

УДК 004.43, 004.932

Б.П. Русин, д.т.н., проф.,
В.Б. Яців, аспірант
Фізико-механічний інститут ім. Г.В. Карпенка НАН України, м. Львів, Україна
rusyn@ipm.lviv.ua, vicja@list.ru

Основні підходи до розв'язку задачі масштабування зображень у веб-орієнтованих технологіях

Розглядаються теоретичні та практичні аспекти використання SVG та HTML5 Canvas в сучасних веб-орієнтованих технологіях. Запропоновано метод реалізації алгоритму масштабування зображення перед загрузкою на сервер на стороні клієнта.

Ключові слова: HTML5 Canvas, SVG (Scalable Vector Graphics), HTML (Hyper Text Markup Language), масштабування зображення, веб-сторінка, File API.

Вступ

Багато стандартів, як наприклад HTML, підтримують метод динамічного представлення інформації – розміщення таблиць, тексту і т.д. Однак, що стосується зображень, то вони далеко не такі гнучкі і в кращому випадку можуть приймати певні наперед визначені розміри, або змінюватись тільки пропорційно. При необхідності відобразити зображення на дисплеї певного розміру, чи надрукувати його на папері відповідного формату і в ряді інших задач виникає потреба масштабування зображення, часто зі зміною співвідношення сторін.

Часто одне і те ж зображення відображається на веб-сторінці в різних розмірах. Наприклад, зменшена версія для попереднього перегляду і повнорозмірна. Зараз цього добиваються масштабуванням зображення на сервері та збереженням в окремий файл. Але так можна відобразити зображення тільки з одним з наперед заданих розмірів, а при збільшенні кількості варіантів збільшується навантаження на процесор сервера при масштабуванні та використовується більше дискового простору.

Інша проблема виникає при завантаженні користувачем зображень на сервер. Сучасні фотоапарати роблять знімки до 10 мегапікселів і тому вони навіть у стисненому форматі займають більше 3МБ. Для відображення на веб-сторінці такі великі зображення не підходять і тому після завантаження на сервері здійснюється масштабування. При масштабуванні таких зображень використовується відносно багато процесорного часу та оперативної пам'яті. Крім того саме завантаження

Для вибору файлів може використовуватись звичайне поле, що задається тегом `<input type="file">`. В HTML5 у цього тегу з'явився новий атрибут – `multiple`, що дозволяє користувачу за раз вибирати декілька файлів. Крім того

триває довго, і використовує більше інтернет-трафіку. Ці проблеми може вирішити попереднє масштабування зображень на стороні клієнта. Але до недавнього часу це було можливо тільки з використанням таких технологій, як Flash та Java, що вимагають встановлення окремих плагінів. Тепер розвиток стандартів дозволяє реалізувати цей метод з використанням звичайного JavaScript.

Аналіз існуючих рішень

Реалізація масштабування зображення на стороні клієнта стала можливою завдяки створенню нової версії стандарту HTML (Hyper Text Markup Language). Він ще не є остаточно затвердженим, але більшість браузерів вже реалізували деякі з нових можливостей. Найважливішим для даної задачі нововведенням в HTML5 є тег `canvas` [1-2]. Він призначений для створення та відображення растрових зображень засобами мови JavaScript. Інтерфейс тегу `canvas` містить функції для створення графічних примітивів, але крім цього дозволяє доступ до зображення як до масиву пікселів. Останній метод дає можливість реалізувати будь-який алгоритм масштабування.

Сам по собі тег `canvas` дозволяє працювати тільки з зображеннями, які вже присутні на сторінці. Для масштабування зображень, що завантажуються на сервер, використано ще один новий елемент стандарту – `File API`. Методи, що використовують цей інтерфейс дозволяють отримувати властивості файлу на комп'ютері користувача та зчитувати його вміст у двійковому або текстовому вигляді. Звісно ж з міркувань безпеки це можна робити не з усіма файлами, а тільки з обраними користувачем.

З'явився повністю новий спосіб вибору – `Drag and Drop API`. При його використанні файли просто перетягуються у вікно браузера, у визначену для цього область сторінки. Це наближає веб-сторінку за зручністю роботи до традиційних програм.

У випадку масштабування векторних зображень (графіки, діаграми, логотипи та ін.) використовують технологію SVG (Scalable Vector Graphics), яка основана на XML [3-4]. SVG-контент можна змінювати за допомогою CSS і визначати поведінку об'єктів за допомогою SVG DOM. Текст усередині SVG зберігається у файлі, де він залишається відносно доступним для використання. Також svg-контент можна вставити безпосередньо в (X)HTML з використанням елементу `object` [5-6]. Файли SVG можна читати і редагувати за допомогою звичайних текстових редакторів.

SVG надає всі переваги XML:

1. Можливість роботи в різних середовищах.
2. Інтернаціоналізація (підтримка Юнікоду).
3. Широка доступність для різних застосувань.
4. Легка модифікація через стандартні API - наприклад, DOM. SVG підтримує стандартизо-

вану W3C об'єктну модель документа DOM, забезпечуючи доступ до будь-якого елементу, що дає широкі можливості з динамічної модифікації елементів, їхніх атрибутів і подій.

5. Легке перетворення таблицями стилів XSLT. Як будь-який заснований на XML формат, SVG дає можливість використовувати для його обробки таблиці трансформації (XSLT). Перетворюючи XML-дані в SVG за допомогою простого XSL, можна легко отримати графічне представлення будь-яких даних, наприклад візуалізувати хімічні молекули, описаних на мові CML (Chemical Markup Language).

Представимо відмінності описаних вище веб-технологій, розглянувши їх переваги та недоліки (табл. 1).

Таблиця 1. переваги та недоліки SVG та Canvas

Canvas	SVG
<ul style="list-style-type: none"> • Висока продуктивність при відтворенні будь-яких 2d об'єктів. • Продуктивність падає тільки при збільшенні роздільності зображення. • Можна зберегти отримане зображення в PNG або JPG файл. • Краще всього підходить для створення растрової графіки (наприклад, в іграх, фракталів і т.п.), редагування зображень і операцій, що вимагають маніпулювання на рівні пікселів. • Не існує API для анімації. Доводиться використовувати індикатори завантаження та інші події для оновлення канви. 	<ul style="list-style-type: none"> • Немає залежності від розрешення - SVG краще підходить для кросплатформених користувацьких інтерфейсів, веб-додатків, оскільки дозволяє масштабувати зображення при різних розрешеннях екрану. Існує можливість збільшити будь-яку частину зображення SVG без втрати якості. • SVG дуже добре підтримує анімацію. Елементи можуть бути анімовані з використанням описового синтаксису або за допомогою Javascript. • Можна отримати повний контроль над кожним елементом, використовуючи SVG DOM API в Javascript. • Низька швидкість рендерінга при збільшенні складності документа (малюнка), оскільки використовується модель DOM. • Можна отримати повний контроль над кожним елементом, використовуючи SVG DOM API в Javascript.

Реалізація алгоритму, результати

Провівши аналіз існуючих рішень для реалізації масштабування зображення на стороні клієнта, які використовуються в сучасних інформаційних веб-орієнтованих технологіях, було розроблено програму для завантаження зображень на сервер. При цьому до надсилання зображення масштабуються за допомогою JavaScript на стороні клієнта.

Після завантаження веб-сторінки викликається процедура `init` для ініціалізації скрипта (рис. 1). Перш за все вона перевіряє, чи підтримуються необхідні технології браузером. Підтримка роботи з канвами перевіряється за наявністю методу `getContext` в елементу `canvas`, а `Files API` – за наявністю класу `FileReader`. У випадку, якщо `Files API` не підтримується, ніяких подальших дій не виконується і файли завантажуються на сервер звичайним способом, через поле вводу з типом `file`.

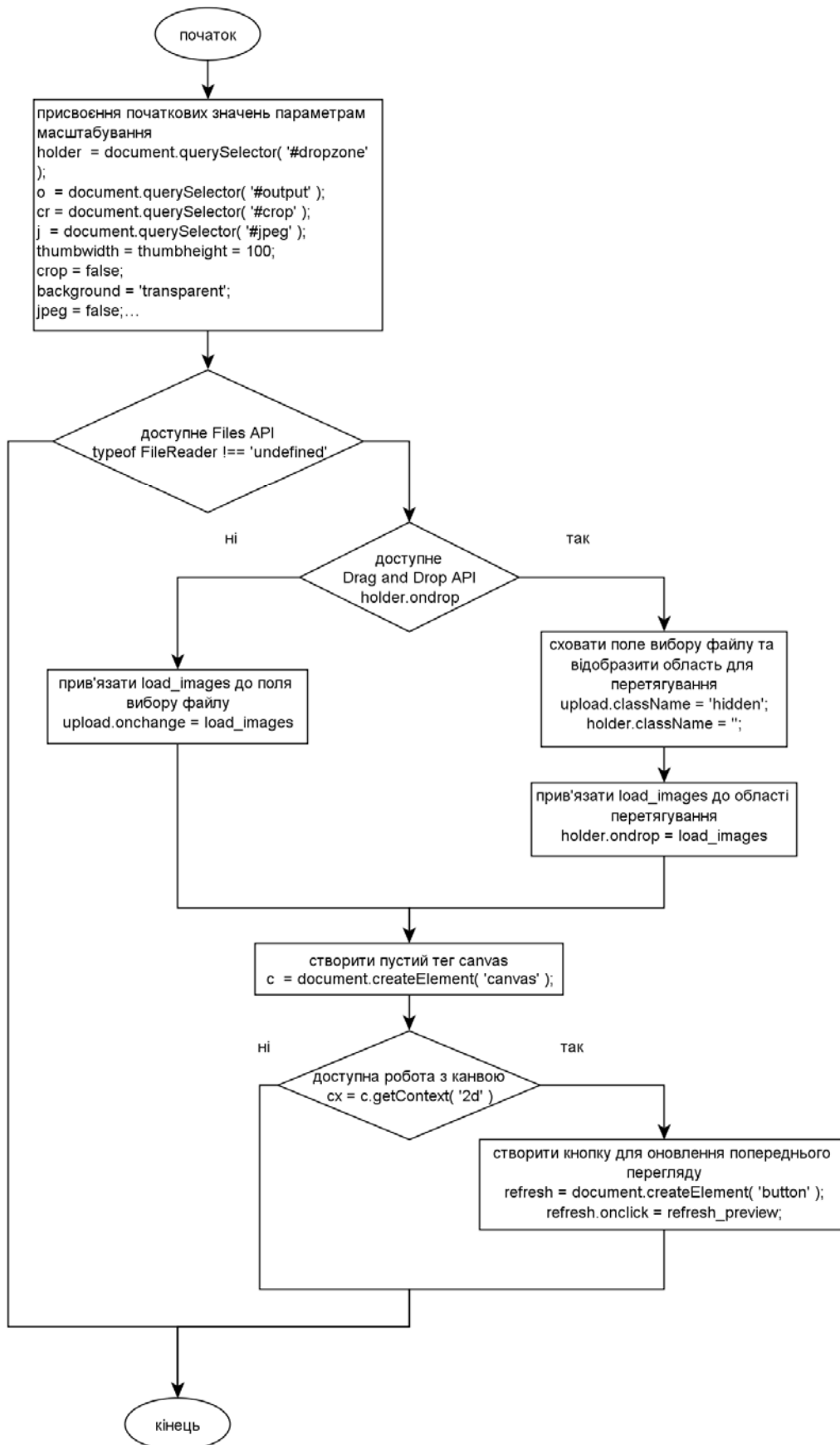


Рисунок 1 – Блок-схема алгоритму масштабування на сторінці клієнта

Якщо робота з файлами підтримується браузером – то перевіряється підтримка технології Drag and Drop для вибору файлів шляхом перетягування у вікно браузера. Якщо ця технологія не підтримується – то процедура `load_images` прив'язується до події `onchange` поля вибору файлів. Якщо ж перетягування підтримується – то стандартне поле ховається, а замість нього відображається прямокутний блок, що служить ціллю для перетягування. Тоді `load_images` викликається при події `drop`.

Процедура `load_images` завантажує та зберігає у пам'яті для подальшої обробки файли, вибрані користувачем (рис. 2). Для цього використовується схований тег `img` та технологія `Data URI`. Вона полягає в тому, що в параметрі `src` знаходиться не адреса зображення а його вміст, закодований спеціальним чином. Для цього створюється об'єкт типу `FileReader`, та використовується його метод `readAsDataURL`. Після завершення зчитування файлу з зображенням відбувається подія `onload` і зчитаний вміст заноситься в параметр `url`.

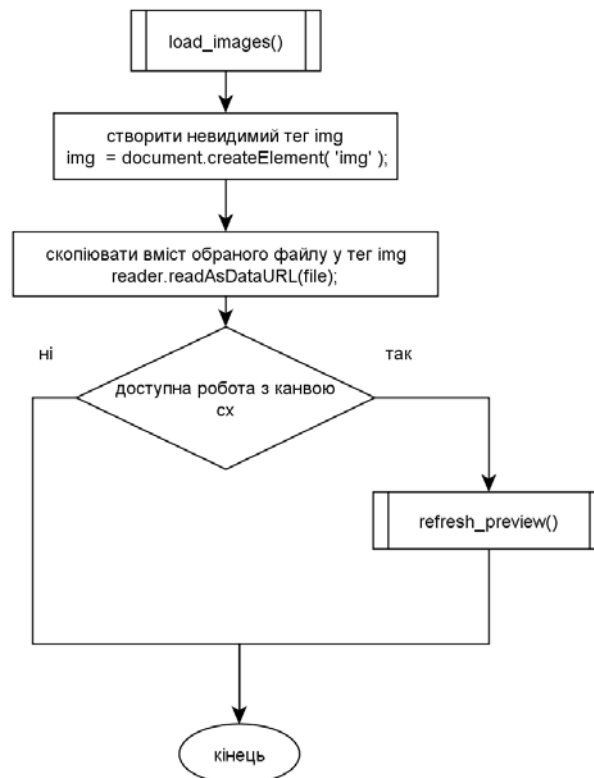


Рисунок 2 – Блок-схема реалізації процедури `load_images`

Процедура `refresh_preview` викликається при виборі користувачем нового файлу, або при зміні параметрів масштабування (рис. 3). Спочатку в ній параметри масштабування зчитуються з полів вводу. Потім обчислюються розміри результуючого зображення. Далі викликається процедура масштабування, що відповідає обраному алгоритму. Результуюче зображення виводиться на канву:

```
cx.drawImage(img, dimensions.x, dimensions.y, dimensions.w, dimensions.h);
```

І в кінці зображення кодується у форматі `Data URI` та вноситься у форму для надсилання на сервер.

Кожний користувач використовуючи запропоновану програму, має можливість змінювати такі параметри як, ширина та висота результуючого зображення, алгоритм масштабування, ак-

тивувати пропорційне масштабування, задати обрізку та ступінь стиснення. При цьому в реальному часі на веб-сторінці відображається та оновлюється попередній перегляд результатів масштабування та об'єм зображень.

Вибір зображень для масштабування здійснюється шляхом перетягування файлів у відмічену область. Прогрес завантаження та масштабування відображається на індикаторі. Після завершення масштабування зображення можна надіслати на сервер, де вони будуть збережені спеціальним PHP скриптом.

У випадку, якщо необхідні технології не підтримуються веб-браузером, відображається звичайне поле для вибору файлу і масштабування здійснюється на стороні сервера.

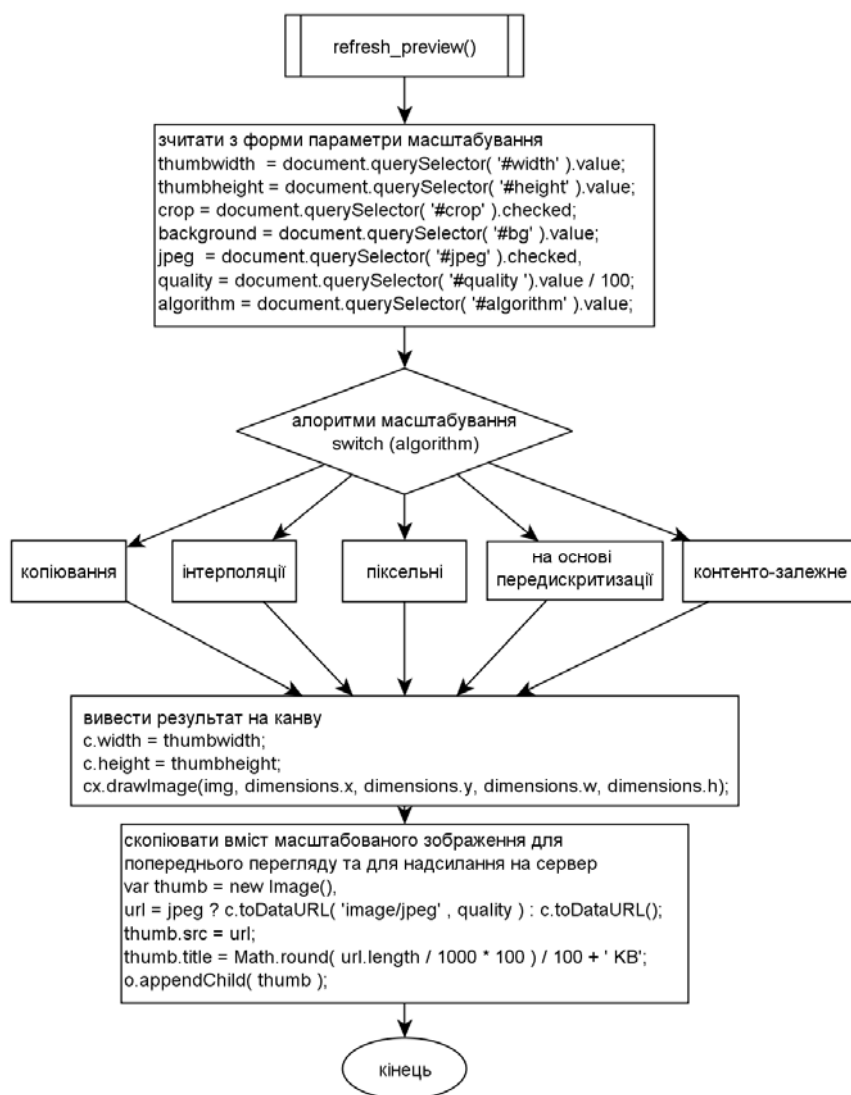


Рисунок 3 – Блок-схема реалізації процедури refresh_preview

Висновок

Програмна реалізація запропонованого алгоритму для масштабування і завантаження на сервер зображень може бути інтегрована та використана в будь-яких сучасних інформаційно веб-орієнтованих технологіях (веб-сайтах, веб-порталах, CMS – Content Management System), де можливе завантаження фотографій користувачами.

Запропонований в статті алгоритм показав себе як зручний інструмент, який дозволяє користувачеві в реальному часі на стороні клієнта відобразити попередній перегляд результатів масштабування, економлячи при цьому інтернет-трафік та знімаючи з сервера навантаження.

Кожна технологія має свої сильні та слабкі сторони, тому користувачеві важливо визначити в запропонованій програмі для себе необхідний функціонал, який можна використати як додаток для своїх алгоритмів і отримати хороші результати.

Список використаної літератури

1. Лабберс П. HTML5 для професіоналів: мощные инструменты для разработки современных веб-приложений / П. Лабберс, Б. Олберс, Ф. Салим. – М.: «Вильямс», 2011. – 272 с.
2. Хольцнер С. HTML5 за 10 минут, 5-е издание / С. Хольцнер. – М.: «Вильямс», 2011. – 240 с.
3. Sucas M. SVG or Canvas? Choosing between the two, June 2010. – <http://dev.opera.com/articles/view/svg-or-canvas-choosing-between-the-two>
4. Офіційний сайт SVG. – <http://www.w3.org/Graphics/SVG/>

5. Учебник по SVG (школы информационных технологий государственного университета Иллинойс (Illinois State University)). – <http://www.itk.ilstu.edu/faculty/javila/SVG/index.htm>

6. Кариев Ч.А. Масштабируемая векторная графика (Scalable Vector Graphics): электронная книга / Ч.А. Кариев. – <http://www.intuit.ru/department/graphics/svg/1/>

Надійшла до редколегії 30.03.2012

Б.П. РУСИН, В.Б. ЯЦИВ

Физико-механический институт им. Г.В. Карпенко
НАН Украины, г. Львов, Украина

B.P. RUSYN, V.B. YATSIV

Karpenko Physical-Mechanical Institute Of the National Academy of Sciences of Ukraine

ОСНОВНЫЕ ПОДХОДЫ К РЕШЕНИЮ ЗАДАЧИ МАСШТАБИРОВАНИЯ ИЗОБРАЖЕНИЙ В ВЕБ - ОРИЕНТИРОВАННЫХ ТЕХНОЛОГИЯХ

Рассматриваются теоретические и практические аспекты использования SVG и HTML5 Canvas в современных веб-ориентированных технологиях. Предложен метод реализации алгоритма масштабирования изображения перед загрузкой на сервер на стороне клиента.

Ключевые слова: HTML5 Canvas, SVG (Scalable Vector Graphics), HTML (Hyper Text Markup Language), масштабирование изображения, веб-страница, File API.

BASIC APPROACHES TO SOLVING THE PROBLEM OF SCALING IMAGES IN WEB-BASED TECHNOLOGIES

The theoretical and practical aspects of SVG, HTML5 Canvas in modern Web-oriented technologies are considered. The method of image scales algorithm implementation before uploading to a server on the client side is proposed.

Keywords: HTML5 Canvas, SVG (Scalable Vector Graphics), HTML (Hyper Text Markup Language), image scaling, website, File API.