

УДК 681.3, 621.372.542

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ ДВУМЕРНЫХ БИХ-ФИЛЬТРОВ В РАСПРЕДЕЛЕННОЙ СИСТЕМЕ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

А. В. Никоноров¹, М. Г. Милюткин¹, В. А. Фурсов¹

Рассматривается информационная технология построения двумерных фильтров с бесконечной импульсной характеристикой (БИХ-фильтров). Для определения их характеристик используются малые тестовые фрагменты, которые формируются из искаженного изображения с использованием априорной информации о геометрической форме регистрируемых объектов. Двумерный БИХ-фильтр строится в виде параллельного соединения физически реализуемых фильтров с опорной областью в виде квадранта. Предложена архитектура распределенной вычислительной системы, в которой учитывается параллелизм как на уровне декомпозиции крупноформатного изображения, так и на уровне формирования структуры самого БИХ-фильтра. Работа выполнена при финансовой поддержке РФФИ (проект № 09-07-00269-а) и программы № 1 фундаментальных исследований Президиума РАН. Статья подготовлена по материалам доклада авторов на международной конференции “Параллельные вычислительные технологии” (ПавТ-2010; <http://agora.guru.ru/pavt>).

Ключевые слова: обработка цифровых изображений, БИХ-фильтр, идентификация, распределенные вычисления.

1. Введение. В связи с созданием и развитием центров приема космической информации крайне востребованной является задача обработки крупноформатных изображений с целью улучшения их качества (например резкости). Для этой цели обычно используются либо фильтры с конечной импульсной характеристикой (КИХ-фильтры), либо БИХ-фильтры (с бесконечной импульсной характеристикой). КИХ-фильтры всегда физически реализуемы и устойчивы. Однако при значительных искажениях требуется опорная область больших размеров, что приводит к существенному возрастанию вычислительной сложности. Кроме того, возможности улучшения качества изображений с помощью КИХ-фильтров ограничены.

При применении БИХ-фильтра можно обеспечить компенсацию сильных искажений с использованием опорной области небольших размеров. Однако эти фильтры для некоторых форм опорной области физически нереализуемы, кроме того, они могут оказаться неустойчивыми. Связано это с тем, что при радиально симметричных искажениях для определения выходного отсчета используются окружающие его отсчеты опорной области, некоторые из которых еще не вычислены. Указанные проблемы рассматривались в работах [1–4], в частности использовалась итерационная схема реализации.

В настоящей статье для обработки крупноформатных изображений рассматривается параллельное соединение физически реализуемых БИХ-фильтров с опорными областями в виде квадрантов. Реализация таких алгоритмов в распределенной вычислительной системе позволяет использовать декомпозицию задачи на двух уровнях. Первый уровень — декомпозиция по данным (разбиение крупноформатного изображения на фрагменты) между узлами вычислительной системы, второй уровень — функциональная декомпозиция потоков, соответствующих реализации отдельных БИХ-фильтров, каждый из которых реализован в “своем” квадранте опорной области.

2. Общая схема формирования двумерного БИХ-фильтра. Предполагается, что характеристики искажающей системы допускают построение восстанавливающих фильтров в классе линейных инвариантных к сдвигу систем. Известно, что для этого класса систем имеют место ассоциативные и дистрибутивные свойства, обеспечивающие возможность последовательного и параллельного соединения подсистем. В частности, применяя правило дистрибутивности для нахождения общего импульсного отклика для двух параллельно соединенных систем, имеем

$$y = (x * h) + (x * g) = x * (h + g), \quad (1)$$

¹ Самарский государственный аэрокосмический университет, факультет базовой подготовки и фундаментальных наук, Московское шоссе, д. 34, 443086, г. Самара; А.В. Никоноров, доцент, e-mail: admin@mcdk.org; М.Г. Милюткин, инженер, e-mail: m.milutkin@gmail.com; В.А. Фурсов, профессор, e-mail: fursov@ssau.ru

где x — общий входной сигнал, y — общий выходной сигнал, h и g — импульсные отклики систем, $(\dots**\dots)$ означает операцию двумерной свертки. Опираясь на свойство (1), представим двумерный фильтр в виде параллельного соединения физически реализуемых БИХ-фильтров.

В общем случае выражение, определяющее способ вычисления выходного отсчета $y(n_1, n_2)$ двумерного БИХ-фильтра, имеет вид

$$y(n_1, n_2) = \sum_{r_1} \sum_{r_2} a(n_1 - r_1, n_2 - r_2) x(r_1, r_2) - \sum_{k_1} \sum_{k_2} b(n_1 - k_1, n_2 - k_2) y(k_1, k_2), \quad k_1 k_2 \neq n_1 n_2. \quad (2)$$

Нетрудно заметить, что для вычисления значения конкретного выходного отсчета выходная маска должна накрывать только известные значения отсчетов (за исключением вычисляемого). На рис. 1а приведен пример опорной маски 5×5 , для которой требование рекурсивной вычислимости не выполняется (вычисляемый выходной отсчет обозначен кружком в центре опорной области).

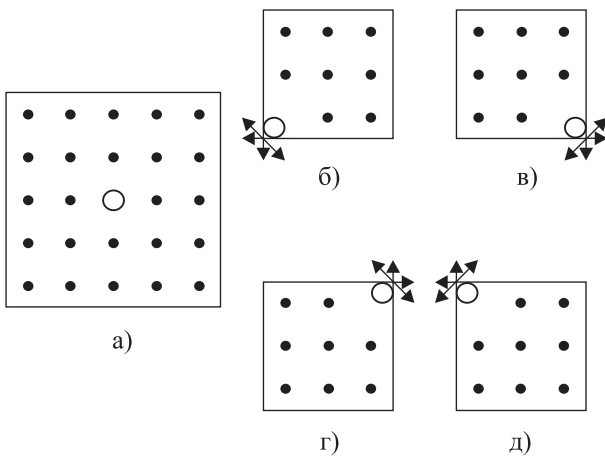


Рис. 1. Исходная маска 5×5 (а), соответствующие ей маски и допустимые направления рекурсии для: 1-го квадранта (б); 2-го квадранта (в); 3-го квадранта (г); 4-го квадранта (д)

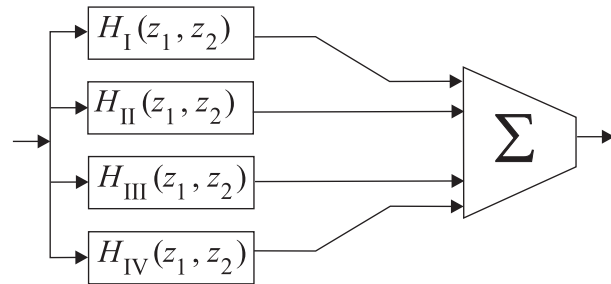


Рис. 2. Параллельное соединение 4-х фильтров одного квадранта, реализующее двумерный БИХ-фильтр на (n_1, n_2) -плоскости

Известно [5], что импульсный отклик симметричного БИХ-фильтра с опорной областью на всей (n_1, n_2) -плоскости можно разбить на четыре отдельных импульсных отклика, по одному на каждый квадрант. На рис. 1б–1д приведены соответственно маски 1-го, 2-го, 3-го и 4-го квадрантов и направления рекурсии для них. С использованием указанного разбиения можно построить фильтры, соответствующие этим квадрантам, и, соединив их параллельно, получить общий импульсный отклик двумерного БИХ-фильтра. На рис. 2 приведена общая схема параллельного соединения БИХ-фильтров одного квадранта, реализующая двумерный БИХ-фильтр на (n_1, n_2) -плоскости.

Пусть построенные на каждом i -м квадранте БИХ-фильтры устойчивы и описываются передаточными функциями вида $H_z^{(i)} = \frac{A_z^{(i)}(z_1, z_2)}{B_z^{(i)}(z_1, z_2)}$, $i = 1, 2, 3, 4$. В соответствии со схемой (рис. 2), составленную из них общую передаточную функцию двумерного БИХ-фильтра на (n_1, n_2) -плоскости можно записать в форме

$$H_z^p(z_1, z_2) = \frac{A_z^p(z_1, z_2)}{B_z^p(z_1, z_2)} = \frac{\sum_{j=1}^4 \prod_{i \neq j} A_z^{(j)}(z_1, z_2) B_z^{(i)}(z_1, z_2)}{\prod_{i=1}^4 B_z^{(i)}(z_1, z_2)}.$$

Нетрудно заметить, что знаменатель общей передаточной функции является произведением знаменателей БИХ-фильтров на квадрантах.

В рамках описанной схемы формирования двумерного БИХ-фильтра наиболее сложная часть задачи заключается в построении устойчивых простых БИХ-фильтров на квадрантах. В следующем разделе описывается использовавшаяся в работах [1–5] и развиваемая здесь технология определения характеристик фильтров путем решения задачи идентификации.

3. Идентификация характеристик простых БИХ-фильтров. Задача построения БИХ-фильтров на квадрантах может быть решена путем идентификации его характеристик по заданным тестовым изображениям. Например, для настройки оптических систем фотоаппаратов часто используются так называемые миры [6]. Идентификация характеристик восстанавливающих БИХ-фильтров в этом случае может осуществляться по всему изображению. Построенный таким образом фильтр, компенсирующий искажения оптики, затем может использоваться для улучшения качества регистрируемых изображений в каждом эпизоде съемки.

Если ставится задача улучшения качества крупноформатных изображений, регистрируемых системами аэрокосмического мониторинга Земли, эталонное изображение, как правило, отсутствует. Тем не менее, можно сформировать “неискаженный” фрагмент изображения, используя априорные сведения о геометрической форме зарегистрированных объектов. Например известно, что граница между крышей здания и отбрасываемой зданием тенью обязана быть резкой. Способ автоматизированного формирования резкой границы на малом фрагменте изображения впервые рассматривался в [7]. В последующих работах [1–4] этот способ неоднократно использовался в различных технологиях.

Для заданной опорной маски в виде квадранта можно выписать соотношение вида (2). Для некоторой пары тестовых фрагментов, один из которых сформирован путем компьютерного регуширования резкой границы, можно записать систему линейных уравнений [8]

$$\mathbf{Y} = S\boldsymbol{\varphi} + \mathbf{o}, \quad (3)$$

где, в соответствии с (2),

$$\mathbf{Y} = \begin{pmatrix} y_1(n_1, n_2) \\ y_2(n_1, n_2) \\ \dots \\ y_N(n_1, n_2) \end{pmatrix}, \quad S = \begin{pmatrix} x_1(r_1, r_2) & \dots & y_1(k_1, k_2) & \dots \\ x_2(r_1, r_2) & \dots & y_2(k_1, k_2) & \dots \\ \dots & \dots & \dots & \dots \\ x_N(r_1, r_2) & \dots & y_N(k_1, k_2) & \dots \end{pmatrix}, \quad \mathbf{o} = \begin{pmatrix} \xi_1(n_1, n_2) \\ \xi_2(n_1, n_2) \\ \dots \\ \xi_N(n_1, n_2) \end{pmatrix},$$

$\boldsymbol{\varphi} = (a(n_1 - r_1, n_2 - r_2), \dots, b(n_1 - k_1, n_2 - k_2), \dots)^T$ — искомый вектор параметров фильтра, N — число отсчетов на тестовых фрагментах, по которым сформирована система (3).

Основная проблема при реализации указанной схемы идентификации состоит в том, что по фрагментам, имеющим перепад функции яркости в одном направлении, может быть идентифицирована передаточная функция невысокого порядка. В то же время, формирование заданной функции яркости для фрагментов, содержащих объекты сложной геометрической формы, представляется не всегда возможным. Связано это обычно как с недостатком априорной информации о деталях изображения, так и с трудностями отыскания резких границ объектов сложной формы. В настоящей работе для идентификации фильтра высокого порядка матрицу S в (3) предлагается составлять из блочных матриц, каждая из которых формируется по простым фрагментам, на каждом из которых функция яркости изменяется в одном направлении, но эти направления для разных фрагментов различны. При этом обеспечивается хорошая обусловленность задачи при идентификации параметров фильтра достаточно высокого порядка.

Для вычисления параметров фильтра по данным системы (3) может использоваться простейшая оценка метода наименьших квадратов (МНК):

$$\boldsymbol{\varphi} = (S^T S)^{-1} S^T \mathbf{Y}. \quad (4)$$

В ряде случаев предпочтительнее более устойчивая к грубым ошибкам типа сбоев оценка метода наименьших модулей (МНМ-оценка).

Заметим, что идентификация восстанавливающего фильтра в классе БИХ-фильтров по соотношениям (3) и (4) в вычислительном отношении существенно выгоднее по сравнению с использованием для этой цели КИХ-фильтров. Связано это с малыми размерами опорных областей. Малые размеры опорной области позволяют формировать тестовые фрагменты малых размеров, что важно с практической точки зрения.

4. Технология обработки цветных изображений. Если исходное искаженное цветное изображение задано в цветовом пространстве RGB (Red, Green, Blue), то каждый отсчет представляется в виде вектора $(r, g, b)^T$. На первом этапе для каждого отсчета цветного изображения осуществляется переход

от модели RGB к модели XYZ по формуле $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix}$ [8]. Затем при

$x_n = 96.422, y_n = 100$ и $z_n = 82.521$ осуществляется переход к модели Lab по формулам

$$L = 116f\left(\frac{y}{y_N}\right) - 16, \quad a = 500\left[f\left(\frac{x}{x_N}\right) - f\left(\frac{y}{y_N}\right)\right], \quad b = 200\left[f\left(\frac{y}{y_N}\right) - f\left(\frac{z}{z_N}\right)\right],$$

$$f(t) = \begin{cases} t^{1/3}, & t > 0.008856, \\ 7.7867t + \frac{16}{116}, & t \leq 0.008856. \end{cases} \quad (5)$$

Известно [5], что компонента L отвечает за уровень яркости, а две другие — за цветообразование, поэтому есть основания полагать, что указанным выше искажениям будет подвергаться именно L -компонента. Следовательно, описанную выше технологию формирования структуры и идентификации параметров простых БИХ-фильтров одного квадранта для улучшения качества цветных изображений достаточно применить к одной L -компоненте. После того как сформирована L -компонента, обработанная параллельными простыми БИХ-фильтрами одного квадранта, осуществляется обратный переход от пространства Lab в пространство RGB. Сначала с использованием формул (5) осуществляется обратный

переход к координатам x, y, z , а затем к r, g, b : $\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$. Это

преобразование осуществляется для каждого отсчета изображения.

5. Архитектура распределенной вычислительной системы. Описанные алгоритмы обработки изображений обладают высокой степенью параллелизма. Этот факт иллюстрируется на примере рассматриваемой ниже реализации соответствующей информационной технологии, в виде распределенной вычислительной системы, подобной рассматривавшейся в работе [9].

Задача обработки изображений хорошо декомпозируется по данным, поэтому для обработки крупноформатного изображения применена простейшая клиент-серверная архитектура. Декомпозицию изображения на фрагменты, с учетом числа и производительности доступных вычислительных узлов, выполняет приложение-сервер, инициирующий распределенные вычисления. Вычисления выполняются приложениями-клиентами локальной сети, которые подключены к серверу. Схема приведена на рис. 3. Важной особенностью реализации алгоритмов фильтрации в данном случае является то, что наряду с декомпозицией по данным на каждом узле реализуется параллельная схема обработки четырьмя БИХ-фильтрами с опорной областью в виде одного квадранта, при этом для каждого элементарного БИХ-фильтра создается свой вычислительный поток, что позволяет эффективно использовать современные многоядерные (multicore) процессоры.

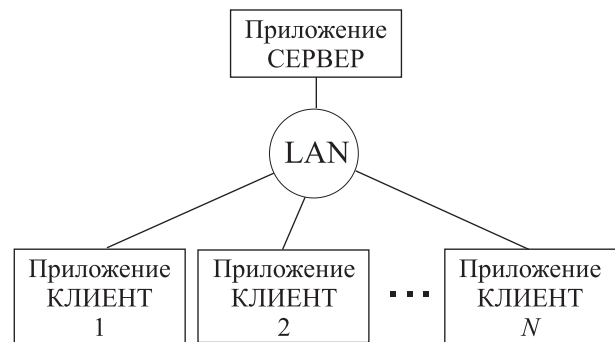


Рис. 3. Схема распределенной системы

Каждый клиент в предложенной схеме выполняет вычисления для своей области данных, такой подход целесообразен для загрузки клиента в фоновом режиме. Приведенное на рис. 1 разбиение на квадранты может быть использовано для многопоточной обработки одного и того же фрагмента изображения.

При таком подходе отдельный поток реализует фильтр одного квадранта. Разбиение, приведенное на рис. 1, описывает 4 квадранта, что естественным образом реализуется четырьмя вычислительными потоками. В случае когда число ядер на узле больше 4-х, например 8 или 16, целесообразно использовать простые БИХ-фильтры с опорной областью в виде сектора, при этом целесообразна структура фильтра, при которой число простых БИХ-фильтров одного сектора совпадает с числом ядер на узле (8 или 16). Описанная схема реализации вычислений на одном многопоточном узле позволит существенно снизить накладные расходы.

Повышение эффективности многопоточного подхода может быть достигнуто также за счет сочетания технологии Java и исполнения native-кода на каждом узле. Такой подход предполагает использование технологии Java JNI (Java Native Interface).

Таким образом, для более эффективной реализации БИХ-фильтров в распределенной многопоточной среде целесообразно использовать два уровня декомпозиции и две технологии для распределения вычислений: декомпозиция данных и Java для распределения вычислений по вычислительным узлам, а также

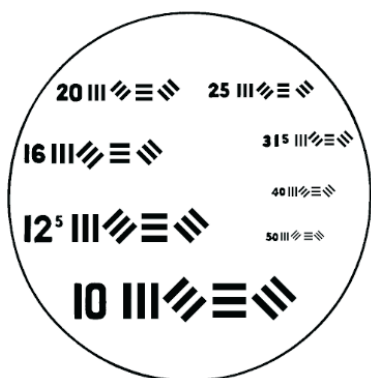


Рис. 4. Исходное изображение



Рис. 5. Искаженное изображение



Рис. 6. Восстановленное изображение



a)



b)

Рис. 7. Спутниковые черно-белые изображения: (a) исходное, (b) восстановленное

многопоточная технология JNI на каждом многоядерном вычислительном узле. С использованием JNI параллельная реализация простых БИХ-фильтров может быть выполнена в среде RDMA (Remote Direct Memory Access).

6. Результаты вычислительных экспериментов. Наиболее просто описанная технология определения характеристик БИХ-фильтров и обработки изображений реализуется в обычной схеме коррекции оптических искажений цифровых фотоаппаратов. Для этого может использоваться тестовое изображение — мира, например показанная на рис. 4. Для идентификации характеристик мира регистрируется фотоаппаратом, а затем изображения исходной и зарегистрированной миры (рис. 5) используются для формирования систем (3) для БИХ-фильтра каждого квадранта.

На рис. 6 приведено полученное суммированием результатов обработки фильтрами, построенными путем идентификации параметров четырех БИХ-фильтров одного квадранта. Восстанавливалось приведенное на рис. 5 изображение, полученное моделированием размытия значений отсчетов гауссианом с радиусом 5 с помощью графического пакета Paint.NET. Для восстановления использовалось параллельное соединение простых БИХ-фильтров одного квадранта, опорные области которых показаны на рис. 16–1д.

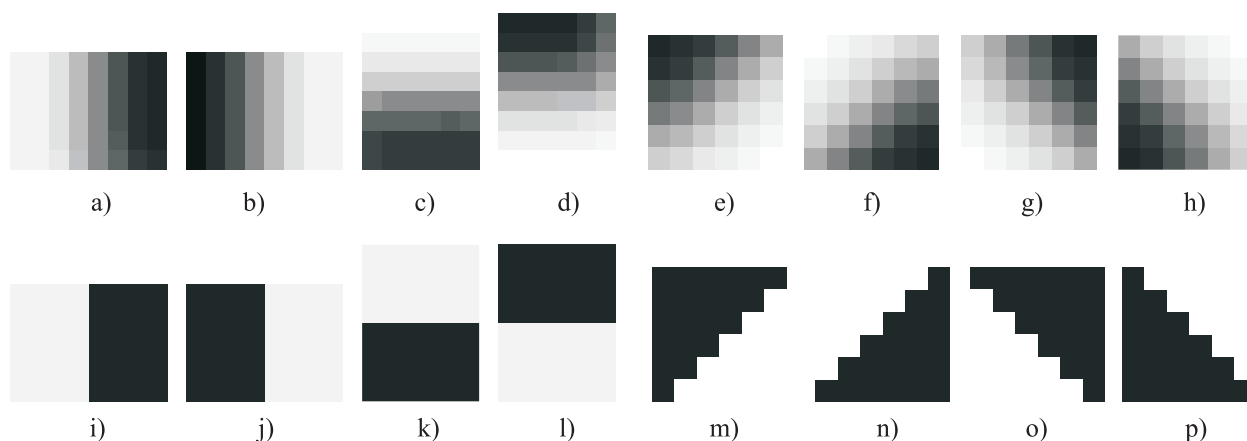


Рис. 8. Тестовые фрагменты: (а)–(h) на исходном изображении, (i)–(p) те же фрагменты после компьютерного ретуширования



Рис. 9. Спутниковые цветные изображения: (а) исходное, (б) восстановленное

На рис. 7 приведен другой пример применения описанной технологии к обработке черно-белых аэрокосмических снимков. Как и в предыдущем случае, использовались четыре параллельно соединенных БИХ-фильтра одного квадранта с опорными областями, показанными на рис. 1б–1д.

Для идентификации параметров фильтров одного квадранта использовались тестовые фрагменты малых размеров, показанные на рис. 8а–8h, с изменением функции яркости в одном из 8-и направлений. Соответствующие им “неискаженные” фрагменты, сформированные путем компьютерного ретуширования, показаны на рис. 8i–8p. В соответствии с количеством пар тестовых фрагментов было сформировано

8 матричных блоков размерности 30×17 , из которых затем составлена матрица S размером 240×17 .

На рис. 9 приведен пример обработки цветных аэрокосмических снимков. Использовалась технология, описанная в разделе 4. В качестве тестовых использовались фрагменты, аналогичные показанным на рис. 8. Отличие состояло в том, что функции яркости на фрагментах формировались только для одного компонента цветного изображения, а именно для L -компонента.

7. Заключение. Рассмотренный подход к построению процедур обработки крупноформатных изображений может служить примером того, как путем отказа от традиционного метода реализации алгоритмов обработки изображений (КИХ-фильтрами) удастся существенно повысить эффективность параллельной вычислительной системы с многоядерными узлами. Рассмотренная в настоящей статье реализация в виде распределенной вычислительной системы связана с используемым способом распределенного хранения крупноформатных изображений. Описанный подход без существенных изменений может быть реализован также на кластере.

Наиболее интересными для дальнейших исследований являются направления, связанные с различными вариантами декомпозиции, при этом возможны различные схемы распределенной реализации как на кластерных системах, так и на системах с общей памятью. Представляет также интерес анализ эффективности параллельной реализации простых БИХ-фильтров в RDMA-системах.

СПИСОК ЛИТЕРАТУРЫ

1. Попов С.Б., Соيفер В.А., Тараканов А.А., Фурсов В.А. Кластерная технология формирования и параллельной фильтрации больших изображений // Компьютерная оптика. 2002. № 23. 75–78.
2. Дроздов М.А., Зимин Д.И., Скуратов С.А., Попов С.Б., Фурсов В.А. Кластерная технология определения восстанавливающих фильтров и обработки больших изображений // Компьютерная оптика. 2003. № 25. 175–182.
3. Зимин Д.И., Фурсов В.А. Технология определения восстанавливающего фильтра и обработки цветных изображений // Компьютерная оптика. 2005. № 27. 170–173.
4. Зимин Д.И., Фурсов В.А. Построение устойчивых алгоритмов обработки изображений путем аппроксимации фильтров с бесконечной импульсной характеристикой // Компьютерная оптика. 2005. № 28. 124–127.
5. Dudgeon D.E., Mersereau R.M. Multidimensional digital signal processing. Prentice Hall: Englewood Cliffs, 1984.
6. <http://ru.wikipedia.org/wiki/>
7. Sergeev V.V., Fursov V.A., Parfyonov C.I. Information technology for evaluating the resolving power of the video channel with the use of a model filter with infinite impulse response // Pattern Recognition and Image Analysis. 1999. 9, N 2. 314–316.
8. Методы компьютерной обработки изображений / Под ред. В.А. Соифера. М.: Физматлит, 2001.
9. Никоноров А.В., Фурсов В.А. Распределенная вычислительная среда коррекции цветных изображений // Тр. XV Всероссийской научно-методической конференции “Телематика 2008”. СПб., 2008. 88–89.

Поступила в редакцию
11.03.2010
