

УДК 004.93'1; 004.932

Д.В. Прадун, Б.А. Залесский

ИСПОЛЬЗОВАНИЕ АЛГОРИТМА МАКСИМАЛЬНОГО ПОТОКА ГРАФА ДЛЯ ФИЛЬТРАЦИИ МУЛЬТИСПЕКТРАЛЬНЫХ ИЗОБРАЖЕНИЙ

Исследуется алгоритм максимального потока графа и возможность его использования для фильтрации мультиспектральных изображений. Приводятся основные понятия и определения, необходимые для описания максимального потока графа. Предлагаются способы сегментации бинарных изображений с помощью алгоритма максимального потока, исследуется возможность их применения для фильтрации полутоновых и мультиспектральных изображений. Дается сравнение результатов фильтрации космических изображений при различных значениях пропускных способностей дуг графа и числа бинарных слоев. Дается оценка результатов работы алгоритма максимального потока в сравнении с другими известными алгоритмами фильтрации.

Введение

Методы теории графов находят свое применение во многих областях: экономике, коммуникационных и транспортных системах, компьютерной химии и т. д. Особенно активно теория графов используется в программировании в силу удобного выражения задач обработки информации на теоретико-графовом языке. Чаще всего с помощью теории графов выполняется моделирование программ автоматизации. Для этого используются управляющие графы, синтаксические деревья, сети Петри и др. С появлением суперкомпьютеров и распределенных сетей возникла проблема эффективной организации параллельных вычислений над большими массивами данных. Оказалось, что графовые методы решают такую проблему довольно эффективно [1].

С развитием систем обработки мультиспектральных космических изображений потребность в использовании теории графов возросла. С помощью методов теории графов были разработаны глобальный подход к обнаружению и связыванию контуров изображений [2], алгоритмы фильтрации бинарных и полутоновых изображений и т. д. Разработанные методы и алгоритмы обработки изображений основываются на поиске различного рода потоков и путей на построенном графе. Например, для выполнения процесса фильтрации используется алгоритм максимального потока, с помощью которого можно получить сегментированное изображение. К сожалению, разработанные алгоритмы фильтрации с помощью определения максимального потока графа предназначены для обработки бинарных и полутоновых изображений и не используются для обработки космических снимков с несколькими каналами цветового спектра.

В статье предложен алгоритм фильтрации мультиспектральных изображений, который использует поиск максимального потока графа, построенного для искомого космического снимка. Данный алгоритм осуществляет фильтрацию всех цветовых каналов изображения по отдельности, что позволяет выполнять вычисления параллельно друг другу. Алгоритм был протестирован на 85 изображениях земной поверхности, и результаты его работы были сравнены с результатами выполнения фильтрации с помощью других известных алгоритмов.

1. Основные понятия и определения максимального потока

Пусть дан связный ориентированный граф $G = (V, E)$, который не имеет петель и удовлетворяет следующим условиям [3]:

1. Существует только одна вершина, которая не имеет входящих дуг. Эту вершину назовем источником и обозначим через s .
2. Существует только одна вершина, которая не имеет исходящих дуг. Эту вершину назовем стоком и обозначим через t .

3. Каждой дуге $e = (i, j)$ графа сопоставлено неотрицательное вещественное число, называемое пропускной способностью дуги из вершины i в вершину j . Обозначим пропускную способность через $c(e)$ или $c(i, j)$. Если не существует дуги e из i в j , то $c(e) = 0$.

Потоком f графа G является функция, сопоставляющая каждой дуге $e = (i, j)$ неотрицательное вещественное число $f(e) = f(i, j)$ так, что выполняются следующие условия [3–5]:

$$f(i, j) \leq c(i, j), \quad i, j \in V; \quad (1)$$

$$\sum_{i \in V} f(i, j) - \sum_{j \in V} f(j, i) = 0, \quad i \neq s, j \neq t; \quad (2)$$

$$\sum_{j \in V} f(i, j) = 0 \quad \text{для всех } i \text{ из } V - \{s, t\}. \quad (3)$$

Величина потока f определяется как сумма потоков по всем дугам, исходящим из источника [3–5]:

$$\text{val}(f) = \sum_{j \in V} f(s, j). \quad (4)$$

Из условия (3) легко увидеть, что общее количество материала, выходящего из источника, равно общему количеству материала, входящего в сток [3]:

$$\text{val}(f) = \sum_{j \in V} f(s, j) = \sum_{i \in V} f(i, t).$$

Для того чтобы определить, какой поток в графе является максимальным, вводится понятие минимального разреза графа. Будем говорить, что разрез $\langle S, \bar{S} \rangle$ в графе разделяет источник s и сток t , если $s \in S$, а $t \in \bar{S}$. Пропускная способность $c(K) = c(S, \bar{S})$ разреза $K = \langle S, \bar{S} \rangle$ определяется выражением

$$c(K) = \sum_{\substack{i \in S \\ j \in \bar{S}}} c(i, j). \quad (5)$$

Можно доказать [3], что для любого потока f и любого разреза $\langle S, \bar{S} \rangle$ в графе

$$\text{val}(f) = f(S, \bar{S}) - f(\bar{S}, S), \quad (6)$$

откуда следует, что

$$\text{val}(f) \leq c(S, \bar{S}).$$

Если для графа определен поток f и такой разрез $K = \langle S, \bar{S} \rangle$, что $\text{val}(f) = c(K)$, то поток f будет называться максимальным, а разрез – минимальным. Другими словами, величина максимального потока графа равна пропускной способности минимального разреза [3, 6].

2. Основные алгоритмы нахождения максимального потока графа

За 60 лет изучения задачи о максимальном потоке был создан ряд алгоритмов. Первым алгоритмом поиска максимального потока, на котором в дальнейшем базировались многие

другие алгоритмы, был алгоритм Форда – Фалкерсона. Данный алгоритм относился к так называемым помечивающим алгоритмам [3, 4]. Он основывался на построении увеличивающей цепи, которая представляет собой путь от истока к стоку. После этого Е.А. Диницом был разработан алгоритм, состоящий из фаз, на которых поток увеличивался сразу вдоль всех кратчайших цепей определенной длины. Для этого на каждой фазе строилась вспомогательная бесконечная сеть, которая содержала все увеличивающие цепи, длина которых не превышала длины кратчайшего пути из s в t [7]. Данный алгоритм позволил значительно увеличить быстродействие разрабатываемых алгоритмов. Модификация алгоритма Диница, предложенная А.В. Карзановым, позволила добиться еще большей оценки быстродействия.

В течение долгого времени алгоритмы Форда – Фалкерсона, Диница и Карзанова были основополагающими для достижения лучшего быстродействия в разрабатываемых алгоритмах поиска максимального потока. Однако появился новый алгоритм Голдберга и Тарьяна, который позволил значительно увеличить скорость работы алгоритмов при поиске максимального потока [7]. Данный алгоритм выполнял две основные операции: наращивание потока и изменение метки вершины графа, используя при этом так называемые предпотоки, понятие о которых было введено еще Карзановым. Существует также параллельная модификация алгоритма Голдберга – Тарьяна для вычислений на нескольких процессорах. Кроме того, Голдбергом и Рао была предложена модификация алгоритма, который по быстродействию превосходил все предыдущие алгоритмы в несколько раз [7].

Различные модификации алгоритма Голдберга широко используются на практике до сих пор. Однако, как показали многочисленные исследования, поиск максимального потока с помощью алгоритма Голдберга при обработке изображений является неэффективным. Поэтому наиболее подходящим для решения данной задачи оказался алгоритм, предложенный Ю. Бойковым [8]. Данный алгоритм основывается на алгоритме Диница и выполняет следующие основные шаги:

- построение поисковых деревьев из источника в сток, формирующих путь из s в t ;
- насыщение полученного пути;
- восстановление поисковых деревьев.

3. Алгоритм Пикарда – Рэтлиффа сегментации бинарных изображений

Алгоритм Пикарда – Рэтлиффа позволяет выполнить сглаживание бинарных изображений [9], основываясь на следующих утверждениях. Любое бинарное изображение можно представить как граф, в котором, например, белые пиксели соединены с источником, т. е. создают определенное множество W , а черные пиксели соединены со стоком, образуя множество B . Таким образом, есть два множества $W, B \subset S$, таких, что $W \cap B = \emptyset$ и $W \cup B = S$, где S – множество всех вершин графа, кроме s и t . Другими словами, множества W и B образуют разрез, отделяющий вершины графа, соединенные с источником, от вершин, соединенных со стоком. Тогда согласно формуле (5) пропускная способность такого разреза будет следующей:

$$C(x) = \sum_{\substack{i \in W \cup \{s\} \\ j \in B \cup \{t\}}} c(i, j), \quad (7)$$

где x – вектор с координатами $x_i = 1$, если $i \in W$, и $x_i = 0$, если $i \in B$. Иначе говоря, множество векторов $x \in X$ представляет собой пиксели бинарного изображения [10].

Допустим, имеется другое бинарное изображение, построенное на исходных пикселях с теми же дугами. Пусть дан вектор y с координатами $y_i = 1$, если дуга (s, i) принадлежит множеству E всех дуг графа, или $y_i = 0$, если $(i, t) \in E$. Тогда пропускную способность разреза $C(x)$ можно представить как

$$U(x) = \sum_{i \in S} \lambda_i \cdot (1 - 2y_i) \cdot x_i + \sum_{\substack{i \in S, \\ j = S}} \beta_{i,j} (x_i - x_j) \cdot x_i. \quad (9)$$

Можно легко заметить, что $C(x) = U(x) + \sum_{i \in S} \lambda_i y_i$. Тогда минимум функции $U(x)$ будет находиться в той же точке, что и минимум функции $C(x)$. Поэтому согласно Пикарду и Рэтлиффу множество векторов, удовлетворяющих равенству $x^* = \arg \min_{x \in X} U(x)$, будет являться минимальным разрезом или максимальным потоком графа, построенного для искомого бинарного изображения.

4. Использование алгоритма максимального потока для полутонновых и мультиспектральных изображений

Любое полутонновое изображение x можно представить в виде суммы $x = \sum_{l=1}^{L-1} x(l)$ бинарных изображений-слоев $x(l)$, где $0 < l \leq L-1$, $2 < L \leq 256$, а координаты каждого вектора $x(l)$ равны 1, если $x_i(l) \geq l$, и равны 0 в противном случае. Полученные бинарные изображения задают монотонно убывающую последовательность $x(1) \geq x(2) \geq \dots \geq x(L-1)$ [10]. Следовательно, для любого полутоннового изображения функцию $U(x)$ можно представить как

$$U(x) = \sum_{l=1}^{L-1} u_l(x(l)), \quad (10)$$

где

$$u_l(x(l)) = \lambda \sum_{i \in S} |y_i(l) - x_i(l)| + \sum_{(i,j) \in S} \beta_{i,j} |x_i(l) - x_j(l)|. \quad (11)$$

При этом $y_i(l)$ – координаты вектора исходного полутоннового изображения, полученные на слое l . Используя формулы (10), (11), можно определить функцию $U(x)$ как

$$U(x) = \sum_{i \in S} \lambda_i \left| y_i - \sum_{l=1}^{L-1} x_i(l) \right| + \sum_{(i,j) \in S} \beta_{i,j} \sum_{l=1}^{L-1} |x_i(l) - x_j(l)|. \quad (12)$$

Минимизировав функцию $U(x)$, т. е. найдя невозрастающую последовательность таких бинарных изображений-слоев $\tilde{x}(l)$, каждое из которых будет выполнять условие

$$\tilde{x}(l) = \arg \min_{x_{bin}} u_l(x_{bin}), \quad l \in \{1, \dots, L-1\}, \quad (13)$$

получим сегментированное изображение [8], т. е. выполним фильтрацию исходного изображения. При этом коэффициенты $\beta_{i,j}$ из формулы (12) определяют степень сглаживания изображения, а коэффициенты λ_i определяют пропускную способность дуг, выходящих из источника и входящих в сток. Для мультиспектральных изображений каждый цветовой канал спектра представляет собой полутонновое изображение, которое может быть сегментировано согласно представленному выше алгоритму.

5. Результаты работы алгоритма максимального потока

Алгоритм фильтрации изображений с помощью определения максимального потока был протестирован на 85 изображениях земной поверхности и городских построек. Реализованный

метод использует алгоритм поиска максимального потока графа, разработанный Ю. Бойковым [8]. Алгоритм Бойкова обеспечивает построение графа для изображения с заданными показателями коэффициентов λ_i и $\beta_{i,j}$ из формулы (12), а также поиск на построенном графе максимального потока для каждого из бинарных слоев, с помощью которых получается отсегментированное изображение.

В результате проведенных тестов было получено, что наиболее приемлемые значения коэффициентов $\beta_{i,j}$ лежат в диапазоне $[0,25, \dots, 3]$. При $\beta_{i,j} < 0,25$ сегментации изображения не происходит. При $\beta_{i,j} > 3$ происходит чрезмерное сглаживание изображения и результат фильтрации не может быть использован при дальнейшей классификации графических объектов. Поэтому для проверки качества работы алгоритма фильтрации были выбраны следующие значения $\beta_{i,j}$: 0,25; 0,5; 1; 2. При этом значения λ_i для источника и стока задавались одинаковыми и были равны 1. Кроме коэффициентов λ_i и $\beta_{i,j}$, для алгоритма фильтрации мульти-спектральных изображений необходимо задать количество бинарных слоев L , на которые будет разбиваться исходное изображение. Для тестирования были выбраны следующие значения L : 10, 50, 100, 200, 256. При этом значения $\tilde{x}(l)$ выбирались на основе гистограммы изображения. Примеры исходных космических снимков представлены на рис. 1.



Рис. 1. Исходные тестовые изображения спутниковой и аэросъемки:
а) городские постройки; б) сельскохозяйственные угодья

На рис. 2 показаны результаты фильтрации исходных изображений при $L = 10$. Следует отметить, что яркость результирующих изображений уменьшилась по сравнению с исходным снимком, что можно объяснить слишком малым значением переменной L . Это может привести к появлению на изображении областей с менее ярким цветовым содержанием. Кроме того, с увеличением значений $\beta_{i,j}$ степень сглаженности изображений также увеличивается, более мелкие детали сливаются с более крупными однородными областями и все сложнее становится даже визуально определить отдельные объекты на изображении. Поэтому для сохранения относительной четкости сегментированного изображения целесообразно использовать малые значения коэффициентов $\beta_{i,j}$.

При увеличении значения L результат фильтрации по своей яркости и содержанию приближается к исходному изображению (рис. 3). Можно сделать вывод, что для получения более точного результата фильтрации необходимо использовать значения L , близкие к 256 или равные ему. Однако это потребует большего времени выполнения процесса фильтрации.

Если сравнить результаты фильтрации с помощью реализованного алгоритма с результатами, получаемыми с помощью других известных методов, можно заметить, что алгоритму максимального потока свойственна потеря незначительных мелких объектов на изображении, хотя по степени сглаженности однородных областей он может конкурировать с адаптивными

алгоритмами фильтрации (рис. 4), например фильтром Smart Blur пакета Adobe Photoshop или алгоритмом адаптивной фильтрации на основе автоматической бинаризации градиента [11].

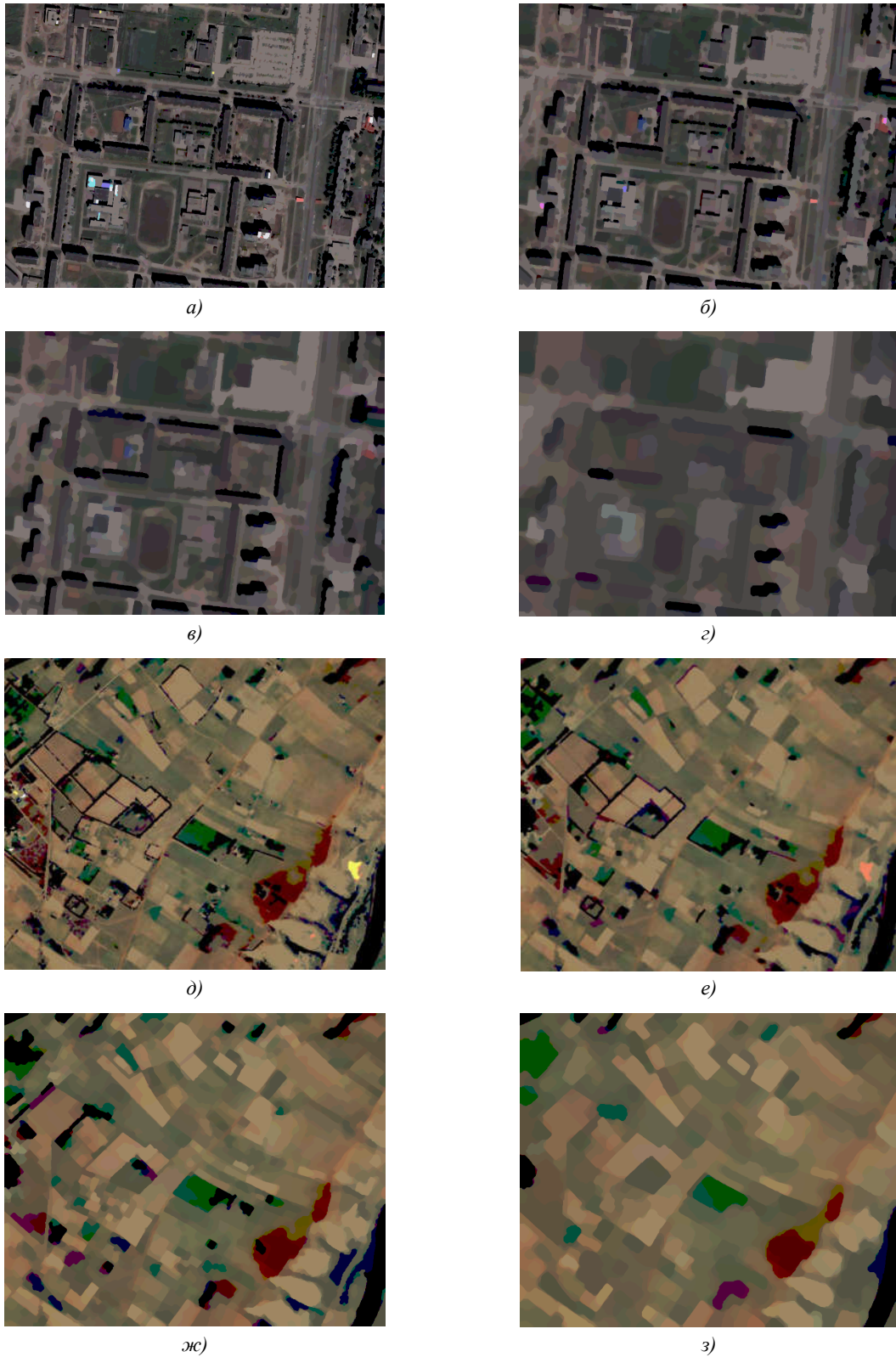


Рис. 2. Результаты фильтрации снимков на рис. 1: а), д) $\beta_{i,j} = 0,25$;
б), е) $\beta_{i,j} = 0,5$; в), ж) $\beta_{i,j} = 1$; г), з) $\beta_{i,j} = 2$

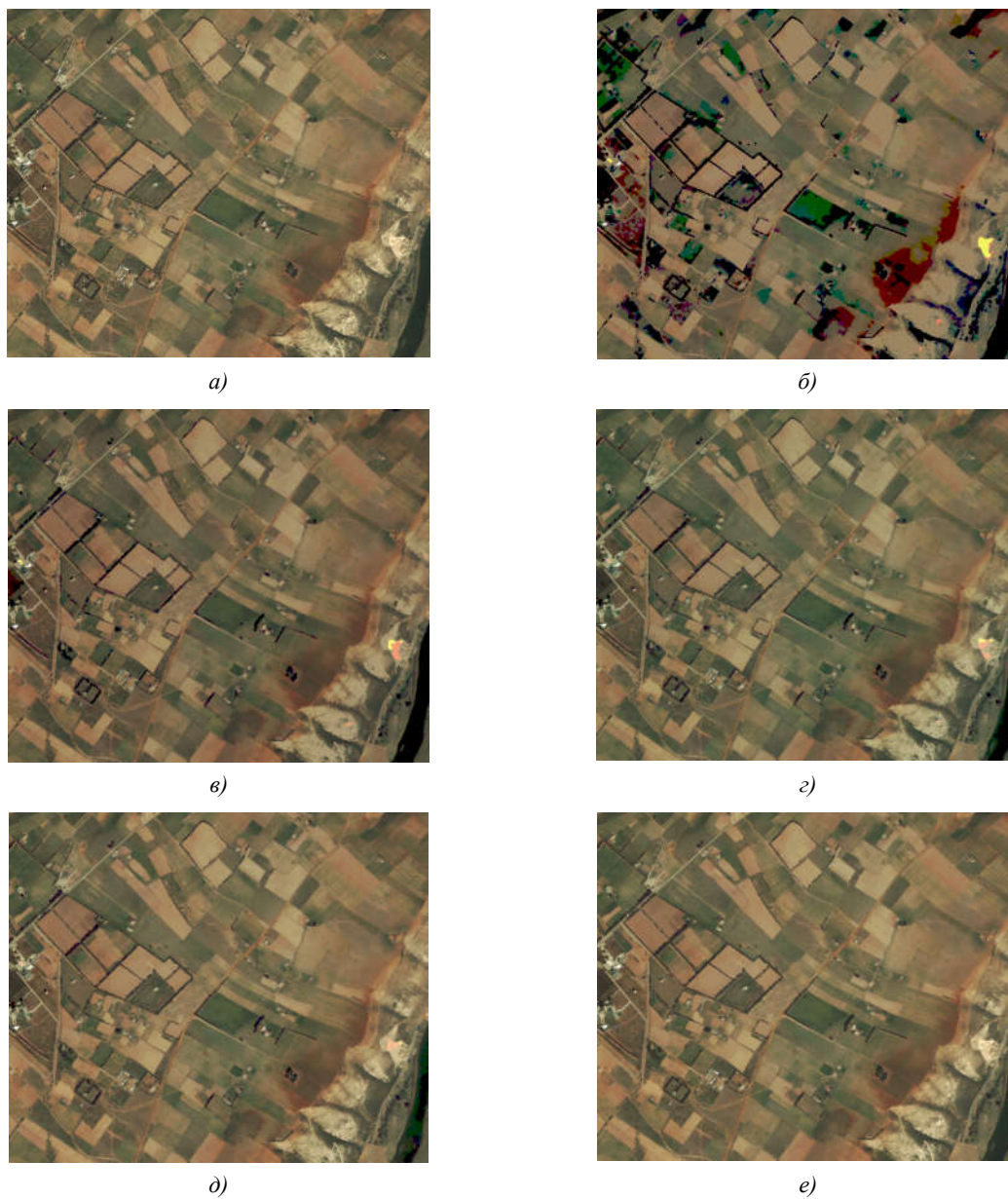


Рис. 3. Результаты фильтрации снимка на рис. 1, б при $\beta_{i,j} = 0,25$:

a) исходное изображение; б) сегментированное изображение при $L = 10$;
в) сегментированное изображение при $L = 50$; г) сегментированное изображение при $L = 100$;
д) сегментированное изображение при $L = 200$; е) сегментированное изображение при $L = 256$

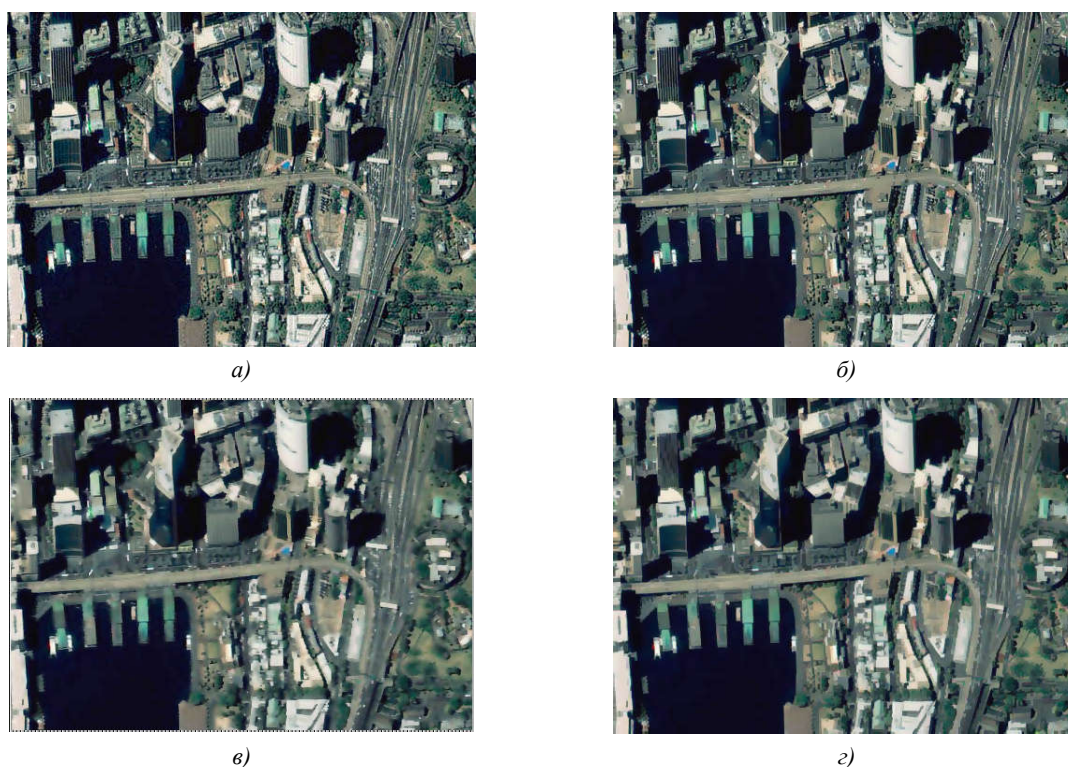


Рис. 4. Результаты фильтрации снимка разными алгоритмами:
а) исходное изображение; б) фильтр Smart Blur; в) алгоритм адаптивной фильтрации;
г) алгоритм максимального потока, $\beta_{i,j} = 0,25$, $L = 256$

Заключение

Тестирование алгоритма максимального потока показало, что его использование для фильтрации изображений является вполне эффективным и перспективным. Несмотря на чрезмерное сглаживание небольших графических объектов на изображениях, а также довольно длительную ручную настройку алгоритма, качество результатов фильтрации сравнимо с результатами фильтрации с помощью других известных алгоритмов. Для улучшения качества сегментации с помощью алгоритма максимального потока необходимо учитывать принадлежность пикселей к границам изображения. Этого можно добиться с помощью динамического определения величины пропускных способностей дуг, объединяющих соседние вершины графа. Одним из решений данной проблемы может стать использование некоторой функции, зависящей от значения градиента в каждом конкретном пикселе. Кроме того, сохраняется проблема уменьшения временных затрат при обработке изображения с помощью алгоритма максимального потока, а также способность алгоритма обрабатывать большие и сверхбольшие мультиспектральные изображения.

Список литературы

1. Касьянов, В.Н. Применение графов в программировании / В.Н. Касьянов // Институт систем информатики им. А.П. Ершова СО РАН : сб. статей [Электронный ресурс]. – 2007. – Режим доступа : http://www.iis.nsk.su/preprints/articles/sbor_kas_07_ru.shtml. – Дата доступа : 12.09.2009.
2. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
3. Свами, М. Графы, сети и алгоритмы / М. Свами, К. Тхуласираман. – М. : Мир, 1984. – 454 с.
4. Майника, Э. Алгоритмы оптимизации на сетях и графах / Э. Майника. – М. : Мир, 1981. – 324 с.

5. Кормен, Т. Алгоритмы. Построение и анализ / Т. Кормен, Ч. Лейзер, Р. Ривест. – М. : МНЦО, 2000. – 893 с.
6. Кристофидес, Н. Теория графов: алгоритмический подход / Н. Кристофидес. – М. : Мир, 1978. – 432 с.
7. Труханов, А. Алгоритмы нахождения максимального потока / А. Труханов // Algolist – алгоритмы, методы, исходники [Электронный ресурс]. – 2000. – Режим доступа : <http://algolist.manual.ru/maths/graphs/maxflows/>. – Дата доступа : 8.10.2009.
8. Boykov, Y. An experimental comparison of min-cut/max-flow algorithms of energy minimization in vision / Y. Boykov, V. Kolmogorov // IEEE Transactions on PAMI. – 2004. – Vol. 26, № 9. – P. 1124–1137.
9. Picard, J.C. Minimum cuts and related problems / J.C. Picard, H.D. Ratliff // Networks. – 1975. – Vol. 5, № 4. – P. 357–370.
10. Zalesky, B.A. Network flow optimization for restoration of images / B.A. Zalesky // Journal of Applied Mathematics. – 2002. – Vol. 2, № 4. – P. 199–218.
11. Залеский, Б.А. Алгоритм адаптивной фильтрации мультиспектральных изображений / Б.А. Залеский, Д.В. Прадун // Информатика. – 2009. – № 2 (22). – С. 31–38.

Поступила 11.06.09

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: pradundv@gmail.com*

D.V. Pradun, B.A. Zalesky

APPLICATION OF MAXIMAL FLOW ALGORITHM FOR MULTISPECTRAL IMAGE FILTRATION

The network max flow algorithm and possibilities of its use for filtration of multispectral images is investigated. The basic concepts and definitions of the max flow are given. The ways of segmentation of binary images by applying max flow algorithm are suggested and possibility of their application for grayscale and multispectral image filtration are investigated. A comparison of the results of filtration of space images with various values of throughputs of the network arches and amounts of binary layers is provided. A comparison of flow algorithm with some known filtration algorithms is given.