

Алгоритм поиска импульсного шума на изображении

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 алгоритм выявления импульсного шума. Изображение представляется в виде графа. На графе ищутся сообщества. Сообщества, состоящие из одного пикселя считаются поврежденными. Проведен компьютерный эксперимент. Алгоритм показал высокую эффективность.

Введение

Поиск на изображении пикселей, поврежденных импульсным шумом, можно отнести к задачам обучения без учителя. Основной подход, используемый в подавляющем количестве алгоритмов, состоит в анализе скользящего окна [1, 2]. Под скользящим окном понимается некоторая локальная область изображения последовательно проходящая все пиксели. Из сравнения характеристик центрального пикселя окна с остальными пикселями окна делается вывод о его повреждении. Отличие методов детектирования поврежденных пикселей друг от друга состоит в выборе размеров скользящего окна и алгоритма принятия решения о повреждении пикселя [3, 4, 5]. Такие алгоритмы работают достаточно быстро, так как имеют линейную трудоемкость по количеству пикселей изображения. Однако эффективность поиска поврежденных пикселей существенно зависит от пользовательских настроек. Например, алгоритм SD-ROM [1], лежащий в основе большого количества других алгоритмов, содержит четыре параметра, выбираемых пользователем. Варьирование данных параметров может менять эффективность поиска поврежденных пикселей от 62% до 96%. При этом остается высоким процент ложных срабатываний.

Улучшение результатов удастся добиться на основе алгоритмов, анализирующих не только небольшую локальную область, но и все изображение в целом. В качестве таких можно отметить алгоритмы на основе ассоциативных правил [6, 7], методах поддержки принятия решений [8], а также основанные на методах сегментации изображений [9, 10]. Во всех перечисленных алгоритмах один или два настраиваемых параметра и достаточно высокая эффективность обнаружения поврежденных пикселей. При этом удастся существенно снизить процент ложных срабатываний. Следует отметить, что перечисленные алгоритмы обладают не линейной, а квадратичной трудоемкостью.

Целью данной статьи является разработка алгоритма выделения поврежденных пикселей на основе графового представления изображения с последующей его кластеризацией.

Из методов на сегментации изображений можно отметить представление изображения на основе взвешенного графа с последующим выделением сообществ (community) на данном графе [11, 12, 13]. Выделение сообществ в большинстве случаев реализуется на основе выращивания областей на исходном изображении. Данный подход имеет квадратичную трудоемкость. Поврежденные пиксели выделяются как сегменты, состоящие из одной точки.

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: Sergey V. Belim, Nadezda F. Bogachenko (eds.): Proceedings of the Workshop on Data Analysis and Modelling (DAM 2016), Омск, Russia, October 2016, published at <http://ceur-ws.org>

1 Постановка задачи и алгоритм решения

Импульсный шум на изображениях проявляется как отдельные случайно расположенные пиксели случайного цвета. При этом информация о прежнем значении пикселя становится утерянной. Обозначим отношение количества измененных пикселей к общему количеству пикселей изображения через p и будем называть процентом зашумления.

На входе алгоритма имеем изображение размером $N \times M$ пикселей, на котором присутствует импульсный шум.

Положение пикселя на изображении может быть задано двумя координатами (x, y) , где x принимает целые значения на отрезке $[0, N - 1]$, y принимает целые значения на отрезке $[0, M - 1]$. Для цветных изображений каждому пикселю с координатами (x, y) соответствует три цветовые составляющие: $r(x, y)$ – красный цвет, $g(x, y)$ – зеленый цвет, $b(x, y)$ – синий цвет.

Представим изображение в виде неориентированного взвешенного графа G . Вершинами графа будут служить пиксели изображения. Ребрами соединим только вершины, соответствующие ближайшим соседним пикселям на изображении. Вес ребра будет определяться отличием цветовых составляющих концов ребра. Как показано в статье [14] наилучшие результаты могут быть получены при использовании экспоненциальной функции. Вес ребра, соединяющего вершины $v_i = (x_i, y_i)$ и $v_j = (x_j, y_j)$, определим в следующем виде:

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

Здесь $r_i = r(v_i, v_j)$, $g_i = g(v_i, v_j)$, $b_i = b(v_i, v_j)$. Параметр h определяет значение разности цветовых составляющих, детектируемое как переход к соседнему сегменту. Данный параметр определяется пользователем и является общим для всего изображения.

Выделение испорченных пикселей будем осуществлять на основе алгоритма сегментации изображения. При сегментации все пиксели разбиваются на группы схожих элементов. В качестве меры схожести выбирается «цветовое расстояние» между пикселями $d(v_i, v_j)$. В результате сегментации зашумленного изображения некоторые сегменты будут состоять из одной точки. С высокой вероятностью в качестве однопиксельных сегментов будут выделены импульсные шумы. Однако стоит отметить, что большинство фотографических изображений содержит элементы, которые в отсутствие шумов также определяются, как однопиксельные кластеры. Наличие таких элементов будет приводить к ложным срабатываниям.

При представлении изображения в виде графа сегментация эквивалентна кластеризации графа G . Граф G разбивается на подграфы, расстояния между вершинами которых значительно меньше, чем с остальными вершинами. Такие подграфы образуют сообщества (community). Для количественной оценки разбиения графа на сообщества используется функция модульности Ньюмана [15, 16].

Для вычисления функции модульности необходимо для графа G задать матрицу весов E . Диагональные элементы матрицы весов E_{ii} показывают вес i -ой вершин. На начальном этапе вес всех вершин нулевой. Недиagonalные элементы E_{ij} ($i \neq j$) имеют вес, равный цветовому расстоянию между пикселями, соответствующим вершинам v_i и v_j . Матрица E будет симметричной относительно главной диагонали, так как у ребер нет ориентации. Для вычисления функции модульности введем приведенную матрицу весов $e = E/m$, где

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

В приведенной матрице весов элемент e_{ij} показывает долю веса заданного ребра в общем весе графа. В дальнейшем будем рассматривать только приведенный вид матрицы весов. Очевидно, что

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

Величина модульности определяется выражением [15, 16]:

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

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 ($b_i = \sum_{j=1, j \neq i}^K e_{ji}$). Так как граф G неориентированный,

входящая и исходящая степень для всех вершин одинаковы ($a_i = b_i; i = 1, \dots, K$). Функция модульности принимает вид:

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

Для выявления сообществ на графе будем использовать 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 $G \setminus H$, 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.

Будем находить сообщества методом выращивания областей.

1. Выбираем произвольный пиксель изображения v , не входящий ни в одно сообщество.

2. Для выбранного пикселя v поочередно рассматриваем ближайшие соседние пиксели $v^{(i)}$ ($i = 1, \dots, 8$) и пытаемся объединить их в сообщество. При каждом объединении вычисляем изменение функции модульности ΔQ_i ($i = 1, \dots, 8$).

3. Если объединение данной вершины в сообщество с другими пикселями не выгодно ($\Delta Q_i < 0, i = 1, \dots, 8$), то соответствующий пиксель относим к поврежденным. Иначе строим сообщество

На выходе алгоритма получим список пикселей, образующих сообщества из одного элемента. С большой вероятностью данные пиксели являются искомыми.

Изменение функции модульности может быть вычислено достаточно быстро на основе текущих характеристик графа. При объединении вершин v_i и v_j изменение модульности будет равно:

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

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

2 Компьютерный эксперимент

В компьютерном эксперименте обрабатывались и искусственные изображения и фотографические файлы размером $N = 512, M = 512$ пикселей. Координаты и цвет pNM точек, моделирующих импульсный шум генерировались с помощью линейного конгруэнтного генератора с равномерным распределением. Эксперименты проводились с уровнем шума от 0% до 70%. Для каждого набора параметров и каждого изображения проводилось по 100 независимых экспериментов.

В каждом эксперименте генерировались pNM координат и цветов пикселей. После этого осуществлялось выявление поврежденных пикселей. Результаты сравнивались со сгенерированными данными. Для оценки качества работы алгоритма использовались две величины. Эффективность работы алгоритма:

$$eff = \frac{N_0}{pNM},$$

где N_0 – количество верно идентифицированных испорченных пикселей. Уровень ложных срабатываний:

$$err = \frac{N_e}{N},$$

где N_e – количество ошибочно идентифицированных испорченных пикселей, N – общее количество пикселей определенное алгоритмом, как испорченные.

Для изображения, представляющего собой область равномерной заливки одним цветом эффективность работы алгоритма при $p = 10\%$ составляет $eff = 99.98\%$ при проценте ложных срабатываний $err = 0.11\%$. Эффективность ниже ста процентов, так как при случайном выборе цвета испорченного пикселя ненулевая вероятность совпадения его с цветом заливки. Эффективность работы алгоритма остается высокой вплоть до больших значений уровня шума. Так при $p = 70\%$ эффективность равна $eff = 99.97\%$ при проценте ложных срабатываний $err = 6.20\%$. Рост процента ложных срабатываний связан с тем, что при высоком уровне шума испорченные пиксели достаточно часто оказываются ближайшими соседями.

Аналогичные эксперименты для искусственных изображений правильных геометрических фигур с четкими границами и присутствием областей градиентной заливки при с уровнем шума от $p = 10\%$ до $p = 70\%$

показали эффективность от $eff = 99.97\%$ до $eff = 99.95\%$. Процент ложных срабатываний, при этом, также снижался от $err = 7.81\%$ до $err = 4.15\%$. Основной вклад в величину процента ложных срабатываний вносили четкие границы, разделяющие области изображения.

Эксперименты на фотографических изображениях показали, что эффективность выявления поврежденных пикселей остается достаточно высокой и медленно убывает с ростом уровня шума. Даже для процента зашумления $p = 50\%$ эффективность $eff = 83.85\%$. Значение процента ложных срабатываний также убывает, что связано с его относительным характером. По абсолютной величине количество ошибочно определенных пикселей остается практически на одном уровне.

3 Анализ результатов и выводы

Предложенный метод выявления пикселей, поврежденных импульсным шумом, обладает достаточно высокой эффективностью даже для сильно зашумленных изображений. Причем количество пикселей, ошибочно отнесенных к поврежденным, растет очень медленно с увеличением зашумления, что приводит к уменьшению уровня ложных срабатываний. Проведем сравнение с результатами других работ. Наиболее распространенный алгоритм SD-ROM позволяет обнаруживать испорченные пиксели в среднем с эффективностью 71% и уровне ложных срабатывания 5% при интенсивности шума 10% [1]. Этот же метод может демонстрировать эффективность до 96% при уровне зашумления 30%, но на шумах типа «соль и перец». Следует отметить, что задача анализа изображений с шумом «соль и перец» является более простой, так как в них интенсивность цвета испорченных пикселей имеет либо максимальное, либо минимальное значение. Причем достижение максимальной эффективности возможно лишь при определенном наборе четырех параметров, подбираемых вручную. При увеличении уровня шума уровень ложных срабатываний в алгоритме SD-ROM растет очень быстро и уже при интенсивности шума 30% достигает 73%. Аналогичные алгоритмы, основанные на анализе локального окружения каждого пикселя, демонстрируют близкие результаты. Так метод, использующий построение ассоциативных правил для наборов соседних пикселей, при интенсивности шума 30%, показывает эффективность 77% и уровень ложных срабатываний 35% [6, 7]. Алгоритм, основанный на методе анализа иерархий позволяет обнаруживать импульсный шум с эффективностью 85% и ложными срабатываниями 55% [8]. Существенным недостатком данных методов является анализ достаточно небольшой области изображения вокруг каждого пикселя, который не позволяет выявлять и анализировать протяженные области на изображении. Частично данный недостаток был устранен в методе выявления импульсного шума, основанном на кластеризации изображений [9, 10], для которого эффективность обнаружения испорченных пикселей 82%, а уровень ложных срабатываний 23% при зашумлении на 10%. При увеличении процента испорченных пикселей эффективность падает и при 30% зашумления составляет 68%, а уровень ложных срабатываний 11%.

Таким образом, предложенный в данной статье алгоритм позволяет получать заметный выигрыш как в эффективности обнаружения испорченных пикселей, так и в уровне ложных срабатываний при значительном зашумлении изображения, выше 30%.

Список литературы

- [1] Abreu E., Lightstone M., Mitra S.K., Arakawa S.K. A new efficient approach for the removal of impulse noise from highly corrupted images. *IEEE Transactions on Image Processing, IEEE Transactions on.* 5:1012–1025, 1996.
- [2] Garnett R., Huegerich T., Chui C., He W. A Universal Noise Removal Algorithm with an Impulse Detector. *IEEE Trans Image Process.* 14(11):1747–1754, 2005.
- [3] Sorokin S.V., Shherbakov M.A. Realizacija SD-ROM fil'tra na osnove koncepcii nechetkoj logiki. (The SD-ROM filter implementation based on the fuzzy logic concept.) *Izvestija vysshih uchebnyh zavedenij. Povolzhskij region.* 3:56–65, 2007.
- [4] Krasovskij G.Ja., Uss M.L. Fil'tracija izobrazhenij, iskazhennyh impul'snymi pomehami tochechnogo i strochnogo tipa, na osnove sistem iterirovanyh funkcij. (Filtration of broken by string and dotted impulse images based on system of iterated functions.) *Radioelektronni i komp'juterni sistemi.* 2:47–55, 2003.
- [5] Chan R., Ho C., Nikolova M. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Trans. Image Proc.* 14(10):1479–1485, 2005.

- [6] Belim S.V., Majorov-Zil'bernagel' A.O. Algoritm poiska povrezhdennykh pikselej i udalenija impul'snogo shuma na izobrazhenijah s ispol'zovaniem metoda asociativnykh pravil. (The algorithm of broken pixel detection and impulse noise removal based on associative rules.) *Nauka i obrazovanie: jelektronnoe nauchno-tehnicheskoe izdanie*. 12, 2014, URL: <http://technomag.bmstu.ru/doc/744983.html>
- [7] Belim S.V., Majorov-Zil'bernagel' A.O. Vosstanovlenie izobrazhenij so staticheskimi propuskami na osnove metoda asociativnykh pravil. (Recovering of images with static gaps based on the method of associative rules.) *Vestnik komp'yuternyh i informacionnyh tehnologij*. 12:18–23, 2014.
- [8] Belim S.V., Seliverstov S.A. Ispol'zovanie metoda analiza ierarxij dlya vyyavleniya impul'snogo shuma v graficheskix ob"ektax. (Using of hierarchy analysis method for impulse noise recognizing in graphical objects.) *Informacionnye tehnologii*. 4:251–258, 2015.
- [9] Belim S.V., Kutlunin P.E. Vyyavlenie povrezhdennykh pikselej na izobrazhenii s pomoshh'yu algoritma klasterizacii (Broken pixels detection in images based on the clusterization algorithm.) *Vestnik komp'yuternyx i informacionnyx tehnologij*. 3:3–10, 2016.
- [10] Belim S.V., Kutlunin P.E. Boundary extraction in images using a clustering algorithm. *Computer optics*. 39(1):119–124, 2015.
- [11] Pourian P., Karthikeyan S., Manjunath B.S. Weakly supervised graph based semantic segmentation by learning communities of image-parts. *The IEEE International Conference on Computer Vision (ICCV)*. 1359–1367, 2015.
- [12] Nair R.S., Vineetha K.V. Modularity Based Color Image Segmentation. *IJIREICE*. 3(1):109–113, 2016.
- [13] Mourchid Y., Hassouni M.E., Cherif H.A. New Image Segmentation Approach using Community Detection Algorithms. *International Journal of Computer Information Systems and Industrial Management Applications*. 8:195–204, 2016.
- [14] Browet A., Absil P.-A., Van Dooren P. Community Detection for Hierarchical Image Segmentation. *IWCIA*. LNCS 6636:358–371, 2011.
- [15] Newman M. E. Analysis of weighted networks. *Physical Review E*. 70(5):056131, 2004.
- [16] Clauset A., Newman M. E. J., Moore C. Finding community structure in very large networks. *Physical Review E*. 70(6):066111, 2004.

Algorithm of an Impulse Noise Detection on an Image

Sergey V. Belim, Stanislav B. Larionov

In article the algorithm of of an impulse noise detection is suggested. The image is presented as a graph. On the graph communities are looked for. Communities from one point are read by the defective pixels. The computer experiment is made. The algorithm shows high effectiveness.