Поиск сообществ на множестве пикселей изображения

С.В. Белим
belimsv@omsu.ru

С.Б. Ларионов
me@stas-larionov.ru

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация

В статье предложен метод анализа изображений с помощью графового представления. Вершинами графа служат пиксели изображения. Ребрами соединены только ближайшие соседи на изображении. Вес ребра определяется цветом, соединяемых вершин. Для полученного графа осуществляется поиск сообществ, позволяющий выявить фрагменты изображения, схожие по цвету. Предложенный подход применяется для подавления импульсного шума на изображении. Проведен компьютерный эксперимент. Показано, что предложенный метод более эффективен, чем традиционные фильтры.

Введение

К импульсному принято относить шум на изображениях, проявляющийся в виде изменения цвета отдельных пикселей, которые в дальнейшем будем называть испорченными. [1]. Устранение импульсного шума актуально не только с точки зрения улучшения визуального восприятия изображения, но как необходимый этап предварительной обработки в задачах сегментации, выделения контуров, распознавания образов и т.д.

Традиционный подход к устранению импульсного шума состоит в использовании сглаживающих фильтров. Наибольшее распространение получили фильтр Винера и медианный фильтр [2]. Главным недостатком сглаживающих фильтров является изменение не только испорченных пикселей, но и всего изображения в целом, что приводит к размытию контуров.

Для преодоления недостатков сглаживающих фильтров в последнее время развивается подход, основанный на предварительном поиске испорченных пикселей и преобразовании только их цветовых характеристик. Для поиска пикселей, подвергшихся влиянию импульсного шума, используются методы Data Mining. В этом случае обработка изображений разбивается на два этапа. Во-первых, осуществляется поиск испорченных пикселей. Во-вторых, вносятся изменения в цветовые характеристики найденных пикселей. На обоих этапах необходим анализ пикселей окружающих данных. На первом этапе для выделения пикселей значительно отличающихся от соседних, на втором этапе – для определения нового значения цвета испорченного пикселя. На сегодняшний день разработано несколько алгоритмов выявления поврежденных пикселей: SD-ROM [3, 4], на основе метода анализа иерархий [5], на основе ассоциативных правил [6], на основе сегментации изображений [7]. Для восстановления цвета испорченного пикселя, как правило используются различные методы интерполяции данных [8, 9].

В данной статье приведен алгоритм устранения импульсного шума на основе использования алгоритма поиска сообществ на графах. Данный подход позволил построить эффективный алгоритм сегментации изображений [10].

In: Sergey V. Belim, Nadezda F. Bogachenko (eds.): Proceedings of the Workshop on Data, Modeling and Security 2017 (DMS-2017), Omsk, Russia, October 2017, published at http://ceur-ws.org

1 Постановка задачи и алгоритм фильтрации

Будем применять алгоритм фильтрации к зашумленному изображению размерами $N \times M$ пикселей. Положение каждого пикселя на изображении может быть задано парой целых чисел (x, y), которые можно считать геометрическими координатами. Число x лежит на отрезке $[0, N-1]$, а число y – на отрезке $[0, M-1]$. Будем использовать цветовую модель RGB. В этом случае для каждого пикселя с координатами (x, y) определены три цветовые функции: $r(x, y)$ – интенсивность красного цвета, $g(x, y)$ – интенсивность зеленого цвета, $b(x, y)$ – интенсивность синего цвета.

Для каждого изображения однозначным образом может быть построен неориентированный взвешенный граф G, показывающий изменения цвета при переходе от одного пикселя к другому. Множество вершины графа совпадает с множеством пикселей изображения. Ребрами будем соединять только ближайшие соседние пиксели. Вес ребра будем вычислять на основе цветовых компонент пикселей, соединенных этим ребром. Для двух соседних вершин $v_i = (x_i, y_i)$ и $v_j = (x_j, y_j)$ вес ребра будет равен:

$$d(v_i, v_j) = \exp(-\frac{1}{h}\sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}).$$

Здесь $r_i = r(x_i, y_i)$, $g_i = g(x_i, y_i)$, $b_i = b(x_i, y_i)$.

Параметр h определяет значение разности цвета между соседними пикселями, соответствующее переходу через границу двух сегментов. Данный параметр определяется пользователем и является общим для всего изображения. Как показано в работах [7, 10] такой вид весовой функции позволяет достаточно точно различать изменение цвета, соответствующее границам областей на изображении. Будем разбивать граф на подграфы, вершины которых связаны между собой сильнее, чем с остальными. Такие подграфы принято называть сообществами (community). Качество разбиения графа количественно может оценено с помощью функции модульности Ньюмана [11, 12]. Чем больше значение функции модульности, тем более качественно осуществлено разбиение. Поврежденными будем считать пиксели, объединение которых с любыми другими сообществами понижает значение функции модульности. Таким образом, будут выделены пиксели, образующие сообщества из одной вершины, то есть сильно отличающиеся от своих ближайших соседей. Опишем эту процедуру формально.

Определим матрицу весов E для графа G. Значения диагональных элементов E_{ii} равно весу вершин. В начале работы алгоритма вес всех вершин нулевой. Остальные элементы матрицы весов E_{ij} $(i \neq j)$ равны весу соответствующего ребра. Следует отметить, что в матрице E_{ij} будет много нулевых элементов, так как в графе G ребрами соединены только вершины, соответствующие ближайшим соседям на изображении. Матрица E будет симметричной относительно главной диагонали, так как граф G неориентированный. Перейдём к приведенному виду матрицы весов $e = E/m$, где

$$m = \sum_{i,j=1}^{MN} E_{ij}.$$

Элемент e_{ij} равен доле веса заданного ребра в общем весе графа. В дальнейшем под матрицей весов будет пониматься именно приведенный вид. Легко видеть, что

$$\sum_{i,j=1}^{MN} e_{ij} = 1.$$

Величина модульности определяется выражением [11, 12]:

$$Q(G) = \sum_{i=1}^{K} e_{ij} - \sum_{i=1}^{K} a_i b_i.$$

где K – количество вершин графа, a_i – приведенная исходящая степень вершины v_i:

$$a_i = \sum_{j=1, j\neq i}^{K} e_{ij},$$

b_i – приведенная входящая степень вершины v_i

$$b_i = \sum_{j=1, j\neq i}^{K} e_{ji}.$$

Рассматриваемый граф является неориентированным, поэтому входящая и исходящая степень для всех вершин одинаковы ($a_i = b_i$; $i = 1, ..., K$). Функция модульности принимает более простой вид:

$$Q(G) = \sum_{i=1}^{K} e_{ij} - \sum_{i=1}^{K} a_i^2.$$

Для поиска сообществ на графе используем процедуру образования стяжек. Под стяжкой понимается преобразование, при котором некоторый подграф H графа G заменяется одной вершиной v_H. Если какая-то из вершин подграфа H была связана ребром с вершиной v из подграфа $G\backslash H$, то вершина v_H также будет связана ребром того же веса с вершиной v. Вес новой вершины будет равен сумме весов ребер и вершин, входящих в подграф H. Новый граф обозначим G_H. Подграф H будем считать сообществом, если $Q(G_H) > Q(G)$. Следует отметить, что при образовании стяжки уменьшается число вершин графа K. Нашей задачей стоит поиск вершин графа, не входящих ни в одно более крупное сообщество. Выявление таких вершин будем осуществлять с помощью следующего алгоритма.

1. Осуществляем последовательный проход по всем пикселям изображения.

2. Для каждого пикселя v поочередно рассматриваем ближайшие соседние пиксели $v^{(i)}$ ($i = 1, ..., 8$). Рассматриваем подграфы, состоящие из двух вершин v и $v^{(i)}$ ($i = 1, ..., 8$) и пытаемся объединить их в сообщество. При каждом объединении вычисляем изменение функции модульности ΔQ_i ($i = 1, ..., 8$).

3. Если при объединении данной вершины в сообщества со всеми ближайшими соседями изменение функции модульности отрицательно ($\Delta Q_i < 0$, $i = 1, ..., 8$), то соответствующий пиксель считаем поврежденным.

Изменение функции модульности достаточно быстро вычисляется из текущих характеристик графа, соответствующего изображению. При объединении вершин v_i и v_j изменение модульности будет иметь вид:

$$\Delta Q = 2(e_{ij} - a_i a_j).$$

Очевидно, что предложенный алгоритм имеет линейную трудоемкость от числа пикселей.

После выявления поврежденных пикселей необходимо выбрать для них цвет из анализа окружающих пикселей. Для этого проанализируем ближайших соседей. Пусть минимальное значение цветовой компоненты пикселей соседних с поврежденным пикселем равно m_1, а максимальное – m_2. Проведем последовательный перебор всех значений цвета поврежденного пикселя от m_1 до m_2. Для каждого значения будем вычислять значение изменения функции модульности ΔQ при объединении данной вершины в сообщество с одним из ближайших соседей. В качестве окончательного цвета оставим тот, который позволяет получить максимальное значение изменения функции модульности.

2 Компьютерный эксперимент

Компьютерный эксперимент проводился как на искусственных изображениях геометрических объектов, так и на цветных фотографиях. Уровень импульсного шума характеризовался величиной p, показывающей процент поврежденных пикселей по отношению к общему количеству пикселей изображения. Импульсный шум моделировался с помощью линейного конгруэнтного генератора псевдослучайных чисел, с помощью которого определялись как координаты поврежденных пикселей, так и их цвет. При проведении компьютерного эксперимента процент испорченных пикселей варьировался от 10% до 70%. Улучшение изображения производилось с помощью предложенного фильтра, а также, для сравнения, с помощью широко распространенного медианного фильтра.

Для сравнения близости изображений использовалась метрика Минковского [13, 14], согласно которой расстояние между изображениями A и C находится по формуле

$$d(A, C) = \max_{mn} \sum_{k=1}^{N} \frac{1}{N} |A_{mn}^{(k)} - C_{mn}^{(k)}|.$$

где A_{mn} и C_{mn} – значения цветов пикселей изображения A и C, N – количество пикселей.

Относительное улучшение изображения вычислялось на основе расстояния $d(orig_{fig}, r_{fig})$ от восстановленного изображения r_{fig} до исходного $orig_{fig}$ и расстояния $d(orig_{fig}, p_{fig})$ от испорченного изображения p_{fig} до исходного изображения $orig_{fig}$:

$$\delta = \frac{d(orig_{fig}, p_{fig}) - d(orig_{fig}, r_{fig})}{d(orig_{fig}, p_{fig})} \cdot 100\%.$$

Эксперимент для прямоугольной области с равномерной заливкой показал, что предложенный фильтр позволяет значительно улучшить изображение. Зависимость относительного улучшения от процента испорченных пикселей для предложенного фильтра и медианного фильтра представлены на рисунке 1.

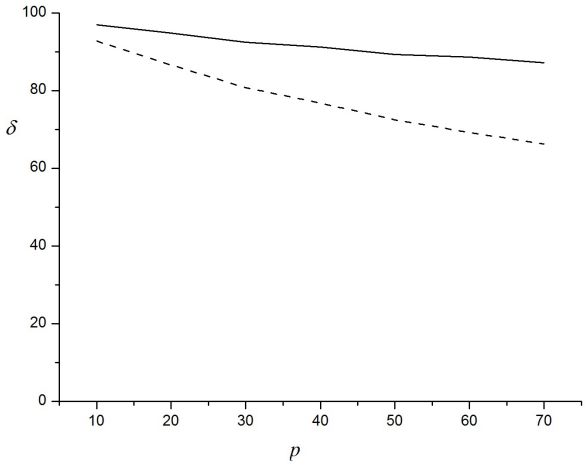

Рис. 1: Зависимость относительного улучшения изображения от процента зашумления для предложенного фильтра (сплошная линия) и медианного фильтра (пунктирная линия)

Результаты применения предложенного фильтра и медианного фильтра для улучшения искусственного изображения с наличием сплошной и градиентной заливки представлены на рисунке 2.

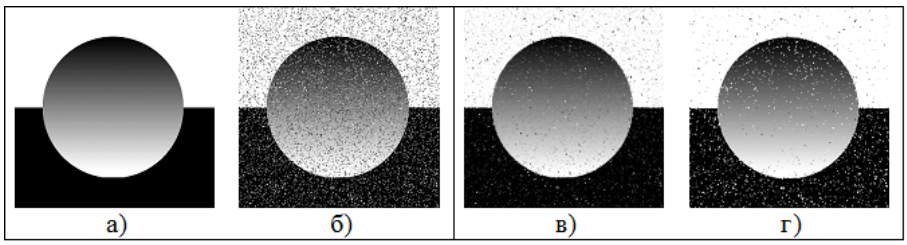

Рис. 2: Результаты применения фильтра к искусственному изображению с уровнем шума p=20%: а) исходное изображение, б) зашумленное изображение, в) изображение восстановленное предложенным фильтром, г) изображение восстановленное медианным фильтром

Как хорошо видно из рисунка 2 результаты предложенного фильтра более выигрышно выглядят для темных участков изображения, тогда как медианный фильтр дает лучший визуальный результат для светлой части изображения. Этот эффект связан с выбором нового цвета испорченного пикселя. При использовании медианного фильтра цвета восстановленных пикселей смещены в область белого цвета. При этом цвета окружающих его пикселей также изменяют свой цвет. Численное сравнение результатов работы показывает значительное преимущество предложенного алгоритма перед медианным фильтром.

Зависимость относительного улучшения от процента зашумления искусственного изображения представлена на рисунке 3.

Также данный фильтр позволяет получить значительно лучшие результаты и для фотографических изображений. Результаты для хорошо известного изображения «Lena» при уровне зашумления $p = 20\%$ представлены на рисунке 4.

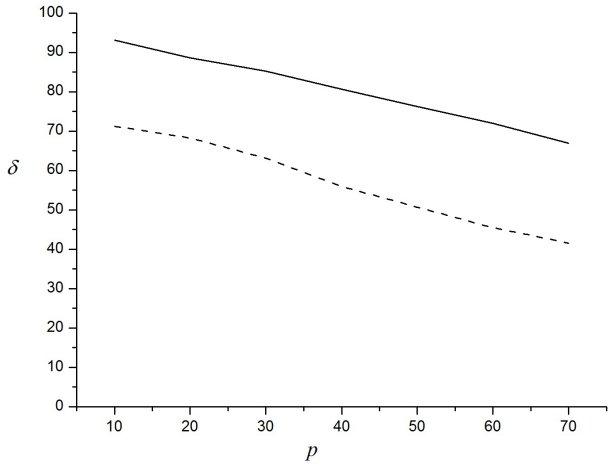

Рис. 3: Зависимость относительного улучшения от процента зашумления искусственного изображения для предложенного фильтра (сплошная линия) и медианного фильтра (пунктирная линия)

Рис. 4: Результаты применения фильтра к изображению «Lena» с уровнем шума $p = 20\%$: а) исходное изображение, б) зашумленное изображение, в) изображение, восстановленное предложенным фильтром, г) изображение, восстановленное медианным фильтром

Хорошо известно, что изображение «Lena» характеризуется большим количеством мелких деталей, которые создают сложности для всех фильтров. Как видно из рисунка 4 предложенный фильтр дает значительно лучшие результаты даже при визуальном сравнении. Зависимость относительного улучшения от процента зашумления представлена на рисунке 5.

3 Выводы

Таким образом, предложенный фильтр обладает хорошими характеристиками при линейной трудоемкости. Как видно из графиков, представленных на рисунках 1, 3 и 5 эффективность данного фильтра примерно на 20% выше, чем медианного при любом проценте испорченных пикселей. Такое заметное преимущество объясняется тем, что обычные фильтры изменяют все пиксели изображения. Исправление испорченных пикселей приближает изображение к оригиналу, но при этом изменение неиспорченных пикселей увеличивает расстояние до оригинала. Данное свойство присуще не только медианному фильтру, но и всем традиционным фильтрам.

Предложенный в данной статье фильтр действует избирательно и изменяет только те пиксели, которые значительно отличаются от окружающих. С высокой вероятностью такие пиксели окажутся поврежденными импульсным шумом. Выбор нового цвета на основе присоединения к одному из соседних сообществ пикселей позволяет формировать группы близких по характеристикам точек.

Список литературы

[1] I. Pitas, A. Venetsanopoulos. *Nonlinear Digital Filters: Principles and Applications.* Boston, MA: Kluwer, 1990, 392 p.

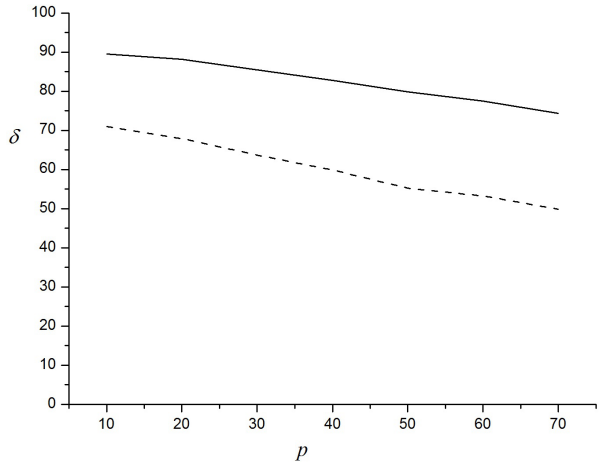

Рис. 5: Зависимость относительного улучшения от процента зашумления изображения «Lena» с помощью предложенного фильтра (сплошная линия) и медианного фильтра (пунктирная линия)

[2] R. Boyle, M. Sonka, V. Hlavac. *Image Processing, Analysis, and Machine Vision, First Edition.* University Press, Cambridge, 2008, 920 p.

[3] E. Abreu, M. Lightstone, S.K. Mitra, S.K. Arakawa. A new efficient approach for the removal of impulse noise from highly corrupted images. *IEEE Transactions on Image Processing*, 5:1012–1025, 1996.

[4] R. Garnett, T. Huegerich, C. Chui, W. He. A Universal Noise Removal Algorithm with an Impulse Detector. *IEEE Trans Image Proccess*, 14(11):1747–1754, 2005.

[5] S.V. Belim, S.A. Seliverstov. Hierarchy Analysis Method as a Way to Detect Impulse Noise on Images. *Information technology*, 4(21):251–258, 2015.

[6] S.V. Belim, A.O. Mayorov-Zilbernagel. Algorithm for Searching the Broken Pixels and Eliminating Impulse Noise in Images Using a Method of Association Rules. *Science and Education of the Bauman MSTU*, 12:716–737, 2014. URL: `http://technomag.bmstu.ru/doc/744983.html`.

[7] S.V. Belim, P.E. Kutlunin. Impulse noise detection in image using a clustering algorithm. *Herald of computer and information technologies*, 3:3–10, 2016.

[8] S.V. Belim, A.O. Mayorov-Zilbernagel. Image Restoration With Static Gaps On The Basis Of Association Rules. *Herald of computer and information technologies*, 12:18–23, 2014.

[9] S.V. Belim, S.A. Seliverstov. The Analytic Hierarchy Method-Based Algorithm for Restoring Broken Pixels on the Noisy Images. *Science and Education of the Bauman MSTU*, 11:521–534, 2014. URL: `http://technomag.bmstu.ru/doc/742145.html`.

[10] S.V. Belim, S.B. Larionov. An algorithm of image segmentation based on community detection in graphs. *Computer Optics*, 40(6):904–910, 2016.

[11] M.E.J. Newman. Mixing patterns in networks. *Phys. Rev. E.*, 67:026126-1–026126-13, 2003.

[12] A. Clauset, M.E.J. Newman, C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004.

[13] V. DiGesu, V.V. Staravoitov. Distance-based Functions for Image Comparison. *Pattern Recognition Letters*, 20:207–213, 1999.

[14] J. Ryu, S. Kim, H. Wan. Pareto front approximation with adaptive sum method in multiobjective simulation optimization. *Proc. of the 2009 Winter Simulation Conference (WSC)*, 623–633, 2009.

The Communities Search on a Pixels Set of the Image

Sergey V. Belim, Stanislav B. Larionov

In article the images analysis method by means of graph representation is suggested. Image pixels are compared to graph vertex. By edges only the closest neighbors on the image are connected. Weight of an edge is defined by color, the connected vertex. For the received graph search of communities is carried out, allowing to reveal the image fragments similar in color. The suggested approach is applied to impulse noise suppression on the image. The computer experiment is made. It is shown that the suggested method is more effective, than traditional filters.