

В. А. Ястребов, В. П. Реутт – студенты кафедры безопасности информационных систем
М. Р. Гильмутдинов (канд. техн. наук) – научный руководитель

МЕТОД ПРОСТРАНСТВЕННОЙ ИНТЕРПОЛЯЦИИ ИЗОБРАЖЕНИЙ

В современном мире каждый из нас постоянно работает с цифровыми изображениями. Порой мы их создаем сами с помощью цифрового фотоаппарата, либо находим их, используя различные ресурсы, например, сеть Интернет. Довольно часто возникает необходимость в пост-обработке таких изображений, например, избавление от дефектов. Таким образом, задача пространственной интерполяции изображений является востребованной в современном мире обработки цифровых изображений. В данной работе представлены результаты реализации собственного алгоритма из класса inpainting.

Методы пространственной интерполяции класса inpainting позволяют решать следующие прикладные задачи:

- удаление нежелательных объектов с изображения;
- удаление наложенного на изображение текста;
- удаление механических дефектов, возникающих, например, при оцифровке старых фотографий (царапины, перегибы, потертости);
- удаление дефектов, вызванных проблемами передачи по сети;
- удаление импульсного шума, наложенного на изображение.

Здесь и далее будем придерживаться общепринятых терминов и определений, характерных для описания алгоритмов класса inpainting.

Исходная область – область на изображении, которая остается неизменной в ходе работы алгоритма.

Целевая область – область на изображении, которую необходимо изменить (интерполировать).

Основными принципами построения алгоритмов пространственной интерполяции изображений являются:

- анализ на основе *текстур*;
- анализ на основе *изофот*.

Под *текстурой* понимается изображение оптической структуры поверхности графического объекта, генерируемое в соответствии с его характером, условиями освещения и наблюдения [1]. Текстура определяет шероховатость или гладкость поверхности объекта. Различные виды текстур представлены на рис. 1,а [2].

Одним из подходов заполнения пикселей целевой области является нахождение такой подходящей замены, которая позволяет наиболее полно сохранить текстуру всего изображения. Другим базовым подходом является заполнение целевой области на основе анализа *изофот* – линий равной интенсивности на изображении. В этом случае, в первую очередь заполняются пиксели, находящиеся на линии значительного перепада яркости. На рис. 1b представлены различные изофоты.

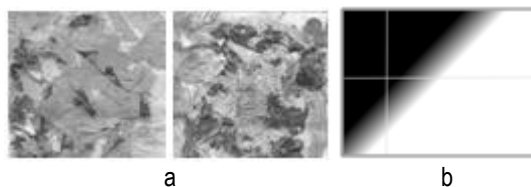


Рис.1

а) примеры текстур; б) примеры изофот

Надо отметить, что выбор порядка заполнения целевой области может значительно сказаться на результирующем изображении. Для задания необходимого порядка обхода каждому из целевых пикселей может быть задан приоритет заполнения. При текстурном анализе наибольший приоритет запол-

нения будет отдан пикселям, которые окружены наибольшим количеством пикселей, определяющих текстуру изображения. При использовании анализа на основе изофот, приоритет заполнения будет отдаваться тем пикселям, которые логически перекрывают продолжение изофоты.

Разработанный алгоритм базируется на алгоритме А. Crimnisi [4]. Основное отличие состоит в том, что за один шаг работы алгоритма производится замена только одного из целевых пикселей, а не нескольких пикселей сразу.

Рассмотрим основные шаги разработанного алгоритма (см. рис. 2).

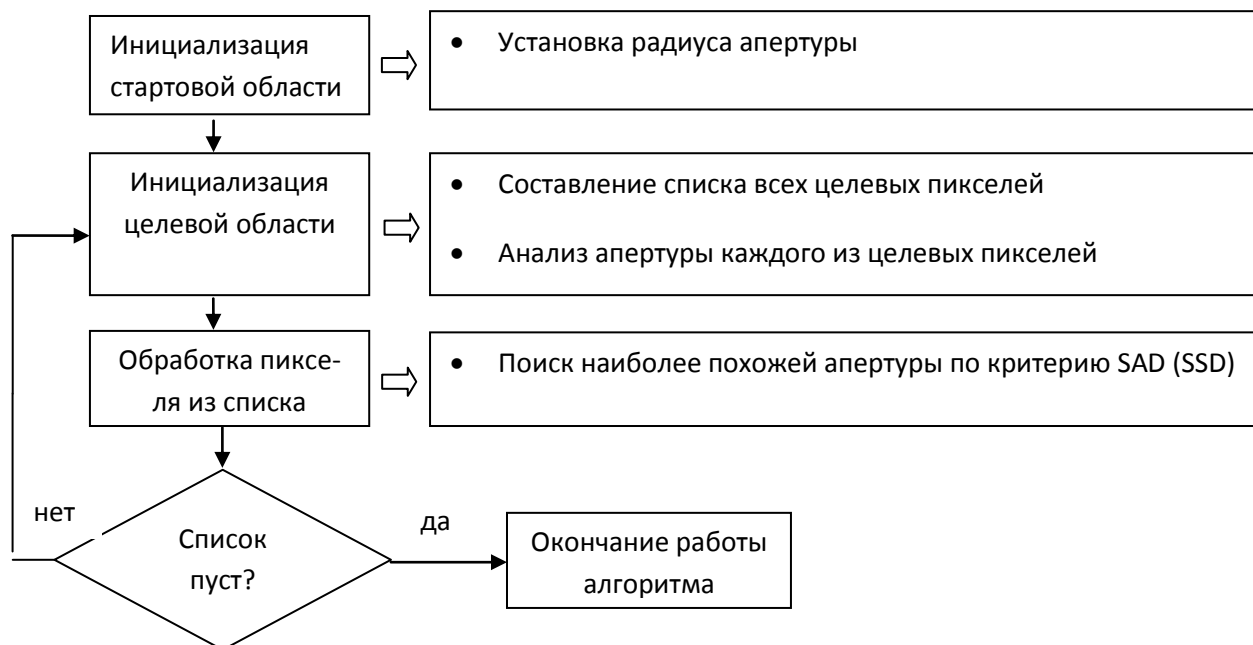


Рис. 2. Основные шаги работы алгоритма

Первым шагом инициализируется исходная область, а именно происходит установка радиуса апертуры и выставление степени недоверия исходным пикселям. Радиус апертуры определяет количество соседей пикселя, участвующих в выборе замены. Степень недоверия для исходных пикселей равна 1. Исходные пиксели пользуются наибольшим доверием.

Вторым шагом является инициализация целевой области. На этом этапе составляется список всех целевых пикселей, и на основе анализа апертуры каждого из них выбирается пиксель, подлежащий замене. Анализ апертуры состоит в определении количества “полезных соседей” пикселя, т.е. подсчете количества пикселей в радиусе апертуры, которые лежат в границах изображения и не являются целевыми.

Следующим шагом является поиск наилучшей замены по критерию SAD или SSD и, наконец, замена целевого пикселя.

До тех пор пока список целевых пикселей не будет пуст, инициализация целевой области будет производиться заново, с последующим выбором пикселя для замены.

В реализованном алгоритме критерием «похожести» двух пикселей является показатель SAD (sum of absolute differences), который рассчитывается на основе данных о соседях. Выражаясь строгим языком, для текущего обрабатываемого пикселя $I(x_1, y_1)$ из целевой зоны происходит поиск по всему изображению другого пикселя с условием $\min(\{SAD(h)\})$:

$$SAD(h) = \sum_{a=-R}^R \sum_{b=-R}^R Q(a, b) * W(x_2 + a, y_2 + b), \quad (1)$$

где $h = (1, \text{размеры нецелевой области})$; $Q(a, b) = |I(x_1 + a, y_1 + b) - I(x_2 + a, y_2 + b)|$ – разность значений яркости пикселя из апертуры исходной и целевой областей, взятой по абсолютному значению; R – радиус апертуры; $I(x, y)$ – значение яркости пикселя в точке с координатами (x, y) ; $W(x, y)$ – коэффициент недоверия для пикселя с координатами (x, y) .

Чем выше SAD для заполненного пикселя, тем больший коэффициент недоверия ему присваивается. Расстановка коэффициентов недоверия позволяет находить замену для целевого пикселя более точно. В случае если минимальный SAD имеют несколько пикселей, то для каждого из них рассчитывается SSD (sum of squared differences) и выбирается пиксель с условием $\min(\{SSD(h)\})$:

$$SSD(h) = \sum_{a=-R}^R \sum_{b=-R}^R (Q(a, b) * W(x_2 + a, y_2 + b))^2 \quad (2)$$

В качестве примера работы алгоритма рассмотрим изображение test.bmp, взятое из стандартного тестового набора изображений и наложим на него текст (см.рис.3). Нашей задачей является удаление текста с изображения.



Рис. 3. Изображение с наложенным текстом

Для сравнения были рассмотрены открытая реализация Q.Zhang [3] алгоритма A.Crimnisi [4] и коммерческий продукт Akvis Retoucher [5], который запускался в автоматическом (экспресс) режиме. Увеличенные фрагменты изображений с результатами работы различных алгоритмов приведены на рис.4.

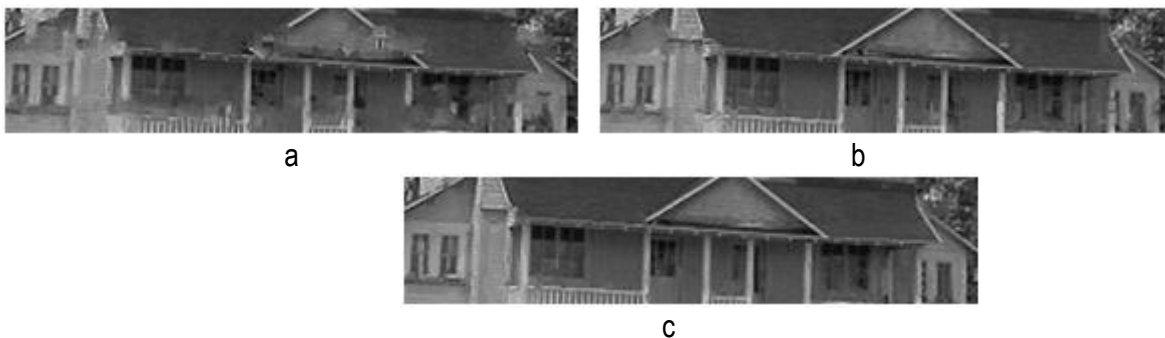


Рис. 4. Результат обработки

а) Akvis Retoucher; б) реализацией Q.Zhang; в) реализованным алгоритмом

Хорошо видно, что продукт Akvis Retoucher нарушил целостность картинки: колонны и фасад дома выглядят разрушенными. Результаты обработки алгоритмом A.Crimnisi выглядят лучше. Фасад и колонны целы, но текстура колонн не однородна и правый скат фасада несколько размыт. Реализованный алгоритм справился наилучшим образом. Все составные части изображения целы и эффект размытия отсутствует.

Теперь удалим наложенный импульсный шум с изображения peppers.bmp, также взятого из стандартного тестового набора. Сравнение работы разработанного алгоритма будем производить с модифицированным медианным фильтром [6] по критерию PSNR [7] (пиковое отношение сигнала к шуму). Модификация фильтра состоит в том, что подвергать изменению будем только зашумленные пиксели, не затрагивая оставшиеся неизмененные пиксели. Поскольку, в используемых изображениях максимальное значение яркости пикселя равняется 255, PSNR рассчитывался по следующей формуле:

$$PSNR = 20 \log_{10} \frac{255}{\sqrt{\frac{1}{n} \sum_{i=1}^n [(x_i) - x'_i]^2}} \quad (3)$$

где n – число пикселей изображения; x_i – i -ый пиксель исходного изображения; x'_i – i -ый пиксель восстановленного изображения.

На рис.5 представлено исходное изображение, а также изображения с различным количеством шумовых пикселей. В ходе работы были сгенерированы и обработаны изображения с 5%, 10%, 15%, 20%, 40% шумовых пикселей.

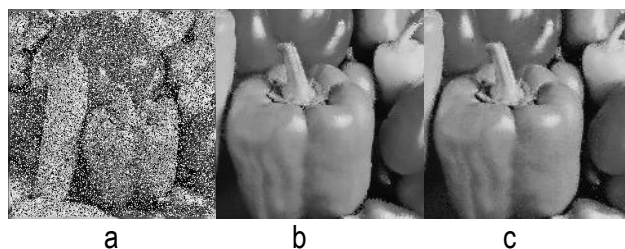


Рис.5.

а) изображение с 40% шума; б) фрагмент изображения, восстановленного с помощью модифицированного медианного фильтра; в) фрагмент изображения, восстановленного с помощью разработанного алгоритма

Результаты сравнения работы алгоритмов приведены на рис. 6.



Рис. 6. Критерий PSNR разработанного алгоритма и медианного фильтра при восстановлении изображения с различным количеством шумовых пикселей

Из графика (рис. 6) видно, что по сравнению с модифицированным медианным фильтром, разработанный алгоритм более точно восстанавливает исходное изображение, т.к. значения PSNR всегда лежат выше. Выигрыш на наиболее сильно зашумленном изображении составляет 0,63 dB.

Таким образом, в данной работе были проанализированы основные сферы применения алгоритмов пространственной интерполяции изображений класса inpainting, рассмотрены базовые подходы к построению таких алгоритмов, описаны шаги разработанного алгоритма. Также приведен сравнительный анализ работы алгоритма на изображении *test.bmp* на качественном уровне и на изображении с наложенным импульсным шумом *peppers.bmp* на основе критерия PSNR.

Библиографический список

1. <http://www.megabook.ru/Article.asp?AID=606874>. Меразнциклопедия Кирила и Мефодия.
2. <http://www.gfxtra.com/dl/texture+crumpled+paper>
3. http://read.pudn.com/downloads137/sourcecode/graph/texture_mapping/587532/inpaint/inpainting.cpp___.htm.
4. A. Criminisi, P. Perez, K. Toyama. Region Filling and Object Removal by Exemplar-Based Image Inpainting. *IEEE Transactions on Image Processing*, vol. 13, No9, Sep. 2004
5. <http://akvis.com/ru/retoucher/index.php>
6. Компьютерное зрение / Л. Шапиро, Дж. Стокман; Пер. с англ. –М.: БИНОМ. Лаборатория знаний, 2006. –752 с.
7. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. –М.: Техносфера, 2005. –1072 с.