Illustration Watermarking: An Object Based Approach for Digital Images

Thomas Vogel, Jana Dittmann

Otto-von-Guericke University of Magdeburg, Germany
Working group: Advanced Multimedia and Security
{thomas.vogel, jana.dittmann}@iti.cs.uni-magdeburg.de

ABSTRACT

For most applications common watermarking techniques usually spread the data to embed over the entire media, since distributing the watermark information promises an improvement in regard to security aspects, data hiding capacity or rather robustness in terms of redundancy. Distribution is controlled by a a visual or psychoacoustical model that takes limitations of the Human Visual System (HVS) and syntactical information about the signal characteristics into account. Therefore in most cases syntactical and not semantical aspects determine embedding. In our paper we introduce an approach for object based annotation watermarking which respects semantical characteristics of digital images, referred to as model for illustration watermarking. By applying a user-assisted segmentation process regions, representing semantical objects within the image, are identified and prepared for embedding. Providing robustness to typical image processing operations like cropping, scaling, compression and rotation, the proposed technique is applicable for binding additional illustrative information to selected objects within the medium. Moreover we identify the requirements of object based watermarking in consideration of imperceptibility, as well as watermark payload, and present first experimental test results.

Keywords: annotation watermarking, illustration watermarking, object based watermarking, annotation browser

1. INTRODUCTION

The technology of digital watermarking is used for a variety of applications, e.g. to protect the copyrights of users, to guarantee the integrity of content or to provide additional information embedded in the media. Each individual watermarking scheme can be assigned to at least one of the following categories[2]:

- *Copyright watermarking* is applied to secure ownership on copyrighted material, detect originators of illegally made copies, monitor the usage of the copyrighted multimedia data and analyse the spread spectrum of the data over networks and servers.
- *Integrity watermarking* aims at the protection of digital content in terms of embedding integrity information in the media for detecting content changes.
- *Annotation watermarking* (sometimes also called caption watermarking) is used to embed supplementary information directly in the media, so that the additional information cannot be separated from the media by accident (e.g. meta data like the ImageDescription field in the TIFF header can easily be taken away by converting the image from TIFF to JPEG).

Irrespective of the application in common watermarking algorithms the embedded information is either spread over the entire medium or concentrated at dedicated positions given by a visual or psychoacoustical model. Syntactic and not semantic aspects determine thus embedding. For ease of exposition, we assume for the rest of this paper that the content being watermarked is a still image, though most statements given in the following are, in principle, in a similar way applicable to audio and video data. Moreover we focus on annotation watermarks since there is a wide range of applications to annotation watermarking that have not been surveyed yet. In comparison to copyright watermarking annotation watermarking does generally not need to be difficult to remove. If an attacker wants to destroy the embedded information there is often no need to keep him from that. Annotated data would lose value and therefore there is in most cases no attack motivation. Thus security issues are less important. Even though annotation watermarking algorithms may be robust to cropping and therefore the embedded information could be extracted after cutting out some parts of an image,

there may be a 'semantical gap' between the original image and the cropped image since the embedded information belongs to the entire image. For example let us assume the original image shows a windmill on a green hill under a blue sky and the annotation watermark is a string containing "Windmill". Thus the content of the original image is additionally described by the annotation watermark. A common application for this form of annotation may be a vocabulary-learning tool. If an attacker cuts out the windmill such that only the green hill and the blue sky remain the relationship between the marked image and the embedded information is violated. The problem is that the embedded information is spread over the whole image and not directly stored in the related objects within the image. In the described situation the information has to be bound rather to single objects than to the entire image. Common watermarking schemes usually do not support object based information embedding. There is one scheme developed by Digimarc Corporation that uses single objects for embedding but the watermark information is limited to URLs and depends directly on the size of the object[5]. Additionally there are algorithms providing region-based watermarking but in this case the main goal is to provide rather integrity than object based annotation[14].

In our paper we focus on annotation watermarks for digital images that are bound to user-defined objects within the image and compare the requirements to the common used watermarking approaches. We present a technology for object based watermarking related to semantical information of images. The used positions for embedding are first defined by the user at a high, semantical level and only in a next step the positions are qualified by the visual model at a lower, syntactial level. In contrast to the watermarking scheme from Digimarc Corporation our approach offers a form of object-size independent embedding so that a virtually unlimited amount of information can be bound to a very small object. Beyond it the embedded watermark is not limited to a special type of information. This paper is organized as follows. In section 2 we introduce the term of illustration watermarking and the requirements for object based watermarking in respect to the watermarking parameters transparency, capacity, security and robustness as described in Jana Dittmann's „*Digitale Wasserzeichen*"[2]. We further introduce a model for illustration watermarking and present the workflow of a digital image in the space of this model. In section 3 we concentrate on the used algorithms and describe the segmentation, feature extraction and remaining preprocessing in more detail. Section 4 addresses implementation issues related to techniques for modelling the Human Visual System (HVS), suitable error correction codes and the demand for confidentiality, as well as the need for embedding references, and section 5 contains information about first test results. Conclusion is given in section 6.

2. ILLUSTRATION WATERMARKING

Since the term *illustration watermarking* has not been widely used in literature until now, we first introduce illustration watermarking in this chapter as a form of object based watermarking with specific requirements. These requirements will be discussed in view of the common watermarking aspects and a formalization of that requirements will be given, too. Afterwards the concept of illustration watermarking will be demonstrated by a model described below. In our former work we used the term *steganographic illustrating* for that model but from a second point of view we decided to drop it since in our opinion steganographic and illustrating are oppositional terms.

2.1 Requirements of object based watermarking

After discussing the basic difficulties of common watermarking schemes in regard to object based watermarking in the introduction we are now able to formulate the requirements of an approach to object based watermarking. An important claim of object based watermarking is the consideration of user acceptance. If there are noticeable artefacts in the marked media the user will probably not accept the media for professional and semi-professional purposes like publishing and presentation. Therefore transparency of the marked media is a fundamental requirement. On the other hand the security properties of the watermarking scheme have a minor relevance. The embedded data itself must be encrypted to protect the content but the watermarking scheme does not need to be secure to statistical detection. An attacker may separate the embedded information from the cover. As long as he cannot decrypt the information it is useless to him. In contrast to security issues robustness to various media transformation processes is highly desired. In our approach we concentrate on cropping, scaling, compression and rotation, since these are typical operations in further processing the marked image. Beside this the capacity of the watermarking scheme is another important demand. It is obvious that capacity generally correlates with the size of the object to mark. Hence the watermarking algorithm must account for using appropriate watermark vectors according to the capacity of the related object. Furthermore solutions have to be developed for binding large sets of information to small objects.

2.2 Formalization of requirements and illustration watermarking

Since we need to integrate our concept of illustration watermarking into the well-known terminology of digital watermarking, we give a formal overview of the most important terms used in this work at first. This formalization is primarily based on the terms used in books of J. Dittmann[2] and I. J. Cox, M. M. Miller, J. A. Bloom[3].

Let C be the set containing all cover work suitable for digital watermarking. According to Figure 1, let $o \in O \subseteq C$ be the original cover work, wm_{encode}: $M \times K \rightarrow C$ a watermark encoder function, $m \in M$ the input message, and $k_p \in K$ the private key.

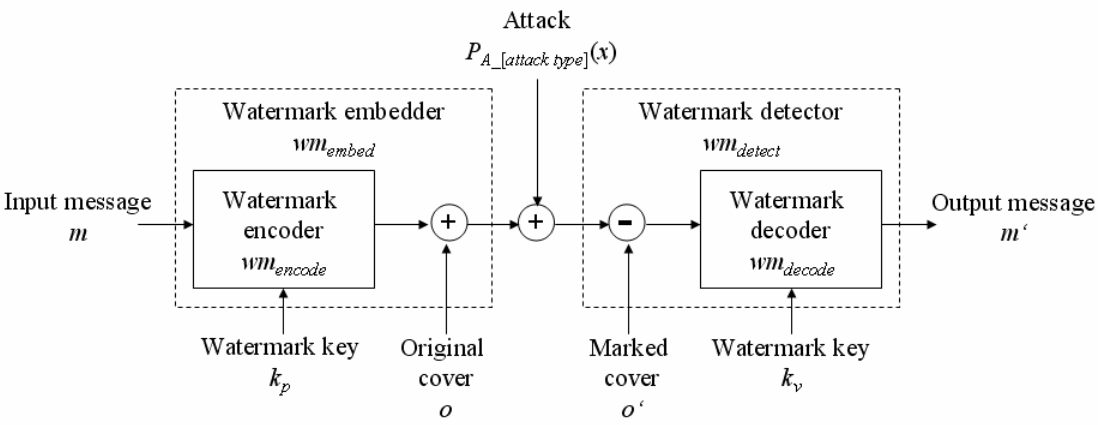

Figure 1: Generic model for digital watermarking.

Then we get the watermarked work $o' \in O' \subseteq C$ by using a watermark embedding function wm_{embed}: $O \times C \rightarrow O'$. Let us denote

$$o' = wm_{embed}(o, wm_{encode}(m, k_p)). \tag{1}$$

Furthermore we use the notation and concept of profiles introduced in the paper of A. Lang, J. Dittmann, E. T. Lin, E. J. Delp[20] for modeling possible attacks on o'. The concept of profiles describes the combination of different attacks for re-enacting certain attack scenarios and achieving a more realistic reflection of the real world situation. For example we would denote the profile $P_{A\text{-}AddNoise}$ representing an add noise attack as $P_{A\text{-}AddNoise}(\textit{in-signal} \parallel \textit{out-signal} \parallel \textit{parameters})$.

Let wm_{detect}: $O \times O'' \rightarrow C$ be an informed detector function, and $o'' \in O''$ the watermarked work, that has been possibly attacked. We achieve the output message m' by applying a watermark decoding function wm_{decode}: $C \times K \rightarrow M$ using $k_v \in K$ for verification. For a non-blind watermarking scheme we denote

$$m' = wm_{decode}(wm_{detect}(o, o''), k_v). \tag{2a}$$

If we use a blind watermark detector, wm_{detect} is equivalent to id and hence

$$m' = wm_{decode}(wm_{detect}(w), k_v) = wm_{decode}(w, k_v). \tag{2b}$$

Definition WATERMARKING SCHEME. Let $N=\{0, 1, 2, ...\}$ be a set of non-negative integers and $WP = (wm_{encode}, wm_{embed}, wm_{detect}, wm_{decode})$ describe the watermarking process, $|.|$: $M \rightarrow N$ determine the length in bits of a message $m \in M$, and $\#_{WS}$: $O \rightarrow N$ define the maximum capacity in bits of $o \in O$ for the embedding function wm_{embed}. Further we denote $a \approx_{s,t} b$ if, and only if, the distance between a and b, measured by an appropriate similarity function s, is smaller than a given threshold t, as well as we denote $(k_p, k_v) \in R_K$ if, and only if, $m = wm_{decode}(wm_{encode}(m, k_p), k_v)$. A watermarking scheme can be now represented by the 4-tuple (O, K, M, WP) and the following constraint:

$WS=(O, K, M, WP)$ is referred to as *watermarking scheme* if, and only if,

$$\forall o \in O\ \forall m \in M\ \forall (k_p, k_v) \in R_K\ .\ (\#_{WS}(o) \geq |m| \wedge m \approx_{s,t} wm_{decode}(wm_{detect}(o, wm_{embed}(o, wm_{encode}(m, k_p))), k_v)\ ). \tag{3}$$

After describing the term *watermarking scheme*, we now specify the desired properties in respect of object orientation, as well as robustness to rotation, scaling and cropping.

Definition OBECT BASED WATERMARKING. Let $WS = (O, K, M, WP)$ be a watermarking scheme. If we take image o as a constitution of l particular objects $o_i \in O$, i.e. $o = \{o_1, o_2, \dots, o_l\}$ we define $W_o = \{o_{i1}, o_{i2}, \dots, o_{iw} \mid 1 \le w \le l,\ 1 \le i_j \le w\}$ as the set of all $w \le l$ objects in o intended for the watermarking process. Let $m_i \in M$ be the message, as well as k_{pi} the private key and k_{vi} the verification key used for embedding in object o_i. With $o_i' = wm_{embed}(o_i, wm_{encode}(m_i, k_{pi}))$ we denote

$$WS \text{ is } \textit{object-based} \text{ if, and only if,}$$
$$\forall i \in \{i_1, i_2, \dots, i_w\}\ \forall o_i \in O\ \forall m_i \in M\ .\ (o' \supseteq \{o_i' \mid 1 \le i \le w\} \wedge m_i \approx_{s,t} wm_{decode}(wm_{detect}(o, o'), k_{vi})). \quad (4)$$

Definition COMPRESSION, ROTATION, CROPPING, SCALING. Let $WS=(O, K, M, WP)$ be a watermarking scheme. We denote

$$WS \text{ is } \delta\textit{-robust} \text{ if, and only if,} \quad (5a)$$
$$o' = wm_{embed}(o, wm_{encode}(m, k_p)) \wedge P_{A\text{-}Compression\ attack}(o' \parallel o'' \parallel parameters) \wedge m \approx_{s,t} wm_{decode}(wm_{detect}(o, o''), k_v).$$

$$WS \text{ is } \varphi\textit{-robust} \text{ if, and only if,} \quad (5b)$$
$$o' = wm_{embed}(o, wm_{encode}(m, k_p)) \wedge P_{A\text{-}Rotaton\ attack}(o' \parallel o'' \parallel parameters) \wedge m \approx_{s,t} wm_{decode}(wm_{detect}(o, o''), k_v).$$

$$WS \text{ is } \lambda\textit{-robust} \text{ if, and only if,} \quad (5c)$$
$$o' = wm_{embed}(o, wm_{encode}(m, k_p)) \wedge P_{A\text{-}Cropping\ attack}(o' \parallel o'' \parallel parameters) \wedge m \approx_{s,t} wm_{decode}(wm_{detect}(o, o''), k_v).$$

$$WS \text{ is } \theta\textit{-robust} \text{ if, and only if,} \quad (5d)$$
$$o' = wm_{embed}(o, wm_{encode}(m, k_p)) \wedge P_{A\text{-}Scaling\ attack}(o' \parallel o'' \parallel parameters) \wedge m \approx_{s,t} wm_{decode}(wm_{detect}(o, o''), k_v).$$

Let us further denote

$$\text{WS is } (\delta,\varphi,\lambda,\theta)\textit{-robust}, \text{ if, and only if, WS is } \delta\textit{-robust}, \varphi\textit{-robust}, \lambda\textit{-robust} \text{ and } \theta\textit{-robust}. \quad (6)$$

Based on this preparatory work we introduce illustration watermarking as an object based form of annotation watermarking carrying additional illustrative information of the medium while providing robustness to typical image processing operations like cropping, scaling, compression and rotation.

Definition ILLUSTRATION WATERMARKING SCHEME. IWS is referred to as *illustration watermarking scheme* if, and only if,

i) IWS is a *watermarking scheme*, q.v. (3),
ii) IWS is *object-based*, q.v. (4),
iii) IWS is $(\delta,\varphi,\lambda,\theta)$*-robust*, q.v. (6),
iv) $\forall o \in O\ .\ \#_{IWS}(o) \ge \#_{IWS'}(o)$ and IWS' is an *illustration watermarking scheme*.

2.3 Model of illustration watermarking

In the following we present our model of illustration watermarking and describe top down the framework that is needed for illustration watermarking. The model (q.v. Figure 2) consists of the authoring tool, the illustration encoder and the annotation browser. The authoring tool is used to load the media o, select the objects within o and choose the information to embed. The selection of the desired objects is supported by a segmentation algorithm but the user can independently choose a region of interest, too. The information to embed can be all kind of data, e.g. simple text, an audio file, an image file or a video file. After the selection of image, objects and information by the user the authoring tool calls the illustration encoder. The encoder merges the objects and the information. If the capacity given by the visual model of an object is large enough the information can be directly embedded in the media. Otherwise the information will be linked to the object in the form of an URL[5] or a (database) identifier. In the later case the encoder generates the appropriate website or the database entry. In order that the watermark can be found by the retrieving process the encoder embeds a synchronization identifier with each watermark. After processing all objects selected for embedding the illustration encoder outputs the marked image o' and the embedding process is finished.

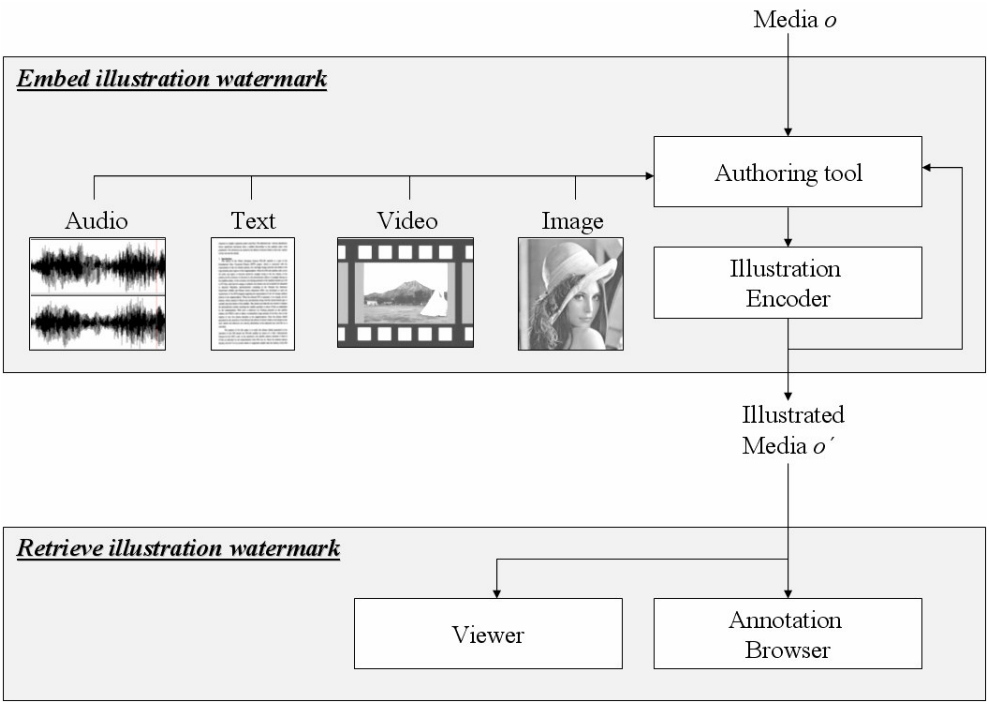

Figure 2: Model of illustration watermarking.

For retrieving the information from the marked image *o'* an annotation browser has to be used. This tool may be implemented in various ways, e.g. as a client-side stand-alone application, a web browser plug-in or a server-side web application. The annotation browser locates the watermark using the synchronization identifier. Then it extracts and decodes the information and displays the associated content. Subject to the application the displayed information does not need to be static so that the user can interact with the medium.

3. ALGORITHMS

After giving a first overview of the general process by introducing our model of illustration watermarking in the previous chapter we go more into detail concentrating on necessary techniques and the applied algorithms now. First we describe the process of object segmentation.

3.1 Object segmentation

Since the annotation browser has no semantical information about the image and reasonable segmentation of arbitrary digital images is still object of research the process of determining the objects does not rely on completely automated segmentation. Instead the segmentation process is user-assisted which means that the user can control segmentation parameters like texture, color and shape. After choosing a region of interest (ROI) a segmentation algorithm is applied to that region. The segmentation code used provides configurable settings and originally comes from Blobworld, a system for image retrieval from University of California, Berkeley. For a better understanding a short explanation of the functionality is stated here. In the original work[4] you will find a more comprehensive introduction to the algorithm. Generally the system tries to find coherent image regions which roughly correspond to objects by fitting a mixture of Gaussians to the pixel distribution in a joint color-texture-position feature space. Each identified region (named "blob") is then associated with color and texture descriptors. For this purpose the algorithm performs two major steps: Grouping pixels into regions and describing regions by feature vectors. The former can be further divided into three minor steps: Extracting features, combining features and grouping features (q.v. Figure 3). Each pixel is assigned a vector consisting of color coordinates in the L*a*b* color space, texture features like contrast, anisotropy and polarity and the (x,y) position. After feature extraction the 8-D vectors are combined to clusters using the Expectation-Maximization algorithm[19]. All pixels belonging to the same cluster constitute a spatial grouping. At this point the plain segmentation process is

completed. For region description the algorithm stores the color histogram, the mean texture contrast and anisotropy for each "blob" providing a fast alternative to compare different regions of the image.

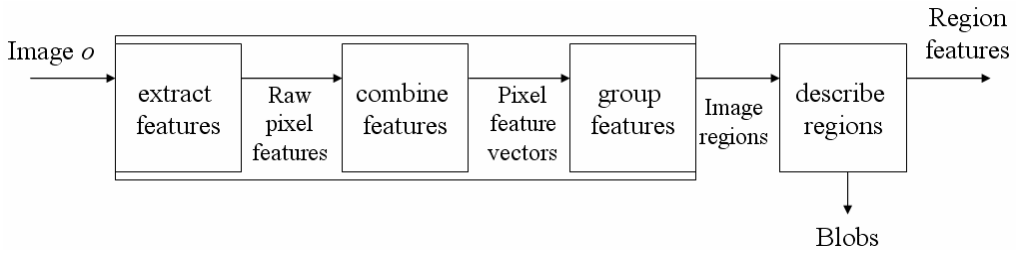

Figure 3: Stages of processing.

An example for applying the algorithm to the illustration watermark scenario is given in Figure 4. The windmill right of the building has been selected by the user as region of interest (ROI). For this region the segmentation borders of all identified objects within the ROI are painted in a bold style. Now the user can easily select one or more objects within that ROI for embedding, e.g. one of the four wings of the windmill.

Figure 4: User-assisted segmentation process and object identification.

3.2 Capacity measurement

For the watermark embedder wm_{embed} it is necessary to know the maximum possible watermark payload, since the algorithm must decide if there is enough capacity for the information itself, or instead the algorithm has to use a reference to that information by using an URL or database identifier. After object segmentation we know which region (i.e. object) the user wants to allow for embedding. Position and size are determined and the watermark embedder can start measuring the maximum capacity of an object by applying our visual model introduced in section 4.3. In a first step we try to identify suitable positions within the chosen region by uniformly spreading the watermark information over all pixels (and frequencies respectively). Applying the quality measures of our visual model we find pixels (and frequencies) which cause the fewest distortion and others that introduce a lot of artifacts to the region. With this information we can establish a ranking of positions and define weights for each position within the object. For a given lower-bound quality index q_{min} the algorithm is able to measure the maximum watermark payload for that object by non-uniformly spreading the watermark using the weights calculated before, such that the quality index of the marked image o' is greater or equal to q_{min}, i.e. $q \geq q_{min}$. Additionally the usable capacity depends on the desired robustness which we plan to take into account by applying the model for Parallel Gaussian Channels described in the work of P. Moulin and M. Kivanç Mihçak[21].

3.3 Watermark embedding and retrieving

At the present time work on our watermarking scheme hasn't finished yet. Our scheme is partly based on the m-band wavelet presented in the work of Y. Fang, N. Bi, D. Huang and J. Huang[6]. Initially the proposed scheme provides basic robustness to JPEG compression and Gaussian noise. Additional robustness to scaling and rotation we plan to realize by templates using steady features of the image. Used features will relate to color, texture and shape. The wavelet transform is widely used in many signal processing applications including image coding and analysis. It also plays an important role in watermarking due to its time-frequency localization characteristics and matching well to HVS features.

Furthermore, the Multi-Resolution Analysis (MRA), as a principle of Discrete Wavelet Transform (DWT), is contributing to the robustness and perceptual effect of digital watermarking. Therefore, we expect there is better promise using the watermarking algorithm in DWT domain than in Discrete Cosine Transformation (DCT) or Discrete Fourier Transform (DFT) domain[3,7,8].

4. IMPLEMENTATION ISSUES

Beside the main algorithms mentioned in the previous chapter there are some more techniques building the framework of the illustration watermarking concept. In the next subsections we will address other important topics related to our model.

4.1 Securing information

Like we constituted in chapter 2, security is not one of the basic requirements of illustration watermarking. Indeed, information embedded in the media might represent valuable assets such that confidentiality must be protected by an cryptographic cipher. For this purpose we use some popular symmetric algorithms. Currently Rijndael[15], Twofish[16] and TripleDES[17] are supported, but more can easily be added due to our modular software architecture. For improved security we apply both Rijndael and Twofish to the input message m before embedding. Combining both Rijndael and Twofish provides a much stronger encryption then using only one algorithm. For the highly unlikely case that a weakness is discovered in one of the algorithms, the use of the second algorithm provides still a sufficient margin of security.

4.2 Applying error correction

In order to decrease the error rate of the retrieved watermark we apply the Bose-Chaudhuri-Hochquenghem (BCH) error correction code for information encoding after reducing the length of the watermark vector by the best-fitting of four compression algorithms, that are in detail gzip[10], bzip2[11], zlib[12] and Huffman[13]. Since BCH codes do not perform well at high channel error rate conditions, we expect further improvements by applying Low Density Parity Check (LDPC) codes[22] as a replacement for BCH codes. LDPC codes are known to be one of the best linear codes performing very near to the Shannon limits and we look forward to optimize capacity or rather robustness characteristics of the applied watermarking scheme.

4.3 Finding suitable positions

The underlying idea of image watermarking is to create a new image which is statistically different but perceptually identical to the host signal[9]. Thus imperceptibility is considered as a very important aspect of watermarking schemes. We use a visual model that is based on known image quality measures and provides a quality index q. Currently we evaluate a feature vector $f = (f_1, f_2, f_3, f_4, f_5)$ covering the features f_i addressed in the next subsections. To achieve an overall measure for image quality we define the quality index q as the weighted mean of f_1 to f_5 (q.v. Figure 5).

$$q = \sum_{i=1}^{5} \omega_i f_i \quad , \qquad \sum_{i=1}^{5} \omega_i = 1$$

Figure 5: Image quality index.

The notation of the following sections still uses o for the original cover and o' for the marked cover. Furthermore $o(i,j)$ represents the pixel value at position (i,j) in image o and, after applying a transformation like DCT or DFT, (u,v) represents the coordinates of a coefficient. The number of bands of an image, i.e. the number of channels, is denoted by m. For example, we get $m = 3$ for a RGB color image. Width and height of an image are denoted by X and Y. Since we split rectangular images with $X \neq Y$ into square blocks we denote the side length of a block by Z and the size, i.e. the total number of pixel, by Z^2. For example, for a 512x512px image we get $X = Y = Z = 512$ and $Z^2 = 262.144$.

4.3.1 HVS absolute norm

In order to obtain a closer relation with the assessment by the human visual system, both the original and watermarked images can be preprocessed via filters that simulate the HVS. One of the models for the human visual system is given as a band-pass filter with a transfer function in polar coordinates[9]. Let $\rho = (u^2 + v^2)^{1/2}$ and $H(\rho)$ defined by Figure 6.

$$H(\rho)=\begin{cases}\frac{1}{20}e^{\rho^{277/500}} & ,\rho<7\\ e^{-9|\log_{10}(\rho)-\log_{10}(9)|^{23/10}} & ,\rho\geq 7\end{cases}$$

Figure 6: Spectral mask.

An image *o* processed through such a spectral mask and then transformed by the inverse Discrete Cosine Transform (DCT^{-1}) can be expressed via:

$$U\{o(i,j)\}=DCT^{-1}\{H((u^2+v^2)^{1/2})DCT\{o(i,j)\}\}$$

Figure 7: Image after processing.

The first measure for the multi-spectral images we use is the normalized absolute error defined in Figure 8:

$$f_1=\frac{1}{m}\sum_{k=1}^{m}\left(\sum_{i=0}^{Z-1}\sum_{j=0}^{Z-1}|U\{o(i,j)\}-U\{o'(i,j)\}|\ /\sum_{i=0}^{Z-1}\sum_{j=0}^{Z-1}|U\{o(i,j)\}|\right)$$

Figure 8: HVS absolute norm.

4.3.2 HVS L2 norm

As addition to the previous mentioned HVS absolute norm[9] we use the HVS L2 norm as defined in Figure 9:

$$f_2=\frac{1}{m}\sum_{k=1}^{m}\left(\frac{1}{Z^2}\sum_{i=0}^{Z-1}\sum_{j=0}^{Z-1}|U\{o(i,j)\}-U\{o'(i,j)\}|^2\right)^{1/2}$$

Figure 9: HVS L2 norm.

4.3.3 Multiresolution error

One limitation of standard objective measures of distance between images is that the comparison is conducted at the full image resolution. Alternative measures can be defined that resemble image perception in the human visual system more closely, by assigning larger weights to low resolutions and smaller weights to the detail image[9]. Such measures are also more realistic in machine vision tasks that often use local information only. Consider the various levels of resolution denoted by $r\geq 1$. For each value of *r* the image is split into blocks b_1 to b_n where *n* depends on scale *r*. For example for $r=1$, at the lowest resolution, only one block covers the whole image characterized by its average gray level *g*. Let g_{ij} be the average gray level of block b_{ij} at the resolution *r*. The distortion at this level is determined by Figure 10.

$$d_r=\frac{1}{2^r}\frac{1}{2^{2r-2}}\sum_{i=1}^{2^{r-1}}\sum_{j=1}^{2^{r-1}}|g_{ij}-g'_{ij}|$$

Figure 10: Distortion index.

Let *R* be the number of all resolution levels. The actual value of *R* (the number of resolution levels) will be set by the initial resolution of the digital image. For example, for a 512x512px image we get $R = 9$. Finally for K-band multi-spectral images the definition can be extended to:

$$f_3=\frac{1}{m}\sum_{k=1}^{m}\sum_{r=1}^{R}d_r^k$$

Figure 11: Multiresolution error.

4.3.4 L*a*b* perceptual error

The choice of color-space for an image similarity metric is important, because the color-space must be uniform, so that the intensity difference between two colors is consistent with the color difference estimated by a human observer. Since the RGB model is not well-suited for this task we decided to use the more appropriate 1976 CIE L*a*b* color-space[9].

One recommended color-difference equation for the L*a*b* color-space is simply given by the Euclidean distance in Figure 12.

$$\Delta L(i,j) = L(i,j) - L'(i,j)$$
$$\Delta a(i,j) = a(i,j) - a'(i,j)$$
$$\Delta b(i,j) = b(i,j) - b'(i,j)$$
$$f_4 = \frac{1}{Z^2} \sum_{i=0}^{Z-1} \sum_{j=0}^{Z-1} \left(\Delta L(i,j)^2 + \Delta a(i,j)^2 + \Delta b(i,j)^2 \right)$$

Figure 12: L*a*b perceptual error.

4.3.5 Spectral phase error

In this category we consider the distortion penalty functions obtained from the complex Fourier spectrum of images[9]. Let the spectra of the Discrete Fourier Transforms (DFT) of the k-th band of the image be denoted by Figure 13:

$$\Gamma_k(u,v) = \sum_{\alpha=0}^{Z-1} \sum_{\beta=0}^{Z-1} o_k(\alpha,\beta) e^{\frac{-2\pi i u \alpha}{Z}} e^{\frac{-2\pi i v \beta}{Z}}, \quad k = 1, \ldots, m$$

Figure 13: Discrete Fourier Transform for band k.

Abstracting from the particular bands we concentrate on the phase spectra for the luminance channel and use $\varphi(u, v)$ as defined in Figure 14.

$$\varphi(u,v) = \arctan(\Gamma(u,v))$$

Figure 14: Phase spectrum.

For our feature vector we determine the spectral phase distortion as stated in Figure 15. Furthermore we think about considering the magnitude phase distortion as an additional vector element[9].

$$f_5 = \frac{1}{Z^2} \sum_{u=0}^{Z-1} \sum_{v=0}^{Z-1} |\varphi(u,v) - \varphi'(u,v)|^2$$

Figure 15: Spectral phase error.

4.4 Embedding references

If the watermark embedder recognizes that the capacity of the chosen region is too small for embedding the total of information a reference to that information has to be generated and stored in the media. The information itself must be made available at the referenced location, of course. There are currently two types of references supported:

- An *Uniform Ressource Locator (URL)* can be embedded for referencing online material[5]. As customary the first part of the address indicates what protocol to use, and the second part specifies the IP address or the domain name where the resource is located. The maximum length of the URL depends on the capacity of the chosen object.
- A *Database Identifier* references to an entry of a table within a database. Only the identifier of the entry itself is embedded, neither the table name, nor any other information about the database. Therefore there must be default values or the annotation browser must have additional information, e.g. location of database or account information.

The second possibility is more challenging since we can provide additional functionality by using database identifiers. While writing illustration information to the database during the embedding process we are able to save extra information, e.g. features of the marked image or point in time of embedding. Providing that additional information to the watermark detector we expect to increase the detection rate since while retrieving the embedded information we will not only have to rely on the information gathered from the marked image, but we can use side information from the database to verify or complete the retrieved watermark data. Currently we operate on a local MySQL database but we plan to provide a publicly accessible database for storing illustration data for testing purposes.

5. FIRST EXPERIMENTAL RESULTS

In this chapter we present some first experimental test results related to imperceptibility and capacity. Since transparency is an important aspect for the acceptance of watermarked media we tested image quality first. For our experiments we used the well known Peak Signal to Noise Ratio (PSNR) that is based on the Mean Squared Error (MSE) for m-band images. Formulas are given in Figure 16. Here MAX_o is the maximum pixel value for all channels of the original cover o, in most cases $MAX_o = 255$.

$$PSNR = 10 \log\left(\frac{MAX_o^2}{MSE}\right) = 20 \log\left(\frac{MAX_o}{\sqrt{MSE}}\right) \qquad MSE = \frac{1}{m}\frac{1}{XY}\sum_{i=0}^{X-1}\sum_{j=0}^{Y-1}|o(i,j) - o'(i,j)|^2$$

Figure 16: Peak Signal to Noise Ratio and Mean Squared Error.

We expect to obtain a significantly higher visual quality by applying the image quality metrics presented in section 4.3 for finding suitable embedding positions, than by simply adding a randomly set dither mask as watermark. The tests have been performed over the six widely used test images shown in Figure 17. Furthermore we determine the maximum capacity for a given PSNR value by executing our algorithm for capacity measurement introduced in section 3.2. Finally some comments on robustness and possible approaches for testing the resistance to cropping, scaling, rotation and compression are denoted in section 5.2.

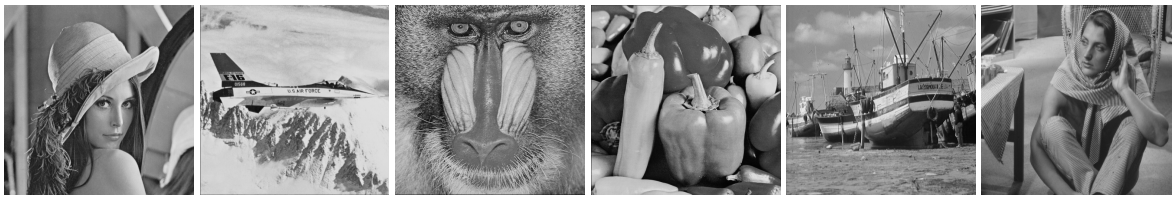

Figure 17: Test images.

5.1 Data hiding capacity and imperceptibility

Due to the fact that development of our DWT watermarking scheme has not finished yet, we performed the watermarking process by a usual LSB watermarking scheme modifying the least significant luminance channel bits of a given image. We tested transparency by applying a 10x10px dither mask to the image. Values of the mask were randomly set depending on the selected capacity in bits per pixel (bpp). For example, choosing a capacity of 0.3 bpp resulted in a mask consisting of 30 bits taken from the watermark vector and 70 bits equal to 0, all values randomly distributed over the entire mask. On the other hand we calculated the quality index q introduced in section 4.3 to find the best positions for hiding data. With this information we built the mask no longer by random but set the watermarking bits accordingly. In both cases the mask was gradually added to the luminance channel of the image. For comparing the two techniques we measured the visual quality decrease in dB using PSNR (q.v. Figure 16). Results are given in Figure 18 by the solid line representing the random mask, and the dashed line representing the visual model approach. The results demonstrate the general effectiveness of our visual model. The greatest differences we observe at higher bit rates. Using the visual model we gain an increase of more than 5 dB at 2.0 bpp.

	Lena	F16	Baboon	Vegetables	Ship	Barbara
20 dB	2.53	2.24	2.48	2.22	2.31	2.61
25 dB	2.14	1.99	2.10	2.01	2.00	2.19
30 dB	1.61	1.41	1.42	1.40	1.40	1.46
35 dB	0.98	0.84	0.91	0.9	0.85	0.82
40 dB	0.40	0.39	0.42	0.44	0.38	0.39

Table 1: Capacity (bits per pixel) against visual quality (decibel) for all test images

In respect of capacity we further tested the watermark payload by applying our algorithm for capacity measurement. For a given PSNR the algorithm estimated the maximum capacity for each test image. Results are given in Table 1. More detailed and most notably more actual information about testing can be found on our Website[23].

5.2 Robustness

For evaluation of robustness we plan to use the stirmark benchmark[18]. We expect a high robustness to gaussian noise and image compression. Applying template information[2] in form of geometric objects will hopefully lead to a moderate robustness to affine transformations and cropping, but tests have not been done yet.

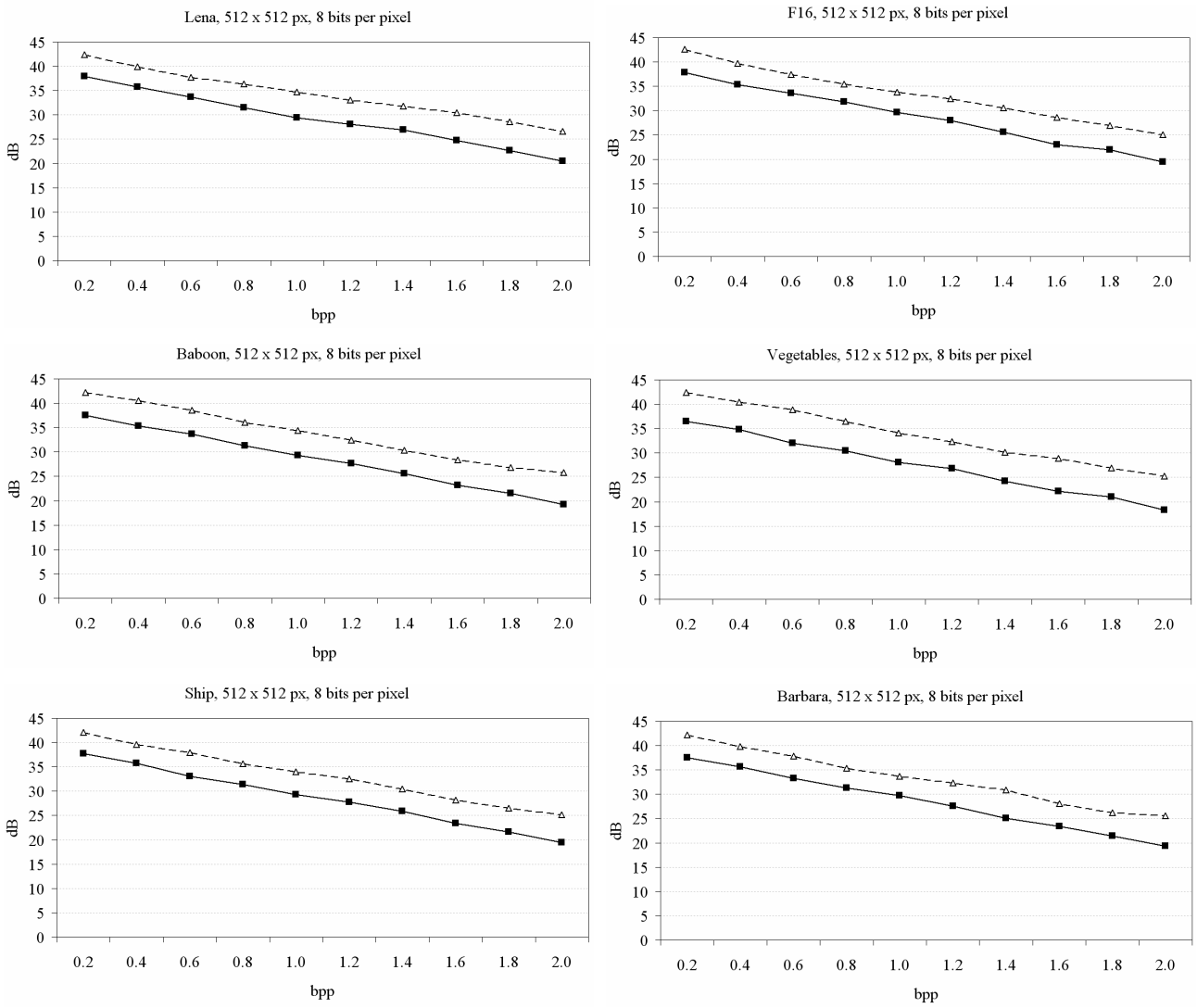

Figure 18: Relation between capacity (bits per pixel) and visual quality (decibel) for all test images.

6. CONCLUSION

We introduced a model for object based annotation watermarking respecting the semantical characteristics of digital images. In contrast to the majority of watermarking schemes our goal is to embed annotation directly into the certain objects by enriching them with further information. Beside the semantic analysis for the illustrations we evaluated the competing properties capacity, imperceptibility and robustness with respect to object-oriented annotation. After discussing the special requirements for object based watermarking we focused on the applied algorithms and described the framework which is necessary for implementing our watermarking scheme. Important goals we already reached are the formalization presented in chapter 2 for introducing illustration watermarking in a technical way and integrating this new form of annotation watermarking into the known watermarking domain, as well as the prototypical implementation

of basic components constituting our proposed model. We provide solutions for embedding references instead of the information itself and presented the first experimental test results in reference to capacity and imperceptibility in chapter 5. But there is still a lot work to do, since many topics mentioned in this paper are still object of our research. We plan to finish the development of the DWT watermarking scheme to verify the hypotheses stated in chapter 3.3. In a next step we will extend the spectrum of the supported algorithms related to encryption, error correction and feature extraction for measuring capacity and visual quality of a given image.

7. ACKNOWLEDGEMENTS

The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. The work described in this paper has been supported in part by the German Research Foundation and the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. Attack profiles for annotation watermarks were sponsored by the Air Force Office of Scientific Research. Air Force Material Command, USAF, under grant number FA8655-04-1-3010. Furthermore the authors wish to acknowledge the support and work of Doreen Rode, Henry Sonnet, Knut Hartmann and Thomas Strothotte, since discussions have been very fruitful at any time.

REFERENCES

1. J. Dittmann, K. Hartmann, H. Sonnet, F. Ritter, T. Strothotte, *Steganographisches Illustrieren: Neue Perspektiven für Try&Buy*, Proc. of DACH 2003.
2. J. Dittmann, *Digitale Wasserzeichen*, Springer Verlag, Berlin, 2000.
3. I. J. Cox, M. M. Miller, J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, 2001.
4. C. Carson, S. Belongie, H. Greenspan, J. Malik, *Blobworld: Image segmentation using Expectation-Maximization and its application to image querying*, University of California, Berkeley, 1999.
5. A. M. Alattar, '*Smart Images' Using Digimarc's Watermarking Technology*, IS&T/SPIE's 12th International Symposium on Electronic Imaging, San Jose, CA, January 25, 2000, volume 3971, number 25.
6. Y. Fang, N. Bi, D. Huang, J. Huang, „*The Multi-band Wavelets in Digital Image Watermarking* ".
7. Y. Fang, J. Huang, Y. Q. Shi, *Image Watermarking Algorithm Applying CDMA*, Proc. of IEEE Int. Sym. on Circuits and Systems, Bangkok, Thailand, 2(5), 2003, 948-951.
8. Y. Fang, J. Huang, S. Wu, *CDMA-based Watermarking Algorithm Resisting to Cropping*, Proc. of IEEE Int. Sym. on Circuits and Systems, Vancouver, Canada, May 23-26, 2004.
9. I. Avcibas, B. Sankur, K. Sayood, *Statistical evaluation of image quality measures*, Journal of Electronic Imaging 11(2), 206-223, April 2002.
10. J. Gailly, M. Adler, *The gzip home page*, <http://www.gzip.org/>, July 2003.
11. J. Seward, *The bzip2 and libbzip2 official home page*, <http://sources.redhat.com/bzip2/>, January 2002.
12. J. Gailly, M. Adler, *zlib Home Site*, <http://www.gzip.org/zlib/>, November 2003.
13. D. A. Huffman, *A method for the construction of minimum-redundancy codes*, Proceedings of the I.R.E., sept 1952, pp 1098-1102.
14. H. Liu, H. Sahbi, L. Croce Ferri, M. Steinebach, *Authentication Using Automatic Detected ROIs*, WIAMIS 2004; 5th International Workshop on Image Analysis for Multimedia Interactive Services, April 21-23, 2004, Instituto Superior Tonico, Lisboa, Portugal.
15. J. Daemen, V. Rijmen, *The Design of Rijndael*, Springer Verlag, 2004.
16. B. Schneier, *Applied Cryptography*, John Wiley & Sons, 1996.
17. National Bureau of Standards, *Data Encryption Standard*, U.S. Department of Commerce, FIPS pub. 46, January 1977.
18. F. Petitcolas, *Stirmark benchmark*, <http://www.petitcolas.net/fabien/watermarking/stirmark/>, May 2004.
19. A. Dempster, N. Laird, D. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. Royal Statistical Soc., Ser. B, 39, 1-38, 1977.
20. A. Lang, J. Dittmann, E. T. Lin, E. J. Delp, *Application Oriented Audio Watermark Benchmark Service*, to appear at SPIE 2005, San Jose, January 2005.
21. P. Moulin and M. Kivanç Mihçak, A Framework for Evaluating the Data-Hiding Capacity of Image Sources, IEEE International Conference on Image Processing, Vancouver, Canada, October 2000.
22. T. Richardson, R. Urbanke, *The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding*, IEEE Transactions on Information Theory, 47, 2001.
23. T. Vogel, J. Dittmann, *IWM test report*, <http://wwwiti.cs.uni-magdeburg.de/iti_amsl/publikationen/iwmrep05.pdf>, January 2005.