

ИНФОРМАТИКА

УДК 621.317.846

**ИСПОЛЬЗОВАНИЕ МЕР СЛОЖНОСТИ ГРАФИЧЕСКИХ ОБРАЗОВ ПРИ
ВНЕДРЕНИИ ВОДЯНЫХ ЗНАКОВ В ЛОГОТИПЫ ПРОГРАММНЫХ СРЕДСТВ**

А.А. РОГОЖНЫЙ, В.Н. ЯРМОЛИК

*Белорусский государственный университет информатики и радиоэлектроники
П. Бровка, 6, Минск, 220013, Беларусь,**Поступила в редакцию 25 октября 2005*

Рассмотрено использование оценки сложности черно-белых графических образов для внедрения водяных знаков в логотипы программных продуктов. Проанализированы основные свойства критериев сложности и экспериментально определены и обоснованы приемлемые параметры внедрения, используемые по умолчанию.

Ключевые слова: логотип программного средства, водяной знак, сложность графического образа.

Введение

В применении к защите интеллектуальной собственности различают следующие направления стеганографии [1]: внедрение водяных знаков (watermarking), цифровых отпечатков (fingerprinting), встраивание заголовков (captioning), идентификационных номеров и скрытая передача информации.

Одним из наиболее перспективных подходов для защиты программного обеспечения видится использование водяных знаков [2]. Современный программный продукт представляет собой сложную систему различного рода текстовой, графической, реже аудио- и видеоинформации. Поэтому для эффективной защиты программного обеспечения имеет смысл применять подходы, которые используют для внедрения секретной информации различные типы данных. Водяные знаки могут внедряться в аудио- или видеосопровождение программного продукта. Главным преимуществом такого подхода является большая емкость внедрения и высокая себестоимость возможных атак. Серьезным недостатком можно считать относительно дорогую процедуру внедрения информации.

Наиболее перспективными кандидатами на роль контейнеров для внедрения водяных знаков можно считать графические логотипы компаний-разработчиков программных продуктов. Преимуществом использования такого рода данных является большое число существующих стеганографических методов, использующих графические изображения. Кроме того, логотипы широко используются и присутствуют практически в каждом современном программном продукте.

В данной работе рассмотрена и проанализирована схема внедрения данных в логотипы программных продуктов с использованием характеристик сложности графических образов.

Внедрение водяных знаков в графические образы

Все множество разработанных методов внедрения водяных знаков в графические изображения можно разделить на 2 группы [3].

К первой можно отнести методы, использующие пространство различных преобразований. При использовании таких подходов изображение предварительно описывается коэффициентами некоторого преобразования: например, преобразования Фурье, косинус-преобразования, вейвлет-преобразования или преобразования Адамара. Внедрение осуществляется изменением полученных коэффициентов. Для проверки факта присутствия водяного знака используется преобразование, обратное преобразованию, использованному при внедрении.

Ко второй группе относится множество методов, которые непосредственно изменяют значения пикселей изображения.

Наибольшее распространение получили методы, основанные на модификации младших битов [4]. Вследствие информационной избыточности графического изображения такие изменения не заметны для человеческого зрения. Существующие подходы вносят информацию изменением одного или двух младших битов. Изменение 3 или 4 битов позволяет значительно увеличить емкость внедрения, но при этом искажение изображения визуально заметно. Можно отметить следующие характерные особенности методов, основанных на модификации младших битов:

- относительно высокая емкость внедрения;
- простота технической реализации;
- низкая устойчивость к атакам злоумышленника.

Метод Patchwork, предложенный в [1], использует статистические свойства изображения. Случайным образом выбираются n пар точек (a_i, b_i) ; яркость в точке a_i увеличивается, в то время как в точке b_i уменьшается. Ожидаемое значение суммы разностей n пар точек равно $2n$ и представляет собой определенное статистическое свойство конкретного изображения. К достоинствам такого подхода можно отнести достаточно высокую устойчивость к атакам злоумышленника. Отличительной чертой метода является низкая емкость внедрения. Фактически внедряется всего 1 бит, свидетельствующий о том, обладает изображение особой статистикой или нет.

Стоит также отметить методы внедрения водяных знаков в длительную постоянную структуру изображения. Часть такой структуры копируется в другую похожую часть изображения. Это приводит к тому, что в изображении присутствует пара идентичных областей. Главным недостатком метода является необходимость участия в процессе оператора для выбора области для копирования и для оценки, насколько такое копирование заметно для человеческого восприятия. Кроме того, такой подход может использоваться лишь для определенного класса изображений, в которых присутствуют продолжительные повторяющиеся области.

Метод ВРС-стеганографии [5] (Bit-Plane Complexity Segmentation Steganography) и его усовершенствованный вариант — ABCDE (A Block Complexity based Data Embedding) [6] используют для принятия решения о внедрении данных оценку сложности черно-белых графических образов.

Данные подходы основаны на особенностях человеческого зрения, которое очень восприимчиво к внесению изменений в продолжительные постоянные регионы изображения, но зачастую не может различить изменения в шумовых областях.

Суть метода заключается в разбиении изображения на множество блоков и определении тех блоков из этого множества, которые могут быть незаметно для человеческого зрения заменены другими, содержащими скрываемую информацию.

Несмотря на то что данный подход разработан для определения сложности черно-белых блоков, он может использоваться и для внедрения данных в цветные изображения.

Цветное изображение вначале разбивается на 3 изображения, содержащие различные градации серого цвета и использующие для хранения каждого пикселя n бит. Каждое изображение соответствует одному из цветовых каналов: красному, зеленому, синему. Затем изобраа-

жения с разными уровнями градации серого цвета разбиваются на n двоичных черно-белых изображений, представляющих битовые слои.

Таким образом, цветное изображение может быть представлено в следующем виде:

$$P = (P_{R_1}, \dots, P_{R_n}, P_{G_1}, \dots, P_{G_n}, P_{B_1}, \dots, P_{B_n}).$$

Меры сложности графических образов

Вводятся 2 меры сложности графических образов:

неравномерность длин серий;

зашумленность границ.

Неравномерность длин серий предназначена для оценки неоднородности черно-белых пикселей внутри блока. Если распределение черных и белых пикселей в блоке имеет регулярную структуру, то данный блок не может считаться сложным.

Определение неравномерности длин серий основано на гистограмме длин серий из белых и черных пикселей по строкам и столбцам.

Определим h_s как величину неравномерности двухцветной последовательности пикселей:

$$h_s = \sum_{i=1}^n h[i] \log_2 p_i, \quad (1)$$

$$p_i = \frac{h[i]}{\sum_{j=1}^n h[j]}, \quad (2)$$

где $h[i]$ — частота появления серий из i черных или белых пикселей; n — максимально возможная длина серии.

Определим неравномерность длин серий для блока.

Пусть размер блока $n \times n$, r_i и c_j соответственно i -я строка и j -й столбец. Неравномерность длин серий β блока, определяется как

$$\beta = \min\{\overline{H_s(r)}, \overline{H_s(c)}\}, \quad (3)$$

где

$$\overline{H_s(r)} = \{h_s(r_0), \dots, h_s(r_{n-1})\}, \quad (4)$$

$$\overline{H_s(c)} = \{h_s(c_0), \dots, h_s(c_{n-1})\}. \quad (5)$$

Неравномерность длин серий принимает значение от 0 до 1. Если значение β для блока достаточно велико, то блок может считаться сложным.

Зашумленность границ γ позволяет проверить, насколько хорошо черные-белые границы распределены в блоке по вертикальному и горизонтальному направлениям. Вычисление меры сложности по зашумленности границ основано на различии между последовательностями смежных двухцветных пикселей в блоке.

Пусть размер блока $n \times n$, r_i и c_j соответственно i -я строка и j -й столбец. Зашумленность границ γ блока, определяется как

$$\gamma = \frac{1}{n} \min\{E_f(P_x(r)), E_f(P_x(c))\}, \quad (6)$$

где

$$P_x(r) = \{p(r_0 \oplus r_1), \dots, p(r_{n-2} \oplus r_{n-1})\}, \quad (7)$$

$$P_x(c) = \{p(c_0 \oplus c_1), \dots, p(c_{n-2} \oplus c_{n-1})\}, \quad (8)$$

p определяет количество 1 в двоичной последовательности x ,

$$E_f(X) = (1, 0 - V(X) / \max\{V(X)\}) \cdot \bar{X}, \quad (9)$$

$$X = \{x_0, x_1, \dots, x_{m-1}\},$$

где $V(X)$ — дисперсия; \bar{X} — среднее значение X .

Определение зашумленности границ γ схоже с определением неравномерности длин серий β . Количество черно-белых границ вычисляется для каждой пары смежных строк и столбцов. Оно представляется множеством $P_x(r)$ для строк и множеством $P_x(c)$ для столбцов в вышеприведенном определении. Затем на их основе рассчитывается средневзвешенное значение $E_f(X)$.

Анализ мер сложности для различных битовых плоскостей

Битовые плоскости изображения отображают монотонно возрастающую сложность от наиболее значимой плоскости к наименее значимой. Наименее значимые битовые плоскости выглядят как белый шум.

На рис. 1 и рис. 2 представлены графики распределения мер сложности β и γ для красной цветовой компоненты блоков различных бинарных плоскостей изображения. Очевидно, что для зеленой и синей цветовой компонент распределение будет иметь похожий вид.

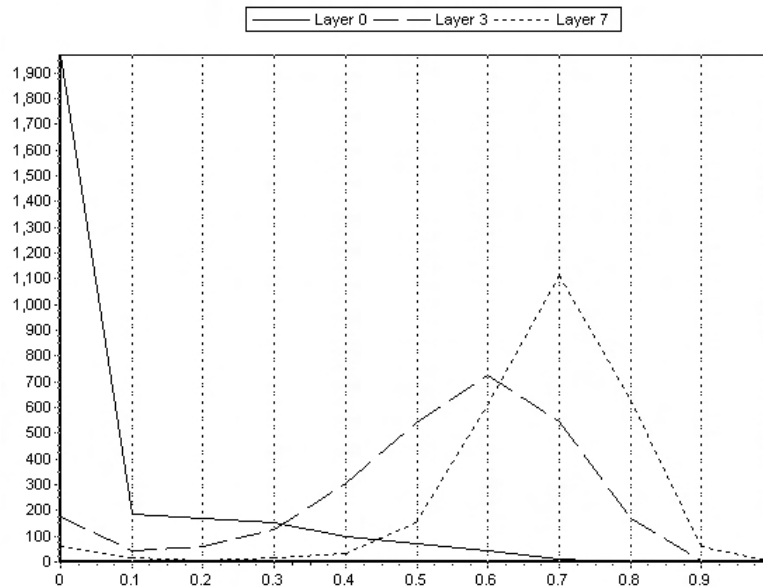


Рис. 1. Распределение β для блоков битовых плоскостей 0,3,7

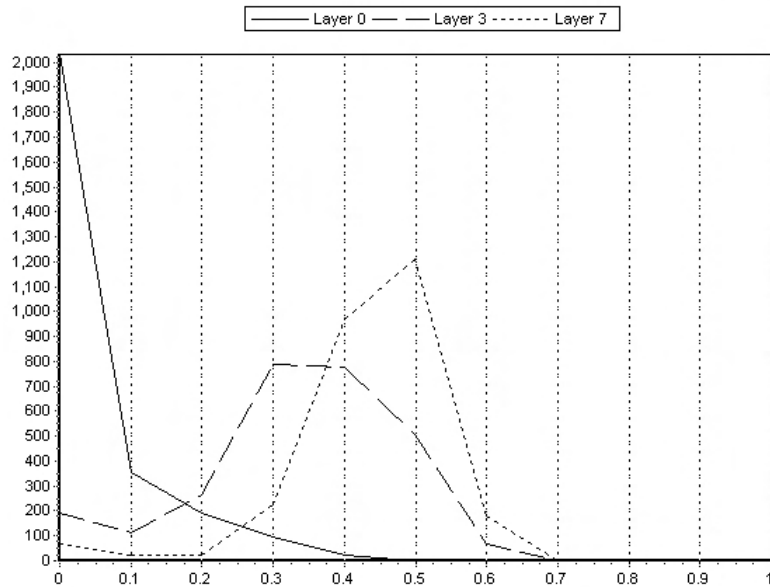


Рис. 2. Распределение γ для блоков битовых плоскостей 0,3,7

Важно отметить, что распределение величин мер сложности β и γ соответствует видимой сложности битовых плоскостей. Заметим, что битовая плоскость 0 является наиболее значимой, информативной областью, а битовая плоскость 7 — наименее значимой и наиболее зашумленной областью. Из рис. 1 и 2 видно, что наиболее значимая плоскость имеет много информативных блоков, для которых значения β и γ малы, в то время как наименее значимая обладает большим количеством сложных блоков.

Таким образом, показатели неравномерности длин серий и зашумленности границ являются достаточными для определения сложности графического образа.

Пороговые значения для определения сложности графического образа

Графический образ B может считаться сложным, если для него справедливы следующие неравенства: $\beta(B) \geq \beta_t$; $\gamma(B) \geq \gamma_t$, где β_t и γ_t — пороговые значения.

Возникает вопрос, какие пороговые значения использовать для корректного определения сложности блоков?

Случайным образом были сгенерированы 10000 графических блоков размером 8×8 для определения средних значений β и γ . Экспериментально полученное распределение представлено на рис. 3.

Установлено, что $\bar{\beta} = 0,75$, $\sigma_\beta = 0,07$; $\bar{\gamma} = 0,50$, $\sigma_\gamma = 0,05$.

Следует задавать пороговые значения β_t^i , γ_t^i отдельно для каждой битовой плоскости i . При этом блок считается шумовым в случае выполнения следующих условий:

$$\beta(B) \geq \beta_t^i,$$

$$\gamma(B) \geq \gamma_t^i.$$

Существуют пороговые значения, используемые по умолчанию [5]:

$$\beta_t^i = \bar{\beta} - i\sigma_\beta \text{ при } 0 \leq i \leq 4,$$

$$\beta_t^i = 0,0 \text{ при } 5 \leq i \leq 7,$$

$$\gamma_t^i = \bar{\gamma} - i * \sigma_\gamma \text{ при } 0 \leq i \leq 4,$$

$$\gamma_t^i = 0,0 \text{ при } 5 \leq i \leq 7.$$

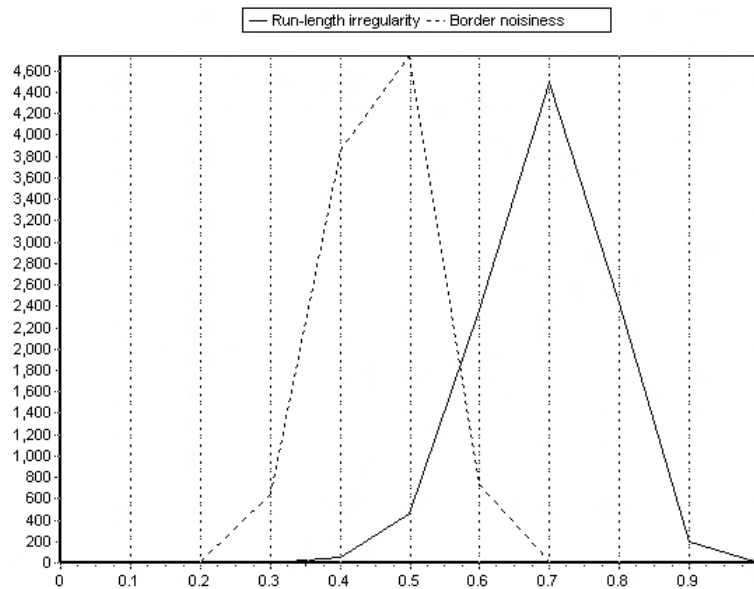


Рис. 3. Распределение β и γ для 10000 блоков, сгенерированных случайным образом

Приведенное выше выражение разрешает использовать для внедрения информации все блоки битовых плоскостей 5, 6 и 7. Такое утверждение основано на факте, что человеческое зрение не в состоянии заметить изменения в этих областях. Однако, как показали эксперименты, такие пороговые значения могут использоваться только для внедрения данных в синюю и красную цветовые компоненты.



Рис. 4. Изображение с данными, внедренными в зеленую компоненту при использовании пороговых значений по умолчанию

Это обусловлено особенностями визуальной системы человека, который лучше всего замечает изменения зеленого и хуже замечает изменения синего цвета. На рис. 4 представлена часть изображения, в котором данные внедрены в зеленую цветовую компоненту с использованием пороговых значений по умолчанию.

В этом случае изменения, внесенные в изображение, визуально заметны, что недопустимо.

Поэтому пороговые значения для внедрения следует определять для каждой цветовой компоненты в отдельности. Как показали эксперименты, для синей и красной составляющих цвета достаточно использовать предлагаемые значения по умолчанию.

Для зеленой компоненты используются следующие значения:

$$\beta_t^i = \bar{\beta} - i\sigma_\beta \text{ при } 3 \leq i \leq 7,$$

$$\gamma_t^i = \bar{\gamma} - i\sigma_\gamma \text{ при } 3 \leq i \leq 7.$$

Таким образом, старшие 3 бита для внедрения не используются.

Пороговые значения по умолчанию для внедрения не являются строго фиксированными значениями, которые всегда стоит использовать. Это значения, позволяющие внедрить достаточный объем информации, при этом искажения изображения нельзя определить визуально. Пороговые значения могут быть изменены в каждом конкретном случае, что позволяет варьировать между качеством изображения и емкостью внедрения.

Результаты эксперимента

В качестве эксперимента проводилось внедрение в изображение случайно сгенерированных данных. Для определения сложности блока использовались пороговые значения по умолчанию. Значения для красной и синей цветовой составляющих приведены в табл. 1, для зеленой — в табл. 2.

Таблица 1. Пороговые значения β и γ для красного и синего цветовых каналов

	0	1	2	3	4	5-7
β	0,75	0,68	0,61	0,54	0,47	0
γ	0,5	0,45	0,40	0,35	0,30	0

Таблица 2. Пороговые значения β и γ для зеленой цветовой компоненты

	0	1	2	3	4	5-7
β	1	1	1	0,54	0,47	0,4
γ	1	1	1	0,35	0,30	0,25

Размер изображения-контейнера 315×115 пикселей. Размер графического файла формата bmp 99 594 байт.

Емкость внедрения определяется процентным отношением размера внедренных данных к размеру контейнера:

$$\frac{E}{Size} 100\% . \quad (10)$$

Очевидно, что емкость внедрения прямо пропорциональна количеству сложных блоков в изображении, количество которых в свою очередь зависит от используемых пороговых значений и характера изображения.

На рис. 5 представлено оригинальное изображение, на рис. 6 — изображение с внедренными данными.



Рис. 5. Оригинальное изображение



Рис. 6. Изображение с внедренными данными

Визуально данные изображения идентичны. При этом емкость внедрения составила 34 %.

Эксперимент по внедрению данных в фотографическое изображение продемонстрировал еще большую емкость — 50 %. На практике внедряемые данные предварительно архивируют, при этом обеспечивается достижение сразу 2 целей:

значительное увеличение объема внедряемых данных;

архив может быть защищен с помощью пароля, что повышает секретность внедрения

Таким образом, эксперимент показал относительно большую емкость внедрения. При этом, несмотря на то что большое число блоков изображения заменено, искажение изображения не наблюдается.

Заключение

Описанная схема внедрения позволяет добиваться компромисса между емкостью внедрения и качеством изображения после внедрения данных путем задания больших или меньших пороговых значений.

При этом установлено, что при задании пороговых значений для определения сложности блока стоит учитывать не только номер битовой плоскости, но и цветовую составляющую.

Для метода характерны недостатки подходов, основанных на модификации младших битов. Данный подход обеспечивает определенную надежность с предположением о том, что изображение не будет подвергнуто тщательному анализу путем разложения на битовые плоскости и цветовые компоненты.

Следует отметить, что большое влияние на функционирование схемы внедрения оказывает вид изображения, используемого в качестве контейнера. Одним из направлений дальнейшего изучения является более точный автоматизированный анализ изображения для оценки приемлемости использования его в качестве контейнера и для определения наилучших пороговых значений.

USAGE OF GRAPHICAL PATTERNS COMPLEXITY MEASURES FOR WATERMARKS EMBEDDING IN LOGO IMAGES

A.A. ROGOZHNY, V.N. YARMOLIK

Abstract

Usage of black-and white patterns complexity evaluation for watermarks embedding in logo images has been presented. Main properties of complexity criteria have been analyzed and acceptable default embedding parameters have been experimentally defined and approved.

Литература

1. *Bender W., Gruhl D., Morimoto N., Lu A.* // IBM Systems Journal. 1996. Vol. 35. P. 316–320.
2. *Collberg C.S., Thomborson C.* // Watermarking, Tamper-proofing, and Obfuscation — Tools for Software protection. Computer Science Technical Report #170. University of Auckland, Auckland, New Zealand. February 10. 2000. P. 740.
3. *Fridrich J.* // Applications of data hiding in digital images. Tutorial for the ISPACS'98 Conference in Melbourne. Australia, 1998.
4. *Swanson M.D., Tewflik A.H.* // Proceedings of the IEEE. Vol. 86. No. 6. June 1998. P. 1072.
5. *Kawaguchi E., Eason R.O.* // Proceedings of SPIE: Multimedia Systems and Applications. 1998. Vol. 3528. P. 464–472.
6. *Hirohisa H.* // Proc. Pacific Rim Workshop on Digital Steganography 2002. P. 30–47.