Digital image Watermarking

by Gleb V. Tcheslavski: gleb@ee.lamar.edu
http://ee.lamar.edu/gleb/dip/index.htm

Why watermark images?

Image compression makes the distribution of digital images and video materials over the Internet and on media practical. Such digital materials can be copied and redistributed uncontrollably. Digital image watermarking – the process of inserting data into an image – can be used to protect the rights of their owners in a variety of ways:

1) **Copyright identification** – provide a proof of ownership;
2) **User identification** (fingerprinting) – encode identity of legal users to encode sources of illegal copies;
3) **Authenticity determination** – if the watermark will be destroyed by modification in an image, its presence quarantines authenticity;
4) **Automated monitoring** – monitor when and where images are used (for royalty collection or the location of illegal uses);
5) **Copy protection** – they can specify rules of image usage and copying (for instance, for DVD players only…)

Spring 2009 ELEN 4304/5365 DIP
Visible watermarks

A Visible watermark is an opaque or semi-transparent sub-image or image that is placed on top of another image (that is watermarked) so that is obvious to the viewer. An example: a logo placed by TV networks. Typically, performed in the spatial domain.

Denoting the original image as $f$, the watermark as $w$, and the watermarked image as $f_w$

$$f_w = (1 - \alpha) f + \alpha w$$

Visible watermarks

The constant $\alpha$ controls the relative visibility of the watermark. If $\alpha = 1$, the watermark is opaque and the underlying image is completely obscured. As $\alpha$ approaches 0, more of the underlying image and less of the watermark is seen. In general:

$$0 < \alpha \leq 1$$

In the previous example, $\alpha = 0.3$. the last image is the scaled by 128 difference between the watermarked image and the original one.

The underlying image is clearly visible through the “semi-transparent” watermark.
Invisible watermarks

Invisible watermarks cannot be seen with the naked eye but they can be recovered with an appropriate decoding algorithm. The invisibility is assured by inserting them as visually redundant information (something that human visual system does not perceive): watermarked image after high quality JPEG compression and the extracted watermark…

In the example, the watermark image was inserted in two least significant bits according to:

\[ f_w = 4 \left( \frac{f}{4} \right) + \frac{w}{64} \]

Note that – assuming the unsigned integer arithmetic –
1) dividing and multiplying by 4 sets the two least significant bits of \( f \) to 0;
2) dividing \( w \) by 64 shifts its two most significant bits into the two least significant bit positions;
3) adding two results generates the \( LSB \) watermarked image.
Invisible watermarks

The watermark can be recovered by zeroing 6 most significant bits of the image and scaling the remaining values to the full intensity range.

An important property of invisible watermarks is their resistance to attempts (both intentional and accidental) to remove them... This type of watermarks is referred to as fragile invisible watermarks.

If the watermarked image is compressed and then decompressed using lossy JPEG, the watermark is destroyed. As shown in the last figure, although the visual information was preserved, the watermark was unusable. This can be used for image authentication.

Watermarking techniques

Robust invisible watermarks must survive image modification (attacks) including image compression, linear or non-linear filtering, cropping, resampling, rotation, printing/rescanning, adding noise, etc.

Typical watermarking system:

Encoder

Decoder
Watermarking techniques

The encoder inserts watermark $w_i$ into image $f_i$, producing watermarked image $f_{w_i}$.

The decoder extracts and validates the presence of $w_i$ in watermarked input $f_{w_i}$ or unmarked input $f_i$. If the watermark $w_i$ is visible, the decoder is not needed. Otherwise, the decoder may or may not require a copy of $f_i$ and $w_i$ to do its job. If $f_i$ and/or $w_i$ are used, the watermarking system is called a private (restricted-key) system; otherwise, the system is public or unrestricted-key system.

Since the decoder must process both marked and unmarked images, $w_0$ is used to denote the absence of a mark. Finally, the decoder needs to correlate the extracted watermark $w_j$ with $w_i$ and compare the result to a predefined threshold that sets the degree of similarity accepted as a match.

Watermarking techniques

*Watermark insertion* and *extraction* can be performed in spatial domain as shown previously, or in the transform domain.

Two watermarked images computed using DCT and the intensity-scaled differences between the original and watermarked images.
Watermarking techniques

The DCT-based watermarking approach:
1. Compute the 2D DCT of the image to be watermarked;
2. Locate its $K$ largest coefficients $c_1, c_2, \ldots, c_k$ by magnitude;
3. Create a watermark (for instance, by generating a $K$-element pseudo-random sequence of numbers $w_1, \ldots, w_k$ with zero mean and unit variance);
4. Embed the watermark into the $K$ largest DCT coefficients as
   \[ c_i' = c_i \cdot (1 + \alpha w_i) \quad 1 \leq i \leq K \]
   The constant $\alpha > 0$ controls the extent to which $w_i$ alters the coefficients.
5. Replace $c_i$ by $c_i'$ and compute IDCT.

Watermarking techniques

Spreading watermarks across an image’s perceptually significant frequency components, $\alpha$ can be made small to reduce watermark visibility. At the same time, watermark security is high since:
1. Watermarks are composed of pseudorandom numbers with no obvious structure,
2. Watermarks are embedded in multiple frequency components with spatial impact over the entire 2D image (their location is not obvious),
3. Attacks against them tend to degrade image as well since the image’s most important frequency components must be altered to affect the watermarks.
Watermarking techniques

To determine whether a particular image is a copy of an image previously watermarked with watermark \( w_1, \ldots, w_k \) and with DCT coefficients \( c_1', \ldots, c_k' \), the following procedure can be used:

1. Compute the 2D DCT of the image in question.

2. Extract the \( k \) DCT coefficients in the positions corresponding to \( c_1', \ldots, c_k' \) in the watermarking procedure and denote the coefficients as \( \hat{c}_1, \ldots, \hat{c}_k \).
   - If the image in question is the previously watermarked (and not modified) image, then \( \hat{c}_i = c_i' \).
   - If the image in question is a copy of a previously watermarked and modified image, then the equality will be approximate.
   - If the image was not watermarked or was watermarked with another watermark, the coefficients will be completely different.

3. Compute watermark as
   \[
   \hat{w}_i = \hat{c}_i - c_i \quad 1 \leq i \leq K
   \]

4. Measure the similarity of \( \hat{w}_1, \ldots, \hat{w}_k \) and \( w_1, \ldots, w_k \) form the watermarking procedure using some metric such as the correlation coefficient as specified below:
   \[
   \gamma = \frac{\sum_{i=1}^{k} (\hat{w}_i - \bar{\hat{w}})(w_i - \bar{w})}{\sqrt{\sum_{i=1}^{k} (\hat{w}_i - \bar{\hat{w}})^2 \cdot \sum_{i=1}^{k} (w_i - \bar{w})^2}} \quad 1 \leq i \leq k
   \]
   where bars indicate means of the \( k \)-element watermarks.
Watermarking techniques

5. Compare the measured similarity to a predefined threshold $T$ and make a binary detection decision

$$D = \begin{cases} 1 & \text{if } \gamma \geq T \quad \text{– watermark is present} \\ 0 & \text{otherwise } \text{– no watermark} \end{cases}$$

Using this procedure, the watermarked “Lena” image (seen in Slide 10) – measured against itself – yields a correlation coefficient of 0.9999, which is a definite match.

The correlation coefficient computed between two images in slide 10 was 0.0417; therefore, we conclude that these two images do not contain the same watermark.

Watermarking techniques

The DCT-based watermarking approach is fairly resistant to the attacks.

- lossy JPEG, $\text{rms} = 7$
- lossy JPEG, $\text{rms} = 10$

- Smoothing by spatial filtering
- add Gaussian noise
- Histogram-equalized
- rotation