

# БЛОЧНЫЙ АЛГОРИТМ ПОСТРОЕНИЯ СВЕРХРАЗРЕШЕНИЯ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ФИЛЬТРА КАЛМАНА

А. А. Сирота, А. Ю. Иванков

*Воронежский государственный университет*

Поступила в редакцию 18.10.2012 г.

**Аннотация.** В работе представлен метод обработки последовательности изображений, позволяющий построить итоговое изображение со сверхвысоким разрешением. В основе метода лежит использование фильтрации Калмана в блочной форме, позволяющей сократить размерность матриц, используемых при реализации алгоритма фильтрации.

**Ключевые слова:** обработка изображений, сверхразрешение, фильтр Калмана.

**Annotation.** This paper represents the approach to processing image sequence for superresolution restoration. This method based on the Kalman filtering in block form that can reduce the size of the matrices used in the implementation of algorithms.

**Keywords:** image processing, superresolution, Kalman filtering.

## ВВЕДЕНИЕ

Графическое разрешение ограничено физическими параметрами системы. Это является проблемой во многих приложениях, связанных с получением изображений. Графические системы получают искаженные или неполные изображения, если плотность их чувствительных матриц недостаточна высока. Одним из вариантов увеличения разрешения изображений является увеличение чувствительных элементов матрицы, однако при этом возрастает стоимость оборудования. Альтернативный подход – это реализация методов обработки изображений, позволяющих повысить разрешение. Одним из таких методов является метод многокадрового сверхразрешения (СР) [1–6]. При его использовании достигается эффект восстановления изображения с высоким разрешением (ВР) из некоторого количества изображений с низким разрешением (НР). При этом НР-изображения представляют собой различные взгляды на одну и ту же сцену и смещены на доли пикселя.

Целью данной работы является синтез алгоритма обработки последовательности изображений, позволяющий построить изображение со сверхвысоким разрешением на основе принципов калмановской фильтрации. В работах [2–6] рассматривается подход к построению СР-изображений, с применением фильтра Калмана. В данной работе предложен способ ис-

пользования фильтра Калмана применительно к задаче построения сверхразрешения изображения в блочной реализации, позволяющей сократить размерность матриц, используемых при построении алгоритма восстановления изображения с высоким разрешением.

## МОДЕЛЬ ИЗОБРАЖЕНИЯ В ЗАДАЧЕ СВЕРХРАЗРЕШЕНИЯ

Изначально полагается, что в качестве наблюдаемых данных выступает последовательность НР-изображений  $\mathbf{y}_k$ , где  $k = 1, \dots, K$ . Изображения имеют размер  $M \times M$  пикселей и представлены в ходе последующего анализа как одномерный вектор, полученный после соответствующей развертки исходной матрицы. В качестве оцениваемых данных рассматривается последовательность изображений  $\mathbf{x}_k$  с высоким разрешением, где каждое изображение имеет размер  $L \times L$ , ( $L > M$  в  $t$  раз). Каждое НР-изображение связано с соответствующим ВР-изображением соотношением

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \quad (1)$$

где  $\mathbf{H}_k$  – матрица прореживания, которая осуществляет преобразование изображения с высоким разрешением  $\mathbf{x}_k$  в изображение с низким разрешением  $\mathbf{y}_k$  и моделирует процесс пространственного усреднения при получении наблюдаемого изображения;  $\mathbf{v}_k$  – вектор аддитивного гауссовского шума с нулевым математическим ожиданием и матрицей ковариации  $\mathbf{R} = \sigma_R^2 \mathbf{I}$ .

Далее предполагается, что последовательность ВР-изображений  $\mathbf{x}_k$  удовлетворяет следующему уравнению:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k, \quad (2)$$

где  $\mathbf{F}_k$  – матрица сдвига, характеризующая геометрические взаимные деформации (смещения) изображений  $\mathbf{x}_{k+1}$  и  $\mathbf{x}_k$ , соответствующие перемещению камеры и/или объекта в процессе получения изображений;  $\mathbf{w}_k$  – гауссовский шум с нулевым математическим ожиданием и матрицей ковариации  $\mathbf{Q} = \sigma_Q^2 \mathbf{I}$ .

Рассмотрим системные матрицы, участвующие в (1) и (2). Для понижения разрешения

$$h_{r,t}^{(k)} =$$

$$= \begin{cases} 1, r = ((i-1)m + (q-1))M + (j-1)m + \\ + p, t = ((i-1)Mm + (j-1)m + 1, \\ 0, r \neq ((i-1)m + (q-1))M + (j-1)m + \\ + p, t \neq ((i-1)Mm + (j-1)m + 1, \end{cases} \quad (3)$$

где  $i = 1, \dots, M$ ,  $j = 1, \dots, M$ ,  $p = 1, \dots, m$ ,  $q = 1, \dots, m$ . Например, матрица, понижающая разрешение изображения размером  $4 \times 4$  пикселя в 2 раза, выглядит следующим образом:

$$\mathbf{H}_k = \begin{pmatrix} 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, \end{pmatrix}$$

ВР-изображения используется матрица прореживания  $\mathbf{H}_k$ . Вид этой матрицы определяет способ получения НР-кадра из ВР-изображения. Предполагается, что матрица остается неизменной от кадра к кадру. В данной работе матрица подобрана таким образом, что, каждому пикселю в  $\mathbf{y}_k$  соответствует значение левого верхнего пикселя в блоке, размера  $m \times m$  пикселей, в  $\mathbf{x}_k$ , что показано на рис. 1.

Заполнение матрицы прореживания для изображения размер  $M \times M$  пикселей при понижении разрешения в  $m$  раз происходит в соответствии со следующей формулой:

$$\mathbf{H}_k = \left\| h_{r,t}^{(k)} \right\|;$$

Матрица сдвига  $\mathbf{F}_k$  характеризует перемещение датчика относительно сцены от кадра к кадру. Это перемещение осуществляется по

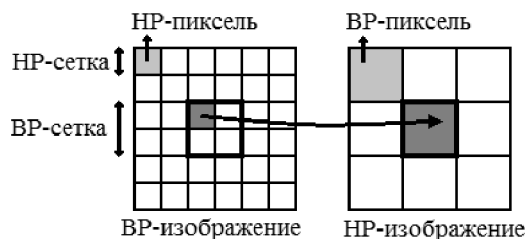


Рис. 1. Иллюстрация работы матрицы прореживания  $\mathbf{H}_k$  при понижении разрешения в 2 раза



$$\begin{aligned} \mathbf{U}_k &= \mathbf{M} \left[ \left( \mathbf{y}_k - \tilde{\mathbf{y}}_{k|k-1} \right) \left( \mathbf{y}_k - \tilde{\mathbf{y}}_{k|k-1} \right)^T \right] = \\ &= \mathbf{M} \left[ \left( \mathbf{H}_k \boldsymbol{\varepsilon}_k + \mathbf{v}_k \right) \left( \mathbf{H}_k \boldsymbol{\varepsilon}_k + \mathbf{v}_k \right)^T \right] = \\ &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}, \\ \mathbf{P}_{k|k} &= \mathbf{M} \left[ \left( \tilde{\mathbf{x}}_{k|k} - \mathbf{x}_k \right) \left( \tilde{\mathbf{x}}_{k|k} - \mathbf{x}_k \right)^T \right] = \\ &= \mathbf{P}_{k|k-1} - \mathbf{V}_k \mathbf{U}_k^{-1} \mathbf{V}_k^*. \end{aligned}$$

Рекуррентное выражение для оценки  $\tilde{\mathbf{x}}_{k+1|k}$  на  $k+1$  шаге имеет вид

$$\begin{aligned} \tilde{\mathbf{x}}_{k+1|k} &= \mathbf{M}[\mathbf{x}_{k+1} | \mathbf{y}^k] = \mathbf{F}_k \tilde{\mathbf{x}}_{k|k} = \\ &= \mathbf{F}_k \tilde{\mathbf{x}}_{k|k-1} + \mathbf{F}_k \mathbf{W}_k \left( \mathbf{y}_k - \tilde{\mathbf{y}}_{k|k-1} \right). \end{aligned} \quad (5)$$

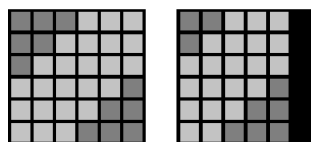
Рекуррентное выражение для матрицы  $\mathbf{P}_{k+1|k}$  на  $k+1$  шаге имеет вид

$$\mathbf{P}_{k+1|k} = \mathbf{M}[\boldsymbol{\varepsilon}_{k+1} \boldsymbol{\varepsilon}_{k+1}^T] = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^T + \mathbf{Q}. \quad (6)$$

На первом шаге  $\tilde{\mathbf{x}}_{1|0}$  и  $\mathbf{P}_{1|0}$  известны как начальные условия. Начальное значение оценки изображения и матрицы ковариации ошибки определяется из априорной информации об оцениваемом изображении. Если о нем ничего неизвестно, можно предположить, что каждый пиксель изображения с одинаковой вероятностью может принимать любое доступное значение. Таким образом, значения пикселей оцениваемого изображения представляют собой стандартную равномерно распределенную случайную величину. Если считать, что диапазон значений такой величины равен  $\Delta$ , то её среднее значение  $m$  и дисперсия  $D$  вычисляется как

$$m = \frac{\Delta}{2}, \quad D = \frac{\Delta^2}{12}.$$

Изначально известен диапазон значений  $\Delta$ , которые может принимать каждый пиксель изображения (от 0 до 255 или от 0 до 1, как в данном эксперименте). Тогда  $\tilde{\mathbf{x}}_{1|0}$  представляет собой вектор, каждый элемент которого равен среднему значению, то есть  $m$ . Матрица ковариации ошибки  $\mathbf{P}_{1|0}$  определяется как



исходное изображение      смещенное изображение

а)

$\mathbf{P}_{1|0} = \mathbf{D}\mathbf{I}$ ,  
где  $\mathbf{I}$  – единичная матрица.

### БЛОЧНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ФИЛЬТРАЦИИ

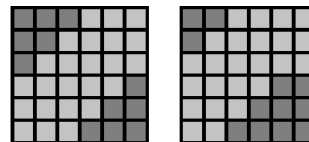
Одной из проблем реализации данного алгоритма является необходимость учета краевых эффектов, которые возникают при применении формулы (5) из-за того, что крайний столбец смещаемого изображения (оценки) заполняется нулевыми элементами (пиксели  $\mathbf{x}_{k+1}$ , которые не могут быть получены из  $\mathbf{x}_k$ ). Для изображения, смещаемого влево, эту ситуацию иллюстрирует рис. 2,а. Так же, нулями заполняется и соответствующая часть матрицы ковариации ошибки. Это неверно, так как в этом случае выходит, что ошибка в крайнем столбце между оценкой и оцениваемым изображением равна нулю, что не соответствует действительности. Это повлияет на расчет матрицы весов – получается, что оценка в этом месте корректироваться не будет.

Одним из вариантов решения – учесть новые значения пикселей в  $\mathbf{w}_k$ , как это сделано в [2]. При этом вектор  $\mathbf{w}_k$  содержит значения крайних пикселей для  $\mathbf{x}_{k+1}$ , а в матрице  $\mathbf{P}_{k+1|k}$  соответствующие значения учитываются с помощью матрицы  $\mathbf{Q}$ .

Другим вариантом является дополнение соответствующих элементов матрицы  $\mathbf{F}_k$  единичными значениями, что эквивалентно инициализации соответствующих элементов  $\mathbf{x}_{k+1}$  и  $\mathbf{P}_{k+1|k}$ . Так для матрицы сдвига изображения размером  $M \times M$  на 1 пиксель влево, единицы дополняются по диагонали:

$$\begin{aligned} \mathbf{F}_k &= \left\| f_{i,i}^{(k)} \right\|; \\ f_{i,i}^{(k)} &= \begin{cases} 1, & i = (M-1)M + 1, \dots, M^2, \\ 0, & i = 0, \dots, (M-1)M. \end{cases} \end{aligned} \quad (7)$$

В этом случае будет дублироваться соседний столбец, как это показано на рис. 2, б.



исходное изображение      смещенное изображение

б)

Рис. 2. Смещение изображения за счет матрицы  $\mathbf{F}_k$   
а) без дополнения матрицы единицами; б) с дополнением матрицы единицами

Матрица сдвига, дополненная единицами по диагонали имеет вид:

читается соответственно размеру обрабатываемых изображений. Таким образом, теряться будут

$$\mathbf{F}_k = \begin{pmatrix} 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, \end{pmatrix}$$

По большому счету не важно, как заполняются пустые элементы в матрице сдвига. Дублируемые пиксели в оценке будут пересчитаны при коррекции, если в весовой матрице  $\mathbf{W}_k$  для соответствующих элементов не будет нулевых строк. Чтобы избежать нулевых строк в весовой матрице, следует не допустить их в матрице ковариации ошибки. Предлагаемая структура матрицы сдвига позволит предотвратить этот эффект.

При использовании нескольких матриц сдвига, например, для смещения влево и вправо, крайние столбцы изображения-оценки могут теряться. Так при смещении изображения влево, левый столбец теряется, а при последующем смещении изображения вправо вместо утерянного левого столбца дублируется соседний столбец. Подобная ситуация возникает с матрицей ковариации ошибки, так как в её пересчете так же участвует матрица сдвига (6).

Чтобы решить эту проблему, можно искусственно расширять каждое НР-изображение и изображение-оценку на  $m$  пикселей с каждой стороны, а после обработки вырезать из центра полученного изображения-оценки регион, соответствующий исходному размеру изображения. Размер системных матриц так же увели-

искусственно дополненные столбцы и строки, что не скажется на изображении-оценке. В общем случае ширину границ определяет вид матрицы  $\mathbf{H}_k$ .

Еще одна проблема реализации метода калмановской фильтрации – чрезмерно большой размер матриц, которые нужно обрабатывать при получении ВР-изображений. На практике это приводит к возникновению эффектов расходимости алгоритма фильтрации. Решением служит обработка изображений блоками. В этом случае НР-кадры делятся на блоки, которые обрабатываются как отдельные изображения. При этом обрабатывается не единая последовательность НР-изображений, а параллельно несколько последовательностей блоков, на которые делятся НР-изображения. В итоге получится оценка для каждой последовательности блоков, которая по окончании процесса обработки собирается в одно результирующее изображение. Данная ситуация продемонстрирована на рис. 3. При этом все блоки одного изображения используют одну и ту же матрицу ковариации ошибки, так как она меняется при переходе к другому изображению, но не меняется при переходе между блоками одного изображения.

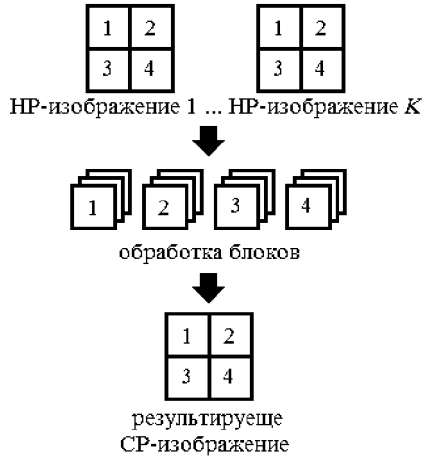


Рис. 3. Блочная обработка последовательности HR-изображений

При поступлении очередного HR-изображения  $\mathbf{g}^k$  размером  $M \times M$  пикселей, оно преобразуется в изображение  $\mathbf{f}^k$ , расширенное на  $m$  пикселей с каждой стороны (размером  $(M + 2m) \times (M + 2m)$ ):

$$\mathbf{f}_{m+i,m+j}^k = \mathbf{g}_{i,j}^k, \quad i = 1, \dots, M, \quad j = 1, \dots, M. \quad (8)$$

Изображение делится на  $B$  блоков по вертикали и по горизонтали, каждый блок имеет размер  $b \times b$ , где  $b = M / B$ . Соответственно, имеется массив изображений оценок  $\tilde{\mathbf{x}}$  для каждого блока, длина массива  $B \times B$ , каждый элемент в нем расширен на  $m$  пикселей с каждой стороны и развернут в вектор размером  $(b + 2m) \times 1$ . При обработке блока из расширенного HR-изображения  $\mathbf{f}^k$  извлекается соответствующий регион  $\mathbf{s}^{p,q}$  размер  $(b + 2m) \times (b + 2m)$ :

$$\mathbf{s}^{p,q} = \mathbf{f}_{m+(p-1)b-m+i, m+(q-1)b-m+j}^k, \quad (9)$$

где  $i = 1, \dots, b + 2m, j = 1, \dots, b + 2m, p = 1, \dots, B, q = 1, \dots, B$ . Регион  $\mathbf{s}^{p,q}$  раскладывается в вектор по столбцам, таким образом преобразуется в вектор  $\mathbf{y}^{p,q}$ . Уравнение (4) в данном случае имеет вид

$$\tilde{\mathbf{x}}^{p,q} = \mathbf{F}_k \tilde{\mathbf{x}}^{p,q} + \mathbf{F}_k \mathbf{W}_k (\mathbf{y}^{p,q} - \mathbf{H}_k \tilde{\mathbf{x}}^{p,q}), \quad (10)$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^T + \mathbf{Q}. \quad (11)$$

После обработки всех HR-изображений, каждый элемент массива  $\tilde{\mathbf{x}}^{p,q}$  преобразуется в двумерный вектор, у которого обрезаются  $m$  пикселей с каждой стороны. Из полученных элементов формируется результирующее изображение-оценка. Точность восстановления изображения на краях блоков зависит от вида матрицы  $\mathbf{H}_k$ , а

так же от количества пикселей, на которое блок расширяется с каждой стороны.

## РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТА

Прежде всего необходимо доказать достоверность алгоритма, основанного на фильтрации Калмана. Этим доказательством может послужить совпадение теоретической и практической ошибки оценивания и восстановления изображений. В качестве теоретической ошибки выступает след матрицы ковариации, рассчитанной в ходе работы алгоритма по формуле (11). В качестве практической ошибки выступает след усредненной по  $N$  реализациям реальной матрицы ковариации ошибки, вычисленный как

$$d_k = \frac{1}{N} \sum_{t=1}^N \sum_{i=1}^{M^2} (\mathbf{x}_i^{(t)} - \tilde{\mathbf{x}}_{k|k,i}^{(t)})^2, \quad (12)$$

где  $k$  – номер последнего шага фильтрации;  $t$  – номер реализации,  $\tilde{\mathbf{x}}_{k|k,i}$  – оценка  $i$ -го пикселя, полученная на  $k$ -м шаге в  $t$ -й реализации;  $\mathbf{x}_i$  – значение  $i$ -го пикселя изображения (сгенерированного случайного поля), оцениваемого в процессе  $t$ -й реализации эксперимента. При каждой реализации в качестве исходного изображения генерируется случайное поле, затем из него создается последовательность HR-изображений, которую обрабатывает алгоритм. В результате работы алгоритма имеем оценку исходного изображения по серии HR-изображений (SR-изображение) и матрицу ковариации ошибки. При проведении опыта показана хорошая точность вычислений.

На рис. 4, а представлена зависимость следа матрицы ковариации ошибки, полученной экспериментальным и теоретическим путем от числа реализаций. На рис. 4, б показана зависимость средней квадратичной ошибки между исходным и восстановленным изображением от величины шума  $\sigma_R$

На рис. 5, б представлен результат повышения разрешения в 4 раза из 16 HR-изображений, полученных из оригинального изображения рис. 5, а. HR-изображения смещены друг относительно друга на 0.25 пикселя. Величина шума:  $\sigma_R = 0.05$ .

## ЗАКЛЮЧЕНИЕ

В данной работе приведен метод построения сверхразрешения нескольких HR-изображений, основанный на процедуре фильтрации Калмана. За счет блочной обработки изображений сокра-

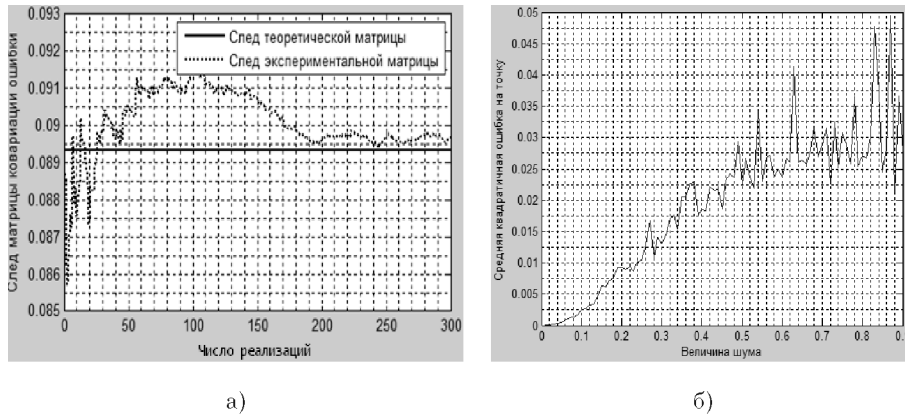


Рис. 4. Графики зависимостей:  
 а) следов матриц ковариации ошибки, полученной экспериментальным и теоретическим путем от числа реализаций при усреднении;  
 б) средней квадратичной ошибки между исходным и восстановленным изображением от величины шума  $\sigma_R$



Рис. 5. Повышение разрешения изображений в 4 раза: а) исходное оцениваемое изображение; б) восстановленное изображение; в), г) НР-кадры, полученные из исходного ВР-изображения. Изображения в), г) смещены друг относительно друга на 0.25 пикселя. Величина шума:  $\sigma_R = 0.05$

щается размерность матриц, используемых при реализации алгоритма фильтрации. При этом появляется возможность параллельной обработки блоков одного изображения.

#### СПИСОК ЛИТЕРАТУРЫ

1. Super-resolution image reconstruction: a technical overview / Sung Cheol Park [и др.] // Signal Processing Magazine, IEEE, 2003. – Т. 20, №. 3. – С. 21–36.

2. Elad M. Super-resolution reconstruction of continuous image sequences / M. Elad, A. Feuer // International Conference on Image Processing (ICIP 99), 1999. – Т. 3. – С. 459–463.

3. Elad M. Super-resolution reconstruction of continuous image sequences: adaptive filtering approach / M. Elad, A. Feuer // Image Processing, IEEE Transactions on, 1999. – Т. 8, №. 3. – С. 387–395.

4. A new motion-compensated reduced-order model kalman filter for space-varying restoration of progressive and interlaced video / A. Patti [и др.] //

Image Processing, IEEE Transactions on, 1998. – Т. 7, №. 4. – С. 543–554.

5. Modified Kalman Filtering for Image Super-Resolution / C. Newland [и др.] // Image and Vision Computing New Zealand 2006 (IVCNZ06), 2006. – С. 79–84.

6. Time Invariant Steady-State Kalman Filter for Image Super-Resolution / C. Newland [и др.] // Image and Vision Computing New Zealand 2005 (IVCNZ05), 2006. – С. 381–387.

**Сирота Александр Анатольевич** – доктор технических наук, профессор кафедры информационных систем Воронежского государственного университета. Тел.: 89030306943. E-mail: sir@cs.vsu.ru

**Sirota A. A.** – doctor of technical sciences, professor VSU information system department. E-mail: sir@cs.vsu.ru, Tel: 89030306943

**Иванков А. Ю.** – аспирант кафедры информационных систем Воронежского государственного университета

**Ivankov A. J.** – VSU information system department postgraduate student