# A Video Watermarking System for Broadcast Monitoring

Ton Kalker, Geert Depovere, Jaap Haitsma and Maurice Maes

Philips Research, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands

## ABSTRACT

This paper presents a video watermarking technology for broadcast monitoring. The technology has been developed at the Philips Research Laboratories in Eindhoven in the context of the European ESPRIT project VIVA (Visual Identity Verification Auditor). The aim of the VIVA project is to investigate and demonstrate a professional broadcast surveillance system. The key technology in the VIVA project is a new video watermarking technique by the name of JAWS (Just Another Watermarking System). The JAWS system has been developed such that the embedded watermarks (i) are invisible, (ii) are robust with respect to all common processing steps in the broadcast transmission chain, (iii) have a very low probability of false alarms, (iv) have a large payload at high rate, and (v) allow for a low complexity and a real-time detection. In this paper we present the basic ingredients of the JAWS technology. We also briefly discuss the performance of JAWS with respect to the requirements of broadcast monitoring.

**Keywords:** Multimedia, Watermarking, IPR protection, DSP, Broadcast monitoring

## 1. INTRODUCTION

Globally the distribution of TV products is worth many billions of dollars. News items, such as those distributed by companies such as Reuters, CNN and Associated Press, have a value of over 100,000 USD per hour, which make them very vulnerable to Intellectual Property Rights (IPR) violation. The same is true for entertainment programs and TV dramas. In the European ESPRIT project VIVA the use of digital watermarks is investigated to protect IPR of the content owners. For that purpose the JAWS* watermark technology has been developed.

A watermark is embedded before transmission on behalf of the content owner. A monitoring site installed within a transmission area will extract the watermark and its associated payload. This allows the determination of the content owner and other data. This information is fed back to a central database for various applications such as IPR surveillance (e.g. in the case of news clips), transmission verification (e.g. in the case of commercials) and statistical data collection and analysis.

The insertion of the watermark into the video is such that the precise identity of a particular video clip can be established. JAWS is designed such that it is practically impossible to remove the watermark, either intentionally or due to regular processing. The effect of JAWS watermarking on picture quality is imperceptible to the human observer. On the monitoring side, an extractor will recognize the watermark and precisely identify the source clip and also the time, location and channel of the broadcast. The watermark extraction process has a very small error rate (false negatives as well as false positives), even in the case that the original watermarked content was submitted to a combination of common signal processing operations, such as

- compression
- D/A and A/D conversion
- editing (subtitle or logo insertion)
- format conversion
- change of aspect ratio

Correspondence: Email: kalker@natlab.research.philips.com, 31-40-2743839 (tel), 31-40-2744675 (fax)

*JAWS = Just Another Watermarking Systems. Originally we intended to use the name YAWS (Yet Another Watermarking System) as a pun to some well known UNIX utilities. When we found out that YAWS has some undesirable other meaning, we changed the name slightly.

JAWS is also robust to signal degradations caused by transmission such as noise accumulation and bit errors.

In the following sections of this paper we will present JAWS in more detail. The outline of the paper is as follows. In Section 2 we discuss the basic philosophy of the JAWS watermarking system. In Section 3 we will discuss the basics and the motivation (with an emphasis on simplicity) of the JAWS embedding scheme and in Section 4 the basics of detection theory. In Section 5 we focus on a very important robustness issue, viz. the requirement that watermark detection be robust with respect to arbitrary shifts. By introducing translational symmetry in watermark patterns, JAWS can be made shift invariant. In Section 6 we show that this shift invariance can be used to increase the payload of the embedded watermarks. In Section 7 we show that this invariance can also be used to associate a reliability measure with every watermark detection. In Section 8 we discuss the complexity of the detection algorithm and in Section 9 we briefly discuss the robustness of JAWS. Finally, we end with the conclusion in Section 10.

## 2. BASIC VIDEO WATERMARKING PHILOSOPHY

The basic premise at the start of the development of JAWS was to design a watermarking system, which was both simple and satisfied all the requirements with respects to perceptual quality and robustness.

Several issues had to be addressed. Firstly we had to decide upon the basic format in which the watermark was to be detected. Currently video is mostly broadcasted in the analog domain (PAL, NTSC). In the near future there will be a shift to broadcasting in the digital domain, mainly using the MPEG-2 compression standard. As the VIVA broadcast monitoring system is to be operational in the near future, it seemed inevitable that the watermarking scheme should at least be able to detect in the (analog or digital) base-band domain. A watermark detection scheme directly operating on analog base-band signals would be ideal, but we are not aware of the existence of such a system. We concluded therefore that JAWS detection should operate in the digital base-band domain. A consequence of this decision is that, without special tricks, watermark detection on digital MPEG video needs at least a partial MPEG decoder.

Secondly, we had to decide in which representation to detect the watermark. Browsing through the literature, one finds basically three kinds of approaches. In the simplest approach no transformation is performed, and the watermark is directly detected in the base-band video using some correlation-like method.[1–3] At the other end of the spectrum watermarks are embedded and detected in some type of frequency domain. Embedding and detection is therefore preceded by a frequency domain transform. Well-known transforms are a Fourier Transform (FT),[4] a Discrete Cosine Transform (DCT)[5] and a wavelet transform (WT).[6] Again by using some correlation-like method, the watermark is then detected in the transform domain. Although these latter approaches tend to yield very reliable watermark detection, we decided not to pursue this direction due to the complexity of the global transform, very likely prohibiting real-time detection. The third approach addresses this complexity issue by performing frequency transforms on a block-by-block basis.[7,8] The problem with such an approach is its vulnerability to spatial image shifts, a very common and cheap processing step. Spatial shifts cause a miss-alignment of block-boundaries and therefore a failure to detect the watermark. Based upon this analysis we decided to rely on the first approach, i.e. simple spatial correlation. Representing a pixel (luminance) value at position $i$ (both spatial and temporal) by the symbol $y_i$ and the correlation pattern by $w_i$, watermark detection can be described by the formula Eq. (1),

$$d = \frac{1}{N} \sum_i y_i w_i, \tag{1}$$

where $N$ is the number of pixels involved in the correlation. The system is designed such that a large value of $d$ indicates the presence of the watermark $W = \{w_i\}$, and a small value indicates the absence of the watermark. In this manner it is possible to embed a one-bit payload. Note that watermark detection is not performed on chrominance values, as the system is required to be robust to gray-scale conversions. For the remainder of this we will therefore ignore any chrominance data, and assume that all content is gray-scale only.

Thirdly, we had to decide upon the utilization of the temporal axis. For reasons of complexity we decided upon the use of a purely spatial watermark pattern $W$ and to embed $W$ repeatedly in every frame of the video. This choice amounts to treating video as a sequence of still images. Watermark detection can now succinctly be described by Eq. (2),

$$d = \frac{1}{NT} \sum_i \left( \sum_t y_{t,i} \right) w_i, \tag{2}$$

where $t$ and $i$ denote the temporal and spatial position of a pixel, respectively. The symbol $N$ and $T$ denote the number of pixels in a single video image (note the difference with the previous interpretation of $N$) and the number of video frames, respectively. By first accumulating in time, as indicated by the brackets in Eq. (2), the complexity of watermark detection is reduced by decreasing the number of multiplications. The following section goes into some more details of JAWS watermark embedding.

## 3. BASIC WATERMARK EMBEDDING

In the previous section we concluded that – for our watermarking purposes – video is best considered as a sequence of stills, a video sequence is then marked by embedding the same watermark in a number of consecutive frames. By changing the watermark pattern at a low rate, we can also realize payload along the temporal axis, but for the current discussion this is of no relevance. We therefore focus on watermark embedding in a single video frame

Given our preferred watermark detection scheme, viz. correlation with a watermark pattern $W = \{w_i\}$, the optimal embedding scheme consists of adding a scaled version of $W$ to an original image $X = \{x_i\}$. That is, a watermarked image $Y = \{y_i\}$ is obtained by Eq. (3),

$$y_i = x_i + sw_i, \tag{3}$$

where $s$ is a global scaling parameter. In other words, a watermark in JAWS is simply additive noise. The samples of the watermark pattern $W$ are independently drawn from a normal distribution $\mathcal{N}(0,1)$ with mean and standard deviation equal to 0 and 1, respectively. In particular, the sample values of $W$ are floating point values. Many existing schemes in the literature use integer valued watermark patterns.[1,9] There are however a number of advantages to using floating point values.

1. Under the condition that the average energy of the watermark per pixel is equal to 1, a normal (floating point) probability distribution has the largest entropy.[10] From a standpoint of security, a normal distribution is therefore optimal, i.e. most difficult to guess.

2. A floating point watermark pattern yields a better linear relationship between the global scaling parameter $s$ and image distortion. For integer valued watermarks this relationship is much more step-like. At this point we need to remark that the embedding formula Eq. (3) needs to be modified slightly to include rounding and clipping to reflect the fact that luminance can only be integer valued (typically between 16 and 235)

$$y_i = \text{RoundClip}(x_i + sw_i), \tag{4}$$

As the sample values of the watermark pattern are independently drawn, it follows that the watermark pattern $W$ is spectrally white. It is a priori not clear that "white" is the optimal choice for the spectral color of watermark patterns. On the one hand – as (natural) images tend to be highly correlated – one might argue for using a correlated watermark pattern. Such a pattern can for example be obtained by low-pass filtering a spectrally white pattern. Experimentally we have found that low-pass watermarks are indeed more robust than white watermarks, but also that it is difficult to avoid visual artifacts. On the other hand – as the Human Visual System (HVS) is less sensitive to high frequency patterns than it is to low frequency patterns – one could also argue for using a high-pass watermark. A drawback of such an approach is that such a watermark detection is less robust. Weighing the pros and cons of the low-pass and high-pass approaches, we decided to compromise upon a spectrally white watermark pattern.

If watermark embedding is performed directly as described in Eq. (4), one easily finds that artifacts appear in image regions where there is little activity, e.g. in regions with little texture. A solution to this problem is the incorporation of a local scaling factor $\Lambda = \{\lambda_i\}$,

$$y_i = \text{RoundClip}(x_i + s\lambda_i w_i). \tag{5}$$

The value of $\lambda_i$ should be small in image regions where there is little activity (e.g. flat regions in cartoons), and large in regions where there is much activity (e.g. in textured regions or at edges). A satisfactory local scaling factor has been experimentally found by filtering the image with a Laplacian high-pass filter $L$ and taking absolute values, i.e.

$$\Lambda = |L \otimes X|, \tag{6}$$

where "$\otimes$" denotes convolution, and where is $L$ is defined by

$$L = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} /9. \tag{7}$$

## 4. BASIC WATERMARK DETECTION

We recall from Section 2 that watermark detection is performed by spatial correlation. If watermark embedding is performed as in Eq. 5, we can write

$$Y = X + s\Lambda W, \tag{8}$$

where $X$ is the original image, $s$ is the global scaling parameter, $\Lambda$ the local activity measure and $W$ the watermark pattern. Performing watermark detection by correlation, the resulting decision $d$ consists of two terms,

$$d = d_{\text{org}} + d_{\text{wmk}} = \frac{1}{N}\sum_i x_i w_i + \frac{1}{N}\sum_i \lambda_i w_i^2 \tag{9}$$

It is not difficult to show that the expected value $\mathrm{E}[\,d_{\text{org}}\,]$ contributed by the original unmarked image is equal to $0^{\dagger}$, and that under very general conditions the standard deviation of $d_{\text{org}}$ is given by

$$\sigma_{d_{\text{org}}} = \frac{\sigma_X}{\sqrt{N}}, \tag{10}$$

where is $\sigma_X$ the standard deviation of the original image.[11] The contribution of the watermark is given by

$$d_{\text{wmk}} = s\mu_1(\Lambda), \tag{11}$$

where $\mu_1(\Lambda)$ denotes the first moment (or mean) of the local activity. It follows that for a given false positive rate $\rho$ and associated threshold $T_\rho = \mathrm{erfc}(\rho)$, where erfc is the complementary error function, the value of $s$ should be larger than[12]

$$s \geq \frac{\sigma_X T_\rho}{\mu_1(\Lambda)\sqrt{N}}. \tag{12}$$

In practice $s$ has to be chosen considerably larger in order for the watermark to survive common video processing.

A boost in detection performance can be obtained by applying *matched filtering* before correlation.[11] For the purpose of this paper a matched filter is a decorrelating and zero-phase FIR filter say $A$. Application of $A$ should significantly remove the correlation between neighbouring image pixels. A good example of a filter $A$ is given by,

$$A = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} /4. \tag{13}$$

If we put $Z = A \otimes Y$ and apply correlation detection to $Z$, then the contribution of the original image (i.e. of $A \otimes X$) will be much reduced in value due to the reduction in standard deviation. It is easily checked that, under some mild restrictions, the contribution of the watermark part (i.e. $sA \otimes (\Lambda W)$) remains unchanged. Replacing $X$ by $Z$ in Eq. (12) we see that a smaller scale factor is required to meet a given false positive rate, or, that for a given scale factor $s$, a more reliable detection is obtained.

Matched filtering can also be understood in the frequency domain. It is well-known that the optimal way of detecting a signal $S$ in additive noise $\nu$ is to filter the noise with a (matched) filter $H$ such that $H \otimes \nu$ noise is spectrally white.

$^{\dagger}$We assume normalization to zero mean before correlation.

## 5. INCORPORATING SHIFT INVARIANCE

In the foregoing we have assumed that during detection the watermark and the image are perfectly aligned. In practice we can not rely on this. During normal processing the position of the image may easily vary a little. Moreover, in order to circumvent watermark detection, a malevolent hacker can easily and cheaply induce spatial shifts, even on a frame by frame basis. It is therefore strictly required that the watermark system is resistant to spatial shifts. The most simple approach to achieve this invariance is exhaustive search for the correct alignment of the watermark. That is, for each allowed spatial shift $k$ the decision variable $d_k$,

$$d_k = \frac{1}{N} \sum_i y_i w_{i-k} \tag{14}$$

has to be computed. For ease of presentation, we have neglected boundary problems in this formula. For the same reason matched filtering is omitted from Eq. 14.

This search over all possible spatial shifts is computationally prohibitive if we aim for real-time watermark detection. The solution adopted in JAWS is to introduce translational symmetry in the watermark pattern $W$. The particular choice made in JAWS requires that

$$w_{i+k} = w_i, \tag{15}$$

for every vector $k$ whose components are multiples of $M$, where $M$ is referred to as the *tile size*. A practical choice for the value of $M$ is $M = 128$. In other words, the watermark pattern $\{w_i\}$ is completely determined by an $M \times M$ matrix $\underline{w}_i$, $i \in 0, ..., M-1 \times 0, ..., (M-1)$ of (pseudo) random values. The full watermark pattern $\{w_i\}$ is obtained by *tiling* (possibly with truncation) the matrix $\{\underline{w}_i\}$ over the extent of the image.

With these assumptions the exhaustive search over all possible shifts is greatly simplified. As the watermark is repeated over vectors which are multiples of $M$, one can first *fold* the suspect image data $Y$ to matrix a $B = \{b_i\}$ of size $M \times M$,

$$b_i = \mathrm{fold}(Y)_i = \sum_{j=(j_1 M, j_2 M)} y_{i+j}, \quad i \in 0, ..., M-1 \times 0, ..., (M-1), \tag{16}$$

where $j_1$ and $j_2$ are the indices of the individual tiles. Due to the folding and neglecting boundary problems, we now only need to search over *cyclic shifts* $k$, where $k$ is in $0, ..., M-1 \times 0, ..., (M-1)$. More mathematically this can be expressed as

$$d_k = \frac{1}{M^2} \sum_i b_i \underline{w}_{i-k}, \tag{17}$$

where the subtraction in the index $i-k$ of $\underline{w}$ is computed modulo $M$.

Inspecting Eq. (17) more closely, we easily see that in fact we have to compute a two-dimensional cyclic convolution. Letting $\underline{w}_i^*$ denote the spatial inversion of $\underline{w}$, i.e. $\underline{w}_i^* = \underline{w}_{-i}$, Eq. (17) can be written as

$$D = B \otimes \underline{W}^*, \tag{18}$$

where – by abuse of notation – "$\otimes$" now denotes cyclic convolution. It is well known that a cyclic convolution is most efficiently computed in the frequency domain.[13] The computation of the matrix $D$ then proceeds as follows.

1. Pre-compute, using a Fast Fourier Transform (FFT), the Fourier transform $\underline{\hat{W}}$ of the matrix $\underline{W}$.
2. Compute the Fourier transform $\hat{B} = \mathrm{FFT}(B)$ of the fold buffer $B$.
3. Perform a point-wise multiplication of $\hat{B}$ and $\underline{\hat{W}}^*$ to obtain the matrix $\hat{D}$. Note that in this context the $*$-operator denotes complex conjugation.
4. Compute $D$ by applying the inverse FFT (IFFT) to $\hat{D}$.

More concisely,

$$D = \mathrm{IFFT}(\mathrm{FFT}(B)\,\mathrm{FFT}(\underline{W})^*). \tag{19}$$

As argued in Section 4, detection performance can be improved by preceding correlation by matched filtering. The goal of matched filtering is to decorrelate the suspect image $Y$ to obtain an approximately spectrally white version of $Y$. Matched filtering is usually performed in the spatial domain (using some simple and cheap decorrelation filter), but can in our current set-up also be computed in the Fourier domain. Moreover, we need not be satisfied with an approximately white signal. By only retaining the *phases* of $\hat{B}$ we obtain a *purely* white signal. Experimentally we have found that the best detection is obtained by also ignoring the magnitude information in $\underline{\hat{W}}$, resulting in the detection formula

$$D = \mathrm{IFFT}(\mathrm{phaseOnly}(\mathrm{FFT}(B))\,\mathrm{phaseOnly}(\mathrm{FFT}(\underline{W})^*)), \tag{20}$$

where $\mathrm{phaseOnly}(x) = x/|x|$ for $x \neq 0$ and $\mathrm{phaseOnly}(0) = 1$. This method of detection is actually well-known in the field of pattern recognition and is referred to as Symmetrical Phase Only Filtering[14] (SPOMF). It is the preferred correlation method in JAWS.

## 6. INCREASING THE PAYLOAD

In the previous section we have seen that SPOMF detection is an excellent method to detect the presence/absence or sign of a watermark, whether the watermark is shifted or not. This allows us to embed a one-bit payload. For the broadcast monitoring application at hand, this one-bit payload is by far not sufficient. The payload can be increased along essentially two – non-exclusive – axes: the spatial and the temporal axis, respectively. More precisely, we can increase the payload spatially by using more basic patterns, and temporally by varying the absence/presence or sign of these patterns over time. Disregarding the temporal domain, a payload of $n$ bits can be obtained by using $n$ basic patterns $W_l$ of size $M \times M$, $l = 0, \ldots, n-1$. Each pattern will then correspond to one bit. This is the method as originally proposed by Digimarc.[3] (Note: for the sake of simplicity no underlining is used to denote $M \times M$ tiles!)

There are two disadvantages to the above approach. Firstly, the energy of the total embedded watermark is linear in the number of bits of the payload. This may be a cause for visible artifacts. Secondly, $n$ SPOMF detections are required to retrieve an $n$-bit payload. For complexity reasons this is not a favored solution. Experimentally three, or maybe four, is found to be the maximum number of basic patterns. For the VIVA application at hand the resulting payload-rate is too low. A way out to this dilemma was the insight that the inherent shift invariance of JAWS could be used to increase to payload to $n$ bits with less than $n$ basic patterns!

Because of the shift invariance of JAWS, an embedded pattern $W$ will be found whatever its position in the image. The same is true if this pattern is embedded several, say $m$, times, but at different positions. Performing detection by SPOMF, all $m$ copies of the watermark will be found. If the whole image is shifted before detection, the absolute positions of the correlation values will change cyclically. However, *the relative positions will remain unchanged*, at least if computed with modular arithmetic to the base $M$. We can therefore embed information in the relative position of the correlation peaks. This basic idea needs some refinement to really make it work.

Let us consider the example of one pattern which is embedded twice and that the tile size $M$ is equal to 128. Moreover, let's assume that the pattern is embedded at the origin $(0,0)$ and at position $(8,8)$. Upon detection two correlation peaks will be detected (see also Figure 1). However, we don't know which of the two peaks corresponds to the pattern embedded at the origin. Therefore we can only determine the relative position of the two peaks up to a sign, i.e. we cannot distinguish between $(-8,-8)$ and $(8,8)$. A simple calculation shows that therefore we can only distinguish between $(M^2)/2 + 1 = 8193$ different relative positions. This amounts to a little more than 13 bits for one single SPOMF detection. In practice this payload can not be achieved because the position of the peaks is susceptible to jitter. In particular, positional jitter occurs for video over poor analog links. Experimentally it was found that a reliable detection of payload can be obtained if we require that relative positions are a multiple of the so called *grid size* $G$. A practical value for the grid size is $G = 8$. This restriction allows correction for position jitter, but at the same time reduces the effective payload of the watermark. In our example only 7 bits will remain.

Fortunately we can exploit another degree of freedom. We note that we are at liberty to embed a watermark with either a positive or negative sign. This sign is correctly retrieved by SPOMF detection. Clearly the sign is shift invariant, and can therefore also be used as part of the information carrier. Continuing our example, we now

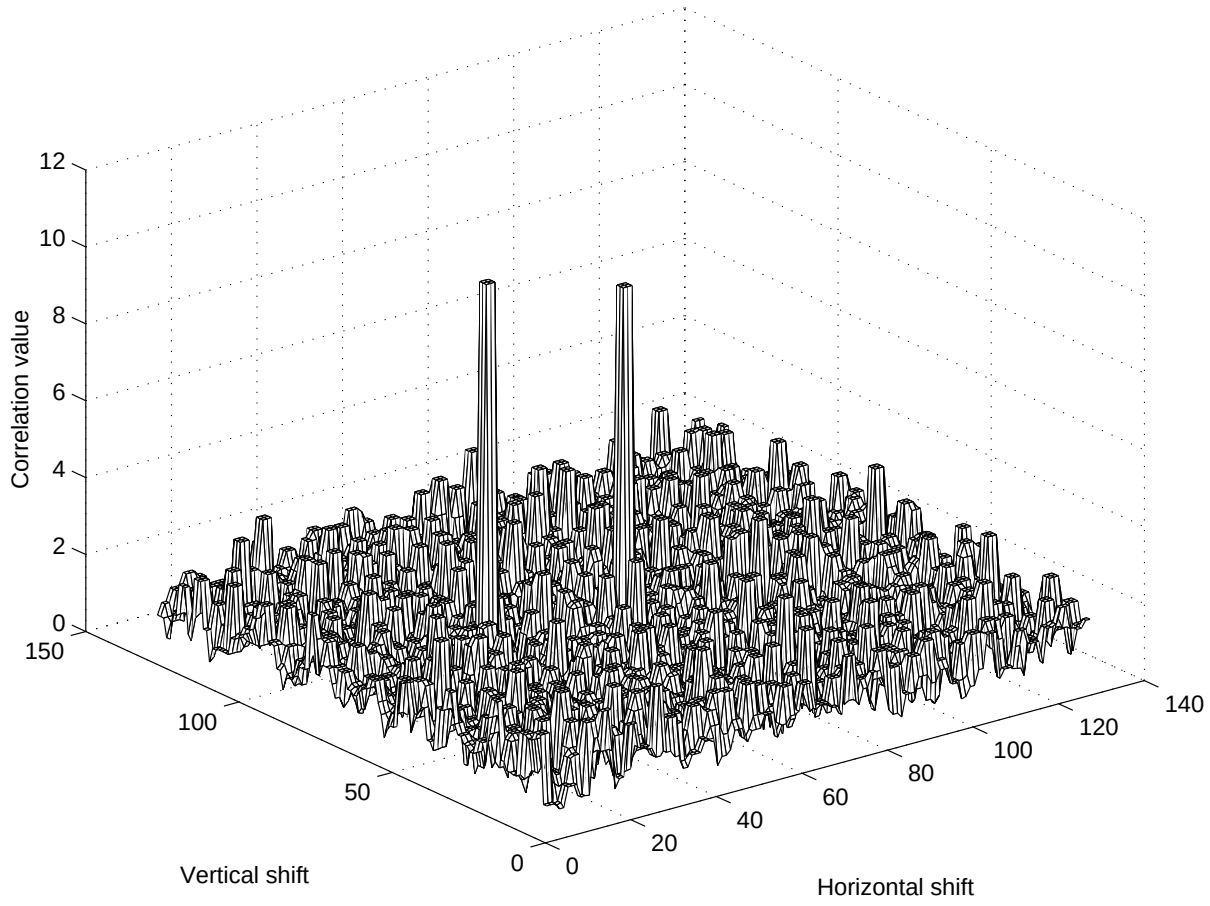


**Figure 1.** An example of SPOMF detection with one pattern of multiplicity two.

embed the pattern at the origin with a positive sign, and the shifted pattern with a negative sign. In this setup, the SPOMF detector is able to distinguish between the peak corresponding to the pattern at the origin and the peak corresponding to the shifted pattern. The sign ambiguity is therefore resolved and, using a grid size $G = 8$, we can now embed 8 bits. The above reasoning can of course be generalized to embedding $m$ patterns $W_i$, each with a certain multiplicity $v_i$. It is an interesting mathematical problem to find in this general setting the relationship between the size of the payload, the total number $m = \sum_i m_i v_i$ of embedded patterns, the grid size $G$ and the number $n$ of basic patterns. This, and methods for the actual association of bit patterns to sets of relative positions, is the topic of another paper.[15] At this point it suffices to say that a sufficient number of bits can be embedded for the broadcast application at hand.

## 7. FALSE POSITIVE ANALYSIS

We recall that the retrieval of watermark payload is in essence achieved by looking for large positive or negative peaks in the correlation buffer $D$. We are interested in the rate of false positives. There are actually two types of false positives. A "true" false positive occurs when a watermark is detected when no watermark has been embedded. An *invalid* positive occurs in case a watermark has been embedded but the wrong payload is retrieved. Both types of false positives are highly undesirable because they may lead to "sending an unjustified bill to the honest guys". A good false positive analysis is therefore essential for the broadcast monitoring application.

An intermediate result in a JAWS detection event is a $M \times M$ buffer $D$ of correlation values. The payload of the watermark is determined by the (relative) positions of a number of extremal values in that buffer. The key insight is now that the non-extremal values can be considered as watermark detections for non-watermarked images: by correlating the watermark with the image at non-embedding positions, the image appears to the watermark as an

original, non-marked image. Experiments have confirmed that these non-extremal values are normally distributed. In fact one can prove that for SPOMF correlation, under very general conditions, the mean and the standard deviation are 0 and $1/M$, respectively. By setting the threshold for peak detection at $5/M$ we achieve a probability for invalid peak detection of $P = 5.7 * 10^{-7}$.

Continuing the example of the previous section (one pattern with multiplicity 2, a grid size $G$ equal to 8 and an allowed jitter $J$ equal to 1) the false positive rate for unmarked images can be computed. A false positive occurs if there are precisely two extremal values in the correlation buffer, one positive and one negative, at positions which differ by a multiple of 8, give or take 1. It is not difficult to derive that the false positive rate $Q_0$ is (approximately) given by

$$Q_0 \approx \left(\frac{2J+1}{G}\right)^2 M^4 P^2 \approx 1.2 * 10^{-5}. \tag{21}$$

As we are dealing with video we can accumulate several (say $T_1$) of these *micro-decisions*. The probability that more than $T_2$, $0 < T_2 \leq T_1$, of these micro-decisions yield the *same result* (i.e. not just a valid payload) is given by the formula

$$Q_{(T_1,T_2)} \approx \left(\frac{2J+1}{G}\right)^2 \sum_{T_2 \leq T_3 \leq T_1} \binom{T_1}{T_3} \left(\frac{2J+1}{M}\right)^{2(T_3-1)} M^{4T_3} P^{2T_3}. \tag{22}$$

For example, by setting $T_1 = 5$, $T_2 = 3$, the probability of a false positive can be reduced to $2.9 * 10^{-20}$. For all practical purposes this false probability rate is more than sufficient. But if necessary, it can be reduced even more by choosing appropriate values for $T_1$ and $T_2$.

A similar reasoning can be applied to estimate the probability of invalid positives. In fact, it is not difficult to see that the probability computed in Eq. 22 is a good approximation.

Note that the previous reasoning assumes that micro-decisions are independent events. Experiments have confirmed that for most video scenes this is true. For certain scenes, such as extremely long stills, this assumption might not hold true.

## 8. COMPLEXITY ANALYSIS

Figure 2 gives an overview of the embedding procedure in JAWS. Given a payload $K$ a pattern $W$ is computed as a sum of cyclicly shifted (and possibly inverted) basic patterns $W_i$. This pattern $W$ is then tiled over the extent of a video frame and locally scaled by means of the local activity measure. After globally scaling with the parameter $s$, the result is added to the video frame. Finally a watermarked video frame is obtained by rounding and clipping. The payload $K$ needs to be kept constant for a sufficient number of video frames to allow reliable detection. By changing the payload at a sufficiently low rate (as not to violate the constraint of the previous sentence), payload can be embedded along the temporal axis.

The most complex operation for watermark embedding is the computation of the local activity measure. The computational complexity per pixel is quite low, but the computations have to be performed at video rate. To show feasibility, a real-time watermark embedder has been implemented on a TriMedia platform.[16] This shows that with "modest" means the complexity of embedding can be surmounted.

Figure 3 gives an overview of the detection procedure in JAWS. Detection starts with the accumulation of sufficiently many video frames. The frames are folded, summed and stored in an $M \times M$ buffer $B$. When a sufficient amount of data has been accumulated, SPOMF correlation with the basic patterns $W_i$ is applied. The resulting correlation buffers are examined for extremal values, and, if present, a payload $K$ is returned. A nice feature of JAWS detection is that the computations at video rate are simple (mainly additions) and that the complex computations (SPOMF) operate at a much lower rate. Moreover, the computational complexity of SPOMF can be reduced at the cost of extra memory resources by pre-computing the phase-only representations of the basic patterns.

To show feasibility, a real-time watermark detector has been built on three different platforms, viz. on a high-end Silicon Graphics workstation, on a TriMedia processor board[16] and on an FPGA based board.

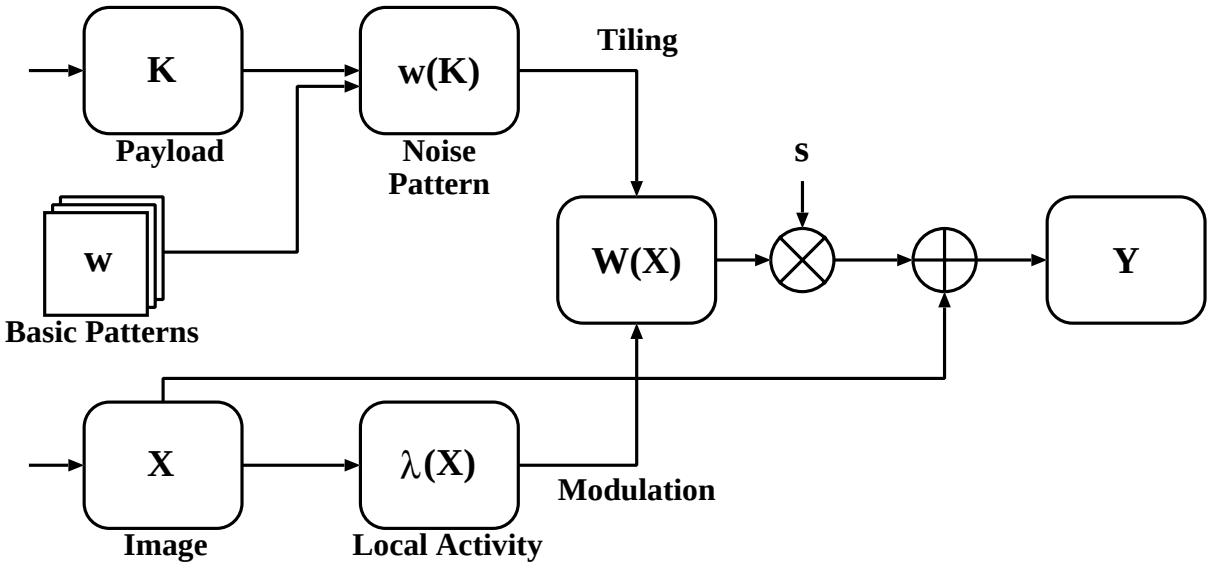


**Figure 2.** Overview of JAWS embedding.

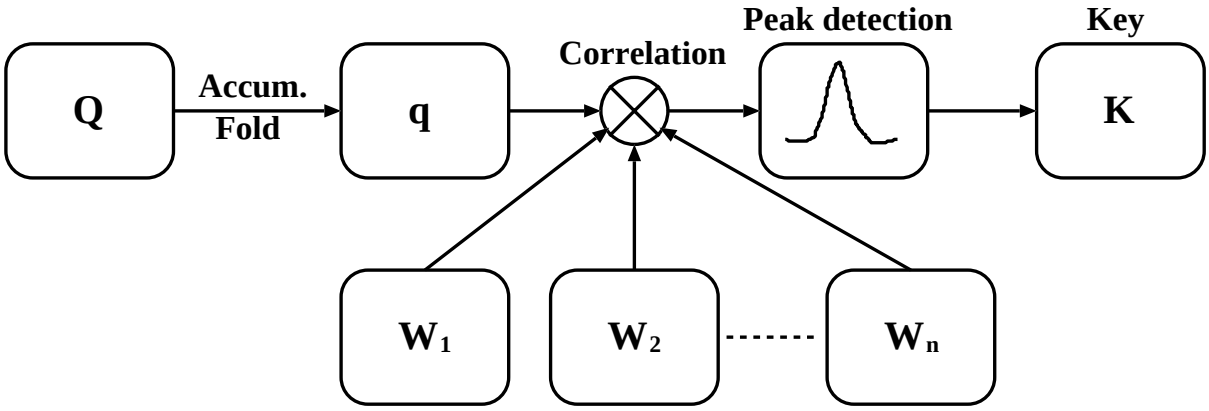


**Figure 3.** Overview of JAWS detection.

## 9. ROBUSTNESS TESTS

Many experiments have been performed to test the robustness of the JAWS system. It has been shown that JAWS survives MPEG-2 compression down to at least 2 Mb/s, MJPEG compression, DA/AD conversion, PAL conversion, noise addition, quantization, subtitling and logo insertion, cropping, frame erasure, speed-ups and transmission errors. A more detailed description of these robustness tests is reported in.[16]

More extensive testing will be done by the end of 1999, when watermarked and subsequently processed video content will be broadcast from a central site by satellite to local broadcasting stations, re-broadcast to the end-users and received by local monitoring stations for watermark detection.

## 10. CONCLUSIONS

We have presented a new watermark technology that can be applied in broadcast monitoring. Given the requirements of broadcast monitoring, we have tried to argue the necessity of a nuber of technical choices. The watermark solution provided by JAWS is unique in the way that it exploits shift invariance to obtain a high payload and a reliability measure with every detection. JAWS has been implemented on several platforms, showing the feasibility of real-time embedding and detection. Tests within VIVA have shown that JAWS scores well on both visibility and robustness.

An important issue for future work is to improve the robustness of JAWS with respect to scaling and rotation.

## ACKNOWLEDGMENTS

The authors would like to acknowledge the support through the VIVA project from the ESPRIT program of the European commission. Similarly, we acknowledge with all partners of the VIVA projects for many fruitful discussions: Pascale Termont, Lieven De Strycker, Jan Vandewege, Andreas Langell, Claes Alm, Per Norman, Gerry 'O Reilly, Herby Rana, Bob Howes, Henk Vaanholt, Pat Donnely and Andy Hudson.

## REFERENCES

1. N. Nikolaidis and I. Pitas, "Copyright protection of images using robust digital signatures," in *Proceedings of the ICASSP*, pp. 2168 – 2171, May 1996.
2. J. Delaigle, C. de Vleeschouwer, F. Coffin, B. Macq, and J.-J. Quisquater, "Low cost watermarking based on a human visual model," in *Multimedia Applications, Services and Techniques (ECMAST '97)*, S. Fdida and M. Morganti, eds., vol. 1242 of *Springer Lecture Notes in Computer Science*, p. 153, Springer-Verlag, Heidelberg, Germany, 1997.
3. G. Rhoads, "Identification/authentication coding method and apparatus." WIPO Patent no. WO 95/14289, 1995.
4. J. O. Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking," in *Proceedings of the ICIP*, vol. 1, pp. 536 – 539, (Santa barbara, California), Oct. 1997.
5. I. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing* **6**, pp. 1673 – 1687, Dec. 1997.
6. X.-G. Xia, C. Boncelet, and G. Arce, "A multiresolution watermark for digital images," in *Proceedings of the ICIP*, vol. 1, pp. 548 – 551, (Santa barbara, California), Oct. 1997.
7. J. Zhao and E. Koch, "Embedding robust labels into images for copyright protection," in *Proceedings of the International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*, (Vienna, Austria), Aug. 1995.
8. J. Zhao and E. Koch, "A generic digital watermarking model," *Computers and Graphics* **22**(4), pp. 397 – 403, 1998.
9. W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal* **35**(3/4), pp. 313 – 336, 1996.
10. J. R. Hernández, F. Pérez-Gonzálves, J. M. Rodríguez, and G. Nieto, "Performance analysis of a 2-d-multipulse amplitude modulation scheme for data hiding and watermarking of still images," *Journal of Selected Areas in Communications* **16**, pp. 510 – 524, May 1998.
11. G. Depovere, T. Kalker, and J.-P. Linnartz, "Improved watermark detection using filtering before correlation," in *Proceedings of the ICIP*, vol. I, pp. 430 – 434, (Chicago), Oct. 1998.
12. T. Kalker, J.-P. Linnartz, and G. Depovere, "On the reliability of detecting electronic watermarks in digital images," in *Proceedings of Eusipco-98*, vol. I, pp. 13 –17, September 1998.
13. A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression*, Applications of Communications Theory, Plenum, 1998.
14. L. Brown, "A survey of image registration techniques," *ACM Computing Surveys* **24**, pp. 325–376, Dec. 1992.
15. M. Maes, T. Kalker, J. Haitsma, and G. Depovere, "Exploiting shift invariance to obtain a high payload in digital watermarking," in *Proceedings of the ICMCS'99*, 1999. Submitted.
16. P. Termont, L. D. Strycker, J. Vandewege, J. Haitsma, T. Kalker, M. Maes, G. Depovere, A. Langell, C. Alm, and P. Norman, "Performance measurements of a real-time digital watermarking system for broadcast monitoring," in *Proceedings of the ICMSC'99*, 1999. Submitted.