

УДК 004.93.1

АЛГОРИТМИ І МЕТОДИ ВИЗНАЧЕННЯ СКЛАДНИХ КОНТУРІВ ДИНАМІЧНИХ ОБ'ЄКТІВ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ WEB-ГРАФІКИ

А. Є. Козир, Г. В. Славко

Кременчуцький національний університет імені Михайла Остроградського
вул. Першотравнева, 20, м. Кременчук, 39600, Україна. E-mail: kozyr060195@gmail.com

Розроблено алгоритми і методи, що використовують особливості web-графіки, для дослідження динамічних процесів (у тому числі і швидкоплинних), для яких вдається отримати відеозапис. Це може бути, наприклад, зростання кристалів, пилові вибухові викиди тощо. Використання web-технологій для обробки відео потоків у реальному часі зумовлено вдосконаленням стандартів браузерів у поєднанні із збільшенням пропускної здатності інтернет-каналів. В результаті роботи розроблено web-застосування, що дозволяє аналізувати динаміку об'єктів зі складними контурами. Використано алгоритми для виділення контурів динамічних об'єктів, застосовано svg-фільтри для виконання матричних операцій фільтрації зображень, алгоритми для векторизації контурів об'єктів, отримано параметричні рівняння виділених контурів, візуалізовано обчислені характеристики динаміки досліджуваних об'єктів. Переваги розробленого web-застосування у можливості практичного застосування, наприклад, для дистанційної оцінки параметрів і характеристик вибухових робіт на Полтавському ГЗК.

Ключові слова: обробка зображень, динаміка об'єктів, векторизація, SVG, параметричні рівняння.

АЛГОРИТМЫ И МЕТОДЫ ОПРЕДЕЛЕНИЯ СЛОЖНЫХ КОНТУРОВ ДИНАМИЧЕСКИХ ОБЪЕКТОВ ИСПОЛЬЗУЯ ТЕХНОЛОГИИ WEB-ГРАФИКИ

А. Е. Козырь, Г. В. Славко

Кременчугский национальный университет имени Михаила Остроградского
ул. Первомайская, 20, г. Кременчуг, 39600, Украина. E-mail: kozyr060195@gmail.com

Разработаны алгоритмы и методы, которые используют особенности web-графики, для исследования динамических процессов (в том числе и быстропротекающих), для которых есть возможность получения видеозаписи. Это может быть, например, выращивание кристаллов, выбросы пыли во время взрывов и т.д. Использование web-технологий для обработки видеопотоков в реальном времени обусловлено совершенствованием стандартов браузеров в сочетании с увеличением пропускной способности интернет-каналов. В результате работы разработано web-приложение, которое позволяет анализировать динамику объектов со сложными контурами. Используются алгоритмы для выделения контуров динамических объектов, применены svg-фильтры для выполнения матричных операций фильтрации изображений, алгоритмы для векторизации контуров объектов, получены параметрические уравнения выделенных контуров, визуализированы вычисленные характеристики динамики исследуемых объектов. Преимущества разработанного web-приложения в возможности практического применения, например, для дистанционной оценки параметров и характеристик взрывных работ на Полтавском ГОК.

Ключевые слова: обработка изображений, динамика объектов, векторизация, SVG, параметрические уравнения.

АКТУАЛЬНІСТЬ РОБОТИ. У сучасній науковій і виробничій діяльності виникає потреба у створенні програмного забезпечення для дослідження динамічних процесів (у тому числі і швидкоплинних), для яких вдається отримати відеозапис. Це може бути, наприклад, зростання кристалів, пилові вибухові викиди і тому подібне.

Використання хмарних технологій та вдосконалення стандартів браузерів у поєднанні зі збільшенням пропускної здатності інтернет-каналів викликає підвищений інтерес до використання обробки відеопотоків у реальному часі. З'явився широкий клас задач, ефективно розв'язання яких можливе з пошуком та дослідженням нових підходів. Серед таких задач можна виділити й обробку відео з метою визначення контурів складних об'єктів та їх подальший аналіз [1, 2]. Складність аналізу відеопотоку у реальному часі зумовлена тим, що необхідно проводити обробку 25 і більше кадрів за секунду. Ці труднощі у першу чергу пов'язані з тим, що потрібно проводити складні математичні операції над великими обсягами даних [3, 4].

В останні роки посилюється інтерес науковців і практиків до питань використання WEB-технологій як засобів реалізації алгоритмів фільтрації зобра-

жень, розпізнавання та класифікації образів. Це зумовлено тим, що клієнт використовує в якості засобу для доступу до цих функцій лише веб-браузер. Таким чином користувачеві не потрібно встановлювати на свій ПК спеціальні програмні продукти та налаштовувати їх.

Розглянувши декілька offline програмних продуктів для трекінгу динамічних об'єктів та знаходження їх фізичних характеристик Tracker (Video Analysis and Modeling Tool), WINalyze, FreeTrack, виникла думка застосувати існуючі методи та алгоритми для створення подібної системи використовуючи технології web-графіки.

Метою роботи є реалізація алгоритмів та методів визначення складних контурів динамічних об'єктів з використанням технологій web-графіки та визначення характеристик динамічних процесів, відтворених на відео.

МАТЕРІАЛ І РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ. Використовуючи технології web-програмування необхідно реалізувати програмний комплекс для визначення контурів динамічних об'єктів на відео та визначення їх фізичних характеристик у реальному часі.

Для вирішення поставленої проблеми були сформульовані наступні задачі:

1. Застосування svg-фільтрів для виконання попередньої обробки зображення та виділення контурів.
2. Векторизація контурів об'єктів використовуючи svg-формат.
3. Отримання параметричних рівнянь виділених контурів.
4. Розробка алгоритму порівняння отриманого векторизованого контуру та параметричного рівняння для аналізу динаміки об'єктів.
5. Реалізація розглянутих та розроблених алгоритмів для аналізу динамічних об'єктів.

Під час трекінгу об'єктів на відео необхідно обробляти велику кількість потокових даних, що потребує значної кількості обчислювальних ресурсів, а значить і чималих витрат часу на обробку таких даних [3]. За останні роки було запропоновано чимало підходів щодо розв'язання цієї задачі. Багато з них накладають певні обмеження на оброблювальні дані такі, як статичний фон і фіксований ракурс, знання про тип об'єкту, за яким ведеться спостереження або наявність декількох камер [5]. Велика частина запропонованих підходів не освітлює у повній мірі проблему аналізу відеоряду в реальному часі.

Програмний комплекс повинен складатися з наступних модулів:

1. Розбиття відеозапису на кадри.

Користувач завантажує відеозапис до web-застосування. Відеозапис розбивається на кадри для обробки кожного зображення в реальному часі.

2. Обробка кожного кадру для видалення шумів та спотворень, виділення контурів об'єктів.

Даний модуль містить набір готових функцій для попередньої обробки зображень (видалення шумів та спотворень, підготовка нечіткого, розривного контуру для подальшої обробки). Виділення контуру об'єкта за допомогою фільтра Кенні [6].

3. Конвертація отриманих контурів до svg-формату.

Кожний отриманий контур векторизується. Для цього використовується відстежуючий алгоритм, який полягає у знаходженні першої точки об'єкту та подальшому відстеженні границі. Кожна знайдена точка складає певний сегмент, який записується до svg-файлу за допомогою елемента path.

4. Отримання параметричних рівнянь виділених контурів.

На вхід даного модуля надходить попередньо оброблене зображення з виділеними контурами. Таким чином отримано всі точки, які можуть бути з'єднані криволінійними сегментами. Для кожного сегмента кривої знаходимо ряд Фур'є. Отримаємо ключову параметризацию.

5. Аналіз контурів для визначення характеристик динамічного об'єкта.

Даний модуль виконує обчислення швидкості зміни площі досліджуваного об'єкта за одиницю часу та відображає графічне представлення отриманих даних.

На рис. 1 зображений порядок та взаємодія модулів програмного модуля.

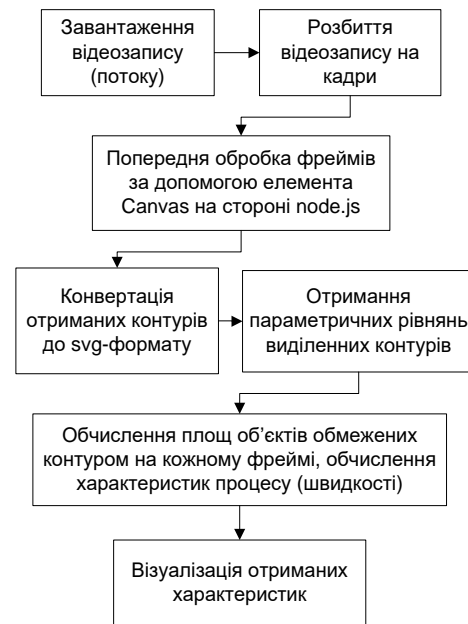


Рисунок 1 – Структурна схема програмного комплексу

Під час виконання роботи було розглянуто кілька технологій, для яких можливе застосування фільтрів. Для роботи з векторною графікою використовується контейнер SVG [7]. Для об'єктів векторної графіки можливо задавати події. За допомогою подій можна організовувати інтерактивну взаємодію користувача з контентом. Робота з растровими зображеннями виконується за допомогою контейнеру Canvas.

Для проведення обробки, в якості графічного контейнеру було використано Canvas. Це елемент HTML5, призначений для створення та редагування растрового двовимірного зображення використовуючи скрипти (переважно JavaScript).

На наступному етапі виконання роботи було використано фільтр Кенні для виділення границь на зображенні. Цей алгоритм має декілька головних кроків: розмиття зображення для видалення шуму, пошук градієнтів, пошук локальних максимумів на зображенні, подвійна порогова фільтрація, трасування області неоднозначності [8]. Розглянемо деякі кроки більш детально.

Розмиття зображення виконувалося за допомогою фільтра Гауса. Цей фільтр змінює кожен поточний шару, роблячи її значення рівним середньому значенню усіх точок в певному радіусі від розглянутої точки. Значення цього радіуса можна змінити. Чим більше радіус, тим сильніше буде розмиття зображення.

Розмите зображення оброблюється за допомогою фільтра Собеля. Границі відмічають там, де градієнт набуває найбільшого значення. Вони можуть мати різні напрямки, тому алгоритм Кенні використовує чотири фільтри для визначення горизонтальних, вертикальних і діагональних ребер у розмитому зображенні [9]. Кут нахилу градієнту округлюється і може приймати значення 0, 45, 90, 135.

На наступному етапі відбувається трасування області неоднозначності. Остаточні границі визначаються шляхом видалення всіх наборів пікселів, незв'язаних з основним контуром.

У роботі досліджується використання svg-фільтрів для попередньої обробки зображення (розмиття зображення, підвищення контрасту) та грубого виділення контурів за допомогою оператора Собеля. Було вирішено дослідити такий підхід через значну швидкість обробки кадрів відеоряду у реальному часі без затримок на обчислення.

SVG-фільтри підтримуються майже всіма сучасними браузерами, включаючи мобільні. Виняток становлять IE9 і Android native browser версії нижче 4.4.

Для фільтрації зображення можливо використовувати як вбудовані фільтри (рис. 2) так і задавати матрицю згортки зображення (рис. 3).

```
<svg width="400" height="300"
viewbox="0 0 400 300">
<defs>
<filter id="myFilter1">
<feGaussianBlur stdDeviation="10"/>
</filter>
</defs>
<circle cx="200" cy="150" fill="#3498db"
filter="url(#myFilter1)" r="80" /> </svg>
```

Рисунок 2 – Приклад використання вбудованого svg-фільтру (фільтр Гауса)

```
<feConvolveMatrix order="3"
kernelMatrix="-1 -1 -1 -1 7 -1 -1 -1 -1" /></feConvolveMatrix>
```

Рисунок 3 – Приклад запису матриці згортки зображення

Після використання svg-фільтрів передбачалася подальша обробка зображення за допомогою технології Canvas [10]. На цьому етапі виконання роботи виникли проблеми через неможливість отримання матриці оброблених пікселів. SVG-фільтри виступають у ролі «лінз» і після їх застосування до зображення не відбувається зміни вхідної матриці.

Намагаючись вирішити цю проблему було обрано інший підхід. Після застосування svg-фільтру до окремого кадру відеоряду планувалося зробити скріншот обробленого зображення.

Розглянуто декілька javascript-бібліотек для створення скріншотів на стороні клієнту. Для вирішення цієї задачі було обрано javascript-бібліотеку html2canvas, яка дозволяє створювати скріншот цілої сторінки або її частини [11]. Скріншот базується на основі DOM і не дає дійсного відображення, але базується на основі доступної інформації на сторінці.

Даний підхід також не дав очікуваних результатів. Ця бібліотека не підтримує багато css-властивостей (як і інші розглянуті бібліотеки), однією з яких є filter, що використовується для застосування svg-фільтру до певного елемента. Це стало поштовхом для пошуку іншого розв'язання цієї проблеми.

Після застосування svg-фільтрів до окремого кадру було вирішено робити скріншот на стороні серверу. Для цього використано Node.js – програмна платформа, заснована на движку V8 (транслює JavaScript в машинний код), що перетворює JavaScript з вузькоспеціалізованої мови в мову загального призначення та PhantomJs – консольний WebKit, повноцінний браузер без графічної оболонки. В результаті було отримано очікуваний результат. Було застосовано декілька фільтрів для виділення контуру (рис. 5, 6) до вхідного фрейму (рис. 4) та отримано наступні результати (рис. 7, 8).

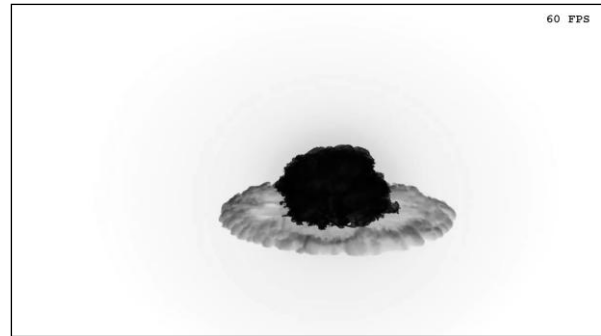


Рисунок 4 – Вхідний фрейм отриманий з відеозапису тестового динамічного процесу (вибуху)

```
<filter id="edge">
<feConvolveMatrix order="3"
preserveAlpha="true"
kernelMatrix="1 1 1 1 -8 1 1 1 1" />
</filter>
```

Рисунок 5 – Ядро згортки дискретного лапласіана

```
<filter id="edge">
<feColorMatrix type="luminanceToAlpha" />
<feConvolveMatrix order="3"
preserveAlpha="true" in="SourceGraphic"
result="a" kernelMatrix="-1 0 1 -2 0 2 -1 0 1" />
<feConvolveMatrix order="3"
preserveAlpha="true" in="a"
kernelMatrix="-1 -2 -1 0 0 0 1 2 1" />
</filter>
```

Рисунок 6 – Оператор Собеля

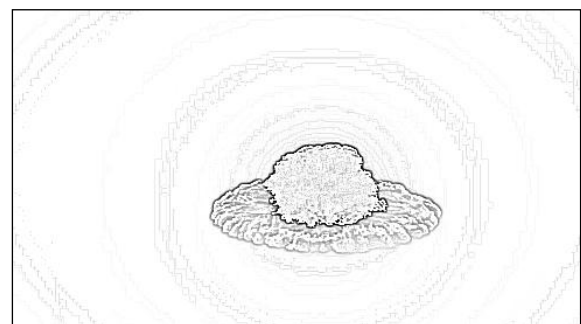


Рисунок 7 – Зображення оброблене дискретним лапласіаном

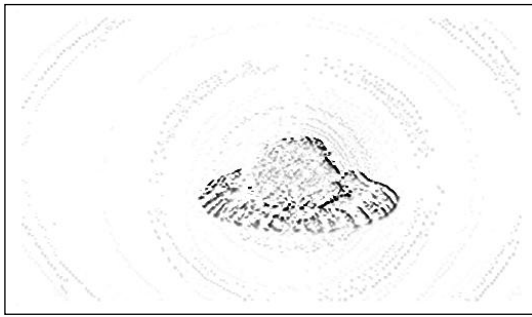


Рисунок 8 – Зображення оброблене фільтром Собеля

Після цього отримані результати обробляються з використанням технології Canvas, для видалення спотворень та остаточного виділення контуру.

На наступному етапі відбувається отримання параметричних рівнянь виділених контурів [12]. Можна показати, що будь-яка гладка крива $y_n(x)$ може бути як завгодно добре апроксимована на будь-якому інтервалі $[x_1, x_2]$ за допомогою рядів Фур'є. Причому для гладких кривих значення коефіцієнтів $\sin(kx)$ та $\cos(kx)$ прямує до нуля при великих значеннях k . Тепер, розглядаючи параметричну криву $f(t) = \{x(t), y(t)\}$, ми можемо використовувати такого роду суперпозиції функцій синусів і косинусів незалежні для горизонтальної $x(t)$ і вертикальної $y(t)$ складових.

$$x(t) = \sum_{k=1}^n (a_k \cos(kt) + b_k \sin(kt)), \quad (1)$$

де a_k, b_k – дійсні константи;

$$y(t) = \sum_{k=1}^n (c_k \cos(kt) + d_k \sin(kt)), \quad (2)$$

де c_k, d_k – дійсні константи.

Якщо візьмемо розклад у ряд Фур'є кривої на перших n членах ряду (1, 2), то отримаємо $4n$ вільних параметрів. Таким чином при деяких значеннях коефіцієнтів, розклад буде мати вигляд складних фігур за допомогою яких можна описати виділені контури. Однак, навіть невеликі зміни в коефіцієнтах розкладу сильно змінюють зображення.

Щоб продемонструвати як знайти прості, засновані на ряді Фур'є формули [13], які наближають дані фігури, ми почнемо з прикладу (рис. 9).

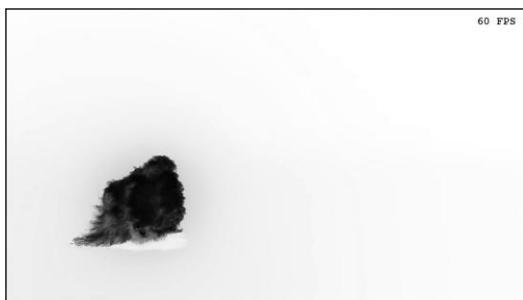


Рисунок 9 – Вхідний фрейм отриманий з відеозапису тестового динамічного процесу (горіння)

Отримаємо набір усіх точок, що описують границю об'єкту (рис. 10):



Рисунок 10 – Зображення з виділеними контурами

Тепер, коли є точки, які задають контур, з'єднаємо їх криволінійними сегментами. Для кожного сегмента кривої потрібно знайти многочлен Фур'є (горизонтальну $x(t)$ і вертикальну $y(t)$ складові), який апроксимує сегмент. Побудуємо B-сплайни кривої для кожного її сегмента. Параметризована змінна B-сплайна кривої буде змінною інтегрування.

Використаємо швидке перетворення Фур'є (FFT) для отримання коефіцієнтів Фур'є.

Для отримання більш однорідних кривих виконаємо ще один крок: повторно параметризуємо інтерпольовану криву сплайна, що складається з набору сегментів кривої, за допомогою нового параметра – довжини дуги. Враховуємо чи відкритий сегмент кривої або закритий, щоб уникнути явища Гіббса [14].

Для неперервної функції очікуємо середнє значення швидкості спадання k^{-2} для k -ого коефіцієнта ряду Фур'є. Цей ефект повинен зберігатися і для розрахованих коефіцієнтів ряду Фур'є. Це дає можливість записати многочлен Фур'є на не надто високих порядках.

Необхідно, щоб остаточна формула для кривих виглядала настільки коротко і просто наскільки це можливо. Тому замінюємо суму у формулі $a \cos(kt) + b \sin(kt)$ на $A \sin(kt + \varphi)$, а також округляємо наближені коефіцієнти ряду Фур'є до найближчих раціональних чисел.

Тепер є все необхідне, щоб записати заключну параметризацію $\{x(t), y(t)\}$ зображення.

Для рисунку (рис. 10) маємо параметричне рівняння та побудуємо його [13]:

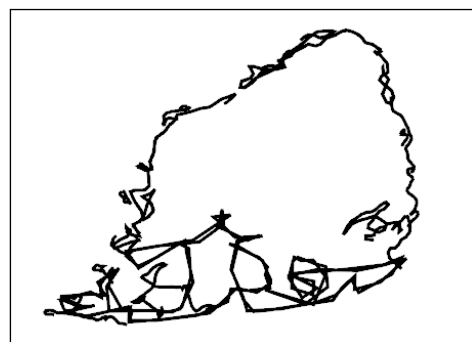


Рисунок 11 – Побудоване параметричне рівняння для вхідного фрейму (рис. 9)

Описані алгоритми і методи були реалізовані з використанням Wolfram Mathematica на мові JavaScript. Також, для обробки зображення та виділення контурів було використано бібліотеки з відкритим вхідним кодом jsfeat, tracking.js.

Далі кожний отриманий контур векторизується (рис. 12).

```
<svg xmlns='http://www.w3.org/2000/svg'
version='1.1' width='445' height='251'>
<path d='M130,125L131,125 132,125 141,129
142,129 143,130 143,131 144,132 145,133
145,134 145,135 145,136 144,137 143,138
144,139 145,140 146,141 147,142 147,143
147,144 147,145 147,146 148,147 149,148
150,149 150,150 151,151 151,152 151,153
152,154 152,155 151,156 150,157 151,158
151,159 152,160 152,161 152,162 152,163
152,164 152,165 151,166 151,167 151,168
152,169 153,170 153,171 153,172 153,173
153,174 153,175 152,176 152,177 151,178
151,179 151,180 150,181 149,182 148,183
148,184 147,185 146,186 146,187 147,188
148,188 149,188 150,188 150,189 151,190
150,191 149,191 148,191 147,191 146,191
145,191 144,191 143,191 142,191 141,191
140,192 139,192 138,193 137,193 136,194
135,194 134,194 133,195 132,195 131,195
130,196 130,197 129,198 128,198 127,198
126,198 125,197 124,198 123,198 122,198
121,199 120,199 119,199 118,200'
fill='white' stroke='black'/>...
```

Рисунок 12 – Фрагмент коду отриманого svg-зображення

Після цього попередньо оброблене зображення з виділеними контурами (рис. 10) передається до наступного модуля для отримання параметричних рівнянь отриманих контурів.

Далі аналізуємо контури для визначення характеристик динамічного об'єкта. Приводиться приклад виділення контуру поширення полум'я на початковому фреймі (10 мс) (рис. 13), після майже 2 секунд поширення полум'я (рис. 14), завершальний етап поширення полум'я (3750 мс) (рис. 15). Виконуємо обчислення зміни площі досліджуваного об'єкта за одиницю часу та відображаємо графічне представлення отриманих даних (рис. 16).

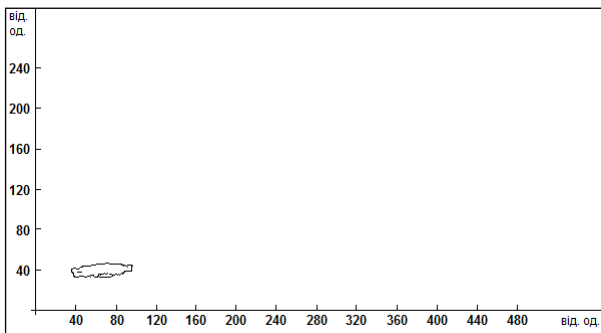


Рисунок 13 – Контур об'єкта на початковому фреймі (10 мс)

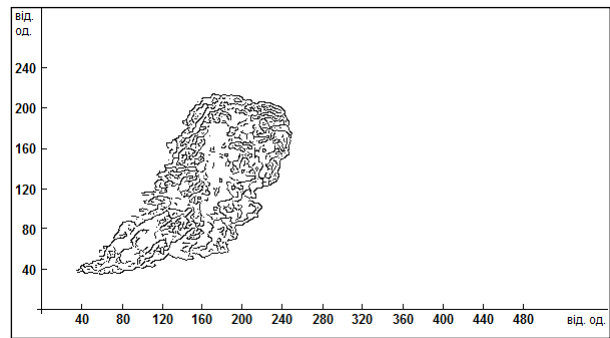


Рисунок 14 – Контур об'єкта після 2000 мс

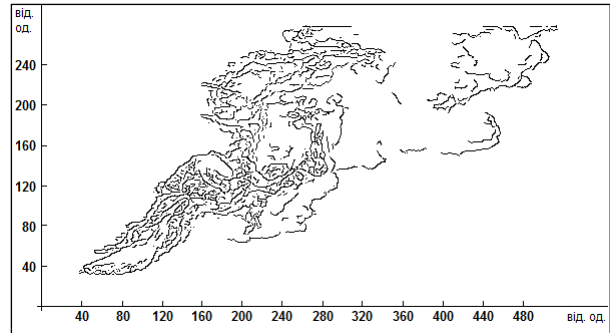


Рисунок 15 – Контур об'єкта на останньому фреймі (3750 мс)

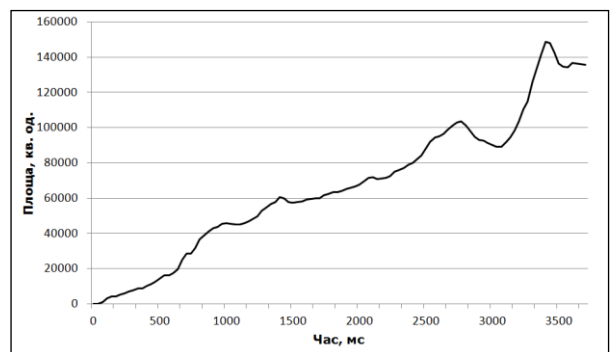


Рисунок 16 – Зміна площі поширення полум'я відносно часу

На наступній гістограмі відображена середня швидкість обробки поточного фрейму (рис. 17).

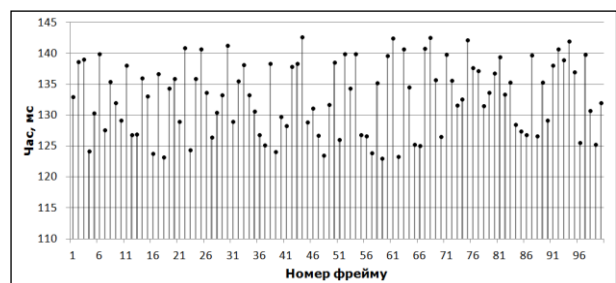


Рисунок 17 – Гістограма середньої швидкості обробки одного фрейму

ВИСНОВКИ. В результаті роботи було реалізовано web-застосування для аналізу динаміки складних об'єктів. Використано алгоритми для виділення контурів динамічних об'єктів, застосовано svg-фільтри для виконання матричних операцій фільтрації зображень, алгоритми для векторизації контурів об'єктів використовуючи svg-формат, отримано параметричні рівняння виділених контурів, візуалізовано обчислені характеристики динаміки досліджуваних об'єктів.

Перевагами векторизованого контуру є можливість його вільної трансформації – векторний об'єкт можна обернути і масштабувати без втрат його якості, швидкість роботи з таким контуром.

Недоліком розробленого методу застосування svg-фільтрів для обробки зображення є його швидкість. Це особливо помітно при застосуванні даного підходу для фільтрації фреймів відеоряду. На сьогоднішній день відсутня можливість виконувати фільтрацію зображення використовуючи тільки svg-фільтри без технології canvas. Існує можливість застосовувати фільтри напряму до canvas за допомогою CanvasRenderingContext2D.filter, але це експериментальна технологія і вона не підтримується у багатьох версіях сучасних браузерів.

Надалі планується вдосконалити існуючі модулі та використати їх у розробці онлайн-лабораторії для динамічного аналізу об'єктів на відео.

ЛІТЕРАТУРА

1. Cohen I., Medioni G. Detecting and Tracking Moving Objects for Video Surveillance // University of Southern California Institute for Robotics and Intelligent Systems. – 1999. – P. 1–6.
2. Dynamic Object Detection, Tracking and Counting in Video Streams for Multimedia Mining / L. Vibha, H. Chetana, S. Deepa // International Journal of Computer Science. – 2008. – P. 1–10.
3. Козир А.Є., Славко Г.В. Обробка зображень фільтром Собеля на основі аналізу структури графічних файлів // Вісник Кременчуцького національного університету імені Михайла Остроградського. – 2014. – № 3. – С. 59–63.
4. Козир А.Є., Славко Г.В. Виділення контурів об'єктів на зображенні за допомогою технологій web-графіки: матеріали Всеукр. наук. конф. «Математичне моделювання та математична фізика», 22-25 вересня 2015, Кременчук, Україна / Кременчуцький національний університет імені Михайла Остроградського. – 2015. – С. 41–42.
5. Розробка алгоритму відокремлення багатозв'язних областей графічних зображень динамічних процесів у відеопотоці / С.О. Геращенко, С. І. Малахов, Г. В. Славко // Вісник Кременчуцького національного університету імені Михайла Остроградського. – 2009. – № 1. – С. 13–16.
6. Прэтт У. Цифровая обработка изображений: пер. с англ. – М.: Мир, 1982. – Кн. 2 – 480 с.
7. Jump Start HTML5 / Т.В. Brown, К. Butters, S. Panda. – Melbourne: SitePoint, 2014. – 276 с.
8. Р. Гонсалес, Р. Вудс. Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с.
9. Яне Б. Цифровая обработка изображений. – М.: Техносфера, 2007. – 584 с.
10. Pfeiffer S. The Definitive Guide to HTML5 Video / Silvia Pfeiffer. – New York: Apress, 2010. – 321 p.
11. Herten, N., html2canvas [Електронний ресурс] – Режим доступу до ресурсу: <https://html2canvas.herten.com/>.
12. Trott M. Making Formulas... for Everything – From Pi to the Pink Panther to Sir Isaac Newton [Електронний ресурс] / Michael Trott. – 2013. – Режим доступу до ресурсу: <http://blog.wolfram.com/2013/05/17/making-formulas-for-everything-from-pi-to-the-pink-panther-to-sir-isaac-newton/>.
13. Отримане параметричне рівняння вхідного фрейму [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: http://primat.org/files/png/parametric_equation.png.
14. Кудрявцев Л. Д. Курс математического анализа. – Москва: Дрофа, 2006. – 351 с.

ALGORITHMS AND METHODS FOR DETECTION OF COMPLEX EDGES OF DYNAMIC OBJECTS USING WEB-GRAPHICS TECHNOLOGIES

A. Kozyr, G. Slavko

Kremenchuk Mykhailo Ostrohradskyi National University

vul. Pershotravneva, 20, Kremenchuk, 39600, Ukraine. E-mail: kozyr060195@gmail.com

Purpose. The purpose is to implement algorithms and methods for determination of complex edges of dynamic objects using web-graphics technologies and determination the characteristics of dynamic processes on video playback.

Methodology. In this paper, a software package for determination of edges of dynamic objects from video and determination of their physics characteristics in real time is implemented using web-programming technologies. SVG-filters is used to image pre-processing and edges detection. Detected edges are vectorized using an algorithm, which is based on SVG format structure features. For getting parametric equations of detected edges Fourier series are used. Vectorized edges and parametric equations are compared and analyzed for getting physics characteristics of dynamic objects. **Results.** In this paper, the algorithms and methods for the analysis of the dynamics of complex objects are implemented. For the problem solving SVG filtering technology is used to applying matrix operations of image processing. The algorithm for vectorization of edges objects and the algorithm for getting parametric equations of detected edges are implemented. **Originality.** Implementation of software for researching of dynamic processes (include fleeting), for which can be obtained videos, is an important issue. Such software is implemented in the work. **Practical value.** The web-application, which is implemented in the work, is used to researching of fleeting processes (an explosion, burning of a match, hydraulic cutting etc.). For example, the web-application can be used to remote evaluation of parameters and characteristics of explosive works at Poltava mining and beneficiation complex. References 14, figures 17.

Key words: image processing, dynamics of objects, vectorization, SVG, parametric equations.

REFERENCES

1. Cohen, I., Medioni, G. (1999), "Detecting and Tracking Moving Objects for Video Surveillance", *University of Southern California Institute for Robotics and Intelligent Systems*, pp. 1–6.
2. Vibha, L., Chetana, H., Deepa, S. (2008), "Dynamic Object Detection, Tracking and Counting in Video Streams for Multimedia Mining", *International Journal of Computer Science*, Vol. 35, pp. 1–10.
3. Kozyr, A.E., Slavko, G.V. (2014), "Image processing by Sobel filter based on the analysis of a graphic files structure", *Transactions of Kremenchuk Mykhailo Ostrohradskyi National University*, iss. 3, no. 86, pp. 59–63.
4. Kozyr, A.E. (2015), "Edge detection of the objects in the image using web-graphics technologies", *Matematychni modeliuvania ta matematychna fizika. Zbirnyk naukovykh prats Vseukrainskoi naukovoï konferentsii* [Mathematical modeling and mathematical physics. Conference proceedings of Ukrainian conference], Kremenchuk, KrNU, September 22–25, 2015.
5. Gerashchenko, S.O., Malahov, S.I., Slavko, G.V. (2009), "Development of the algorithm separation multiply graphic image dynamic processes in the video stream", *Transactions of Kremenchuk Mykhailo Ostrohradskyi National University*, vol. 1, no. 54, pp. 13–16.
6. Pratt, W. (1982), *Cifrovaia obrabotka izobrazhenii* [Digital image processing], Mir, Moscow, Russia.
7. Brown, T.B., Butters, K., Panda, S. (2014), *Jump Start HTML5*, Site Point, Melbourne, Australia.
8. Gonsales, R., Woods, P. (2005), *Cifrovaia obrabotka izobrazhenii* [Digital image processing], Tehnosfera, Moscow, Russia.
9. Jahn, B. (2007), *Cifrovaia obrabotka izobrazhenii* [Digital image processing], Tehnosfera, Moscow, Russia.
10. Pfeiffer, S. (2010), *The definitive guide to HTML 5 video*, Apress, Berkeley, USA.
11. Herten, N. (2015), "Html 2 canvas", available at: (accessed January 15, 2016).
12. Trott, M. (2013), "Making Formulas... for Everything—From Pi to the Pink Panther to Sir Isaac Newton—Wolfram Blog", available at: <http://blog.wolfram.com/2013/05/17/making-formulas-for-everything-from-pi-to-the-pink-panther-to-sir-isaac-newton/> (accessed January 21, 2016).
13. "The result parametric equation of the input frame", (2016), available at: http://primat.org/files/png/parametric_equation.png (accessed March 02, 2016).
14. Kudryavtsev, L. (2006), *Kurs matematicheskogo analiza* [Mathematical Analysis Course], Drofa, Moscow, Russia.

Стаття надійшла 20.05.2016.