УДК 6

# ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ДЛЯ ОТЫСКАНИЯ ВОДЯНЫХ ЗНАКОВ В ИЗОБРАЖЕНИИ

**@ 2006 г. Е.Л. Столов, д.т.н.**

Казанский государственный университет

Предложен метод связывания с изображением специальной информации («водяных знаков»), предназначенной для идентификации изображения, и быстрый параллельный алгоритм для извлечения этой информации. Показана устойчивость предложенных знаков к операциям кадрирования и изменения размера изображения.

## *Введение*

Проблемы компьютерной безопасность привлекают в последнее время самое пристальное внимание исследователей. Это связано, прежде всего, с все более широким использованием электронных документов вместо бумажных. Среди аспектов этой проблемы – задачи о проверки целостности документа и вопросы, связанные с соблюдением авторских прав. Для чисто текстовых документов эта проблема решается с помощью введения электронной подписи. Если же дело идет о документах, содержащих графическую информацию, то задача становится значительно более трудной. Дело в том, что графическая информация сохраняет свою ценность и после различных преобразований, поэтому обычная технология, основанная на цифровой подписи, здесь не применима. В настоящее время публикуется большое число работ, посвященных указанной проблеме. Они объединены общим названием «водяные знаки», или watermarks. Речь идет о внедрении в графическое изображение дополнительной информации или о связывание с таким документом некоей сопутствующей информации, с помощью которой могут быть

решена проблема целостности документа, проблема подтверждения авторства и другие, относящиеся к данной тематике. Параллельно возникла область знаний в задачах компьютерной безопасности, нацеленная на обнаружение и удаление водяных знаков. Поскольку обе указанные задачи требуют значительных вычислительных ресурсов, использование для их решения параллельных вычислений представляется весьма уместным.

В данной работе решается следующая задача. Среди большого количества изображений нужно отыскать данное изображение. В процессе поиска вычисляются специальные характеристики изображения (водяные знаки), с помощью которых и производится идентификация изображения. Найденные значения могут служить и доказательством принадлежности исследуемого изображения данному автору. Основная проблема, связанная с внедрением водяных знаков, заключается в сохранении этих знаков после различных преобразований: всевозможных процедур сжатия, кадрирования рисунка, малых локальных искажений. Достаточно полный обзор возникающих здесь проблем и применяемых методов представлен в [1], где представлена обширная библиография. Чтобы обозначить место данной статьи среди остальных работ, остановимся вкратце на технологиях, используемых в данной области. Первоначально наибольшую популярность получили методы, основанные на модификации младших битов изображения специальным образом. Это может быть внесение сигнала с известной частотой, либо распределение битов некоторого сообщения согласно значениям выбранной хэш-функции. Такое преобразование не приводит к искажению восприятия изображения. Атака на внедренный водяной знак осуществляется добавлением аддитивного шума ко всем пикселям изображения. Большинство обсуждаемых в настоящее время методов основаны на коэффициентах дискретного косинус преобразования (DCT) или дискретного вайвелет преобразования. Изображение разбивается на блоки, например, для создания изображения в формате JPEG используют $8\times8$ блоки, после чего модифицируют специальным образом коэффициенты блока. Предложены многочисленные методы помещения информации в эти

коэффициенты. Упомянутые методы устойчивы к методам компрессии и сжатия. Атака заключается в кадрировании изображения, в результате чего меняется разбиение его на блоки указанного размера. Другой тип атак заключается в аффинных преобразованиях малых блоков , что также не меняет восприятие, но изменяется коэффициенты блоков. Устойчивость к упомянутым атакам повышается путем применения избыточного кодирования для исправления возможных ошибок. Следующий набор методов основан на рассмотрении изображения в целом. Вычисляются коэффициенты Фурье, DCT или вайвелет преобразований всего изображения, и тем или иным способом передаются значения наиболее значимых из них. К этому направлению примыкает данная статья. Как и выше, основная проблема заключается в проверке устойчивости хранимой информации к различным преобразованиям. На примерах показывается, что предлагаемая техника обладает необходимой устойчивостью.

Любое изображение можно рассматривать как совокупность областей различной интенсивности, расположенных специальным образом друг относительно друга. Метод основан на выделении этих областей и определении их взаимного расположения. Процедура извлечения предлагаемых характеристик требует значительных вычислительных ресурсов. Разработан параллельный алгоритм, реализующий данную процедуру. Поскольку изображение имеет смысл лишь до тех пор, пока после всех трансформаций сохранены его основные свойства, предлагаемый метод идентификации останется эффективными , если преобразованное изображение содержательно адекватно оригиналу.

**Математическая модель**

Как указано выше, в основе методов, рассматривающих изображение в целом, лежит идея разложения этого изображения по некоторому базису. Поскольку устойчивость такого разложения к различным преобразованиям не

очевидна, мы предлагаем другой подход, согласно которому изображение представляется как объединение некоторых стандартных областей.

Путь имеется евклидово пространство $L$ размерности $n$ со скалярным произведением $(\alpha,\beta)$ и вектор $\alpha \in L$. Пусть, далее, имеется множество нормированных векторов $\{B : \beta_j \in L,\ j = 0,...,m\}$. Будем предполагать, что $B$ содержит подмножество, образующее базу в $L$. Рассмотрим следующую процедуру, дающую возможность аппроксимировать вектор $\alpha$ с помощью векторов из $B$.

1. Положим $k = 0$ и $\alpha_k = \alpha$
2. Среди векторов $B$ найдем такой вектор $\beta$, для которого $|(\alpha_k, \beta)|$ достигает максимума. Обозначим этот вектор через $\gamma_k$
3. Положим $\alpha_{k+1} = \alpha_k - (\alpha_k, \gamma_k)\gamma_k$
4. Если выполнено условие остановки, алгоритм заканчивает работу, иначе положим $k = k + 1$ и перейдем к шагу 2.

Обозначим для краткости $(\alpha_k, \gamma_k)$ через $c_k$. В результате работы алгоритма получается последовательность векторов

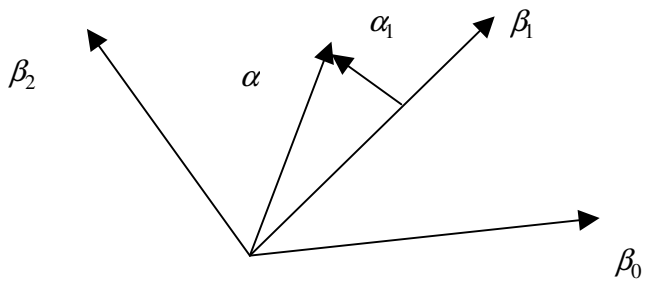


Рис. 1

$$\delta_k = \sum_{i=0}^{k} c_k \gamma_k \qquad (1)$$

Геометрическая интерпретация указанного алгоритма приведена на Рис. 1. Из определения следует, что последовательность $|\alpha - \delta_k|$ монотонно убывает.

Поскольку система векторов $B$ содержит базу пространства, эта последовательность стремится к нулю. Другими словами, мы получаем аппроксимацию (1) исходного вектора с помощью системы векторов из $B$. Следует отметить, что в отличие от разложения вектора по базису, последовательность ненулевых векторов (1) будет, как правило, бесконечной. Последовательность $c_k$ не обязана быть монотонно убывающей, но сходится к нулю. С точки зрения общей теории аппроксимации, приведенная выше процедура менее удобна, чем обычное разложение вектора по базису, хотя она обладает рядом схожих свойств. Например, если векторы из $B$ при отыскании наибольшей проекции всегда рассматриваются в одном и том же порядке, а при наличии двух векторов из $B$, проекции на которые имеют одинаковые длины, выбирается первый из векторов, то последовательность (1) строится однозначно. Если вектор $\alpha$ заменить вектором $d\alpha$, где $d$ - произвольное отличное от нуля число, то вся последовательность (1) также умножится на это число. Покажем, что в случае, когда вектор является изображением, а векторы $B$ подобраны специальным образом, эта аппроксимация обладает рядом полезных дополнительных свойств.

Как обычно, ограничимся рассмотрением только серых изображений. В результате данного соглашения каждое изображение представлено $m \times n$ матрицей с вещественными элементами. Каждая такая матрица $A$ является вектором в линейном пространстве $L$. Предположим, что множество $B$ состоит из изображений колоколообразной формы разных размеров и с различным положением точки максимума. Применяя указанный выше алгоритм в данной ситуации, мы получим представление исходного изображения в виде объединения областей колоколообразной формы. Ограничившись в (1) определенным заранее числом членов, получим водяной знак в виде параметров, характеризующих размер области и положение точки максимума.

Скалярное произведение двух матриц $A_1, A_2$ из рассматриваемого пространства определяется формулой

$$(A_1, A_2) = tr(A_1 A_2^T) \qquad (2)$$

Реализация алгоритма выглядит следующим образом. Полагаем в (2) $A_1 = A$, а в качестве $A_2$ перебираются все матрицы из множества $B$. После чего отмечается та из матриц $A_2 \in B$, для которой (2) принимает максимальное по модулю значение. После этого совершаем дальнейшие шаги, как указано в алгоритме. Цель последующих рассмотрений заключается в уменьшении числа операций для вычисления значений в (2). Если в качестве $A_2$ берутся изображения одной формы, отличающиеся лишь положением максимума, то такие матрицы можно рассматривать как результат сдвига одной матрицы. В этом случае вычисления можно ускорить стандартным образом путем перехода к двумерному преобразованию Фурье. Покажем, что специальный выбор векторов из $B$ позволяет ограничиться одномерным преобразованием Фурье и существенно сохранить объем необходимых вычислений.

Пусть $rank(A_2) = 1$. В этом случае $A_2 = \alpha^T \beta$, где $\alpha$ есть $m$-вектор, а $\beta$ $n$-вектор, которые рассматриваются, как строки. В силу известного свойства следа матрицы можем написать

$$tr(AA_2^T) = tr(A(\alpha^T \beta)^T) = tr(A\beta^T \alpha) =$$

$$tr(\alpha A \beta^T) \qquad (3)$$

Построим семейство векторов вида $Vec(N, w, center)$. Для этого, прежде всего, определим векторы семейства $Sn(w)$. Вектор $\gamma \in Sn(w)$ задан формулой

$$\gamma(i) = \sin\left(\frac{\pi(i+1)}{2w}\right), i = 0,...,2w-2 \qquad (4)$$

Вектор $\alpha \in Vec(N, w, center)$ имеет длину $N$. Он получен из нулевого вектора путем помещения в него целиком вектора $Sn(w)$ таким образом, что координата с номером $w+1$ вектора $\gamma$ становится координатой с номером $center$ вектора $\alpha$. После этого вектор нормируется, чтобы его длина стала равной 1. Пример вектора семейства $Vec(20,5,6)$ (до нормировки) показан на Рис. 2. Множество векторов $B$ состоит из матриц вида $\alpha^T \beta$, где

$\alpha \in Vec(m,w,c)$, $\beta \in Vec(n,w',c')$, где параметры $w,c,w',c'$ пробегают все воз-

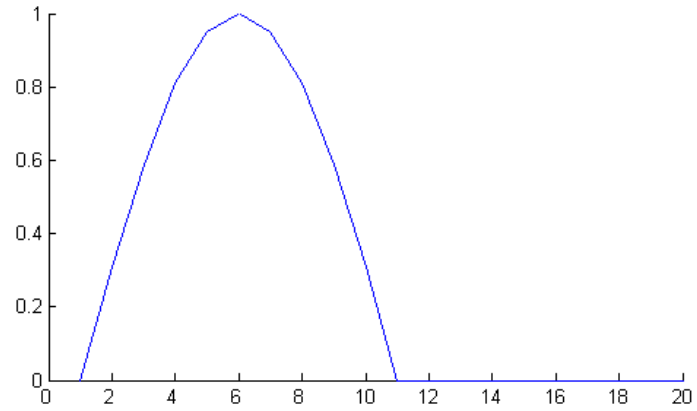


Рис. 2

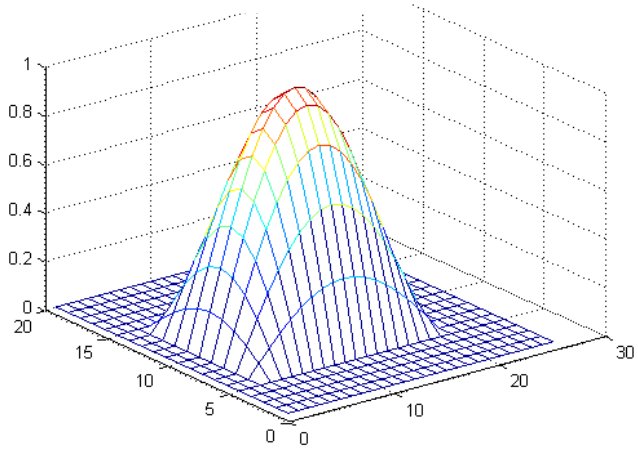


Рис. 3

можные значения, для которых существуют соответствующие семейства $Vec$

Пример матрицы из $B$ (до нормировки) приведен на Рис. 3. Таким образом, с помощью предложенного алгоритма получается аппроксимация изображения суммой матриц типа изображенных на этом рисунке. Преимущество предлагаемой аппроксимации заключается в следующем. Если производится сжатие изображения путем перехода к какому-либо графическому формату, то наиболее значимые компоненты аппроксимации остаются прежними. При кадрировании изображения, когда отрезается малая часть изображения с какого-либо края, слагаемые в (1) лишь меняют положение центра. При изменении

размера изображения происходить соответствующее шкалирование параметров слагаемых в разложении (1).

*Алгоритм быстрого вычисления коэффициентов разложения*

Как было указано выше, основная проблема реализации предлагаемого алгоритма заключается в отыскании процедуры для быстрого вычисления выражений вида (3). Для заданных $m,n$ необходимо осуществить полный перебор всех возможных наборов параметров $w,c,w',c'$. Сначала дадим неформальное описание предлагаемого алгоритма. Первое очевидное замечание состоит в том, что, подсчитав векторы $\alpha A$, где $\alpha \in Vec(m,w_0,c_0)$, мы используем эти же значения для всех векторов $\beta \in Vec(n,w',c')$. На этом шаге можно произвести распараллеливание вычислений, поделив векторы $\alpha$ между разными процессами. Однако выбранная структура векторов $\alpha,\beta$ допускает более эффективную вычислительную процедуру. Два вектора $\alpha_0 \in Vec(m,w_0,c_0), \alpha_1 \in Vec(m,w_0,c_1)$ получаются один из другого с помощью сдвига. Пусть $A[*,j]$ -столбец матрицы $A$ с номером $j$, а $A[k,*]$ - строка с номером $k$. Вычисление произведений

$$\alpha_i A[*,j], \quad \alpha_i \in Vec(m,w_0,c_i) \qquad (5)$$

можно рассматривать как подсчет ковариации между вектором $A[*,j]$ и движущимся вдоль него вектором $Sn(w_0)$. Обозначим через $F(\alpha)$ преобразование Фурье от вектора $\alpha$, подсчитанное по $m$ точкам. Известен способ быстрого подсчета произведений (5) на основе преобразования Фурье, обоснование которого можно найти, например, в [2]. Перемножаем компоненты с одинаковыми индексами векторов $F(\alpha_0)$ и вектора, сопряженного к вектору $F(A[*,j]$, после чего находим обратное преобразование Фурье. Применяя эту процедуру для всех $j$, находим векторы $\lambda_i = \alpha_i A$. Для завершения подсчета по формулам (3) теперь надо найти все произведения $\lambda_i \beta$ для всех векторов

$\beta \in Vec(n, w', c')$. Зафиксируем одно из значений параметра $w'$. Снова, все векторы из множества $\beta_j \in Vec(n, w'_0, c'_j)$ можно рассматривать как результат сдвига одно вектора из этого семейства. Для подсчета всех произведений $\lambda_i \beta_j$ используем тот же прием, что и выше.

Прежде, чем переходить к формальному описанию алгоритма, введем дополнительные обозначения.

$FA -$ $m \times n$ матрица, столбцы которой являются преобразованием Фурье от столбцов матрицы $A$.

$FVec(n, w, c)$ - множество, состоящее из преобразований Фурье от векторов множества $Vec(n, w, c)$. Заметим, что при заданном $n$ достаточно хранить лишь один вектор семейства для каждого значения параметра $w = w_0$. Остальные векторы получаются из данного умножением компонентов на множители вида $\exp(-2\pi ik/n)$.

$Num(n, w)$ - количество векторов вида $Vec(n, w, c)$ при всех возможных значениях параметра $c$. Напомним, что вектор вида $Sn(w)$ должен целиком находиться во всех векторах множества $Vec(n, w, c)$

Перед началом работы алгоритма выбирают значения параметров $w_0, \ldots, w_R$ для векторов из $Vec(m, w, c)$ и $w'_0, \ldots, w'_C$ для векторов из $Vec(n, w', c')$. При желании можно использовать все допустимые значения параметров $w$, однако, как правило, в этом нет необходимости, поскольку желаемая точность обеспечивается весьма ограниченным набором этих параметров. В то же время, если какое-то значение $w$ выбрано, то параметр $c$ принимает все возможные значения.

*Пример вычисления «водяных знаков» изображения*

В данном пункте приведем пример использования изложенной техники для построения «водяных знаков» изображения. Исходное изображение представлено на Рис. 4. Это серое изображение размера $128 \times 128$. Поскольку

нас интересует лишь построение водяного знака, а не полная аппроксимация изображения, мы использовали лишь одно значение параметра $w=16$, как для векторов $\alpha$, так и для векторов $\beta$. Найденные 20 первых членов разложения составляют изображение представленное на Рис. 5. На этом рисунке видны 20 пятен, соответствующие наиболее ярким местам оригинального изображения. Наша цель выяснить, насколько построенные знаки устойчивы к различным преобразованиям. Сравнение двух знаков должно проводиться в аналитической форме на основе значений параметров indI, indK, indJ, indL. В данной работе мы не будем останавливаться на этом моменте. Вместо этого ограничимся графическим представлением водяных знаков. Первое преобразование заключается в «кадрировании» исходного изображения. Из оригинала удалили две первых строки и первый столбец, заменив их нулевыми и сохранив, тем самым, размер изображения (Рис. 6). Соответствующий водяной знак представлен на Рис. 7. Хотя знаки на Рис. 5 и Рис. 7 не совпадают, визуальное сходство между ними очевидно. Следующее преобразование заключалось в изменении размера. Новое изображение получено уменьшением размера вдвое и имеет размер $64\times64$. Водяной знак построен по той же методике, но теперь параметр $w=8$. Его графическое изображение представлено на Рис. 8. И в этом случае очевидна устойчивость по отношению к данному преобразованию.

## СПИСОК ЛИТЕРАТУРЫ

1. Bas P., Chassery J.-M., Macq B. Image watermarking:an evolution to content based approaches. Pattern Recognition vol. 35 pp. 545–561, 2002

2. Отнес Р., Эноксон Л. Прикладной анализ временных рядов. М.: Мир, 1982

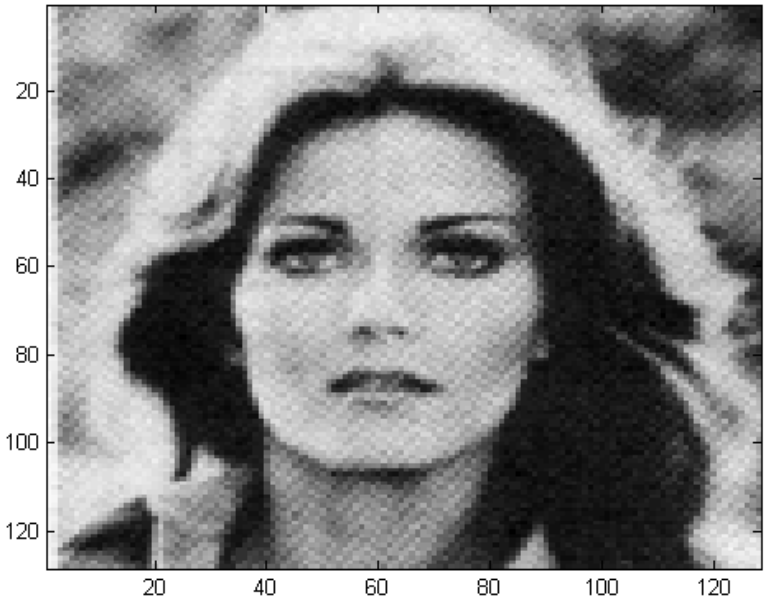
20
40
60
80
100
120
20
40
60
80
100
120

Рис. 4

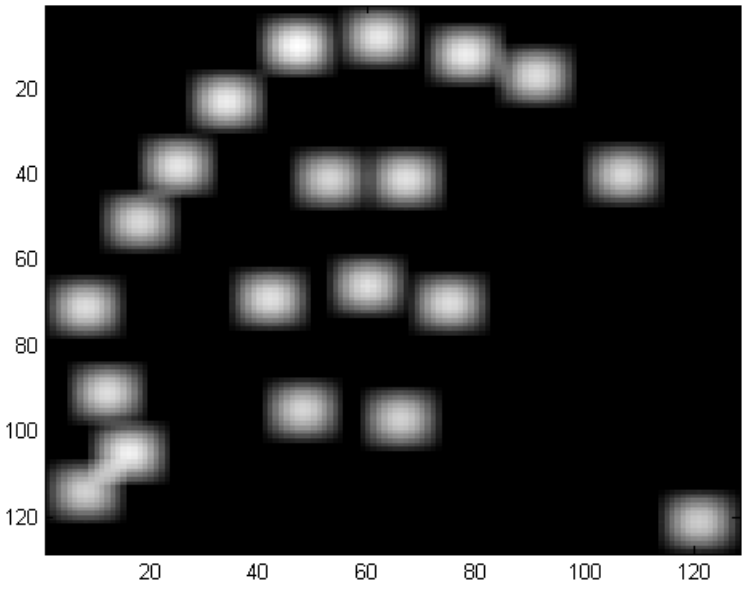
20
40
60
80
100
120
20
40
60
80
100
120

Рис. 5

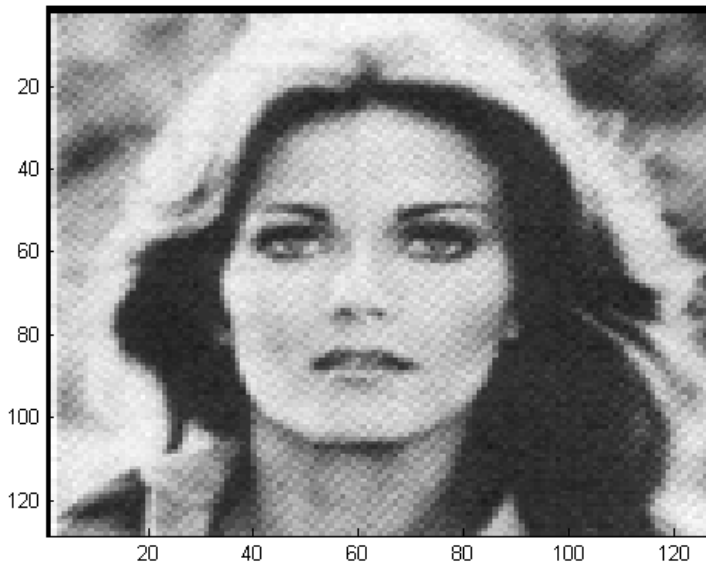
20
40
60
80
100
120
20
40
60
80
100
120

Рис. 6

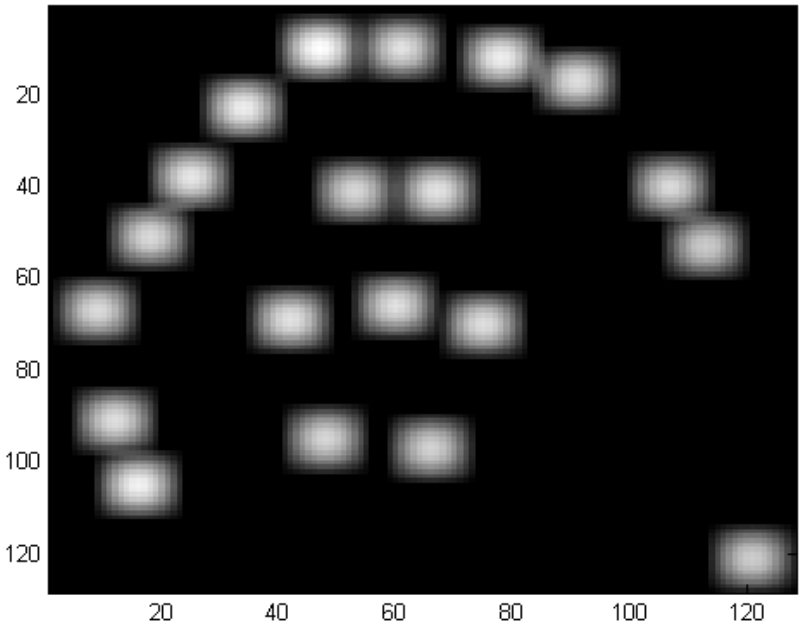
20
40
60
80
100
120
20
40
60
80
100
120

Рис. 7

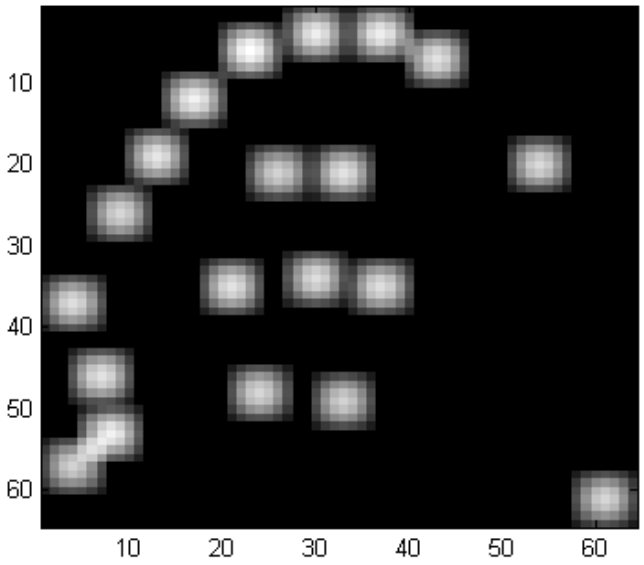


Рис. 8