

КЛАСТЕРНАЯ ТЕХНОЛОГИЯ ОПРЕДЕЛЕНИЯ ВОССТАНАВЛИВАЮЩИХ ФИЛЬТРОВ И ОБРАБОТКИ БОЛЬШИХ ИЗОБРАЖЕНИЙ

М.А. Дроздов, Д.И. Зимин, С.Б. Попов, С.А. Скуратов, В.А. Фурсов

Институт систем обработки изображений РАН,
Самарский государственный аэрокосмический университет

Введение

Одним из важных этапов формирования изображений подстилающей поверхности Земли по данным аэрокосмического мониторинга является их обработка с целью улучшения качества. При этом возникают, по крайней мере, две проблемы: синтез структуры и параметров восстанавливающего фильтра и организация обработки больших изображений.

Проблема синтеза фильтров хорошо известна [1]. Для построения оптимального в некотором смысле восстанавливающего фильтра, строго говоря, требуется знание структуры и параметров модели искажающей системы. Для их определения необходимо проведение испытаний регистрирующей системы с использованием тестовых изображений. Это требует значительных затрат, связанных с созданием и поддержанием на поверхности Земли тестового изображения (миры). Вторая проблема связана с организацией обработки больших изображений на многопроцессорных вычислительных системах. Известно, что эффективность и достигаемое ускорение при распараллеливании задач обработки в значительной степени зависят от согласования архитектуры и числа используемых процессоров с размерами обрабатываемого изображения.

В настоящей работе указанные две проблемы решаются в рамках разработанной авторами сквозной автоматизированной технологии. Идея, положенная в основу разработки, была опубликована в работе [2]. Затем она была применена к решению задачи оценки разрешающей способности видеотракта по фрагментам регистрируемых изображений [3]. В настоящей работе эта схема применена к решению задачи улучшения качества больших изображений, обрабатываемых на вычислительном кластере.

1. Описание технологии построения фильтров и обработки изображений

Кратко перечислим основные этапы технологии определения параметров фильтров и обработки изображений, описанной в работе [2]:

1. Выбор на искаженном изображении фрагментов, наиболее подходящих для формирования тестовых образцов, на которых должны обучаться алгоритмы обработки.
2. Формирование из отобранных фрагментов тестовых образцов (компьютерное ретуширование).
3. Обучение (определение параметров) алгоритмов обработки на тестовых изображениях.
4. Обработка исходного искаженного (большого) изображения.

Обсудим методы и алгоритмы указанных этапов технологии, следуя работе [3], при этом уделив особое внимание особенностям и отличиям их реализации в контексте настоящей работы.

На первом этапе на исходном изображении отбираются фрагменты, у которых изменение функции яркости $f(x,y)$ имеет вид ступенчатой функции (рис. 1) с размытием вдоль средней линии фрагмента (линия А-В). Подходящими для этого считаются участки изображения, с резким перепадом яркости вдоль некоторой линии (например, граница между освещенной крышей здания и отбрасываемой зданием тенью).

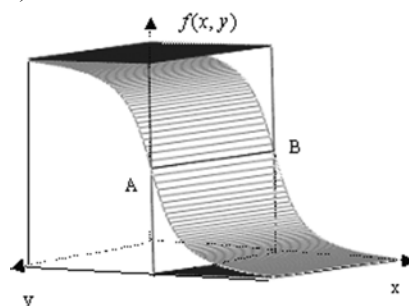


Рис. 1. Вид функции яркости на фрагменте

Отбор информативных фрагментов осуществляется так, как описано в работе [3]. Вначале, для каждого фрагмента изображения определяется перепад яркости. Из полученного множества фрагментов выбираются те, у которых перепады яркости выше заданного. Для каждого отобранного таким способом фрагмента вычисляется СКО функции яркости в «темной» и «светлой» областях и затем отбираются один или несколько фрагментов, для которых полусумма СКО минимальна на множестве допустимых значений.

Подчеркнем, что качество последующего оценивания параметров фильтра по отобранным тестовым фрагментам обычно оказывается выше, если средняя линия (А-В) на фрагменте точно совпадает с направлением строк или столбцов ПЗС-матрицы регистрирующего устройства.

Цель компьютерного ретуширования (этап 2) – формирование на фрагменте изображения ступенчатой функции яркости (рис. 1). Задача заключается в определении расположения линии перепада яркости. Для этого используется априорная информация о геометрической форме объектов, фрагменты изображений которых отобраны в качестве тестовых.

Для определения расположения линии перепада яркости на фрагменте строится гистограмма отсчетов функции яркости. Обычно она имеет вид, представленный на рис. 2. По гистограмме вычисляется среднее значение яркости фрагмента:

$$\bar{c} = \sum_i f_i \cdot i, \quad i = 0,255, \quad (1)$$

где f_i – отношение количества отсчетов с i -м уровнем яркости к общему числу отсчетов изображения.

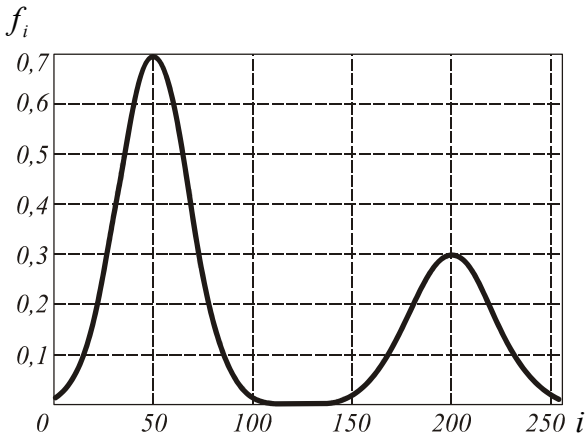


Рис. 2. Пример гистограммы отсчетов функции яркости на фрагменте

Затем подсчитывается количество отсчетов, значения яркости которых меньше среднего:

$$n_1 = N \cdot \sum_i f_i \cdot i, \quad i = 0, \text{int } \bar{c}, \quad (2)$$

где N – общее число отсчетов на фрагменте. По значениям n_1, N вычисляется координата границы функции яркости (расстояние от средней линии фрагмента):

$$x_0 = x \cdot \left(\frac{N+1}{2} - n_1 \right), \quad (3)$$

где x размер стороны фрагмента, перпендикулярной границе.

Подготовленные описанным способом пары тестовых фрагментов используются для определения параметров восстанавливающего фильтра. В работе [4] показано, что эту задачу целесообразно решать в классе линейных фильтров с бесконечной импульсной характеристикой (БИХ-фильтров), для которых связь отсчетов выходного $g(n_1, n_2)$ и входного $f(n_1, n_2)$ изображений описывается соотношением

$$g(n_1, n_2) = - \sum_{(m_1, m_2) \in Q_g} a_{m_1, m_2} g(n_1 - m_1, n_2 - m_2) + \sum_{(m_1, m_2) \in Q_f} b_{m_1, m_2} f(n_1 - m_1, n_2 - m_2), \quad (4)$$

где $a_{m_1, m_2}, b_{m_1, m_2}$ – коэффициенты фильтра.

Важнейшим доводом в пользу БИХ фильтра является то, что даже при значительных динамических искажениях размеры опорной области могут быть небольшими по сравнению со случаем использования фильтра с конечной импульсной характеристикой (КИХ-фильтра). В данном случае этот факт имеет принципиальное значение, т.к. размеры тестовых фрагментов невелики.

Проведя измерения значений отсчетов яркости для N различных положений опорной области на тестовых фрагментах с использованием модели (4) можно записать матричное уравнение:

$$\mathbf{y} = \mathbf{X} \cdot \mathbf{h} + \mathbf{o}, \quad (5)$$

где $N \times 1$ – векторы \mathbf{y} и \mathbf{X} составлены из отсчетов исходного фрагмента и ошибок измерений соответственно, $N \times M$ – матрица $\mathbf{X} = [x_1, x_2, \dots, x_N]^T$ составлена из отсчетов обоих тестовых фрагментов, а $\mathbf{h} = [\dots a_{ij} \dots, \dots b_{kl} \dots]^T$ – $M \times 1$ – вектор искоемых параметров фильтра ($M < N$).

Оценка \mathbf{h} вектора параметров искажающей системы \mathbf{h} , фигурирующего в (5), может быть получена по методу наименьших квадратов или с использованием методов, изложенных в работе [2]. В работе [2] показано, что поменяв входное и выходное изображения местами, по этой же схеме можно получить оценки параметров $\mathbf{C} = [\dots \epsilon_{ij} \dots, \dots \delta_{kl} \dots]^T$ восстанавливающего (инверсного) фильтра:

$$\mathcal{F}(n_1, n_2) = - \sum_{(m_1, m_2) \in Q_f} \epsilon_{m_1, m_2} \mathcal{F}(n_1 - m_1, n_2 - m_2) + \sum_{(m_1, m_2) \in Q_g} \delta_{m_1, m_2} g(n_1 - m_1, n_2 - m_2). \quad (6)$$

Поскольку в соответствии с (6) для вычисления отсчетов выходного изображения используются отсчеты как входного, так и выходного изображений, в рассматриваемой технологии применяется итерационная процедура, которая может оказаться неустойчивой, даже если сам БИХ-фильтр устойчив. Для обеспечения вычислительной устойчивости алгоритма в вычислительную схему вводится дополнительный параметр λ . При этом итерационная схема реализации БИХ-фильтра (6) принимает вид:

$$\mathcal{F}_k(n_1, n_2) = \mathcal{F}_{k-1}(n_1, n_2) - \lambda \sum_{(m_1, m_2) \in Q_f} \epsilon_{m_1, m_2} \mathcal{F}_{k-1}(n_1 - m_1, n_2 - m_2) + \lambda \sum_{(m_1, m_2) \in Q_g} \delta_{m_1, m_2} g(n_1 - m_1, n_2 - m_2), \quad (7)$$

где $g(m, n)$ – функция яркости исходного искаженного изображения, $\mathcal{F}_k(m, n), \mathcal{F}_{k-1}(m, n)$ – функции яркости восстановленного изображения на k -м и $(k-1)$ -м шагах итерационной процедуры.

При увеличении параметра λ восстановление достигается на меньшем числе итераций, но при этом могут появляться артефакты, которые усиливаются при увеличении числа итераций. Уменьшение параметра λ может привести к недостаточному восстановлению изображения.

Для решения задачи выбора параметра λ в рассматриваемой технологии предложено использовать имеющиеся исходный и подвергшийся компьютерному ретушированию фрагменты. В частности, параметр λ выбирается таким, чтобы в результате обработки искаженного фрагмента получалось изображение, максимально «близкое» к желаемому отретушированному фрагменту.

Для оценки степени близости в рассмотрение вводится функция вида

$$F(\lambda) = \sum_i \sum_j \left| \mathcal{F}_{i,j}(\lambda) - f_{i,j} \right|, \quad (8)$$

где $f_{i,j}(\lambda)$ – отсчеты изображения, восстановленного с использованием алгоритма (7), а $f_{i,j}$ – отсчеты неискаженного изображения. Суммирование ведется по всем соответствующим отсчетам фрагментов изображений. Ищется

$$\lambda^* : F(\lambda^*) = \min_{\lambda \in \Lambda} F(\lambda), \quad (9)$$

где Λ – область допустимых значений параметра λ [1].

В данном случае, учитывая тот факт, что размеры фрагментов изображений и диапазон значений параметра λ невелики, задача (9) может быть решена путем непосредственного сравнения значений функции (8).

2. Общая схема распараллеливания

В рассматриваемом случае при построчной организации хранения отсчетов изображения очень естественной представляется одномерная построчная декомпозиция [5], при которой каждая задача обрабатывает фрагмент изображения, содержащий смежные строки. Фрагменты перекрываются между собой на n строк так, чтобы каждая задача на текущей итерации имела все данные для обработки граничных точек. С этой же целью после каждой итерации задачи обмениваются перекрывающимися строками. Для определенности будем полагать, что изображение имеет размеры $N \times M$ ($N > M$), размер фрагментов перекрывающихся полос – $(N/k + 2n) \times M$, где k – число параллельных процессов, а n – число перекрывающихся строк соседних полос (см. рис. 3), которыми узлы должны обмениваться.

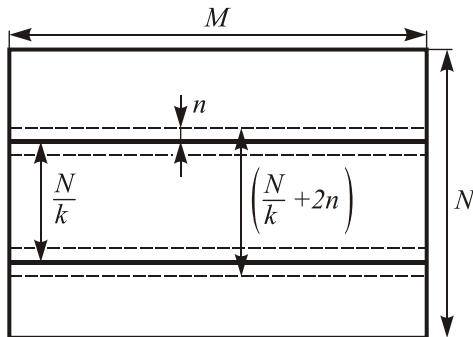


Рис. 3. Разбиение на фрагменты (одномерная декомпозиция)

Будем также полагать, что известны константы: τ_p – время расчета при обработке одного отсчета изображения и τ_n – среднее время, затрачиваемое на передачу информации, связанной с одним отсчетом изображения.

Время выполнения полной обработки одного изображения складывается из времени рассылки фрагментов по параллельным задачам, времени выполнения I шагов обработки, времени обмена перекрывающимися строками и времени формирования результирующего изображения из обработанных фрагментов.

Время рассылки фрагментов изображения определяется следующим образом:

$$T_0 = (k-2) \left(\frac{N}{k} + 2n \right) M \tau_p + \left(\frac{N}{k} + n \right) M \tau_n.$$

Время выполнения одного шага обработки составляет

$$T_{pi} = \frac{NM}{k} \tau_p.$$

Время обмена перекрывающимися строками после выполнения одного шага обработки –

$$T_{ni} = 2nM \tau_n.$$

Время формирования результирующего изображения из обработанных фрагментов –

$$T_{I+1} = (k-1) \frac{NM}{k} \tau_n.$$

Полное время обработки по данной технологии составляет

$$T_{1D} = T_0 + T_{I+1} + \sum_{i=1}^I T_{pi} + \sum_{i=1}^{I-1} T_{ni},$$

т.к. обмен строками после последнего шага не производится, или окончательно

$$T_{1D} = 2 \frac{k-1}{k} NM \tau_n + (2k-3)nM \tau_n + \frac{INM}{k} \tau_p + 2n(I-1)M \tau_n.$$

Относительная эффективность данного варианта параллельной обработки определяется по формуле

$$E_{1D} = \frac{INM \tau_p}{k T_{1D}} = \frac{1}{1 + 2\delta \left(\frac{k-1}{I} + \frac{kn}{IM} (k+I-2,5) \right)},$$

где $\delta = \tau_n / \tau_p$ – отношение отрезков времени, необходимых для пересылки данных, к времени обработки в расчете на один отсчет изображения. Это отношение тесно связано с вычислительной сложностью алгоритма обработки, чем сложнее алгоритм, тем меньше это отношение.

Локальность обработки на каждом шаге обработки изображения позволяет применить альтернативную стратегию декомпозиции, основанную на функциональном подходе. Тот факт, что определенная часть информации готова значительно ранее полного завершения текущего шага обработки, позволяет использовать схему с параллельным выполнением всех шагов алгоритма и непосредственной передачей по мере готовности строк изображения от одной операции к другой.

В данном случае единицей разбиения служит операция, выполняющая один шаг обработки. Все операции выполняются параллельно. Данные передаются от одной операции к другой по мере готовности, т.е. после завершения обработки строки изображения на i -ом шаге, она передается другой задаче и является исходной для выполнения $(i+1)$ -ого

шага. Фактически формируется поток данных изображения, элементы которого на i -ом процессоре подвергаются преобразованию в соответствии с алгоритмом i -ого шага и передаются последующему процессору для дальнейшей обработки. При этом результаты всего процесса обработки формируются с той же скоростью, с которой производится и ввод, но, после задержки, пропорциональной числу выполняемых шагов обработки.

Время работы одной задачи складывается из времени обработки N строк изображения и пересылки их другой задаче:

$$T_p = NM\tau_p + NM\tau_n.$$

Начало обработки i -ой задачи задерживается на время

$$T_{idle i} = (i-1)(n+1)M(\tau_p + \tau_n)$$

в случае $I \leq k$, т.е. число шагов обработки не превышает число процессоров. Если число шагов обработки превышает число процессоров, то задержка i -ой задачи составит

$$T_{idle i} = \left(i - \left\lfloor \frac{i-1}{k} \right\rfloor k - 1 \right) (n+1)M(\tau_p + \tau_n) + \left\lfloor \frac{i-1}{k} \right\rfloor NM(\tau_p + \tau_n),$$

где $\lfloor x \rfloor$ – обозначает целую часть числа x .

Общее время обработки по данному варианту параллельной обработки определяется временем завершения обработки на последнем шаге

$$\begin{aligned} T_{pipe} &= T_{idle I} + T_p = \\ &= (I - \psi k - 1)(n+1)M(\tau_p + \tau_n) + \\ &+ (\psi + 1)NM(\tau_p + \tau_n), \end{aligned}$$

где $\psi = \lfloor I - 1/k \rfloor$.

Относительная эффективность потокового (конвейерного) варианта параллельной обработки определяется по формуле

$$\begin{aligned} E_{pipe} &= \frac{INM\tau_p}{kT_{pipe}} = \\ &= \frac{1}{(1 + M\delta) \left(\frac{k(n+1)}{IN} (I - \psi k - 1) + \frac{k}{I} (\psi + 1) \right)}. \end{aligned}$$

Сравнение относительной эффективности параллельных реализаций рассматриваемой технологии обработки изображения показывает [6], что потоковый вариант предпочтительнее распараллеливания с использованием одномерной декомпозиции по данным только в случае, когда число шагов обработки близко, но не превышает число процессоров параллельной вычислительной системы.

Дополнительные резервы повышения эффективности распараллеливания по данным связаны с применением двумерной декомпозиции, которая более эффективна, по сравнению с одномерной,

поскольку разбиение большого изображения на полосы не всегда экономично с точки зрения объема пересылаемых данных. Представляет интерес установить размеры фрагментов, при которых вычислительные ресурсы кластера используются наиболее эффективно.

Для простоты рассмотрим случай, когда исходное изображение квадратное $X \times X$, где X – число отсчетов одной стороны изображения. При фиксированных значениях величин τ_p и τ_n наименьшая доля пересылок имеет место при форме фрагмента в виде квадрата.

Разобьем изображение на квадраты одинаковых размеров и обозначим x – сторону квадрата. Потребуем, чтобы величина x удовлетворяла неравенству

$$(4 \cdot \delta)/x \leq \Delta_{don}, \quad (10)$$

где Δ_{don} – допустимая величина отношения времени пересылок к времени обработки. Эта величина задана для процессоров, обрабатывающих квадратные (не примыкающие к границам) фрагменты, и определяется желаемым уровнем эффективной загрузки процессоров. Для граничных – $x_0 \times x$ ($x \times x_0$) и угловых – $x_0 \times x_0$ фрагментов неравенство только усилится. Поэтому есть основания ожидать, что эффективность использования процессоров повысится, если размеры угловых и граничных фрагментов увеличить, не уменьшая размеров внутренних так, как показано на рис. 4.

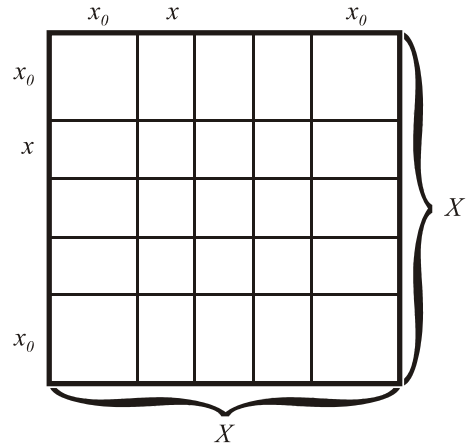


Рис. 4. Разбиение на фрагменты (двумерная декомпозиция)

При малых размерах опорных областей БИХ-фильтра объем передаваемых данных можно считать пропорциональным длине границ (сторон). Тогда суммарное время обработки с учетом пересылок:

а) для внутренней области:

$$T_{вн} = x^2 \cdot \tau_p + 4 \cdot x \cdot \tau_n, \quad (11)$$

б) для граничной:

$$T_{гп} = x \cdot x_0 \cdot \tau_p + 2 \cdot x_0 \cdot \tau_n + x \cdot \tau_n, \quad (12)$$

в) для угловой:

$$T_{угл} = x_0^2 \cdot \tau_p + 2 \cdot x_0 \cdot \tau_n. \quad (13)$$

Положим

$$x_0 = k \cdot x, \quad (14)$$

где

$$1 < k < 1,5 \quad (15)$$

– коэффициент увеличения угловой (и соответственно граничной) области, который необходимо выбрать из условия балансировки процессоров. Легко проверить, что неравенства (15) устанавливают диапазон, за пределами которого балансировка невозможна.

Точную балансировку, при одновременном сокращении времени обработки, можно осуществить лишь для части процессоров. Если сбалансировать работу процессоров, обрабатывающих внутренние и граничные области, вычислительные затраты на обработку угловых областей возрастут, что приведет к простоям остальных процессоров. Поэтому потребуем, чтобы выполнялось равенство времени обработки внутренних и угловых областей:

$$k^2 \cdot x + 2 \cdot k \cdot \delta = x + 4 \cdot \delta. \quad (16)$$

Для выбора k , удовлетворяющего равенству (16), необходимо исключить величину x . Сделаем это исходя из того, что при увеличении сторон угловых квадратных фрагментов в k раз для предельного значения $k = 1,5$, число полос n , на которые может быть разбита сторона исходного (квадратного) изображения со стороной X , определяется равенством

$$n = (X/x) - 1. \quad (17)$$

Соотношение для выбора k , полученное из (16) с учетом (17), принимает вид

$$X(k^2 - 1) - (4 - 2k)(n + 1)\delta = 0. \quad (18)$$

Заметим, что при k , удовлетворяющем равенству (18), неравенство (10) не нарушается.

Для заданного в соответствии с (17) n величина \mathcal{E} стороны квадратного внутреннего фрагмента, уточненная (увеличенная по сравнению с x) с учетом найденного из равенства (18) масштабного коэффициента k , определяется как

$$\mathcal{E} = X / (n + 2 \cdot (k - 1)). \quad (19)$$

Можно показать, что для найденного \mathcal{E} время обработки граничного фрагмента с учетом пересылок не будет превышать время обработки внутренних и угловых областей, если выполняется дополнительное условие

$$k > 0,5 + \sqrt{0,25 + \delta/\mathcal{E}}. \quad (20)$$

Для большинства известных итерационных алгоритмов обработки изображений величина δ/\mathcal{E} мала, при этом граница (20) почти совпадает с левой границей в (15). Следовательно, практически для всех допустимых k время обработки граничных областей не превышает время обработки внутренних и угловых областей, что обеспечивает достаточно полную загрузку соответствующих процессоров.

Соотношения (17), (18), (19) могут использоваться для выбора начального разбиения обрабатываемого изображения, поскольку для их получения

использовался ряд упрощающих предположений. Для более полного учета влияния всех факторов, которые не принимались во внимание в указанной упрощенной постановке, может использоваться технология итерационного планирования распределения ресурсов, описанная в работе [7]. В данном случае ее применение не вызовет дополнительных усложнений по сравнению с вариантом, описанным в указанной работе, поскольку задача выбора k – однопараметрическая.

3. Пример реализации

Описанная технология апробировалась на космических снимках города. В частности, решалась задача восстановления изображения, которое было получено моделированием размытия значений отсчетов гауссианом с радиусом 7 с помощью графического пакета “Gimp”. Сильно уменьшенное исходное искаженное изображение показано на рис. 5.



Рис. 5. Искаженное изображение

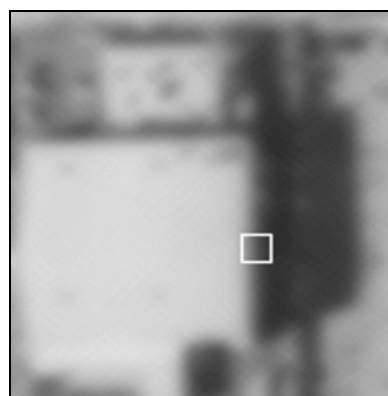


Рис. 6. Участок изображения с отобранным фрагментом

Увеличенный участок изображения, на котором был отобран тестовый фрагмент для определения фильтра, показан на рис. 6. На рис. 7 показаны

(существенно увеличенные): исходный отобранный фрагмент и тот же фрагмент после компьютерного ретуширования.

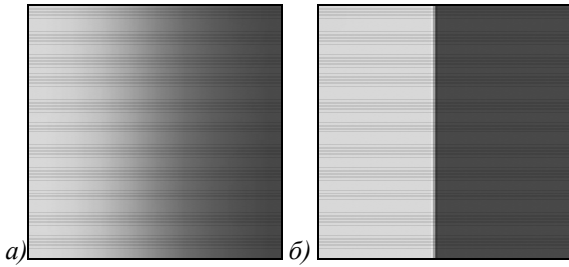


Рис. 7. Тестовые фрагменты: а) исходный; б) после компьютерного ретуширования

По отсчетам функции яркости указанных фрагментов формировалась система типа (5). Для составления матрицы X и вектора y использовалась опорная область 3×3 , показанная на рис. 8.

-1,-1	0,-1	1,-1
-1,0	0,0	1,0
-1,1	0,1	1,1

Рис. 8. Опорная область

Для каждого положения опорной области элементы i -й строки матрицы X (в предположении радиальной симметрии искажений) вычислялись следующим образом:

$$\begin{aligned} X(i,0) &= f(0,0), \\ X(i,1) &= (f(1,0) + f(0,1) + f(-1,0) + f(0,-1)) / 4, \\ X(i,2) &= (f(1,-1) + f(1,1) + f(-1,1) + f(-1,-1)) / 4, \\ X(i,3) &= (g(1,0) + g(0,1) + g(-1,0) + g(0,-1)) / 4, \\ X(i,4) &= (g(1,-1) + g(1,1) + g(-1,1) + g(-1,-1)) / 4, \\ y(i) &= g(0,0), \end{aligned}$$

где $f(i,j)$ – отсчеты опорной области на исходном фрагменте, а $g(i,j)$ – отсчеты опорной области на фрагменте, подвергшемся компьютерному ретушированию.

По полученным матрице X и вектору y вычислялась МНК-оценка $\mathcal{E} = [X^T X]^{-1} \cdot X^T y$ 4×1 – вектора параметров БИХ-фильтра вида (7):

$$\mathcal{E} = [\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_2] = [1, 1,903919; -0,927044; 59,011925; -0,921362; -0,045139].$$

Как указывалось выше, следующий этап технологии должен заключаться в выборе параметра λ для заданных структуры и параметров алгоритма по сформированным малым тестовым фрагментам. Для

этого осуществлялось восстановление исходного фрагмента. Затем вычислялась функция (8), в которой в качестве неискаженного изображения использовался фрагмент, подвергшийся компьютерному ретушированию. Для точек, соответствующих минимальным значениям критерия при различном числе итераций, строилась зависимость числа λ от числа итераций – m (рис. 9).

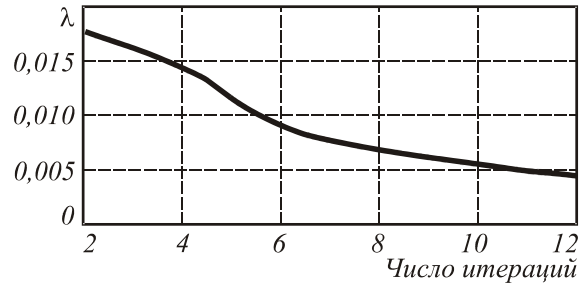


Рис. 9. Зависимость λ от числа итераций

С использованием полученных зависимостей были приняты следующие характеристики фильтра: число итераций – 6, значение параметра λ – 0,01.

Как указывалось выше, выбор параметров алгоритма должен осуществляться также с учетом вычислительных затрат. Для обоснования решений, обеспечивающих компромисс между качеством восстановления и вычислительной сложностью, было проведено исследование зависимости критерия (8) от числа итераций m при различных значениях параметра λ . В частности, для каждого числа итераций устанавливалась величина λ , при которой значение критерия (8) минимально. На рис. 10 приведен вид полученной в работе зависимости критерия (8) от числа итераций, при указанных λ . Легко заметить, что, начиная с некоторого числа итераций (в данном случае 4) уменьшение ошибки восстановления – незначительно.

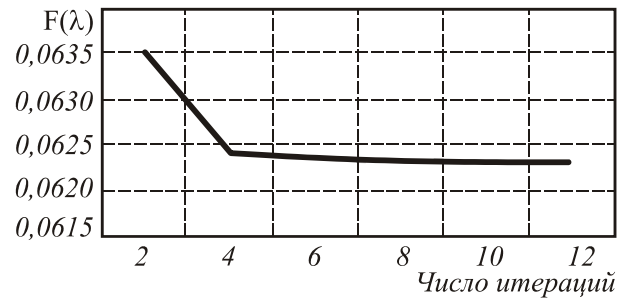


Рис. 10. Зависимость функции (8) от числа итераций

Описанным фильтром была проведена обработка искаженного изображения. На рис. 11а-в представлены для сравнения участок исходного (рис.11а), искаженного, полученного путем моделирования (рис. 11б), и восстановленного (рис. 11в) изображения. Обработка проводилась на многопроцессорной системе (кластере).

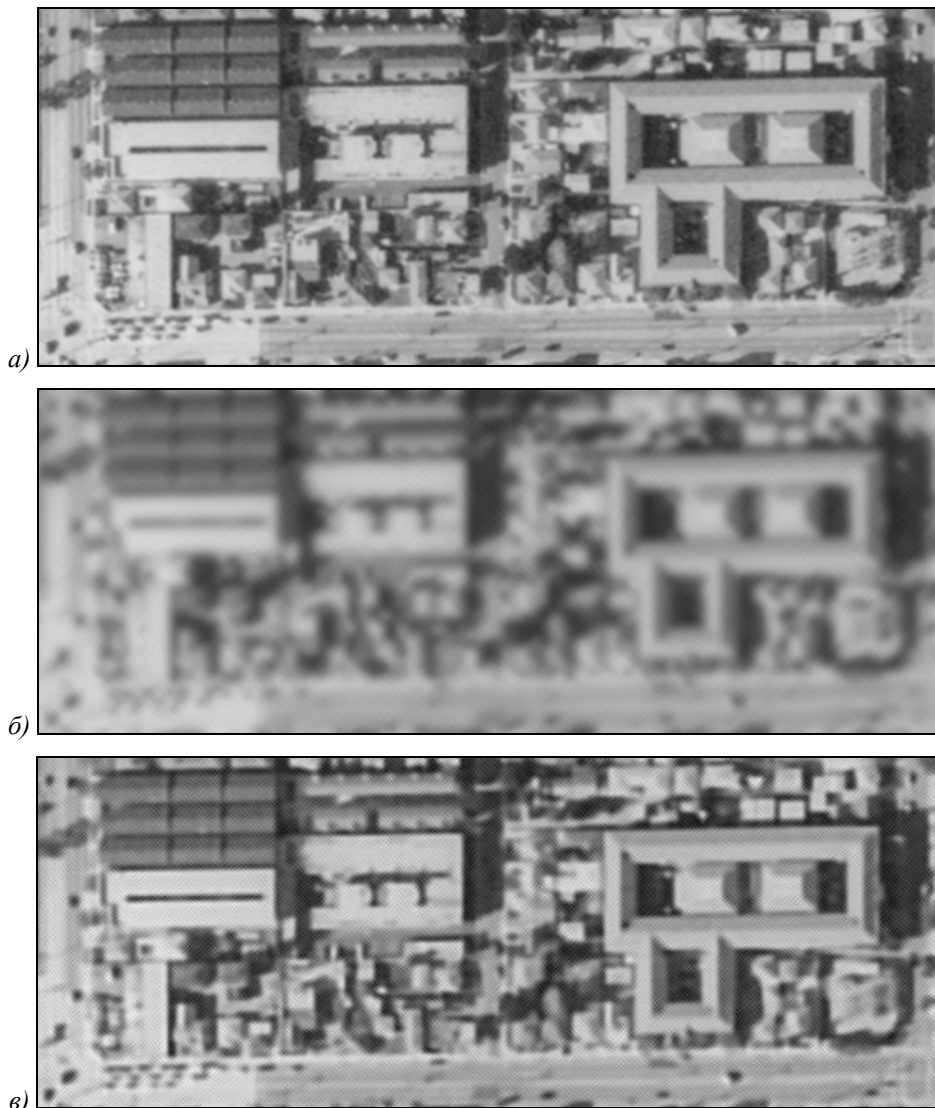


Рис. 11. Участок изображения: а) исходного; б) искаженного; в) восстановленного.

Для описанного восстанавливающего фильтра проводилось исследование эффективности его параллельной реализации на кластере. Исследовалась зависимость ускорения $k = t_1 / t_n$ от количества процессоров. Здесь t_1 – время обработки одним процессором, t_n – время обработки n процессорами. На рис. 12 приведены графики зависимости ускорения от числа процессоров для изображений разных размеров при числе итераций $m = 6$ и одномерной декомпозиции данных.

На рис. 13 приведен построенный по этим же данным график зависимости рекомендуемого (с точки зрения эффективности использования вычислительных ресурсов кластера) числа процессоров от размеров изображения $N \times N$, где $N=1024, 2048, 4096, 8192$.

Для построения графика использовались точки на рис. 12, в которых производная ускорения меняет знак с плюса на минус.

Нетрудно заметить, что это граничные точки, после которых дальнейшее увеличение числа процессоров не приводит к ускорению.

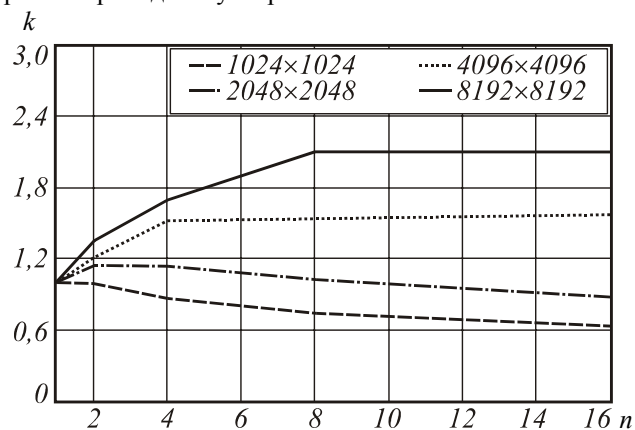


Рис. 12. Зависимость ускорения от количества процессов при одномерной декомпозиции для различных размеров изображений

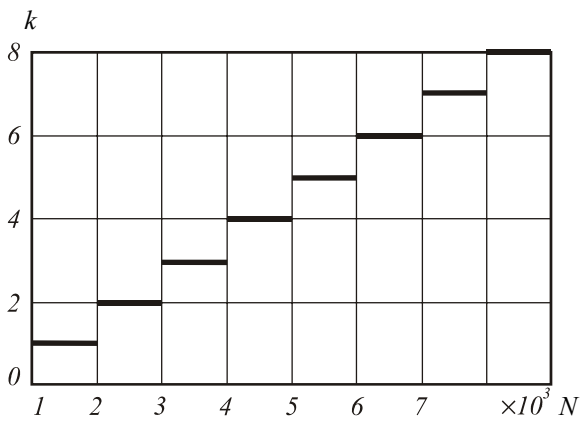


Рис. 13. Зависимость числа процессоров от размеров изображения

На рис. 14 приведены графики зависимости ускорения от числа процессоров для изображений разных размеров при числе итераций $m=6$ и двумерной декомпозиции данных. Из сравнения рис. 12 и 14 видно, что эффективность двумерной декомпозиции превышает эффективность одномерной декомпозиции.

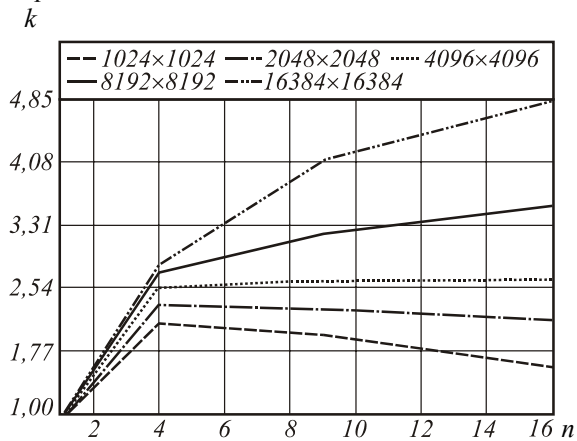


Рис. 14. Зависимость ускорения от количества процессоров при двумерной декомпозиции

Заключение

Полученные результаты показывают принципиальную возможность оперативного построения восстанавливающего фильтра с использованием только исходного искаженного изображения. Кроме того, в рамках рассматриваемой технологии для заданного класса фильтров установ-

лены рекомендуемые значения числа параллельных процессов для различных размеров изображения. Эти результаты могут использоваться при организации оперативной обработки данных аэрокосмического мониторинга Земли.

Благодарности

Работа выполнена при поддержке грантов РФФИ (№ 01-01-00097 и № 03-01-00109), Президента РФ (№ НШ-1007.2003.01) и российско-американской программы «Фундаментальные исследования и высшее образование» («BRHE»).

Литература

1. Методы компьютерной обработки изображений / Под ред. Сойфера В.А., Москва, Физматлит, 2001.
2. Фурсов В.А. Идентификация моделей систем формирования изображений по малому числу наблюдений. Самара 1998.
3. Сергеев В.В., Фурсов В.А., Парфенов С.И. Оценка разрешающей способности видеотракта по фрагментам регистрируемых изображений // Автометрия, 5, 2001. С.25-36.
4. Фурсов В.А., Попов С.Б. Параллельная фильтрация изображений // Труды 2-го международного научно-практического семинара «Высокопроизводительные параллельные вычисления на кластерных системах», Нижний Новгород, 26-29 ноября 2002. С. 307-313.
5. Попов С.Б., Сойфер В.А., Тараканов А.А., Фурсов В.А. Кластерная технология формирования и параллельной фильтрации больших изображений // Компьютерная оптика. N 23, 2002. С. 75-78.
6. Попов С.Б., Скуратов С.А. Пространственное и потоковое распараллеливание в технологиях обработки изображений скользящим окном // Высокопроизводительные параллельные вычисления на кластерных системах. Материалы третьего Международного научно-практического семинара /Под ред. проф. Р.Г. Стронгина. Нижний Новгород: Изд-во Нижегородского госуниверситета, 2003. С.135-140.
7. Фурсов В.А., Шустов В.А., Скуратов С.А. Технология итерационного планирования распределения ресурсов гетерогенного кластера. Труды Всероссийской научной конференции "Высокопроизводительные вычисления и их приложения".- Труды Всероссийской научной конференции г. Черноголовка, 30 октября - 2 ноября 2000 г. с.43-46.