

SCIENTIFIC METHODS AND TECHNOLOGIES

UDC 004.92

Mihalkevitch V.A. Dynamic particle interactive visualization algorithms to enhance user engagement

Алгоритмы интерактивной визуализации динамических частиц для повышения вовлеченности пользователя

Mihalkevitch Vladislav Alexeevich

Student, Department of Instrumental and Applied Software Engineering,
MIREA – Russian Technological University,
Russia, Moscow

Scientific supervisor - **Miftakhov Eldar Nailevich**

Professor of Instrumental and
Applied Software Engineering,
MIREA – Russian Technological University,
Russia, Moscow

Михалькевич Владислав Алексеевич
студент, кафедры инструментального и прикладного программного обеспечения,
МИРЭА – Российский технологический университет,
РФ, г. Москва

Научный руководитель - Мифтахов Эльдар Наилевич
профессор кафедры инструментального и
прикладного программного обеспечения,
МИРЭА – Российский технологический университет,
РФ, г. Москва

Abstract. Object: Development of a mathematical model and software implementation of an interactive visualization algorithms for dynamic particles based on the Canvas API to improve user engagement.

Methods: Vector normalization methods for particle motion control, modular arithmetic for canvas boundary handling, and a frame rate control algorithm for performance optimization were used. The software implementation is written in TypeScript using the Canvas API.

Findings: An object-oriented model of the particle system was developed, and an algorithm for mouse cursor interaction through vector normalization was implemented. Tests confirm stable system operation at 10,000 particles with 45–55 FPS, as well as the effectiveness of frame rate control for reducing the load on the user device.

Conclusions: The proposed algorithmic solutions demonstrate high performance and can be used in the development of interactive web applications requiring dynamic 2D visualization with a large number of independent objects to enhance user engagement.

Keywords: interactive visualization; dynamic particles; Canvas API; vector normalization; modular arithmetic; performance optimization.

Аннотация. Цель: Разработка математической модели и программной реализации алгоритмов интерактивной визуализации динамических частиц на базе Canvas API для повышения вовлеченности пользователя.

Методы: Использованы методы векторной нормализации для управления движением частиц, модульной арифметики для обработки границ холста и алгоритм контроля частоты кадров для оптимизации производительности. Программная реализация выполнена на языке TypeScript с использованием Canvas API.

Результаты: Разработана объектно-ориентированная модель системы частиц, реализован алгоритм взаимодействия с курсором через векторную нормализацию. Проведённые тесты подтверждают стабильную работу системы при 10 000 частиц с частотой 45–55 FPS, а также эффективность контроля частоты кадров для снижения нагрузки на пользовательское устройство.

Выводы: Предложенные алгоритмические решения демонстрируют высокую производительность и могут быть использованы при создании интерактивных веб-приложений, требующих динамической 2D-визуализации с большим количеством независимых объектов для повышения вовлеченности пользователя.

Ключевые слова: интерактивная визуализация; динамические частицы; Canvas API; векторная нормализация; модульная арифметика; оптимизация производительности.

Рецензент: Бюллер Елена Александровна – кандидат экономических наук, доцент.
ФГБОУ ВО «Адыгеский государственный университет»

ВВЕДЕНИЕ

Современные веб-технологии активно развиваются, что приводит к повышению требований пользователей не только к функциональности, но также к визуальному оформлению интерфейсов [1]. Интерактивные фоны, анимированные переходы, динамические эффекты, реагирующие на движения курсора, становятся неотъемлемой частью успешных цифровых продуктов.

Внимание молодой аудитории, которая является основным будущим потребителем цифровых продуктов, в условиях информационного перенасыщения легче удерживать динамическими элементами, нежели статичным контентом [2, 3]. Исследования показывают, что среднее время концентрации внимания у представителей новых поколений составляет около 8 секунд, что требует от разработчиков интерфейсов использования эффективных визуальных инструментов для привлечения и удержания внимания [4].

На рынке существуют решения, упрощающие создание подобных эффектов [5-7]. Однако они предлагают ограниченный набор поведений, зачастую не ориентированных на алгоритмическую сложность, не предоставляют облачного хранилища для сохранения и обмена конфигурациями, а также привязаны к конкретным платформам.

Таким образом, возникает противоречие между потребностью рынка в уникальных визуальных решениях и отсутствием инструмента, сочетающего алгоритмическую базу с удобным управлением. Разработка веб-сервиса с открытой библиотекой и модульной архитектурой поведений является актуальной задачей [8].

Актуальность разработки интерактивных систем на основе частиц подтверждается недавними публикациями, в частности работой С. Тенкхоффа и соавторов, представленной на конференции Eurographics 2025, где описывается платформа для симуляции частиц в браузере с возможностью настройки параметров в реальном времени [9].

Настоящая статья посвящена математической модели и программной реализации алгоритмов интерактивной визуализации динамических частиц, которые лягут в основу создаваемой библиотеки.

Материалы и методы

Для визуализации выбран Canvas API как сбалансированное решение, сочетающее производительность при обработке большого числа объектов, гибкость работы с пикселями и кроссбраузерную поддержку, в отличие от более сложного WebGL и менее производительного SVG.

Для реализации выбран TypeScript, обеспечивающий строгую типизацию математических вычислений, и requestAnimationFrame для синхронизации анимации с частотой обновления экрана.

Тестирование производительности проводилось на устройстве с процессором Intel Core i5-12450H и GPU Intel ARC Graphics с использованием встроенного инструмента Chrome DevTools Performance Tab, версии браузера 149.0.7827.104. Измерялись средняя частота кадров FPS и время отрисовки одного кадра при различном количестве частиц N: 1000, 5000, 10000, 15000.

Математическая модель движения частиц

Метод моделирования нечётких объектов с помощью систем частиц был впервые предложен У.Т. Ривзом в 1983 году [10]. Данный подход предполагает представление сложного динамического объекта, например, огня, дыма, взрыва, в виде множества простых элементов – частиц, каждая из которых обладает собственными координатами, скоростью и другими возможными параметрами. Именно этот принцип лёг в основу разработанной модели.

В разработанном решении используется объектно-ориентированная модель, где каждая частица представлена отдельным классом. Вектор состояния частицы описывается параметрами, как показано в формуле 1.

$$S = (x, y, v_y, v_x, w, h, c), \quad (1)$$

где x, y – координаты;

v_y, v_x – компоненты вектора скорости;

w, h – размеры;

c – цвет.

Такой подход позволяет независимо управлять каждой частицей и легко масштабировать систему.

Обновление положения частицы за временной интервал Δt осуществляется по формулам равномерного прямолинейного движения, как представлено в формуле 2.

$$\{x(t + \Delta t) = x(t) + v_x \cdot \Delta t \quad y(t + \Delta t) = y(t) + v_y \cdot \Delta t \quad (2)$$

В программной реализации это выражается следующим кодом, представленном в листинге 1.

Листинг 1. Класс Particle

```
class Particle {  
    constructor(x, y, vx, vy, size, color) {  
        this.x = x;  
        this.y = y;  
        this.vx = vx;  
        this.vy = vy;  
        this.size = size;  
        this.color = color;  
    }  
  
    update() {  
        this.x += this.vx;  
        this.y += this.vy;  
    }  
}
```

Обработка границ – модульная арифметика

Для создания эффекта бесконечного пространства используется принцип модульной арифметики. При выходе частицы за границу холста её координата переносится на противоположную сторону, как представлено в формуле 3.

$$x' = \begin{cases} x + W + \omega, & \text{если } x < -\omega \\ x - W - \omega, & \text{если } x > W \\ x, & \text{иначе} \end{cases}, \quad (3)$$

где W – ширина холста;

ω – ширина частицы.

Реализация алгоритма, основанного на данной математической модели, представлена в листинге 2.

Листинг 2. Эффект бесконечного пространства

```
checkBounds() {  
    const { width, height } = canvas;  
  
    if (this.x > width) this.x = -this.size;  
    if (this.x < -this.size) this.x = width;  
    if (this.y > height) this.y = -this.size;  
    if (this.y < -this.size) this.y = height;  
}
```

Взаимодействие с курсором – векторная нормализация

При движении курсора для каждой частицы вычисляется направляющий вектор от частицы к курсору, как представлено в формуле 4.

$$\vec{d} = (x_{cursor} - x, y_{cursor} - y) \quad (4)$$

Длина вектора вычисляется по теореме Пифагора, как показано в формуле 5.

$$|\vec{d}| = \sqrt{d_x^2 + d_y^2} \quad (5)$$

Нормализация вектора заключается в делении каждой компоненты на длину, как представлено в формуле 6.

$$\vec{d}_{norm} = \left(\frac{d_x}{|\vec{d}|}, \frac{d_y}{|\vec{d}|} \right) \quad (6)$$

Новый вектор скорости устанавливается пропорционально нормализованному вектору с коэффициентом $k = 3$, как показано в формуле 7.

$$\vec{v}_{new} = k \cdot \vec{d}_{norm} \quad (7)$$

Алгоритм реализуется с оптимизацией для избегания лишнего вычисления квадратного корня при $|\vec{d}| = 0$, как представлено в листинге 3.

Листинг 3. Алгоритм расчёта движения частиц при движении мыши

```
canvas.addEventListener('mousemove', (event) => {  
  const mouseX = event.clientX - canvas.offsetLeft;  
  const mouseY = event.clientY - canvas.offsetTop;  
  
  for (const particle of particles) {  
    const dx = mouseX - particle.x;  
    const dy = mouseY - particle.y;  
  
    // Вычисление квадрата длины для оптимизации  
    const magnitudeSq = dx * dx + dy * dy;  
  
    if (magnitudeSq > 0) {  
      const magnitude = Math.sqrt(magnitudeSq);  
      particle.vx = (dx / magnitude) * 3;  
      particle.vy = (dy / magnitude) * 3;  
    }  
  }  
});
```

На рисунке 1 представлена интерактивная визуализация следования массива частиц за курсором, основанная на описанной математической модели.



Рисунок 1. Интерактивная визуализация

Оптимизация производительности

Для обеспечения стабильной частоты кадров при обработке большого количества частиц $N = 10^4$ используется алгоритм ограничения FPS. Кадр отрисовывается только при превышении заданного временного интервала, как представлено в формуле 8.

$$\Delta t \geq \frac{1000}{FPS_{target}}, \quad (8)$$

где FPS_{target} – целевая частота кадров.

Реализация анимационного цикла с контролем FPS указана в листинге 4.

Листинг 4. Анимационный цикл с контролем FPS

```
let lastTimestamp = 0;
const TARGET_FPS = 100;
const FRAME_INTERVAL = 1000 / TARGET_FPS;

function animate(currentTimestamp) {
  const deltaTime = currentTimestamp - lastTimestamp;

  if (deltaTime >= FRAME_INTERVAL) {
    ctx.clearRect(0, 0, canvas.width, canvas.height);

    for (const particle of particles) {
      particle.update();
      particle.draw(ctx);
    }
    lastTimestamp = currentTimestamp;
  }
  requestAnimationFrame(animate);
}
```

Однако, современные исследования в области оптимизации систем частиц, выполненные Дж. Сэндином, подтверждают, что ключевым узким местом производительности является не вычислительная сложность алгоритмов, а накладные расходы на отрисовку через Canvas API [11].

Результаты тестирования производительности

Проведено тестирование разработанной системы на устройстве с процессором Intel Core i5 и GPU intel ARC Graphics. Результаты тестирования представлены в таблице 1.

Таблица 1

Результаты тестирования

Количество частиц	Средний FPS	Время отрисовки кадра, мс
1 000	65	~15
5 000	60	~16
10 000	50	~20
15 000	45	~22

В качестве инструмента тестирования используется DevTools браузера Google Chrome версии 149.0.7827.104, как представлено на рисунке 2.

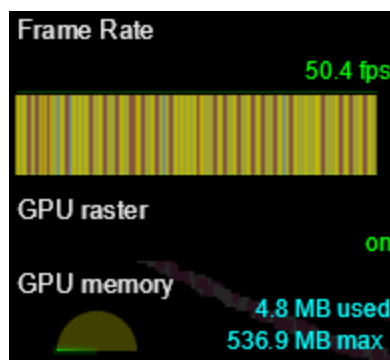


Рисунок 2. Показатели Frame Rate для массива 10 000 частиц
 На рисунке 3 показан детальный отчёт тестирования производительности.

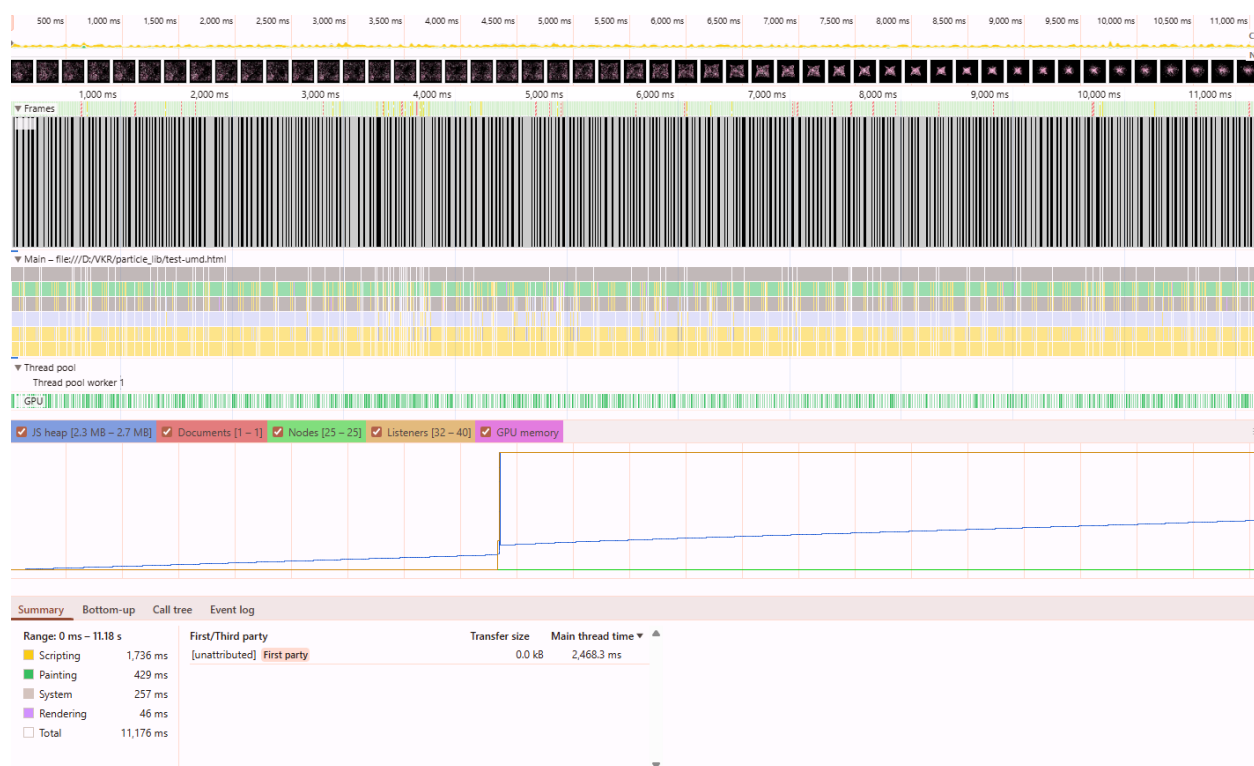


Рисунок 3. Детальный отчёт тестирования массива 10 000 частиц

Полученные результаты показывают, что система сохраняет стабильную производительность при количестве частиц до 10 000 с средним FPS 50. При увеличении числа частиц до 15 000 частота кадров снижается до 45 FPS, однако для большинства практических сценариев использования 10 000 частиц является достаточным.

Обсуждение

На основе разработанной модели могут быть реализованы более сложные алгоритмы поведения частиц:

1) Гравитационное взаимодействие – сила притяжения между частицами по закон всемирного тяготения Исаака Ньютона, формула 9.

$$F = G \cdot \frac{m_1 m_2}{r^2} \quad (9)$$

2) Поведение в электрическом поле – сила, зависящая от расстояния до курсора и «заряда» частицы, как представлено в формуле 10.

$$\vec{F} = \frac{q \cdot \vec{d}}{|\vec{d}|^3} \quad (10)$$

3) Потокное движение – использование шума Перлина для создания естественных траекторий, как показано в формуле 11.

$$\vec{v}(x, y) = \text{Perlin}(x, y, t) \quad (11)$$

В работе В.А. Литовского подробно рассмотрены методы визуализации взрывов с использованием систем частиц, включая гидродинамику сглаженных частиц, что подтверждает перспективность дальнейшего развития библиотеки в направлении физически достоверных эффектов [12].

Заключение

Разработанная математическая модель и её программная реализация на основе Canvas API подтвердили свою эффективность для интерактивной визуализации динамических частиц. Предложенный подход, основанный на векторной нормализации и модульной арифметике, обеспечивает как высокую производительность, так и простоту реализации, необходимую для быстрой интеграции в веб-проекты. Полученные результаты могут быть использованы в качестве вычислительной основы для более сложных алгоритмов, расширяя возможности визуализации в веб-среде для усиления эффекта вовлечения пользователя.

References

1. Sutcliffe A. Designing for User Engagement: Aesthetic and Attractive User Interfaces / A. Sutcliffe. – Cham : Springer Nature, 2022. – 47 p.
2. Prensky M. Digital Natives, Digital Immigrants / M. Prensky // On the Horizon. – 2001. – Vol. 9, No. 5. – P. 1–6.
3. Twenge J.M. iGen: Why Today's Super-Connected Kids Are Growing Up Less Rebellious, More Tolerant, Less Happy – and Completely Unprepared for Adulthood / J.M. Twenge. – New York : Atria Books, 2017. – 352 p.

4. Microsoft Corp. Attention Spans. – URL: <https://sherpapg.com/wp-content/uploads/2017/12/MAS.pdf> (дата обращения: 01.06.2026).
5. Garreau V. Particles.js [Электронный ресурс]. – URL: <https://vincentgarreau.com/particles.js/> (дата обращения: 03.06.2026).
6. Framer [Электронный ресурс]. – URL: <https://www.framer.com> (дата обращения: 03.06.2026).
7. ReactBits [Электронный ресурс]. – URL: <https://www.reactbits.dev> (дата обращения: 03.06.2026).
8. Bruni M. Why everyone should stop using particles.js right now [Электронный ресурс]. – URL: <https://dev.to/tsparticles/why-everyone-should-stop-using-particlesjs-right-now-5eb6> (дата обращения: 03.06.2026).
9. Tenckhoff S., Stauss S., Neumann F. Interactive Particle System with Integrated Marching Cubes Algorithm // Eurographics 2025 – Education Papers / ed. by R. Kuffner dos Anjos, K. Rodriguez Echavarria. – London : The Eurographics Association, 2025. – 3 p. – DOI: 10.2312/eged.20251011.
10. Reeves W.T. Particle Systems – A Technique for Modeling a Class of Fuzzy Objects / W.T. Reeves // ACM Transactions on Graphics. – 1983. – Vol. 2, No. 2. – P. 91–108.
11. Sandin J. Optimering av partikelsystem för webben : En jämförelse mellan objektorienterad och dataorienterad design på webben : Bachelor's thesis. – Skövde : Högskolan i Skövde, 2023. – 45 p. – URL: <http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-22798> (дата обращения: 05.06.2026).
12. Литовский В.А. Визуализация взрывов с помощью систем частиц // Молодой учёный. – 2020. – № 25 (315). – С. 108–112.