

Feature-Based Watermark Localization in Digital Capture Systems

Vojtěch Holub and Tomáš Filler

Digimarc Corporation, 9405 SW Gemini Drive, Beaverton, OR 97008, USA

ABSTRACT

The “Internet of Things” is an appealing concept aiming to assign digital identity to both physical and digital everyday objects. One way of achieving this goal is to embed the identity in the object itself by using digital watermarking. In the case of printed physical objects, such as consumer packages, this identity can be later read from a digital image of the watermarked object taken by a camera. In many cases, the object might occupy only a small portion of the the image and an attempt to read the watermark payload from the whole image can lead to unnecessary processing. This paper proposes a statistical learning-based algorithm for localizing watermarked physical objects taken by a digital camera. The algorithm is specifically designed and tested on watermarked consumer packages read by an off-the-shelf barcode imaging scanner. By employing simple noise-sensitive features borrowed from blind image steganalysis and a linear classifier, we are able to estimate probabilities of watermark presence in every part of the image significantly faster than running a watermark detector. These probabilities are used to pinpoint areas that are recommended for further processing. We compare our adaptive approach with a system designed to read watermarks from a set of fixed locations and achieve significant savings in processing time while improving overall detector robustness.

1. INTRODUCTION

Digital watermarking is an art of embedding auxiliary payload into cover media such that the change remains invisible to humans but enables machines to reliably extract the payload even after common signal-processing operations. This allows machines to uniquely identify objects they “see” or “hear”. Such a form of robust data hiding has been used extensively over the last two decades¹ for applications including media content protection, track and trace, etc.

We focus on applications where watermarks are printed on consumer packages and carry information similar to that found in barcodes*. Because most of the package surface can be watermarked, users no longer need to search for barcodes at checkout, thus leading to increased checkout speed. In this application, the watermark has to be located and decoded as fast as possible while the package is swiped in front of a digital camera (checkout scanner). Assuming a fixed time budget is allocated for each frame produced by the imaging scanner, only a limited area can be examined with an algorithm designed to read the watermarks (watermark reader). Due to the time constraint, watermark reading framework should be designed as a chain of modules, each providing low missed-detection rate and decreasing false alarms so that unmarked areas are rejected as early as possible while not missing any marked area.

The goal of this paper is to design the very first module in the watermark reading framework that quickly analyzes the image and pinpoints areas that most likely contain watermark. More precise watermark detection and decoding algorithms can then be used to read the payload or reject the pinpointed area. We borrow the basic tools from the field of image steganalysis⁵ and, for complexity reasons, deliberately constrain ourselves to image features constructed directly in the spatial domain. Although the tools are available in the literature and have been used for years,⁴ they were not studied in the context of robust detection of noisy signals as in this paper.

E-mail: vojtech_holub@yahoo.com, tomas.filler@digimarc.com; <http://www.digimarc.com>

*Digimarc Discover, see video at <http://www.digimarc.com/discover>

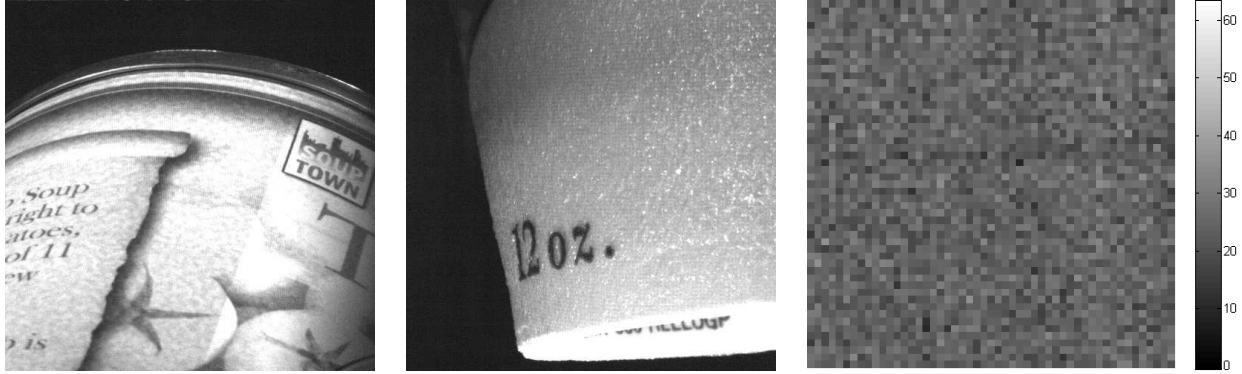


Figure 1. Tomato soup can with unreadable watermark (left), unmarked air insulated coffee cup causing false alarms (middle) and detail of full resolution background noise (right).

For the purpose of this paper, we use a specific instance of the watermark reader[†] which we further treat as a black box. This reader uses 128×128 -pixel image block to detect the presence of a watermark with high accuracy. Our goal is to develop an algorithm that will locate the most appropriate 128×128 block within a captured image for passing to the watermark reader. We do this by using a binary classifier (logistic regressor) trained to distinguish between image blocks with marked and unmarked content.

To apply the trained classifier for watermark localization, an image is decomposed into much smaller square sub-blocks. Carefully chosen features are extracted from each sub-block and the trained logistic regressor is used to estimate the probability of watermark presence. The result of this process is a probability map for a given image. An averaging kernel is then applied to smooth out the map and estimate probability of watermark presence in each 128×128 block. Finally the algorithm then selects a number of locations with watermark probability exceeding a given threshold. These locations are taken as the centers of the blocks considered for further processing by a watermark reader.

Section 2 describes the experimental setup. Information about the feature set, its parameter selection and the classifier is in Section 3. The transition from a binary classification problem to watermark localization is described in Section 4. Section 5 contains localization experiment results and justification for its parameter selection. Finally, a brief summary can be found in Section 6.

2. EQUIPMENT AND COMMON SETUP

Images for all experiments described in this paper were acquired with a commercially available imaging barcode scanner with a monochromatic camera and red LED illumination. The front-facing camera from this scanner provides 1024×1280 -pixel frames at a rate of 40 per second. Each frame captures an item from two different angles using two sets of mirrors causing the image to be split in half. Strobed LED illumination and fast exposure time effectively limit motion blur allowing the scanner to provide sharp images. Obtained images have strong noise component with the peak signal to noise ratio of 35.5 dB as shown in Figure 1. To train and test the proposed approach, we captured tens of thousands of images simulating the checkout process in a grocery store with various marked and unmarked objects. Some of the images contain pure image noise when no object is presented to the camera.

All results presented in this paper are from 341×426 -pixel images obtained by downsampling the original images by a 3×3 box filter. The downsampling factor was empirically determined to provide the best robustness for the watermark reader at hand. Moreover, the downsampling also speeds up the overall watermark extraction and suppresses the noise component.

[†]Spread-spectrum watermark reader compatible with Digimarc Discover.

3. WATERMARK CLASSIFICATION FRAMEWORK

This section describes all components of the watermark classification framework. The next section describes how to apply this for watermark localization.

3.1 Feature Set

A key component of the proposed algorithm is the noise-sensitive feature set that provides enough statistical evidence to distinguish Digimarc Discover noise-like watermark from other noisy content. Since this problem is very similar to the one in digital image steganalysis, we decided to use already developed features from that field. After experimenting with multiple feature sets used in steganalysis and their parameter settings, we decided to use one of the simplest sub-models described in the Spatial Rich Model⁵ (SRM) paper. In the SRM paper, authors introduce a set of 39 linear and non-linear high-pass filters to obtain rich representations of noise residuals. In order to fit into tight complexity constraints we select the simplest of SRM's feature sets – '1a) spam14h' as described in Figure 2 in the SRM paper.⁵ The features are extracted from already downsampled image (1/9 of the original pixels). Using the same image scale for feature extraction and watermark reading speeds up the whole process, suppresses noise and allows modeling of farther spatial-domain dependencies.

We use the symbol \mathbf{X} for an $M \times N$ grayscale image whose pixel values, $x_{i,j} \in \{0, 1, \dots, 255\}$, are represented using the matrix $\mathbf{X} = (x_{i,j})$, $i = 1, \dots, M$, $j = 1, \dots, N$. The horizontal noise residual $\mathbf{Z} = (z_{i,j})$ is computed as $z_{i,j} = x_{i,j} - x_{i,j+1}$. The traditional approach of steganalytic feature set construction now continues with quantization, rounding and truncation of $z_{i,j}$,

$$r_{i,j} = \text{trunc}_T \left(\text{round} \left(\frac{z_{i,j}}{q} \right) \right), \quad \text{trunc}_T(k) = \begin{cases} T & \text{when } k \geq T \\ k & \text{when } -T < k < T \\ -T & \text{when } k \leq -T \end{cases} \quad (1)$$

and forming a D -dimensional co-occurrence matrix⁶ $\mathbf{C} = (c_{d_1, \dots, d_D})$, $d_i \in \{-T, \dots, T\}$ from D horizontal neighboring values of $r_{i,j}$. The process is then repeated with the transpose of the image \mathbf{X} to obtain statistics from the vertically neighboring pixels. Both horizontal and vertical co-occurrence histogram bins are added together and normalized so that all bins sum up to one. The normalization is important for classification of images with different sizes. Finally, by exploiting symmetries in natural images the dimensionality can be reduced by adding up values in co-occurrence bins $c_{d_1, \dots, d_D}, c_{d_D, \dots, d_1}, c_{-d_1, \dots, -d_D}, c_{-d_D, \dots, -d_1}$. For the choice of $T = 2$ and $D = 4$, we obtain $(2T + 1)^D = 625$ -dimensional feature vector that is reduced to the final 169-dimensional feature vector by symmetrization. A more detailed explanation of this feature set is available in Ref. 5[‡].

3.2 Classification

The simplest formulation of the watermark detection problem leads to a binary classification problem – to decide whether the feature vector is extracted from an image containing a readable watermark or not. The idea is to use a machine learning approach, i.e., train a binary classifier on large amount of labeled training samples so that it can successfully generalize and classify previously unseen samples. There are many available off-the-shelf classifiers in literature. We investigated Linear Support Vector Machines (L-SVM), Gaussian kernel SVM (G-SVM), logistic regression and simple Fisher Linear Discriminant (FLD).² While the performance of G-SVM was indeed superior to the other classifiers, we decided to use logistic regression (which performed only slightly worse) due to its fast classification and native ability to output probabilities of detection instead of binary decisions.

The implementation of choice for logistic regression training is LIBLINEAR^{3§} library with option '-s 0'. This implementation minimizes the L2 criterion and uses 5-fold cross-validation to find the optimal value of the regularization parameter. Logistic regressor evaluates probability that a given feature vector $\mathbf{f} = (f_1, \dots, f_{169})$ was obtained from a watermarked object as

$$p(\mathbf{f}) = \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{f}^T}}, \quad \mathbf{w} \cdot \mathbf{f}^T = \sum_{i=1}^{169} w_i f_i, \quad (2)$$

[‡]Feature set implementation is available at http://dde.binghamton.edu/download/feature_extractors/

[§]More information and downloads are available at <http://www.csie.ntu.edu.tw/~cjlin/liblinear/>

where $\mathbf{w} = (w_1, \dots, w_{169})$ are weights obtained from training. In our implementation, elements of the feature vectors were normalized to zero mean and unit variance before being used in the logistic regressor. Feature element means and variances were determined in training phase and kept constant when testing the classifier.

3.3 Training Image Sets

Two image sets were constructed to train the logistic regressor to recognize features extracted from marked and unmarked image blocks. Each set consisted of 5000 images obtained from a full resolution camera frame by downsampling followed by cropping random 128×128 -pixel image block. All of these images were evaluated by a watermark reader and labeled as marked or unmarked. Image sets were handcrafted to include representative samples of various marked and unmarked objects as well as sensor noise. Two image sets are summarized below.

- Unmarked image set (class 0) consisted of image blocks that do not result in watermark detection. The set contained 45% of images of watermarked objects with unreadable watermark (object too far, watermark too weak or the perspective distortion too strong), 45% of images of non-watermarked objects, and 10 % of images with noisy background only.
- Marked image set (class 1) consisted of image blocks with successfully extracted watermark of various signal strength.

3.4 Optimizing Feature Parameters

The feature set described in Section 3.1 has three parameters: the quantization step q , the truncation threshold T and order of the co-occurrence D . Optimal values of these parameters were determined by minimizing probability of error P_E . Error of binary classifier based on logistic regressor P_E is defined as $P_E = \min_{P_{FA}} \frac{1}{2} (P_{FA} + P_{MD}(P_{FA}))$, where P_{FA} is the probability of false alarm and P_{MD} is the probability of missed detection.

Table 1 presents results of search for optimum parameter values. Exhaustive search over more combinations of parameters was done with the same conclusion – the optimal feature set parameters for this problem are $q = 4$, $T = 2$ and $D = 4$ resulting in 169-dimensional feature set.

4. FROM CLASSIFICATION TO LOCALIZATION

The previous section described how to estimate probability of a watermark being present in a given 128×128 image block. Here we use this algorithm as a building block to estimate optimal position of an image block recommended for more detailed processing. The size of the image block used in the training phase was chosen to be compatible with the watermark reader. Due to feature vector normalization, a trained classifier can be used with image block of a different size.

To find the most promising image block for watermark reader, we divided the downsampled image into a grid of non-overlapping 21×21 -pixel sub-blocks. Each sub-block is treated as a separate image and the framework described in Section 3 is used to estimate probability of watermark presence. Algorithm 1 is used to pool the estimated probabilities and obtain the list of most promising locations with probabilities greater than parameter $\alpha \geq 0$. Set of neighboring 6×6 sub-blocks was used to estimate optimal location of 128×128 image block. The sample result obtained by this algorithm is illustrated in Figure 2.

The size of the sub-blocks determines statistical significance of the extracted features, number of sub-blocks used to obtain the final neighborhood in the averaging process and the resolution of final coordinates. The size of 21×21 was determined heuristically as the best trade-off between these properties.

Depending on the application at hand, the number of localized coordinates is driven by two parameters – c and α . Parameter c determines the maximum number of output coordinates per image frame, thus, bounding the complexity required to process one frame. Parameter α determines the sensitivity – the higher the α is, the more confident the algorithm is of the watermark’s presence. For example, when $c = 2$ and $\alpha = 0.5$ up to two coordinates are found and recommended for watermark reading when a watermarked object is swiped in front of the camera. No coordinate is recommended for processing when no object is present.

| quantization step q | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| probability of detection error P_E | 0.134 | 0.128 | 0.119 | 0.114 | 0.116 | 0.118 | 0.123 | 0.125 |

| truncation threshold T | 1 (dim 25) | 2 (dim 169) | 3 (dim 625) |
|--------------------------------------|------------|-------------|-------------|
| probability of detection error P_E | 0.135 | 0.114 | 0.116 |

| co-occurrence order D | 3 (dim 39) | 4 (dim 169) | 5 (dim 819) |
|--------------------------------------|------------|-------------|-------------|
| probability of detection error P_E | 0.139 | 0.114 | 0.116 |

Table 1. Search for optimal values of feature set parameters q (top), T (middle) and D (bottom) as evaluated by the probability of detection error P_E . Lower values of P_E result in better feature set. Two parameters are always fixed to default values $q = 4$, $T = 2$ and $D = 4$. Values of truncation threshold T and co-occurrence order D determine the dimensionality of the feature set before symmetrization.

Algorithm 1 Pseudo code of sub-block pooling algorithm. The algorithm returns a list of image coordinates.

```

while the number of coordinates in the list is smaller than  $c$  do
  compute neighborhood probabilities  $P(\mathcal{N}_i)$  by averaging sub-block probabilities with  $6 \times 6$  sliding window;
  find the neighborhood  $\mathcal{N}_{MAX}$  with largest probability;
  if  $P(\mathcal{N}_{MAX}) > \alpha$  then
    add the center of  $\mathcal{N}_{MAX}$  to the list of coordinates;
  else
    break the loop;
  end
  set the probabilities in all 36 sub-blocks composing  $\mathcal{N}_{MAX}$  to zero;
end

```

5. EXPERIMENTS

In this section, we compare robustness of watermark reading system using the above proposed approach on real-life checkouts. For this test, we captured a database of 10,000 frames simulating the checkout of over 20 different watermarked packages including cans, boxes, and plastic bags. Since the cashier needs time to present an object to the scanner, most of the frames in this database contain solely noise and dark background. This database was created independently of the training database and no image was used for training purposes.

We compare the proposed (adaptive) system with several reference implementations where blocks are statically placed on the frame. The best robustness under the current setup we could achieve is when the watermark reader attempts to extract the watermark from all possible $11 \times 15 = 165$ locations. Due to fixed time budget (25 ms for the case of 40 FPS), this may not always be an option in a practical system. For this reason, we also compared our approach with seven experimentally-designed locations that are fixed for each frame. The number of frames with at least one successfully extracted watermark payload is used as robustness metric for comparison.

Figure 3 shows a dependency between the number of blocks processed by a watermark reader and the number of successfully decoded frames. Several strategies with the number of blocks per frame are shown for parameter α between 0 and 1. When $\alpha = 0$, all pooled probabilities will be greater than zero and the localization algorithm will always select c block positions from the frame. On the other hand, no locations are selected when $\alpha = 1$.

Highlighted points in the figure correspond to points obtained with $\alpha = 0.35$. We consider the value $\alpha = 0.35$ as optimal since smaller values of α do not provide significant improvements in robustness while increasing complexity. Two dashed straight lines show the performance of systems with fixed seven and 165 coordinates. These fixed schemes process 70,000 and 1,650,000 image blocks respectively.

Figure 3 also shows that the system with seven empirically-determined coordinates can be outperformed even with up to $c = 2$ coordinates per frame resulting in more successfully extracted watermark payloads with only 10,500 processed blocks. The number of watermark payloads still increases significantly if more image blocks are examined by the watermark reader (larger value of parameter c). Achieved results allow us to get comparable robustness with approximately one-fifth of the computation time.

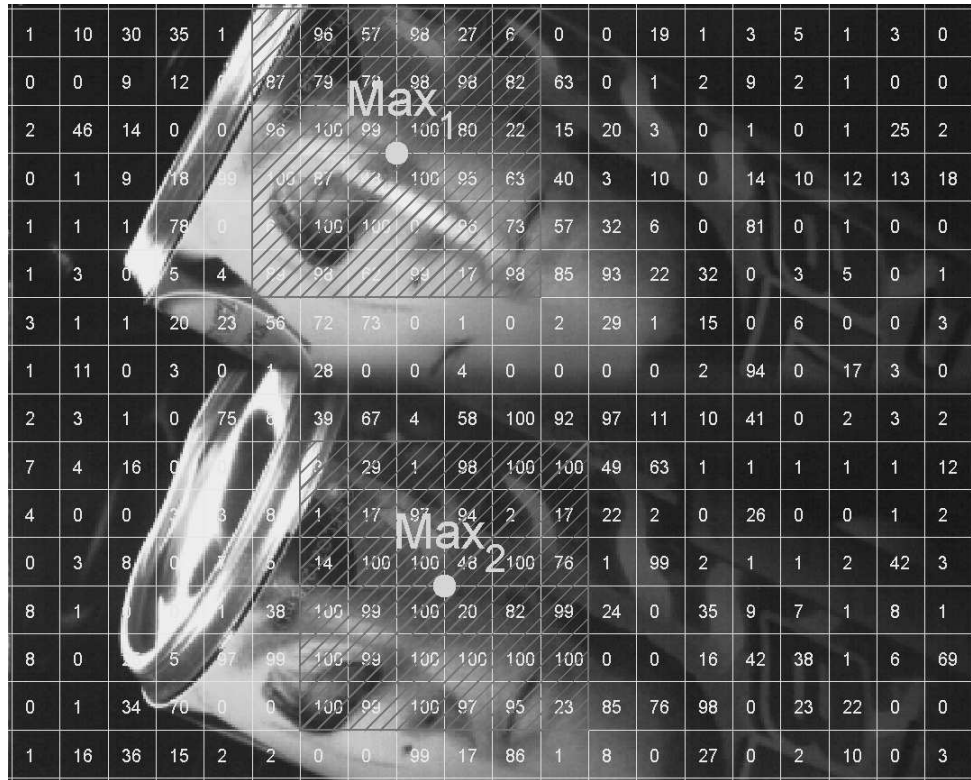


Figure 2. Captured frame of a watermarked package with probability estimates calculated for each non-overlapping sub-block. The two maximums over 6×6 neighborhoods mark the final coordinates for the watermark reader.

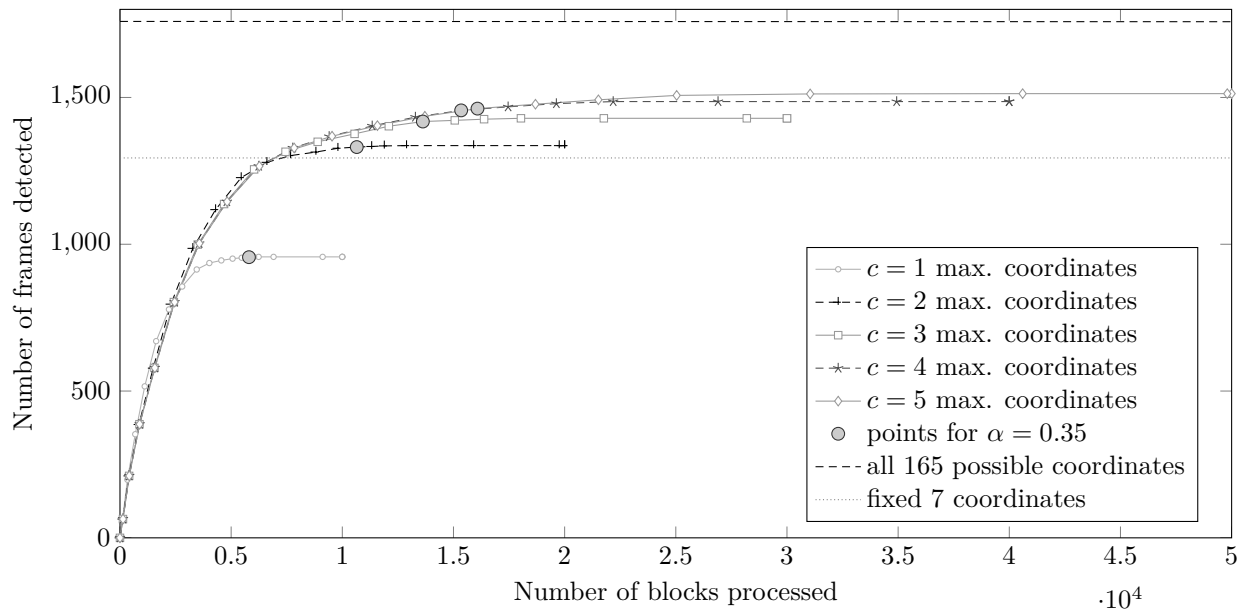


Figure 3. Robustness of the proposed watermark localization system. The graph shows dependency of the number of blocks processed by the watermark reader (complexity) and the number of frames with at least one successfully extracted watermark (robustness) on sensitivity parameter α for multiple c .

Independent of the localization parameters, the whole process of image localization (division into the sub-blocks, feature extraction, classification, finding the best coordinates) takes around 1.5 milliseconds on multi-core Intel 2.5 GHz laptop processor when implemented in C++ using the SSE instructions and utilizing only a single core.

6. CONCLUSION

In this paper, we focused on a problem of real-time watermark extraction on computationally constrained devices. In such cases, a limited amount of processing time is available for watermark reader and thus the whole frame cannot be thoroughly examined by a watermark reader within the budgeted time. One of the main contributions of this paper is a machine learning-based framework that is able to localize areas containing watermark with sufficient precision significantly faster than running watermark reader. In the case of watermarked consumer packages read by an off-the-shelf barcode imaging scanner, the proposed adaptive approach improved performance over an empirically designed system examining fixed image locations. The proposed approach used approximately one-fifth of the computation time. This was achieved in spite of high noise level in the acquired images.

The key component of the framework are noise-sensitive features borrowed from the field of steganalysis which are able to distinguish Digimarc Discover noise-like watermark from other noisy content. These features are classified by a trained logistic regressor that outputs the probability of watermark presence. By evaluating these probabilities from different parts of an image this framework is able to output a set of coordinates that are further processed by the watermark reader.

ACKNOWLEDGMENTS

The authors would like to thank Ravi Sharma and Adnan Alattar of Digimarc Corporation for their contributions to the paper. The authors would also like to thank Wade Jackson, Yang Bai and Kristyn Falkenstern of Digimarc Corporation for editing the manuscript.

REFERENCES

1. I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker. *Digital Watermarking and Steganography*. Morgan Kaufman Publishers Inc., San Francisco, CA, 2007.
2. R. O. Duda, P. E. Hart, and D. H. Stork. *Pattern Classification*. Wiley Interscience, New York, 2nd edition, 2000.
3. R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
4. J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, 2009.
5. J. Fridrich and J. Kodovský. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2011.
6. R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973.