Interactive Images using Illustration Watermarks: Techniques, Study, and Applications

Henry Sonnet* Andrea Unger* Lothar Schlesier* Thomas Vogel†
Tobias Isenberg‡ Thomas Strothotte*

Abstract

The visual information provided by 2D images is typically static and so the entire content of an image is shown at once. When image components are now annotated to communicate the image's content more clearly, the actual information of the image is partially hidden. However, by hiding an image's annotations by default, the image can be turned into an interactive medium for which a user can demand the descriptive information at will. We suggest employing digital watermarking techniques to store auxiliary information directly in the regions the information is associated with, a procedure that we call *Illustration Watermarking*. That way, the additional information becomes part of the image but it is not permanently displayed. In this paper, we describe three illustration watermarking techniques. To analyze and compare these techniques and to determine the maximum amount of data which can be invisibly embedded per pixel, we conducted a user study. In this study, participants searched for coded regions in images of different types and with different amounts of data encoded. One of our results is that under normal viewing conditions at least 5 bits/pixel can be modified without perceivably changing the image. Finally, we present several application fields in which our proposed technique can be applied to enhance the image's communication of its information.

1 Introduction

Digital or printed images can serve as illustrations. Often, the basic content of such illustrations can be divided in two types of layers: the *Background Layer* which is the actual information to be communicated by the image (e. g., a pure photograph) and the *Illustration Layer* which augments the information of the *Background Layer* with auxiliary information that provides additional knowledge or that explains what is depicted in the image (e. g., image caption or integrated annotations). In printed media, these two layers are typically integrated in the same image and permanently visible, which can detract from the background layer due to overlaps and information abundance. In contrast, digital media allow for dynamically changing their contents. Individual parts of the illustration layer can thus be displayed when they are explicitly demanded by the viewer.

In case of digital media, background and illustration layer must be stored in some way, preferably in a compact format. Compactness in this regard means the two layers are combined in a single data file whose format is widely-used and whose size should not increase proportional to illustration layer size increase. This is particularly important when the augmented digital medium is to be distributed or when a

*Otto-von-Guericke University, Dept. of Simulation and Graphics, Magdeburg, Germany
†Otto-von-Guericke University, Dept. of Techn. and Operative Information Systems, Magdeburg, Germany
‡University of Calgary, Dept. of Computer Science, Calgary, Alberta, Canada

multiplicity of such media needs to be administrated. Another important aspect concerning layer storage is to maintain the link between background and illustration layer components, especially when several regions in the background layer are linked with several types of data in the illustration layer (e. g., text, images, or auditory data).

We suggest to exploit digital watermarking techniques for data layer storage. Using watermarking terminology, the background layer serves then as *Cover Medium* whereas the illustration layer data is converted into the *Watermark Message* that is to be encoded [CMB01]. Watermarking allows to invisibly embed data into digital media, which complies with compact data storage. However, since in the traditional watermarking domain watermark messages are basically encoded to provide copyright information or to ensure data origin authenticity, achieving high data storage capacity—which is important for our scenario—is disregarded. We refer to watermarking techniques which can be employed to embed a significant amount of data (*high capacity*) and which do not focus aspects such as security or robustness to criminally intended attacks as *Illustration Watermarking* techniques [SIDS03]. Illustration watermarking is used only to represent illustration layer data which can be any binary data an end-user benefits from. Further aspects which are aimed at by illustration watermarking are to avoid introducing perceivable modifications to the cover medium (*imperceptibility*), to enable data recovery without the need of the original data (*blind detection*), and to embed the watermark message locally (*content-based*), that is, the watermark message is inserted only into the region it is associated with.

In this paper, we demonstrate how digital watermarking can be employed to compactly store background and illustration layer in a single 2D image which, in result, is turned into an interactively explorable medium. The potentialities in terms of data capacity while maintaining imperceptibility were evaluated in the course of a user study. Beside determining the maximum amount of data which can be imperceptibly encoded, we were interested in evaluating a selection of techniques as well as in exploring the effects when different types of cover images are in use. Finally, we demonstrate various application scenarios for our proposed technique.

The paper is organized as follows: After reviewing related work in Section 2, we introduce the basic concept behind our idea as well as the techniques we have used in Section 3. Section 4 contains detailed information about the user study we have conducted. Application scenarios are discussed in Section 5, before Section 6 concludes this work.

2 Related Work

Previous work that relates to the techniques presented in this paper can be found in many fields of traditional watermarking (Section 2.1). Some of those techniques in the field of image watermarking for copyright protection follow a concept of region-based data embedding (Section 2.2). Even though our intention is different, analyzing these methods can be valuable for the content-based approach that we follow. Finally, we review a number of techniques that relate to our concept in that they are used to add information to media for illustrative purposes (Section 2.3).

2.1 Traditional watermarking techniques

Watermarking techniques are typically used to insert a portion of information into media that should not immediately be visible and detectable (see, e. g., COX et al. [CMB01]). In this regard, recent watermark research was mainly focused on the encryption of copyright and other security information. A considerable number of different approaches were published and partly reviewed by many authors (e. g., [BGML96, PAK99, SKT98]. JOHNSON and JAJODIA [JJ98] addressed watermarking and steganography, discussed differences, and introduced several software tools for data encryption.

Watermarking techniques can be classified in spatial and transformed domain techniques. As to transformed domain techniques, many of them have been introduced and successively improved that embed the watermark by applying a Discrete Cosine Transform (DCT) or a Discrete Wavelet Transform (DWT) (e. g., [CKLS97, KZ95, XBA98]). These techniques are mainly applied for reasons such as to withstand attacks or to be robust to image modifications (e. g., image compression). They are hence not capable of storing large amounts of data. However, comparable to what we aimed at with our user study, but under different conditions and with other foci, BARNI et al. [BBRP99] were interested in the maximum number of bits which can be hidden within an image using transformed domain techniques. Their approach is based on theoretical considerations. The actual number of bits which can be encoded per region strongly depends on several variables such as the distribution and magnitudes of the transformed domain coefficients.

Since our work aims at increasing an image's value by including its illustration layer, security issues can be disregarded. Instead, aspects such as high transparency, high capacity, and blind detection are of importance. Modifying the least significant bits (LSBs) of an image's color channels is a simple technique that meets these demands. TIRKEL et al. [TRvS+93] have been one of the first to present a LSB approach to embed a watermark into a gray tone image. KURAK's and MCHUGH's technique [KM92] of hiding an image by replacing the cover image's least significant bits by the watermark image's most significant bits can also be regarded as an LSB approach. Various modifications of the LSB technique exist. LIE and CHANG [LC99], for instance, use a mapping function, which was designed according to human visual sensitivity to contrast, to determine the encodable number of bits. In a similar approach, VAN DROOGENBROECK and DELVAUX [vDD02] divide the cover image into blocks of 8×8 pixels and compute their entropies. Thereafter, a pixel block's capacity can be determined depending on its entropy: the higher the entropy the more data can be embedded.

2.2 Region-based data embedding

When watermark insertion is limited to certain enclosed regions, it happens either to increase the transparency and the robustness of the embedded watermark to image manipulations or to assign a certain amount of data which is semantically linked with that region. Our technique aims at the second purpose.

Concerning techniques that aim at the first purpose, SU et al. [SWK99] state that a watermark is more protected when it is embedded within the most important regions of an image referred to as the regions of interest (ROI). With their technique, they can encode only small data amounts by modifying wavelet coefficients with respect to the afore selected ROI. NIKOLAIDIS and PITAS [NP01] discuss a region-based watermarking method that is robust to several attacks such as compression, filtering, or cropping. In their approach, regions to be watermarked are identified based on an image's salient spatial features. These regions are then approximated by ellipsoids whose bounding rectangles serve as the regions for watermark encoding.

The second aspect has been addressed by BOULGOURIS et al. [BKM+02]. They introduced a technique that aims at indexing images so that data can be retrieved from a database. To this end, two types of watermarks are embedded: one that stores segmentation data and the other that holds the indexing data. The concept presented by RYTSAR et al. [RVEP03] is similar to our approach. Regions in an image are interactively segmented before they are assigned annotations. However, they focus on encoding only short text labels for object identification and on providing pointers into a database.

2.3 Other techniques for linking media with metadata

Our approach can be classified in a digital-only domain, i. e., the cover medium as well as the watermark are virtual. However, there exist several approaches that follow the same idea as ours, namely to enrich a medium with additional information, but are hybrid: they combine real-world media and virtual data. ALATTAR, for example, discusses the concept of *Smart Images* [Ala00]. Those images are either digital or real and contain visually imperceptible data such as Web pointers or contact information. The idea behind smart images is to generate printouts of digital images once they have been augmented with metadata. These printouts are then distributed. After digitization (scanning), the encoded data can be extracted from the digital copy.

DYMETMAN and COPPERMAN [DC98] presented another hybrid approach, the concept of *Intelligent Paper*. Intelligent paper refers to a standard sheet of paper with marks (unique ID and local coordinates) imprinted which can only be detected by specific pointers. When those pointers are connected to the Web, arbitrary information specified by the paper's publisher can be retrieved. *Paper++* is a similar approach (e. g., [NS03]). Here, different virtual layers, arranged on top of a digital copy of a printed document, include regions which are linked to objects in a database. To query the database, a user can point at a printed page with a wand device that reads the page number and the position from a grid of barcodes printed on the paper. An alternative to the widely-used barcodes are *DataGlyphs* which were introduced by HECHT [Hec01]. These are patterns of small strokes whose orientations represent bits. To extract the data represented by those strokes, *Xerox PARC* developed the GLYPH-O-SCOPE system which records and processes the printed document before it displays the encoded information, virtually overlaid on the document.

3 Illustration Watermarks for 2D Digital Images

Images, in general, often need comments or explanations to adequately communicate their contents. This can be achieved by including metadata such as text annotations and image captions in the image. However, in case of static images, a major problem quickly becomes apparent: the presentation space is limited in size. Consequently, the included metadata must often be shortened and partial occlusions of the image must be accepted.

We thus separate original image content (background layer) and associated metadata (illustration layer) and suggest to store illustration layer data as illustration watermarks. This allows to display selective parts of the illustration layer on demand.

3.1 Basic concept

In our approach, we focus on RGB color images which use 24 bits (8 bits per color channel) for color information display. Since the human visual system is not able to distinguish between colors adjacent in the RGB color space, exploiting the color space in its entirety introduces redundancy which enables watermark storage imperceivable to the naked eye. Illustration watermarking benefits from the imperfect human visual system.

An illustration watermark is a sequence of bits which is encoded by replacing bits of each original pixel's color channels. It is typically associated with a specific region in the cover image. Those regions are defined by an ID-buffer which is a 2D raster graphic whose size corresponds to the cover image size and which is represented by eight bits per pixel. Thus, up to 256 different and disjunct image regions, each with a unique color, can be specified. To maintain the link between a region and its watermark, the watermark insertion is confined to this region. As a result, region and watermark form an entity which can then, for example, be copied to a different location while the embedded watermark is preserved. To identify watermarked regions during data extraction, their outline pixels are clearly marked. In addition, each region has a prominent pixel (*seed point*) which is located inside the region and which stores information concerning the watermarking technique used. Seed point detection during the watermark retrieval is enabled by encoding a certain ID in its color values, by specifically choosing its position relative to the region's outline, and by marking its surrounding pixels.

Figure 1 illustrates the basic concept. Two regions in the original image of a human liver (Figure 1(a)) are coded with additional information. The two regions are defined in the ID-buffer (b). The first data item—associated with the ID-buffer's blue region—is a text annotation (c). The data associated with the ID-buffer's red region is an image that shows a transparently rendered version of the liver which reveals a tumor inside (d). Figure 1(e) shows the watermarked image. As the mouse cursor reaches a watermarked image region, the associated embedded data is displayed (f,g).

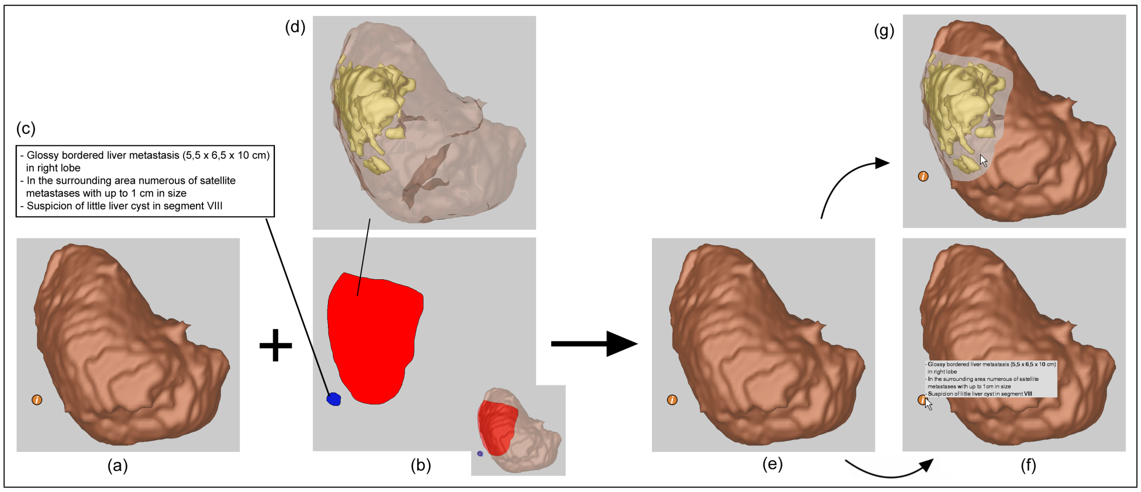

Figure 1: Two types of metadata are inserted into separate regions in the original image of a human liver: text and a part of a second image. After watermark insertion, the encoded information can be displayed by moving the mouse cursor over the corresponding region.

3.2 Traditional LSB approach

The first addressed technique is a simple technique that modifies the least significant bits of a color channel. It does not consider cover image features such as information content or luminance during data encoding. The watermark stream is, hence, inserted in a constant manner, which makes this technique independent from the actual image content.

Watermark stream and its insertion

A watermark stream can include several types of data such as plain text, formatted text, or image data. Formatted text such as speech bubbles can consist of a user-designed geometric shape, words positioned inside the shape, and specific font and color characteristics. As to image data, we limited ourselves to indexed images, which saves storage costs. Indexed images are separated into a color table and color table indices represented as byte pointers. After all data items have been included, the watermark stream is compressed.

During the actual watermarking, the cover image is scanned top down and from left to right to identify those pixels which can be modified. The pixels are selected with respect to the ID-buffer whose differently colored regions are associated with particular watermark streams. For each pixel, at most four bits from each RGB color channel are modified, which depends on the watermark stream size and the space being available. The color channels have varying priorities assigned: the blue channel has the highest priority whereas the green channel has the lowest. These priorities were chosen according to findings about the human visual system which is least sensitive to blue and most sensitive to green when only red, green, and blue color components are considered [Gol96]. To insert, for instance, five bits per pixel, two bits of the blue and red color channels are replaced by watermark bits and one bit of the green channel.

Watermark retrieval

Since the technique is a blind watermarking technique, the encoded data can be retrieved without involvement of the original image. Decoding starts by searching for seed points. Once a seed point has been detected, all remaining pixels of the associated region can be identified. Finally, a region's pixels are traversed to decode the encoded data which is done by reversing the watermarking technique that was applied.

3.3 LSB approach based on entropy and color analysis

The novel watermarking approach proposed in this section encodes data with respect to a *Capacity Map* which is generated by analyzing characteristic image features. This involves an image's information content (entropy) as well as its color intensities. Even though both subtasks entropy and color analysis are not new in the field of digital watermarking (e. g., [vDD02, LC99]), combining and applying them to color images has not been implemented so far. Compared to the traditional LSB approach described in the previous section, this new technique differs in its adaptive data encoding rates which can vary from pixel to pixel depending on the visual information provided by the cover image. With that technique, we aim at improving the imperceptibility of introduced image modifications.

Entropy analysis

The entropy of an image, which is the average number of bits required to represent a sequence of color values, gives hints for the image's information content. High entropy, for example, indicates that an image provides lots of information and has high variance in color tones. It also means that there is a considerable chance that the weaker signals of a watermark stream get masked by the signals transmitted by the cover image.

Illustration watermark insertion is, typically, limited to certain image regions. Thus, the entropy of solely those regions is of interest. But even within those regions, the entropy can vary to a considerable amount. Hence, image regions to be analyzed are confined to pixel blocks of size 8×8.

As described in Section 3.2, watermark insertion can affect at most 4 bits of each color channel. Hence, only the 4 most significant bits can be included in entropy computation which is done separately for each color channel. Algorithm 1 determines which bits can be replaced depending on computed entropies. A capacity value $Cap = 7$, for example, means that at most the first three bits can be replaced. The entropy thresholds were chosen according to the work presented by VAN DROOGENBROECK and DELVAUX [vDD02] and according to informal test results.

Color intensity analysis

The number of bits of a color channel, which can be replaced by watermark bits, significantly decreases when the cover image has a homogeneous texture and when entropy is the parameter which is deciding. But even homogeneous textures, which have the same low levels of entropies, can conceal encoded watermarks differently strong, as Figure 2 illustrates.

Algorithm 1 *Determining watermark bit capacities depending on computed entropies. Note that only the four most significant bits of a color channel serve for entropy computation.*

for all pixel blocks of size 8×8 **do**
    if $entropy > 2$ **then**
        $Cap \Leftarrow 15$
    else if $entropy > 1$ **then**
        $Cap \Leftarrow 7$
    else
        $Cap \Leftarrow 3$
    end if
end for

The figure shows six classes of colors which have at least one color channel highly saturated whereas the remaining channels have a value of 0 assigned. Each of the depicted color squares contains a circle whose color has been slightly altered with respect to the color of its enclosing square. The amount to which a particular color channel was altered is 15.

The figure reveals several interesting aspects, for example:

- Regarding colors red, green, and blue, changing the green color channel in case of green ($Green - \Delta Green$ pair) is more perceivable compared to changing red and blue in cases of $Red - \Delta Red$ and $Blue - \Delta Blue$, respectively.
- For each color class, altering the highly saturated color channel(s) is most perceivable. Thus, red, green, and blue are more suited for watermark insertion than yellow, cyan, and magenta.
- Altering color channels, which are unsaturated (value of 0), has no effect on color perception.

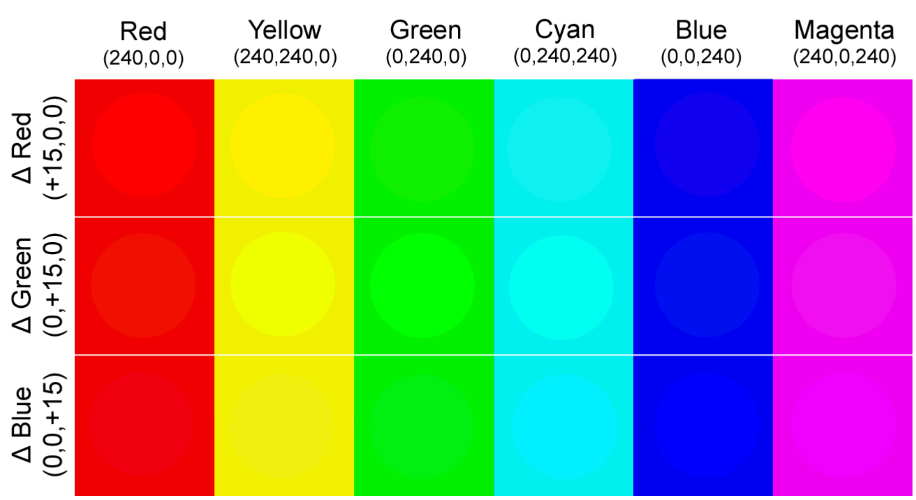

Figure 2: Perception of slight color variations with respect to differently saturated colors. Each square has a circular region whose color was altered by a certain value.

The itemized observations result from perception. However, it is well known that the RGB color space is not qualified for analytically modeling aspects which are in relation to human perception. This is primarily due to the fact that the RGB color space is non-uniform. A color space, on the other hand, which takes human perception into consideration is, for instance, the *CIE L*a*b** color space.

Since watermark insertion proceeds in the RGB color space, RGB and CIE L*a*b* color model must be appropriately combined. The goal in this regard is to find a color $\widehat{C}$ which is perceived similar to the original color C and which is capable of storing a considerable amount of data. For this purpose, two demands should be met (depending on color space):

1. In RGB color space, for each color channel, the distance between C and $\widehat{C}$ should be maximum. This allows for embedding the largest possible amount of data since the distance defines the range within which the value of a color channel can vary during watermark insertion.

2. In CIE L*a*b* color space, the distance between C and $\widehat{C}$ should not exceed a certain predefined value that defines the extent to which differences in perception are acceptable.

The new color $\widehat{C}$ can be found by firstly determining all colors whose color channel values share the 4 most significant bits with those of C. These colors form a class of similar colors according to RGB color space attributes. Altogether, 4096 of such color classes exist, each with 4096 elements. To ascertain similarity in perception, the next step is to compare the colors of a class with each other in the CIE L*a*b* color space. To this end, two colors of a class are converted into the CIE L*a*b* color space and their distance is computed. To find out, for example, whether the four least significant bits of the red color channel can be replaced by watermark bits, all colors of two subclasses are compared: colors of $Subclass\ A$ have their four least significant bits of the red color channel set to 0 whereas colors of $Subclass\ B$ have those bits set to 1. Consequently, both subclasses have 256 elements. All possible color combinations between the subclass elements are then tested. When the distance between two colors exceeds the predefined limit, only less than four bits of the red color channel can be exploited for watermarking, which has to be assured by further testing.

The worst case scenario requires about 16.7 million conversions into CIE L*a*b* color space and comparisons. However, the color classes will never change, which means that the capacity values determined for each class are constant values. They, hence, need to be determined only once and can then be appropriately stored (e. g., as look-up table).

Capacity map generation

The previous paragraphs described two techniques with which capacity values for a selection of pixels in case of entropy analysis and for individual pixels in case of color intensity analysis can be determined. Those capacity values, which serve as guiding values, are computed for each color channel. They can range from 0 to 15, which corresponds to a maximum codable data amount of 4 bits per channel. Capacity values are stored in a capacity map which is a 2D raster graphic of the same size as the cover image whose pixel values represent the capacity values.

To generate the capacity map, the differing capacity values which result from applying the two techniques need to be adequately combined. The goal in this regard is to determine the highest possible amount of data which can be encoded. Algorithm 2 shows how the two techniques were integrated. At first, the entropy of a pixel block is computed, which yields a first capacity suggestion. Thereafter, for each pixel of the block, a second capacity value is determined which results from color intensity analysis. The capacity values recommended by both techniques are then compared to determine the pixel's final suggested capacity.

Algorithm 2 *Determining the capacity Cap_p of a pixel p by combining those capacities suggested by entropy and color intensity analysis.*

for all pixel blocks of size 8×8 **do**
  for each color channel, compute block's entropy and store results in $cap_{entropy}[k]$ with $k = \{red, green, blue\}$
  for all pixels p of the block **do**
    for each color channel, determine pixel's capacity according to perceptual aspects and store results in $cap_{perception}[k]$
    for all color channels k of pixel p **do**
      if $cap_{entropy}[k] > 2 \,\|\, cap_{perception}[k] = 15$ **then**
        $Cap_{p,k} \Leftarrow 15$
      else if $cap_{perception}[k] \geq 7$ **then**
        $Cap_{p,k} \Leftarrow cap_{perception}[k]$
      else if $cap_{entropy}[k] > 1$ **then**
        $Cap_{p,k} \Leftarrow 7$
      else
        $Cap_{p,k} \Leftarrow cap_{perception}[k]$
      end if
    end for
  end for
end for

Figure 3 includes three images: the original, the watermarked, and an image of the capacity map with normalized color values as capacity values. In the capacity image, dark regions indicate low storage capacities whereas bright regions allow for embedding high bit rates.

Figure 3: The original image (left) has been watermarked (center) according to the determined capacity map (right). On average, 7.8 bits were encoded per pixel; in terms of each individual color channel, 2.7 bits (red), 2.66 bits (green), and 2.5 bits (blue) were embedded.

Watermark retrieval

The described technique is a blind method. To retrieve the encoded data, the capacity map must be computed from the watermarked image. Since only the four most significant bits of a color channel, which were not modified during the watermarking procedure, were included in generating the capacity map, it can be ensured that the capacity map computed from the watermarked image is identical to the capacity map computed from the original cover image.

However, the capacity map served only as guideline to suggest potential data encoding rates. The number of bits eventually inserted into the color channels of a pixel was determined depending on the size of the watermark stream to be embedded. Hence, beside those pixels used as region identifiers (compare with Sect. 3.2), four other pixels located at the beginning of a region were exploited to store the size of the watermark stream using a predefined watermarking scheme. Analyzing those reserved pixels enables to detect a watermarked region as well as to decode the amount of encoded data, which together with the capacity map suffices to decode the watermark stream contained in that particular region.

3.4 Approach based on wavelet coefficient modification

Another illustration watermarking approach described in this paper is based on the *Discrete Wavelet Transform (DWT)*. The basic principle behind the DWT is to apply the wavelet transform to a signal (e. g., 2D raster graphic), which results in a transformed representation of that signal. This representation can then be subject to further studies. XIA et al. [XBA98], for example, state that edges in an image are, typically, well-confined to the high frequency subbands which result from DWT appliance. Large coefficients in those subbands, which indicate edges, can thus be exploited for watermark insertion. In the final step, the original representation of the signal can be reconstructed by applying the inverse wavelet transform.

Image decomposition using Haar wavelets

The basic idea behind wavelet transforming a signal is to split up that signal into different time-scale representations, which yields the signal to be represented by linear combinations of wavelets (in this approach *Haar* wavelets).

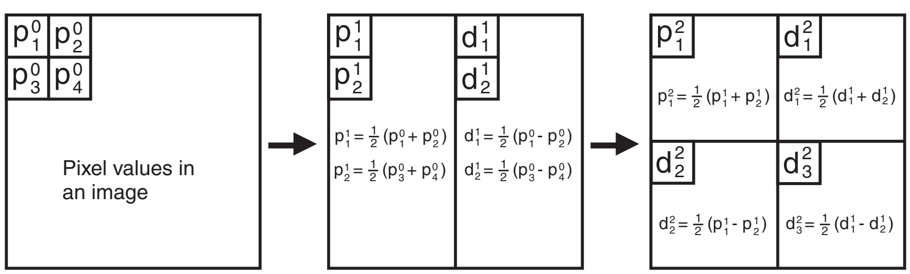

Figure 4: Four pixel values $(p_1^0, p_2^0, p_3^0, p_4^0)$ are transformed line by line (second scheme) and column by column (third scheme) to yield the image part's decomposition.

In case the input signal is a 2D raster graphic represented as a 2D matrix of pixel values, the wavelet transform is applied as a number of successive 1D transformations (for basic details refer to, e. g., [Dau95]). To this end, the 1D wavelet transform can first be applied to each row, which yields a first transformed matrix. Thereafter, the wavelet transform can be applied to each column of the transformed matrix (see Fig. 4 for illustration). As a result, the input image is split up into two equal parts, typically low frequencies (the approximations) and high frequencies (the details). Since the low frequency signal can still contain details a user is interested in, it can be subject to further decompositions. The number of decompositions applied to a signal is referred to as *decomposition levels*.

Figure 5 shows the result of applying a 2-level wavelet decomposition to an image. Here, the low-frequency subband was decomposed a second time. Since edges and regions with varying details can easily be detected in the middle- and high-frequency subbands (HL, LH, HH), these subbands are typically considered when watermarks are to be embedded using a DWT-based approach. To this end, a selection of the values in each subband (called *wavelet coefficients*) are modified.

Figure 5: The original image (left) is wavelet transformed using a 2-level decomposition (middle). The transformation results in low-, middle-, and high-frequency subbands (LL, HL, LH, and HH).

Region-based watermark stream insertion

The watermark stream is encoded by modifying a selection of coefficients in the low- and high-frequency subbands (LH_1, HL_1, and HH_1 in Fig. 5) which result from applying a 1-level wavelet decomposition. The coefficients are selected according to previously specified regions (see Fig. 6(b)), which are those regions associated with metadata, and according to coefficient magnitudes. The proposed technique is similar to the technique discussed by SU et al. [SWK99]. They also embed watermarks by modifying wavelet coefficients with respect to afore selected regions. However, since the authors aim at security aspects, they try to increase the robustness of the encoded watermark by altering solely those coefficients which have specific characteristics (e. g., whose magnitudes are higher than certain thresholds), which restricts them in terms of capacity.

For the approach discussed in this section, watermark insertion proceeds in three steps once image decomposition has been completed:

1. For each subband to be coded, a *Capacity Map* (comparable to the capacity map introduced in Sect. 3.3) is generated which indicates how many bits of a coefficient can be replaced by watermark stream bits. In these maps, all positions of coefficients C_{min}, which correspond to the region to be watermarked, are marked. In addition, among those coefficients, all coefficients C_{max} with a magnitude higher than a specified threshold are specifically marked. These latter coefficients are those which indicate abrupt changes (edges) in the image. The threshold has to be carefully chosen

so that a coefficient does not change its state after watermarking (e. g., from C_{min} to C_{max}). This is done by setting the threshold to the maximum value which can be expressed by those bits to be coded. In case of 4 bits to be watermarked for a coefficient, for example, the threshold would be 15.

Figure 6(c) shows the capacity maps overlayed on their corresponding subband images. Capacity values are provided for each individual coefficient (each pixel in the image has three coefficients, one for each RGB color channel). The brighter a value in the capacity map which is here confined to the region to be watermarked, the more bits can be embedded.

2. For data insertion, the absolute values of coefficients, which are marked in the capacity map, are modified. Embedding data into those coefficients is then equal to embedding data into RGB color channel values by replacing their bits with watermark bits. During the data insertion, coefficients marked as C_{max} are modified stronger than those marked as C_{min}, which yields variations between the amounts of data coded.

3. After watermarking a particular coefficient at a certain position in each of the three subbands, the inverse wavelet transform is emulated to verify that the modified RGB color channel is still valid (i. e., between 0 and 255). If it exceeds its limits, the according original color channel value has to be adapted, i. e., it is increased or reduced accordingly. Thereafter, the corresponding coefficients and capacity map values have to be recalculated and watermarking is repeated.

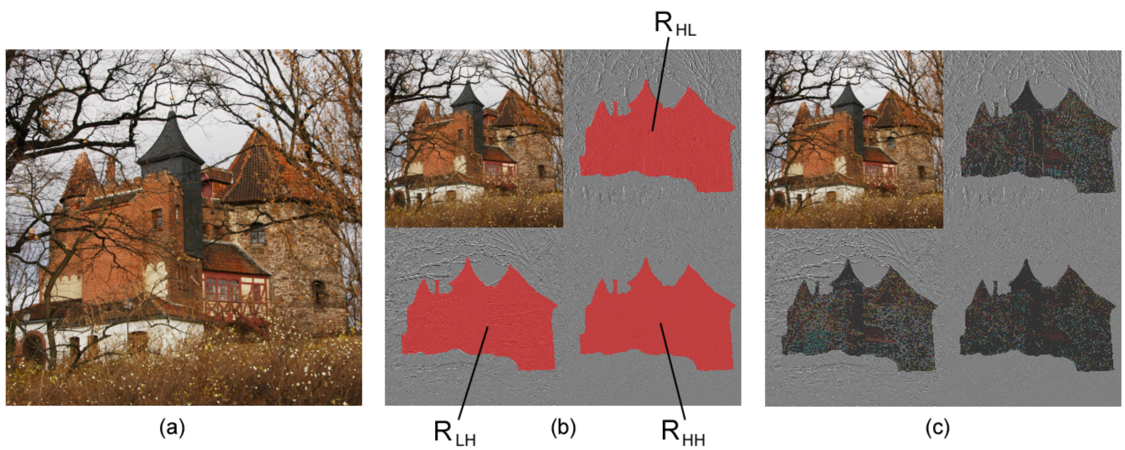

Figure 6: (a) The original image has a certain region which is associated with data to be coded. (b) This region is highlighted in each subband image. (c) Capacity maps overlayed on their corresponding subband images. Dark values indicate lower codable data rates than brighter values within the region.

Figures 7 and 8 show examples with varying data encoding rates. The first image in each figure shows the decomposed image with the corresponding capacity maps overlayed. The second image is the watermarked image. Data encoding was confined to a circular region in the center covering parts of the sky and the trees. Depending on the magnitude of a coefficient, two different bit rates were encoded. The first bit rate $Bits_{min}$ was encoded into all coefficients inside the marked region whose magnitudes were below the specified threshold. The second bit rate $Bits_{max}$ was encoded into the remaining coefficients of a region. Finally, the third image in each figure is a scaled image that illustrates the difference between the original and the watermarked image. Due to the high bit rates ($Bits_{min} = 3$ and $Bits_{max} = 4$ yield an average bit rate of 9.1 bits/pixel), chosen in the watermarked image in Figure 8, the coded region is clearly perceivable.

Figure 7: Left: Decomposed image with capacity maps overlayed. Center: Watermarked image. $Bits_{min} = 1$ and $Bits_{max} = 3$ yield an average encoded bit rate of 2.77 bits/pixel (total number of encoded bits: 130209). Right: Scaled difference between original and watermarked image (the darker a pixel value, the higher the difference).

Figure 8: Left: Decomposed image and capacity maps. Center: Watermarked image. $Bits_{min} = 3$ and $Bits_{max} = 4$ yield an average encoded bit rate of 6.83 bits/pixel (total number of encoded bits: 320448). Right: Difference image.

In Figure 9, the watermark stream was embedded into an image which is computer-generated and which consists of four uniformly colored regions. Modifications introduced during watermarking can thus be detected faster than in images with high entropies. Consequently, already low encoded bit rates are perceivable.

Drawback of this approach

Figure 10(a) shows a 2x2 pixels subregion of the cover image and its four values f_k^0 which represent an arbitrary color channel. Applying the wavelet transform yields the subregion's coefficients (b) whose inverse transformation results in those values depicted in diagram (c). When the coefficients are left unchanged, the

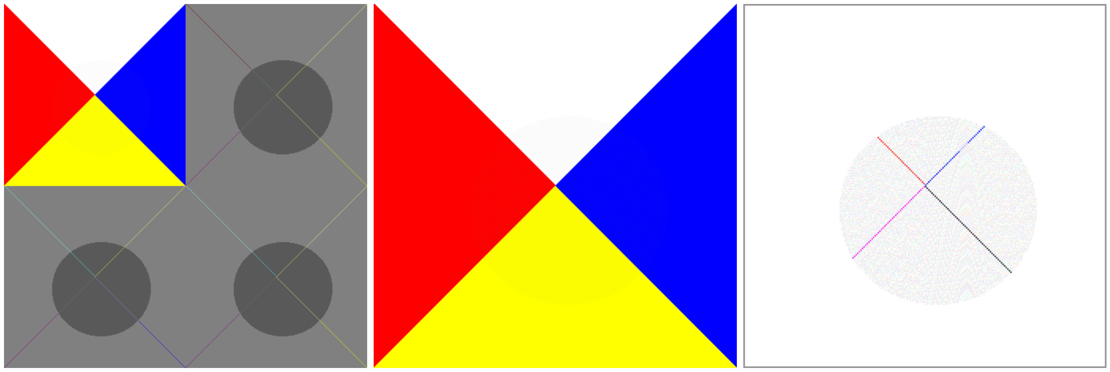

Figure 9: Left: Decomposed image and capacity maps. Center: Watermarked image. $Bits_{min} = 1$ and $Bits_{max} = 3$ yield an average encoded bit rate of 2.29 bits/pixel (total number of encoded bits: 107697). Right: Difference image.

inverse transformation results in the original cover image: $f_k^0 = \widehat{f_k^0}$. Assuming, on the other hand, the coefficients were modified, the inverse transformed image is most likely different from the original image.

In contrast to watermarking techniques that operate in the spatial domain, applying a wavelet domain technique can affect more bits in the cover image than those which were actually modified in the transformed representation. The reason is changes can sum up during the inverse transformation, as the following example illustrates.

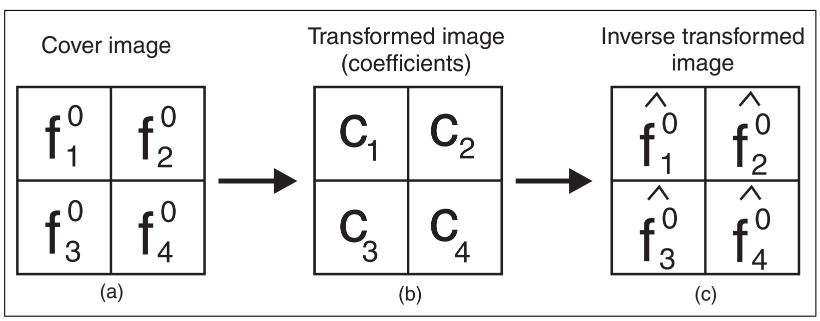

Figure 10: Four color channel values of an image subregion (a) are transformed, which yields the coefficients in (b). These are then inverse transformed (c).

The values $\widehat{f_1^0}$ and $\widehat{f_4^0}$ are determined as follows:

$$\widehat{f_1^0} = (c_1 + c_3) + (c_2 + c_4) \qquad \text{and} \qquad \widehat{f_4^0} = (c_1 - c_3) - (c_2 - c_4) \tag{1}$$

When each middle- and high-frequency coefficient is modified by one of the positive values w_2, w_3, w_4, the maximum differences between f_1^0 and $\widehat{f_1^0}$ and between f_4^0 and $\widehat{f_4^0}$ are:

$$\widehat{f_1^0} \quad = \quad (c_1 + (c_3 + w_3)) + ((c_2 + w_2) + (c_4 + w_4))$$

$$
\begin{aligned}
&= (c_1 + c_2 + c_3 + c_4) + (w_2 + w_3 + w_4) \\
&= f_1^0 + (w_2 + w_3 + w_4) \qquad (2)
\end{aligned}
$$

$$
\begin{aligned}
\widehat{f_4^0} &= (c_1 - (c_3 + w_3)) - ((c_2 + w_2) - (c_4 - w_4)) \\
&= (c_1 - c_2 - c_3 + c_4) - (w_2 + w_3 + w_4) \\
&= f_4^0 - (w_2 + w_3 + w_4) \qquad (3)
\end{aligned}
$$

When a spatial domain technique is used, the bits of a RGB color channel are directly affected. Consequently, the maximum change of a color value correlates with the number of bits modified. In contrast, encoding the same amount of data into a coefficient using a wavelet domain technique can affect the original color value by the data rate encoded into that coefficient plus the data rates encoded into the two remaining coefficients involved, as Equations (2) and (3) show. Hence, encoded data rate and introduced color change do not correlate, which can have a negative effect on the watermark's transparency.

Watermark retrieval

Just as the two other techniques described in the previous sections, the wavelet technique is a blind technique. After decomposing the watermarked image, the coefficients that represent the four subband images are equal to those coefficients the watermarking procedure yielded. The subsequent step is to generate the capacity map. Even though the watermarked coefficients differ from the original (unmodified) coefficients which were used to generate the capacity map during data insertion, both computed capacity maps are equal. The reason is no coefficient changed its state (C_{min} or C_{max}) during data insertion since the threshold was appropriately chosen. Once the capacity map has been generated, the encoded data can be extracted.

4 Evaluation of the Watermarking Techniques

To investigate how much data can be invisibly embedded into a region of an image, we conducted a user study. In addition to the capacity analysis, we wanted to compare the watermarking techniques described in this paper, and we were interested in what effects different types of images would have on perception.

Hypothesis

During the study, participants were shown three types of images varying in their information contents (entropy levels). Beside their differing characteristics, the displayed images differed in their amounts of encoded data and the method which was used for data encoding. We assumed that the image with the highest level of entropy would conceal encoded data most. We also assumed that with an increasing amount of encoded data, changes would be detected easiest. The third assumption we made was that the technique described in Section 3.3 would conceal encoded data most effectively.

4.1 Description of the Study

Participants and experimental setup

112 voluntary participants (36 female, 76 male), aged between 19 and 52 years, took part in the study. The study was conducted in a research lab at the local computer science department.

During the study, a standard PC (MS WINDOWS XP) with a high-resolution flat panel monitor (IBM T221: 22.2 inch viewable diagonal image size, 3840 x 2400 maximum resolution, 204 ppi) was employed. A standard computer mouse served for interaction. The software used was implemented in *PureBasic*. It basically asked for personal data, displayed an image for a particular time, and registered and logged mouse events triggered by the participant.

Though there are various parameters which may have influence on a participant's viewing conditions, it was tried to minimize parameter disparities between participants. For example, each participant performed the task at the same workstation with the same monitor. Also, the room was darkened to avoid differing lighting conditions and reflections on the display.

Watermarking techniques

The three analyzed watermarking techniques were described in the previous sections. The basic *LSB* technique (see Sect. 3.2) was included because it can be used to embed considerable data amounts, which is an important feature with respect to illustration watermarking. However, since it does not consider aspects of human perception, improving this technique may yield better results. The adapted technique *LSB_Ext* (see Sect. 3.3) incorporates human perception characteristics during watermarking. It is therefore expected that the *LSB_Ext* technique will outperform the basic *LSB* technique.

The *Wave* technique (see Sect. 3.4) serves as example of those techniques that insert watermark data into the transformed representation of a cover image. In doing so, these techniques mainly aim at preventing watermarks from being damaged when image operations are applied. High capacity is, in contrast to what illustration watermarking aims at, only a secondary aspect. How well the *Wave* technique will perform in comparison with the two other techniques will be clarified by the study.

Selection of Images

The three images in Figure 11 are the original images whose watermarked copies were presented during the study. These images will be referred to as *Pattern* image, *Landscape* image, and *Flowers* image.

The *Pattern* image is computer-generated and consists of stripes of alternating colors. In contrast to the *Pattern* image, the *Landscape* image is a photograph taken with a digital SLR camera (NIKON D100). Parts of it such as the sky and green plants can be found in many photographs. The *Flowers* image is also a photograph taken with the digital camera. It features high diversification in information content and includes almost no regions of same colors.

The *Pattern* image was chosen based on the assumption that image modifications would be easiest detectable when the image contains large regions of same colors. These colors were selected with respect to the observations discussed in Section 3.3. Regarding their data hiding qualities, the colors—white (RGB: 255,255,255) and green (RGB: 0,255,0)—allow for encoding differing amounts of data. Due to its two unsaturated color channels red and blue, considerably more data can be inserted into the green color than into white.

The two photographs differ in their entropies. The *Landscape* image can be divided in two regions, a sky region which has an almost smooth texture and a ground region which is high in contrast. This image is particularly well suited for evaluating the *LSB_Ext* technique and the *Wave* technique since these techniques consider image characteristics during data encoding provided by their capacity maps. These maps suggest to encode lower data rates into the sky region than into the ground region. The *Flowers* image will, presumably, allow to encode the highest amount of data. The reason is its high-contrast texture which can conceal slight modifications introduced during the watermarking.

Figure 11: The original uncoded images which were selected for evaluation: *Pattern* image (left), *Landscape* image (middle), and *Flowers* image (right).

Watermark Insertion

A specific amount of data (capacity level) was embedded into each image which was shown to a participant during the evaluation. The watermark stream encoded was a random byte stream whose length correlated with the data amount to be inserted.

Data encoding was limited to one circular region in each image. These regions were selected randomly in case of the *Flowers* image. For the two other image types, the regions were selected almost randomly, which means the regions were chosen so that they covered a borderline in the image. The borderline divides sky and ground in the *Landscape* image, and green and white in the *Pattern* image. All regions had the same size. Region locations differed between image types as well as between capacity levels. They did not differ between images of the same type which had the same data amount encoded using one of the techniques.

The amount of data which was inserted into a pixel ranged from 3 bits to 12 bits (capacity levels). In case of capacity map use, varying capacity levels could be suggested for the pixels of a region. The capacity level for a pixel was hence chosen in a way that the averaged capacity level for all pixels approximately corresponded to the level demanded.

The three variables—image type, capacity level, and watermarking technique—yield 90 different images to be evaluated. However, to prevent participants from getting tired of viewing all these images, the number of images was reduced. Images with clearly visible artifacts caused by watermarking were omitted as well as images with no degradation detectable even though the original image could be employed for comparison and the location of the watermarked region was known. In this way, the number of images to be presented to a participant could be reduced to 54 images. Figures 12–14 show for each technique those images included in the study which were watermarked with the lowest and highest capacity levels.

4.2 Task description

At the beginning, participants were asked to answer a pre-study questionnaire including questions about their gender, age, and visual impairments. Thereafter, the task was explained and demonstrated with an example. Each participant was told that the images to be shown would have a circular region which had been modified to a certain degree. Finally, the actual task, the completing of which took about 12 minutes, started.

The basic task was to analyze watermarked images and to find the regions into which a certain amount

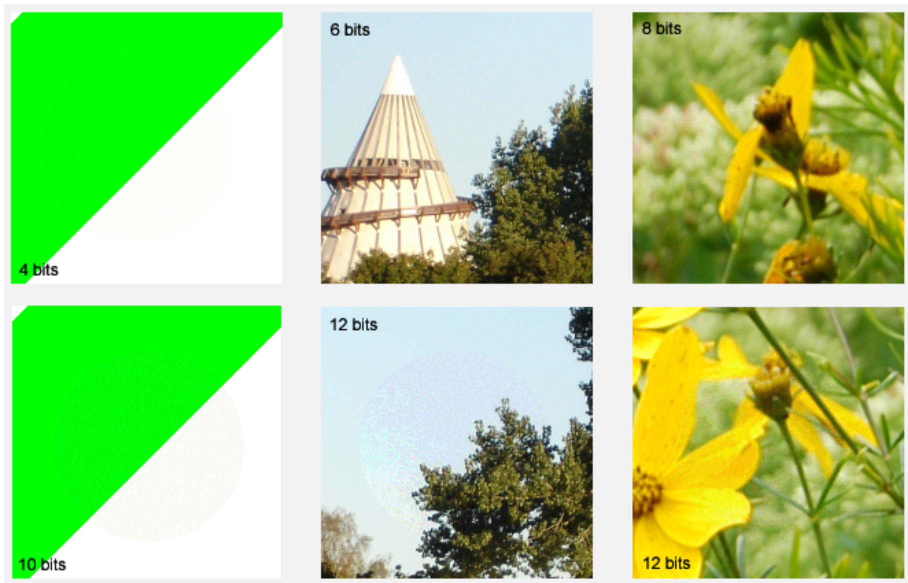

Figure 12: Image parts which were watermarked using the *LSB* technique. The upper row shows evaluated images which have the lowest amount of data encoded, the lower row shows images with the highest data amount encoded. The circular region, into which the watermark stream was inserted, is located in the center of an image.

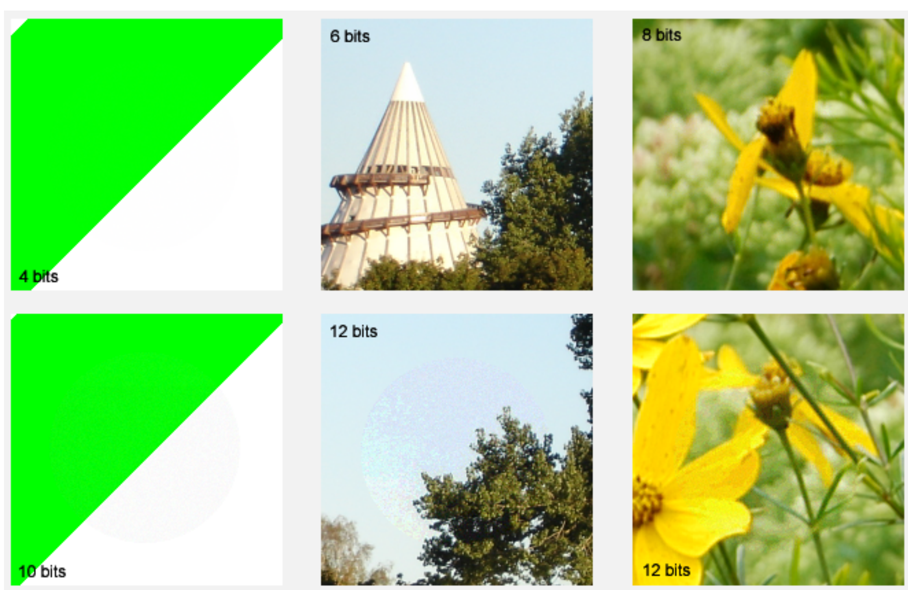

Figure 13: Image parts which were watermarked using the *LSB_Ext* technique.

Figure 14: Image parts which were watermarked using the *Wave* technique.

of data was encoded. To this end, 54 images were successively presented to the participant. Each image was shown for at most 10 seconds. During this time, the participant should find the modified region and select it with the mouse cursor. Once the mouse button was pressed, the next image appeared, regardless of whether or not a region was selected correctly. A region was considered to be hit when the mouse click happened while the cursor position was inside a watermarked region. If no region could be detected within the 10 seconds, which means no mouse click occurred, the next image appeared automatically.

Reasons for design decisions

During performing the task, the participants were asked to pay their complete attention to analyzing the images. It was hence taken heed that a participant was neither distracted by an incident nor by a person. Also, a participant's concentration was not necessarily constant throughout the task duration. To compensate for this inhomogeneity, the sequence in which images were shown was random and varied between participants.

A former study had shown that setting no time limits caused participants to spend different times on analyzing images presented. Consequently, task completing times differed noticeably between participants, which was no acceptable basis for the study analysis. In this study, the maximum time for which each image was shown was hence limited to 10 seconds. During pre-tests, this time span emerged to be appropriate to detect perceivably modified image regions. However, task completing time was also limited to emulate a more realistic scenario in which a user typically does not spend much time on searching for image artifacts.

Another aspect which the former study revealed to be disadvantageous was that, though the image sequence was random, showing same image types consecutively was not avoided. Differences between those images could thus be easily detected. This aspect was prevented in this study. In addition, a black image was shown for a moment between two images.

4.3 Study Results

During the study, a log file recorded data such as personal data (age, gender, colorblindness), image ID and its capacity level, region detection, and time passed until selection. Analyzing this data revealed information about technique efficiency, image type influence, capacity level limits, etc.

The diagram in Figure 15 shows for each image type the total number of correctly selected regions depending on data amounts encoded. The techniques which had been employed for watermark insertion are not distinguished in this diagram. Consequently, 336 correct selections (112 participants and 3 techniques) is the maximum which can be achieved for a capacity level. The *Pattern* image is obviously the type of image which concealed embedded data worst, whereas the texture of the *Flowers* image masked introduced changes most effectively. It even caused participants to not detect 118 watermarked regions in case of 12 encoded bits per pixel. These observations correlate with the entropies of the images: the higher an image's entropy, the more data can be embedded ($H_{Pattern} = 1$, $H_{Flowers} = 7.39$). In case of the *Landscape* image, two regions have to be differentiated: the sky region ($H_{Sky} \approx 3.22$) and the ground region ($H_{Ground} \approx 7.23$). Since both regions were involved in each watermark insertion, the *Landscape* image concealed encoded data worse than the *Flowers* image.

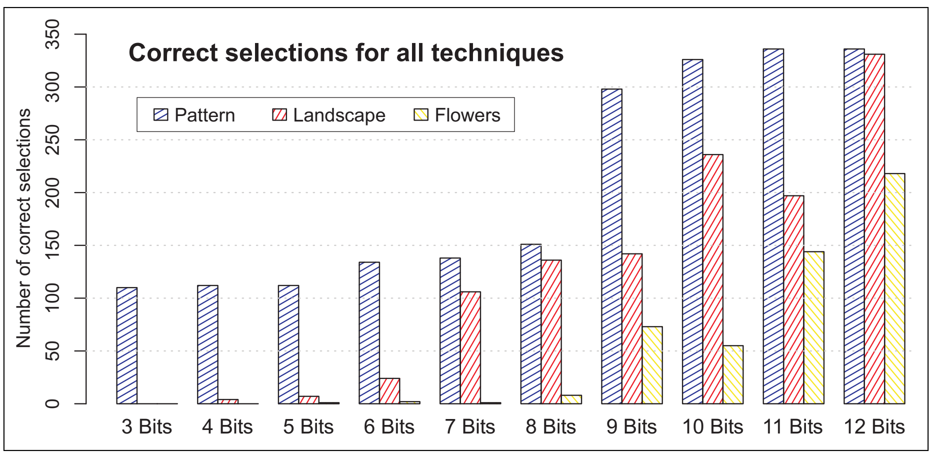

Figure 15: Correctly selected regions separated according to type of image and capacity level.

The diagrams in Figure 16 allow to draw conclusions from analyzing the effects of watermarking technique, image type, and capacity level on correct region detection. Each diagram shows that the *Wave* technique is associated with most correct selections, which indicates that this technique is most ineffective. The diagrams also show that, in most cases, the *LSB_Ext* technique performs better than the standard *LSB* technique. Regarding the effects of technique and image type on region detection, *One-Way ANOVA (Analysis of variance)* yields that both variables have a similar influence: $F_{2,6,tech} = 2.453, p = 0.167$ (technique type) and $F_{2,6,img} = 2.655, p = 0.149$ (image type).

A selection of times which passed until participants selected a region is shown in Figure 17. These times can give hints about the degree to which a particular watermarked region was visible: a low time value, for example, can indicate that a watermarked region was easily detectable. The diagram only considers the *LSB* and *LSB_Ext* technique since these techniques performed better than the *Wave* technique. The *Pattern* image was chosen because it has the lowest entropy compared to the other images so that image texture was less influencing. What can be gathered from the diagram is that elapsed times decreased with increasing

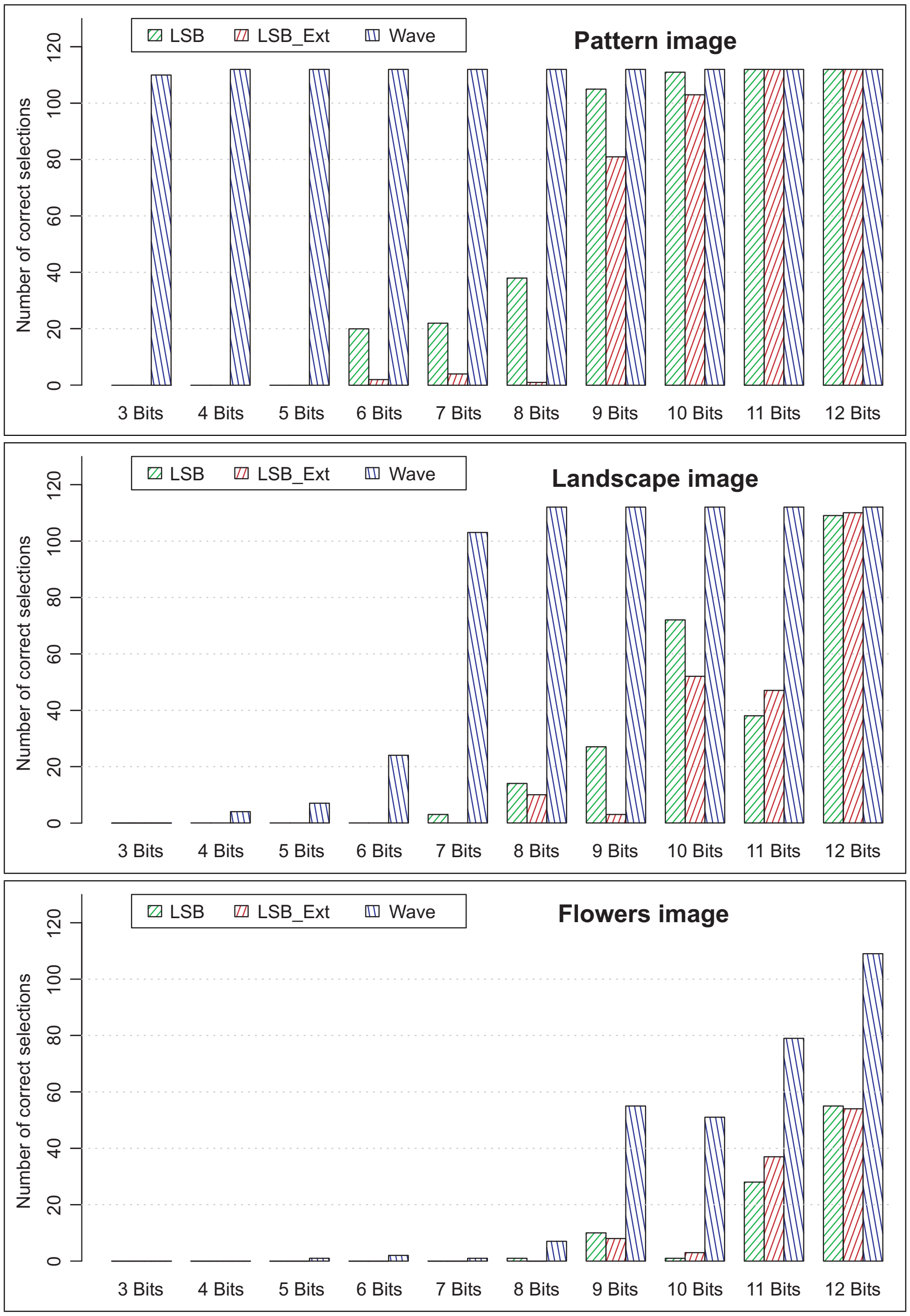

Figure 16: Numbers of correctly selected regions for each image type.

capacity levels and that, for each capacity level, participants needed more time to make a selection when the *LSB_Ext* technique was used. The mean times for both techniques and the four selected capacity levels confirm this observation: $Mean_{LSB} = 5.78\ sec$ and $Mean_{LSB\_Ext} = 7.19\ sec$.

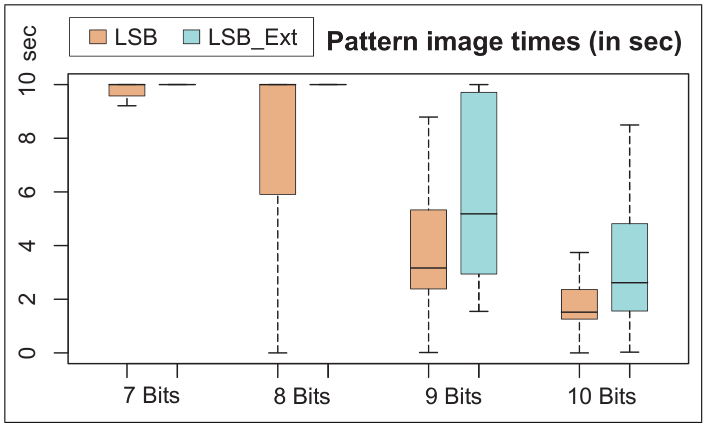

Figure 17: Selective elapsed times for the *LSB* and *LSB_Ext* technique. The boxes provide information about the lower and upper quartiles as well as the medians.

4.4 Discussion

The study has clearly shown that the *Wave* technique is inapplicable as illustration watermarking technique. This can basically be attributed to pixel modifications which can cumulate during coefficient watermarking (discussed in Sect. 3.4). The study has also shown that, even though both the *LSB* and the *LSB_Ext* technique can be employed to embed respectable data rates, the *LSB_Ext* technique outperformed the basic *LSB* method.

One of the primary questions which should be answered by the study was how many bits can be imperceivably embedded into a pixel. Since the *LSB_Ext* technique was the technique that performed best, its results are analyzed to suggest potential capacity limits. The first diagram in Figure 16 shows the number of correct selections participants made for the *Pattern* image which has the most homogeneous texture and which, hence, is best qualified to evaluate artifact visibility. The diagram reveals that at least 5 bits can be inserted per pixel. But even up to 8 bits would be possible since only 1 participant detected the region with this capacity level encoded (results for lower capacity levels: 6 bits: 2 participants, 7 bits: 4 participants).

There are two reasons which argue for raising the capacity limit. At first, illustration watermarking is a technique that aims at embedding descriptive metadata into media which, typically, provide more information than the *Pattern* image does. Real-world images such as the *Landscape* or the *Flowers* image are, hence, rather addressed. They allow for embedding data rates of 8 or 9 bits per pixel. The second aspect is that the participants knew about modified regions they should search for. They were thus highly motivated to find those regions. A viewer, on the other hand, who simply looks at an image and who does not search for artifacts, would presumably be less likely to detect modified regions.

5 Application Scenarios

We see several application scenarios for illustration watermarking. Foremost of these are images on Web pages whose values can be increased by linking them with auxiliary information which semantically correlates with them. So far, those images can either have such information “burnt into” them or they can dynamically change their contents using appropriate software such as JAVASCRIPT or MACROMEDIA FLASH. The advantage of images, on the other hand, that have illustration watermarks embedded is that they are

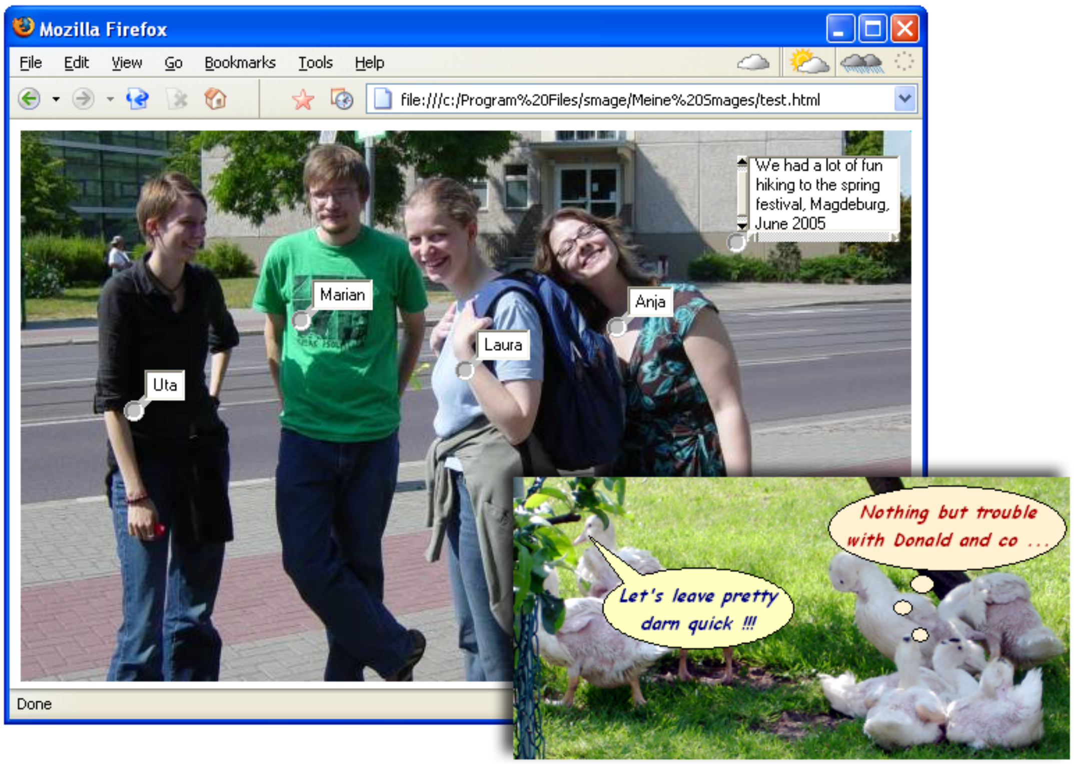

Figure 18: Images with illustration watermarks embedded. The encoded data (simple text annotations and speech balloons) is displayed when the mouse cursor reaches a watermarked region.

still standard raster graphics which can be stored and distributed while the encoded information remains attached, and they can be displayed with any standard Web browser. To explore the encoded data, however, a Web browser plugin is required, as well. Figure 18 shows a browser plugin in action that we have implemented to demonstrate the technique. This and further examples in this section were generated using *Smage*, a software tool which can be downloaded at *http://www.smage.de/*.

When our technique is applied to images, they are turned into images with dynamic contents. In images, for example, that are typically augmented with textual annotations (e. g., comics), the text can be embedded as watermark and displayed only when it is demanded by the user. We have implemented several ways to display embedded text objects being composed of the words and a layout (position, shape, colors, font, etc.). This way, objects such as scrollable text boxes or various kinds of text balloons can be created, embedded, and displayed (see Figure 18).

Images enriched with textual or other information, in general, are useful in many situations. For instance, the illustration in Figure 1 shows the image of an opaque liver. Employing the illustration watermarking technique allows to combine this image with a second image that provides a different view of the liver. Since the resulting image is still a standard raster graphic, one of these two images (the opaque liver as the cover image) can still be displayed using a standard image viewer. This can be useful when image previews are generated, for example, by the WINDOWS EXPLORER. Another example is shown in Figure 19. Here, a map has been encoded to indicate how the place depicted in the photograph can be found.

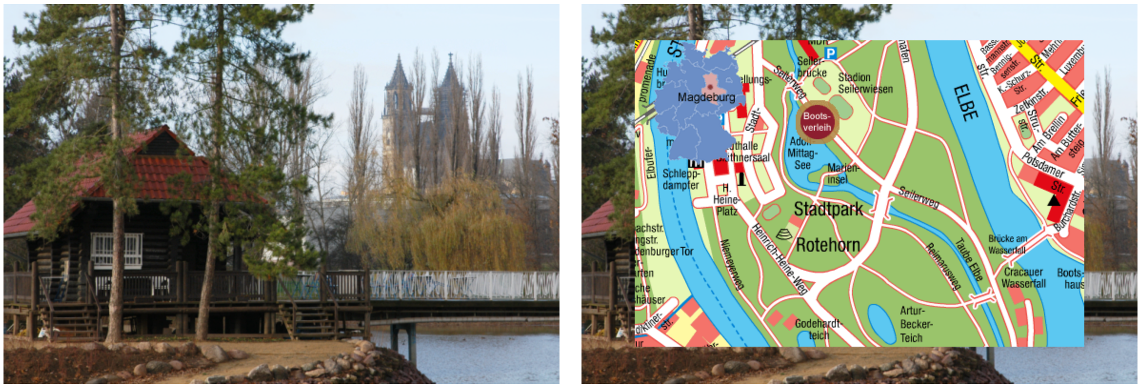

Figure 19: Using illustration watermarks to describe the location that is depicted in the image.

Figure 20: Example that shows annotated objects and their names in German and English.

Another application we envision are interactive pictorial dictionaries. Such a dictionary allows the content author to define objects in the image and to enter their descriptions in a given language. Using a built-in dictionary, these terms can be automatically translated into the languages provided by the system. Thus, when displaying images that contain dictionary content, viewers can switch between object descriptions in different languages (see Figure 20).

6 Summary

In this paper, we discussed three watermarking techniques which can be used to augment a 2D raster graphic with information that adds value to the graphic. These so-called *Illustration Watermarking* techniques insert the watermark data directly into those image regions the data is associated with. The result is that image region and its encoded information become an information unit which allows for end-user interaction, i. e., the encoded information can be displayed and explored at will. The discussed techniques operate in the spatial as well as in the transformed domain. The first technique is a traditional LSB approach which was adapted for our purposes. The second technique is a novel LSB approach since it analyzes several image features to generate a capacity map which is then used for data insertion. The third technique, which also makes use of a generated capacity map, inserts the watermark data in the wavelet transformed representation of the cover image.

In the course of a user study, we compared the three watermarking techniques and analyzed the effects when different types of images are involved. Since we were further interested in the maximum amount of data which can be invisibly embedded per pixel, we varied the encoded data rates. Evaluating the study results revealed that our improved LSB approach performed better than the traditional LSB and the wavelet domain approach. The study also confirmed our expectations concerning the high effect of an image's information content (its entropy) on the transparency of encoded watermark data. As to encodable data rates, the study revealed that 5 bits/pixel can be invisibly encoded into images with uniform textures. When the information content increases, the encodable data rate can increase, as well (8 bits/pixel and more is feasible).

Finally, we proposed a variety of application scenarios for illustration watermarking. For this purpose, we implemented software that enables a user to create explorable images and that allows to integrate those images in personal Web pages so that they can be provided to the public. Potential application scenarios, in this regard, are arbitrary digital photographs which can benefit from assigned comments and descriptions or pictorial dictionaries that can facilitate the study of foreign languages.

References

[Ala00] A. M. Alattar. *Smart Images Using Digimarc's Watermarking Technology*. Proc. of IS&T/SPIE's 12th Int. Symp. on Electronic Imaging, 3971(25):264–273, 2000.

[BBRP99] M. Barni, F. Bartolini, A. De Rosa and A. Piva. *Capacity of the Watermark-Channel: How Many Bits Can Be Hidden Within a Digital Image*. Proc. of SPIE, Security and Watermarking of Multimedia Contents, 3657:437–448, 1999.

[BGML96] W. Bender, D. Gruhl, N. Morimoto and A. Lu. *Techniques for Data Hiding*. IBM Systems Journal, 35(3 & 4):313–336, 1996.

[BKM$^+$02] N. V. Boulgouris, I. Kompatsiaris, V. Mezaris, D. Simitopoulos and M. G. Strintzis. *Segmentation and Content-Based Watermarking for Color Image and Image Region Indexing and Retrieval*. EURASIP Journal on Applied Signal Processing, 4:420–433, 2002.

[CKLS97] I. J. Cox, J. Kilian, T. Leighton and T. Shamoon. *Secure Spread Spectrum Watermarking for Multimedia*. IEEE Trans. on Image Processing, 6(12):1673–1687, 1997.

[CMB01] I. J. Cox, M. L. Miller and J. A. Bloom. *Digital Watermarking: Principles & Practice*. Morgan Kaufmann Publishers, 2001.

[Dau95] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1995.

[DC98] M. Dymetman and M. Copperman. *Intelligent Paper*. In: Proc. of the 7th Int. Conf. on Electronic Publishing, pp. 392–406, 1998.

[Gol96] E. B. Goldstein. *Sensation and Perception*. Brooks/Cole Publishing Company, 1996.

[Hec01] D. L. Hecht. *Printed Embedded Data Graphical User Interfaces*. IEEE Computer, 34(3):47–55, 2001.

[JJ98] N. F. Johnson and S. Jajodia. *Exploring Steganography: Seeing the Unseen*. IEEE Computer, 31(2):26–34, 1998.

[KM92] C. Kurak and J. McHugh. *A Cautionary Note on Image Downgrading*. In: Proc. of the Annual Computer Security Applications Conference, pp. 153–159, 1992.

[KZ95] E. Koch and J. Zhao. *Towards Robust and Hidden Image Copyright Labeling*. In: Proc. of 1995 IEEE Workshop on Nonlinear Signal and Image Processing, pp. 452–455, 1995.

[LC99] W.-N. Lie and L.-C. Chang. *Data Hiding in Images with Adaptive Numbers of Least Significant Bits based on the Human Visual System*. In: Proc. of the Int. Conf. on Image Processing (ICIP'99), volume 1, pp. 286–290, 1999.

[NP01] A. Nikolaidis and I. Pitas. *Region-Based Image Watermarking*. IEEE Trans. on Image Processing, 10(11):1726–1740, 2001.

[NS03] M. C. Norrie and B. Signer. *Web-Based Integration of Printed and Digital Information*. In: Proc. of 2nd Workshop on Data Integration over the Web (DiWeb'02), volume 2590 of Lecture Notes in Computer Science, pp. 200–219. Springer-Verlag, 2003.

[PAK99] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn. *Information Hiding—A Survey*. Proceedings of the IEEE, 87(7):1062–1078, 1999.

[RVEP03] Y. Rytsar, S. Voloshynovskiy, F. Ehrler and T. Pun. *Interactive Segmentation with Hidden Object-Based Annotations: Toward Smart Media*. In: Proc. of the SPIE, Storage and Retrieval Methods and Applications for Multimedia, volume 5307, pp. 29–37, 2003.

[SIDS03] H. Sonnet, T. Isenberg, J. Dittmann and T. Strothotte. *Illustration Watermarks for Vector Graphics*. In: Proc. of Pacific Graphics 2003, pp. 73–82. IEEE Computer Society, 2003.

[SKT98] M. D. Swanson, M. Kobayashi and A. H. Tewfik. *Multimedia Data-Embedding and Watermarking Technologies*. Proceedings of the IEEE, 86(6):1064–1087, 1998.

[SWK99] P.-C. Su, H.-J. Wang and C.-C. J. Kuo. *Digital Image Watermarking in Regions of Interest*. In: Proc. of the Conf. on Image Processing, Image Quality and Image Capture Systems (PICS 1999), pp. 295–300, 1999.

[TRvS+93] A. Z. Tirkel, G. A. Rankin, R. M. van Schyndel, W. J. Ho, N. R. A. Mee and C. F. Osborne. *Electronic Water Mark*. In: Proc. of the 2nd Conf. on Digital Imaging, Computing Techniques, and Applications (DICTA-93), pp. 666–672, 1993.

[vDD02] M. van Droogenbroeck and J. Delvaux. *An Entropy Based Technique for Information Embedding in Images*. In: Proc. of 3rd IEEE Signal Processing Symposium (SPS-2002), pp. 81–84, 2002.

[XBA98] X. G. Xia, C. G. Boncelet and G. R. Arce. *Wavelet Transform Based Watermark for Digital Images*. Optics Express, 3(12):497–511, 1998.