] // LiveJournal : Russian-owned social networking service. 22 february 2018 (<https://nabbla1.livejournal.com/187019.html>).