

Точка M_1^* будет лежать на поверхности, описывающий уровень радиоизлучения, фиксируемой на поверхности Земли.

Рассмотрим для примера шесть расположенных на удалении друг от друга обсерваторий, фиксирующих уровень радиоизлучения. Точка M_1^* имеет координаты $M_1^*(x_{M_1}; y_{M_1}; z_{M_1} + I_0)$. Если числовые значения z_{M_1} и I_0 имеет размерность, отличающуюся на порядок, тогда требуется для удобства обезразмеривание величины I_0 .

Функция:

$$f^2[x_-, y_-, z_-] := \left\{ x + \frac{(-2R - z)}{z - R} \cdot x; y + \frac{(-2R - z)}{z - R} \cdot y; -2R + I \right\}. \quad (12)$$

Поставит в соответствие каждой точке на сфере точку на поверхности S , соответствующей уровню радиоизлучения.

Применяя функцию f^2 к массиву точек, получим шесть точек на поверхности S . Затем строим поверхность, которая проходит через множество точек M_i^* , $i = \overline{1,6}$. Аппроксимирующая поверхность можно построить с помощью вейвлет-преобразования.

Полученная поверхность позволяет определить уровень радиоизлучения в промежуточных точках между обсерваториями.

УДК 330

РЕАЛИЗАЦИЯ МЕТОДОВ ОБРАБОТКИ КРУПНОМАСШТАБНЫХ ИЗОБРАЖЕНИЙ

Виничук О.Н

Белорусский государственный университет информатики и радиоэлектроники

филиал МРК

e-mail: memory1703@gmail.com

***Abstract.** Today it is quite difficult to choose a decent photo editor. There are various ways to customize from desktop versions to online versions. Each is convenient in its own way and not at the same time. Good quality image processing can affect both temporal characteristics and computer memory. Therefore, when developing this web application, first of all I would like to speed up the processing of good quality images, namely large images.*

На сегодняшний день достаточно сложно выбрать достойный фоторедактор. Существуют различные способы установок от десктоп-версий до онлайн версий. Каждые по-своему удобные и одновременно нет. Обработка изображений хорошего качества может затрагивать как временные характеристики, так и память компьютера. Поэтому, при разработке данного веб-приложения, в первую очередь хотелось бы ускорить процесс обработки изображений хорошего качества, а именно изображений большого объема.

Основные технологии, которые использовались при написании веб-редактора:

- HTML5;
- CSS3;
- JavaScript;
- JQuery;
- EaselJS.

HTML5, CSS3, JavaScript и библиотека JQuery – базовый набор «инструментов» при создании любого веб-приложения. Основной функционал, который отвечает за обработку изображений хранится в библиотеке EaselJS. Библиотека EaselJS была выбрана

не случайно, она облегчает работу с элементом canvas – элемент HTML5, предназначенный для создания растрового двумерного изображения при помощи JavaScript. Данная библиотека состоит из полного иерархического дисплейного списка, а именно набора векторов, используемых для построения изображения на экране; базовой модели взаимодействия и вспомогательных классов для значительного облегчения работы с Canvas. Работать с библиотекой EaselJS очень просто: необходимо определить элемент canvas в HTML и сослаться на него в JavaScript. EaselJS использует концепцию Stage, которая является Container высшего уровня для дисплейного списка (сценой будет элемент canvas). Ниже, на рисунке 1, представлена блок-схема, на которой описан алгоритм подключения и настройки библиотеки EaselJS.

У библиотеки EaselJS имеется класс Graphics, который предоставляет легкий в использовании API для создания инструкций векторного рисунка и прорисовывания их на указанный контекст. Команды очень похожи на те, что используются в обычном Canvas HTML5, а также в EaselJS имеются некоторые собственные команды для работы с Canvas.

В приложении реализованы функции по обработке изображений – фильтры. Данные фильтры реализованы посредством класса Filter. Класс Filter – это базовый класс, от которого наследуются свойства всех остальных фильтров. Фильтры применяются к объектам, которые были помещены в кэш при помощи метода cache. Если объект изменяется, необходимо будет поместить данный объект в кэш снова или использовать updateCache(). Хотелось бы отметить, что фильтры должны быть применены перед помещением в кэш. EaselJS предоставляет ряд предварительно созданных фильтров. Вышеописанный процесс представлен на рисунке 2.

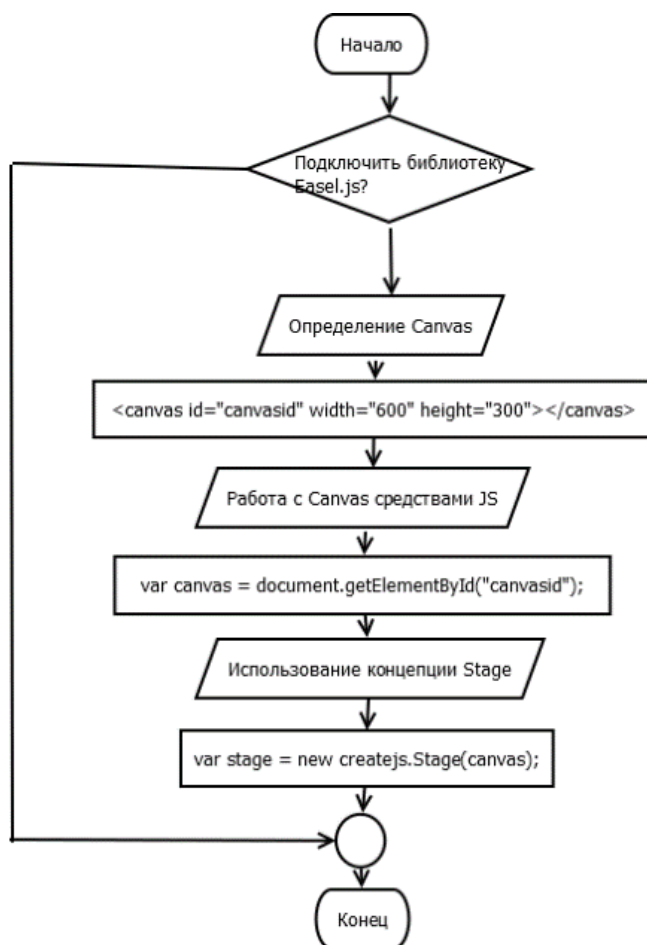


Рисунок 1 – Подключение библиотеки EaselJS.js

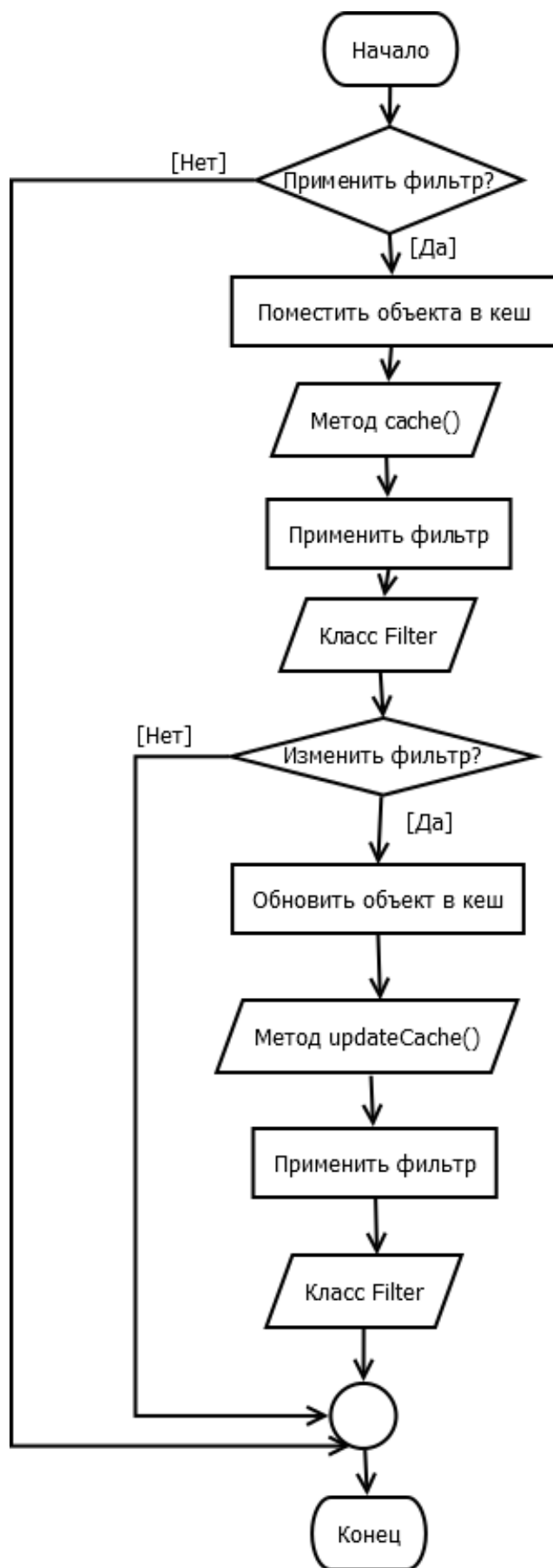


Рисунок 2 – Процесс применения фильтра к изображению

Интерфейс веб-приложения состоит из меню и рабочей области (рисунок 3).

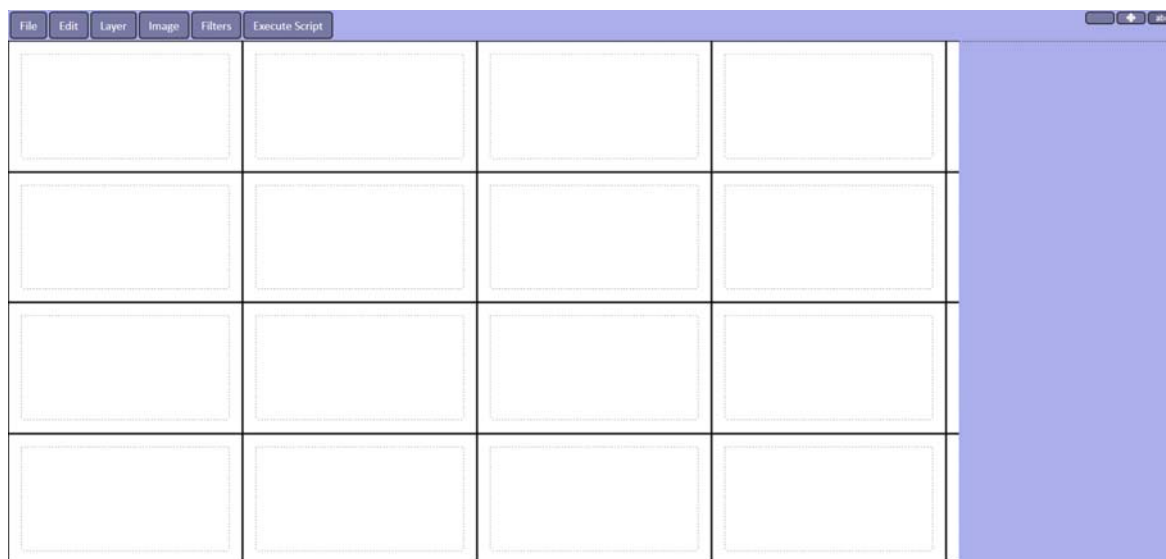


Рисунок 3 – Интерфейс веб-приложения

Справа, при добавлении изображений, отображаются слои (рисунок 4). Если на страницу добавлено сразу несколько слоев, для удобства работы реализованы функции:

- переименовать слой;
- скрыть/показать слой;
- удалить слой.

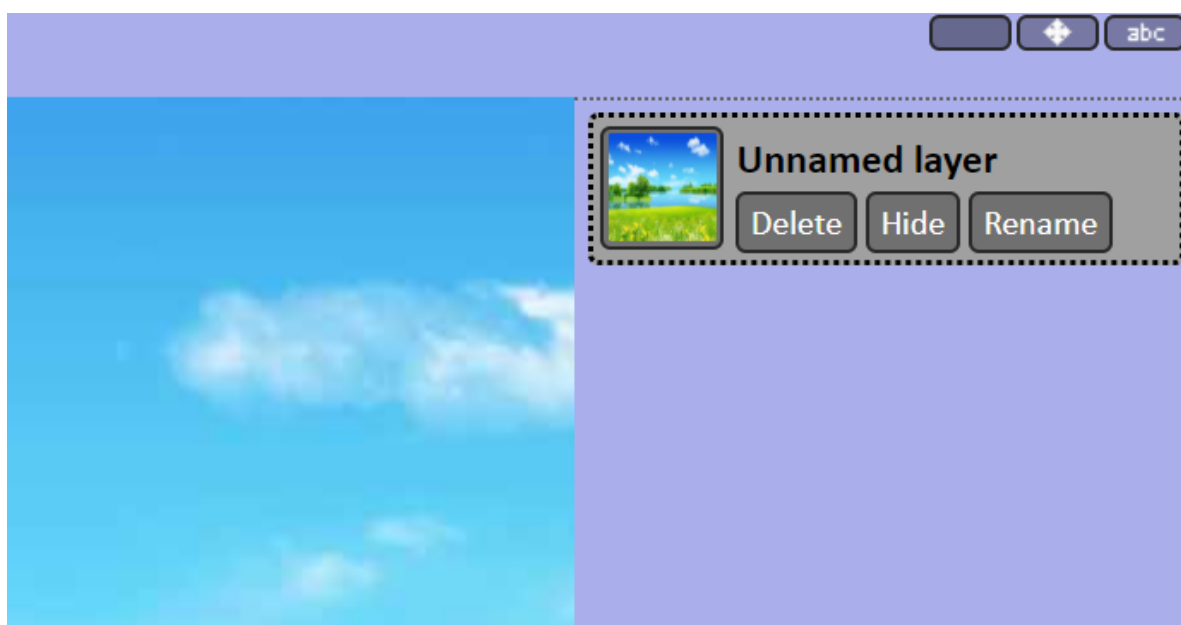


Рисунок 4 – Отображение слоев

С помощью веб-приложения можно легко настроить цвет, размер, яркость изображения. В меню представлены функции трансформирования и обработки изображения:

- Scale – изменение размера слоя/изображения;
- Rotate – поворот слоя/изображения на 2D-плоскости;
- Skew – искажение слоя/изображения;

- Flip vertically – отражение слоя/изображения по вертикали;
- Flip horizontally – отражение слоя/изображения по горизонтали;
- Brightness – изменение яркости изображения;
- Colorify – окрашивание изображения;
- Blur – размытие изображения;
- Gaussian Blur – Гауссовское размытие;
- Emboss – тиснение;
- Sharpen – резкость.

В разработанном веб-приложении встроены самые необходимые функции и методы для редактирования изображения. Можно сказать, что приложение по своим функциям, сравнимо с Photoshop, однако Photoshop потребляет намного больше ресурсов персонального компьютера, чем требуется для запуска веб-редактора. Также основным преимуществом является мобильность веб-приложения, ввиду того, что не требуется дополнительная установка приложения и сторонних программ.

УДК 004.891:004.93

АЛГОРИТМ АВТОМАТИЗИРОВАННОЙ ДИАГНОСТИКИ ЗАБОЛЕВАНИЙ ПОЯСНИЧНОГО ОТДЕЛА ПОЗВОНОЧНИКА НА ОСНОВЕ РЕНТГЕНОВСКИХ ИЗОБРАЖЕНИЙ

Курочка К.С., Панарин К.А., Карабчикова Е.А.

***Аннотация.** Предлагается алгоритм, позволяющий с использованием сверточной нейронной сети на основе регионов Darknet YOLO осуществлять локализацию позвонков на рентгеновских изображениях с последующим определением геометрических параметров с помощью библиотеки компьютерного зрения OpenCV.*

***Abstract.** Technology that allows to localize vertebrae on X-ray images and then determine geometric parameters using the OpenCV computer vision library in conjunction with the convolution neural network Darknet YOLO based on regions is proposed.*

Введение. При диагностике дегенеративных заболеваний поясничного отдела позвоночника человека на основе рентгеновских изображений требуется проведение оценки медицинским работником значительного числа геометрических параметров, получение которых вручную требует значительных временных затрат и не обеспечивает высокой точности [1]. Таким образом актуальной является задача автоматизации данного процесса.

Основным этапом подобной автоматизации является решение задачи о локализации отдельных позвонков позвоночника человека и оценки их взаимного расположения. Для решения данной задачи различными авторами использовались такие подходы как гистограммы ориентированных градиентов [2] и сверхточные нейронные сети с методом скользящего окна [3, 4], однако данные подходы требуют изображения высокой контрастности, а также обладают значительной ресурсоемкостью.

Локализация и параметризация позвонков на изображении. Наличие шумов и побочных объектов на снимках нарушают целостность контуров позвонков, что делает достаточно сложной задачу локализации с использованием таких методов как сегментация и контурный анализ. Для нахождения координат позвонков авторами была использована нейронная сеть на основе регионов Darknet YOLO [5], принцип работы которой состоит в разбиении изображения на фиксированные области, в которых затем происходит поиск наличия объекта. Обнаруженные объекты передаются классификатору, т.е. классификатору передаются лишь фрагменты изображения, гарантированно содержащие объекты, а не все изображение.