MPEG PTY-Marks: Cheap Detection of Embedded Copyright Data in DVD-Video

J. P. M. G. Linnartz and J. C. Talstra

Philips Research, WY8; Prof. Holstlaan 4
5656 AA Eindhoven; Netherlands

Abstract. In this paper we propose a method to watermark digital video content in such a way that detection in consumer electronics equipment is possible with very little hardware (a few thousand gates). The method proposes to modify the MPEG encoding procedure to choose the so-called *Picture Type* of video-frames not from a regular sequence but according to a message one would like to transmit. Removal of this embedded message, the *PTY-Mark*, from the resulting MPEG-stream without jeopardizing video quality is only possible after a complete MPEG decoding and re-encoding cycle. We investigate the modifications to current MPEG encoders which are necessary to accommodate these PTY-Marks. Based on tests we comment on their feasibility. Detection of watermarks without secrets is very reminiscent of "public-key" cryptography. We discuss this relationship by contrasting PTY-marks with pixel-watermarking.
Keywords: Watermarking, copy-protection, MPEG, DVD-video.

1 Introduction

The last few years have seen enormous expansion of the number of multimedia storage options, from ordinary CD to hot newcomers like DVD-R. The common denominator of all these new systems is that they store and disseminate *digital* information, be it text, pictures, audio, video or software. This digital nature poses a very realistic threat to those who provide proprietary or copyrighted content. The need for those providers to protect their legal rights has sparked a flurry of research activity in the field of copyright protection, thereby coming a long way from the days of analog scrambling of premium cable- channels.

The history of anti-copy measures has taught us that as no protection scheme is absolutely secure, the relevant question becomes really one of what level of attacks can be subverted at what price. This question is directly related to what kind of attacks one expects a consumer-device to withstand given the realities of the marketplace. Therefore a few words about levels of piracy and how one would like to guard against them. On the one hand casual home copying can be effectively stopped by fairly simple technical measures. On the other hand, large scale pirates have ample technical means to circumvent any protection. Because the number of these large operations is limited, they can be challenged in court except in countries where the authorities are either non-co-operative

or insufficiently in control. The category in between, viz. the small-scale pirates running cottage or garage factories, may be too small to attack through legal actions. Meanwhile these pirates often have sufficient facilities for tampering with recording devices, to overcome conditional record protection measures of their own equipment However, pirates have no access to the devices installed in the homes of their potential customers. This suggests that the best measure against small-scale piracy is *playback control*, at the expense of (a small amount of) additional logic in consumer equipment. This playback control is preferably conducted in a simple disc drive or other storage device which usually has no facilities to process and interpret the stored "bits and bytes". This shift from record- to playback-control of the anti-copy paradigm is generally regarded as a technological challenge due to the "dumbness" of such drives.

Imposing playback-control implies that the world of playback equipment gets divided into *compliant* and *non-compliant* devices. Given the fact that it will most likely be impossible to root out the members of the second category, the strategy of an anti-copy mechanism should be to keep protected content from being multiplied in the non-compliant world and hurting sales by re-entering the compliant world.

Encryption, as for instance applied to disc sectors, only addresses part of the issue of illegal copying. This applies in particular to the new digital "content scrambling system" or CSS, for DVD-video disks. At some point the encrypted content is read from the disc and becomes available in the clear, either after (legal) decryption, or illegally, after the cryptographic algorithm has been cracked or its key obtained. This content needs further protection against copying and mass-multiplication without loss of quality, an issue that is of particular concern to music- and movie-studios.

One of the solutions which has been pursued in relatively recent years, is that of pixel watermarking. It is possible to mark an image, a video-clip or sound-bite in such a way that marked and unmarked pieces differ in a mean squared sense and are very distinguishable as such by electronic hardware—yet at the same time this deviation cannot be perceived by the human sensory system[1–16]. Embedded signaling in the form of watermarks is much like an electronic "tattoo" in that it ensures that marks are not lost in typical operations, including format conversions. Although the principle of pixel-watermarking seems to put an elegant end to copyright issues, it has a few serious drawbacks.

First of all, a provider who wishes to assert his/her ownership of still-pictures may employ (in principle) arbitrary resources (time and computing power) to detect the watermark that (s)he inserted. On the other hand, for video playback control this ownership has to be determined by a rather limited proxy, viz. *consumer*-equipment that decides whether playing/recording for a particular medium is allowed. This verdict should be reached *on the fly* (say every 10 seconds) and *cheaply* i.e. with very few gates.

Secondly there is the issue of security. With the advent of the personal computer on the film/music scene, care has to be taken that the relatively open bus-structure which is absent in a consumer recorder does not become the Achilles

Heel of a copy-protection system. One can imagine for instance that in a computer, a DVD-drive sends (encrypted) MPEG encoded video-material over the PCI bus to an MPEG decoder video card. That drive learns over the same PCI bus from the card, whether playback should be ceased or not, depending on the state of the watermark in the baseband video-content, a situation that would be vulnerable to a "man in the middle attack". Forestalling this situation by attempting pixel watermark detection already in the drive, would require this relatively dumb playback device to have a partial MPEG- decoder on board! Current estimates of the complexity of such a pixel-watermark detector start around 50,000 gates.

The third undesirable feature of pixel-watermarking is that present methods rely on a pseudo-random number sequence that is embedded in images. The detection of this sequence plays the role of a secret key. An important distinguishing characteristic of watermarks is the level of restriction placed on the ability to read a watermark. For example, in many cases, it is desirable to embed information in audio, image or video content such that this information is readable by any recipient. In an application such as transferring copyright ownership information by watermarking news photographs, any and all receiving users should be capable of reading the embedded information. This has been called "public" watermarking, drawing analogy with public key cryptography. However, this nomenclature is misleading. All currently known watermarking algorithms fall into the category of "secret key" algorithms, in the sense that any expert who knows the algorithm and the key also has all the necessary tools to *remove* that watermark. In the parlance of cryptography this would be called "bringing the content into the open" (allowing it to be copied). The detectors embedded for instance in CE products, therefore have to store this secret in a relatively tamper resistant environment. To the best of our knowledge, no equivalent to public key encryption is currently available for watermarking that would allow public dissemination of a method and key to detect the watermark, without inherently revealing how the watermark can be removed.

In (hypothetical) public watermarking, the embedding algorithm is private. i.e., only known to copyright owners, the detection algorithm is public knowledge. Lacking such systems, typically a secret key algorithm is placed in a tamper-resistant box. It has been shown, however, that even if one assumes that this box is perfectly tamper-proof, it can efficiently be misused as an oracle to reveal the secrets of the watermark[17].

To deal with these three problems and fend off the various attacks associated with them, we would need another "mark", as closely intertwined, with the content as the *pixel-domain* watermark, but one that can already be detected in the *drive*. This mark does not need to survive outside the digital domain, or even after MPEG encoding, as beyond that point, the pixel-domain watermark takes over.

In this paper we will describe how a public watermarking scheme could be designed around MPEG-type compression methods. The asymmetry of embedding and detecting this public watermark relies on the difference in complexity of

MPEG compression versus MPEG decompression. We propose to use the redundancy in the choice of encoding video into MPEG Groups Of Pictures (GOPs) as a carrier for this watermark.

Section 2 will explain the MPEG-watermarking principle. Section 3 will discuss the feasibility modifying existing encoders to accommodate this watermark, using data from trial MPEG-encoding sessions on various public- domain MPEG-1/MPEG-2 encoders. Section 4 will conclude with a comparison of subject of this paper with other similar existing methods and a future outlook.

2 PTY-Marks

As is well known the MPEG-1 and MPEG-2[1] standards define three distinct ways in which a frame in a video stream can be MPEG-encoded, viz. as *I*-,*B*- and *P-Picture Types* (PTY for short). A frame encoded as an *I*-picture type, is *autonomous*: it is essentially encoded as a JPEG picture, exploiting only *spatial* redundancy. *B* and *P* frames were introduced to make better use of *temporal* redundancy: coding a frame as a *P*-picture type, one only describes differences with respect to certain previous frames (of either *I*- or *P*-type). Maximal compression efficiency is achieved with *B*-picture types which code roughly the difference between a given frame and the *interpolation* between the preceding and succeeding *I*- or *P*-frame. A sequence of frames starting with an *I*-type and up to, but not including the next *I*- frame is called a Group Of Pictures, GOP for short[18][2]..

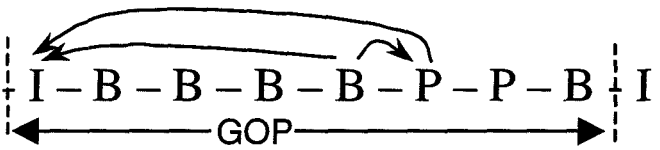

Fig. 1. GOP structure and examples of references of *B* and *P* frames.

As illustrated in Figure 1, *B* frames refer not only to the previous *I* or *P* frame, but also to the nearest *I* or *P* frame in future. Note that *P* and *B* frames cannot be decoded properly if their references are not available.

High coding efficiency is achieved by inserting as many *P*- and *B*-picture types as possible. To code a given frame as an interpolation of two others, implies

[1] Strictly speaking, in MPEG-2 the notion of GOP has been replaced by that of sequence as an autonomous self- referential group of frames/fields. For this paper we will stick to the MPEG-1 nomenclature, but this trivially extends to MPEG-2

[2] Note: the MPEG standards and refinements thereof for DVD impose only mild constraints on the choice of whether to encode a given frame as a *I*,*B* or *P*-picture type: (i) the distance between consecutive *I*'s in the resulting MPEG- stream typically does not exceed 0.6 seconds (15 frames for PAL, 18 for NTSC), and, in early versions, (ii) consecutive P frames should not be more than 3 frames apart.

that the encoder needs to do a (potentially) vast search in this frame for features like a moving car that might occur at another place in previous/later frames. The object that connects this feature to its incarnation in a previous/later frame is called a *motion vector*, and the procedure of finding it carries the name *forward/backward motion prediction.* It is particularly in this motion estimation and selection of the best reference location that MPEG encoders differ from manufacturer to manufacturer. This part of the encoding is seen as the most difficult and computationally intensive task where consumer encoders will lag in performance, compared to professional products.

In principle the degree of freedom of choosing the picture type could be exploited to transmit a low bitrate data-stream, containing e.g. copyright information. We call this deliberate manipulation of picture types a PTY-watermark of PTY-Mark for short.

Of course one might use one of the dedicated user data areas as accommodated by the MPEG-syntax as a copyright channel, but this opens up serious hacking opportunities for potential software-pirates. Conversely, removal of the copyright information embedded in a deliberately chosen sequence of picture types, requires a complete decoding/encoding cycle of the MPEG stream. It is expected that in the upcoming few years, the cost of this cycle (decoding+encoding while maintaining video quality) will remain prohibitive, financially as well as computationally. At the same time, decoding this PTY mark should be possible at little cost (a few thousand gates) as it involves just parsing the MPEG-stream and referring to a look-up table to decode GOP-structures into characters. The PTY-watermark is very much like public-key watermarking, and this allows a detector to become part of a "dumb" device such as a DVD-ROM/RAM player in a PC.

BOX 1: Vulnerability to Attacks

An attacker who is familiar with the MPEG standard can attempt to undo a PTY-mark by rewriting a P frame into a B-frame. This is possible without redoing any motion estimation for that frame. As shown in Figure 2, the new frame could use references to a future frame, but does not need to use these. While the modified frame will be decompressed correctly, artifacts will occur in *other* frames of that GOP, see fig. 2. Neighboring frames will then have incorrect references. The artifacts typically become more severe for frames later in a GOP. Figure 3 gives an example of such an incorrectly decoded frame. Erroneous references create blocks of 8 by 8 pixels with substantial luminance and chrominance errors, particularly in areas with motion.

These artifacts are clearly visible in Figure 4. To correct the artifacts caused by these attacks, the effort needed is comparable to MPEG encoding from scratch. This suggests that the watermarking method can be used in applications

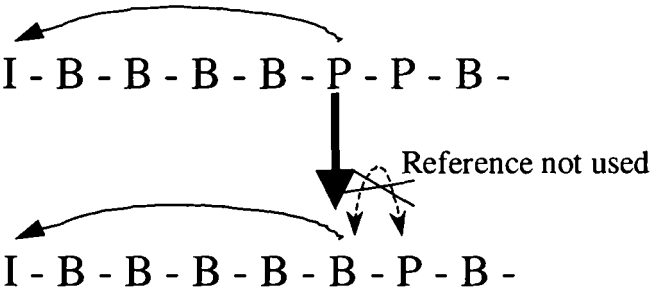

Fig. 2. Original marked GOP structure (above) and attacked sequence (below). One P frame is now written as a B frame by an attacker. The B frame allows references to the next P frame but a hacker would not use these, to avoid having to do motion estimation.

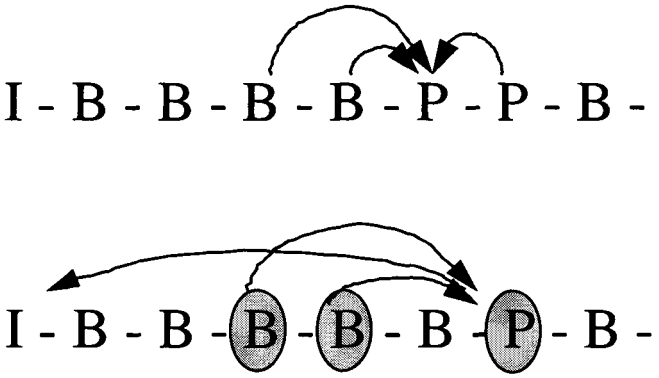

Fig. 3. Originally marked GOP structure (above) and attacked sequence (below). One P frame now is written as a B frame. Frames marked with grey circles have incorrect references and will show severe artefacts.

where conversion to uncompressed digital or analog would have circumvented the copy-protection method anyway.

3 Implementation Issues

To integrate the PTY-mark into a cryptographically secure copyright management system, we would like to allocate 64 bits to it. The MPAA (the consortium of Hollywood Movie Studios) has issued a guideline that watermark-detectors in DVD-players (of either stand-alone or PC type) should detect presence of a watermark, once every 10 seconds.

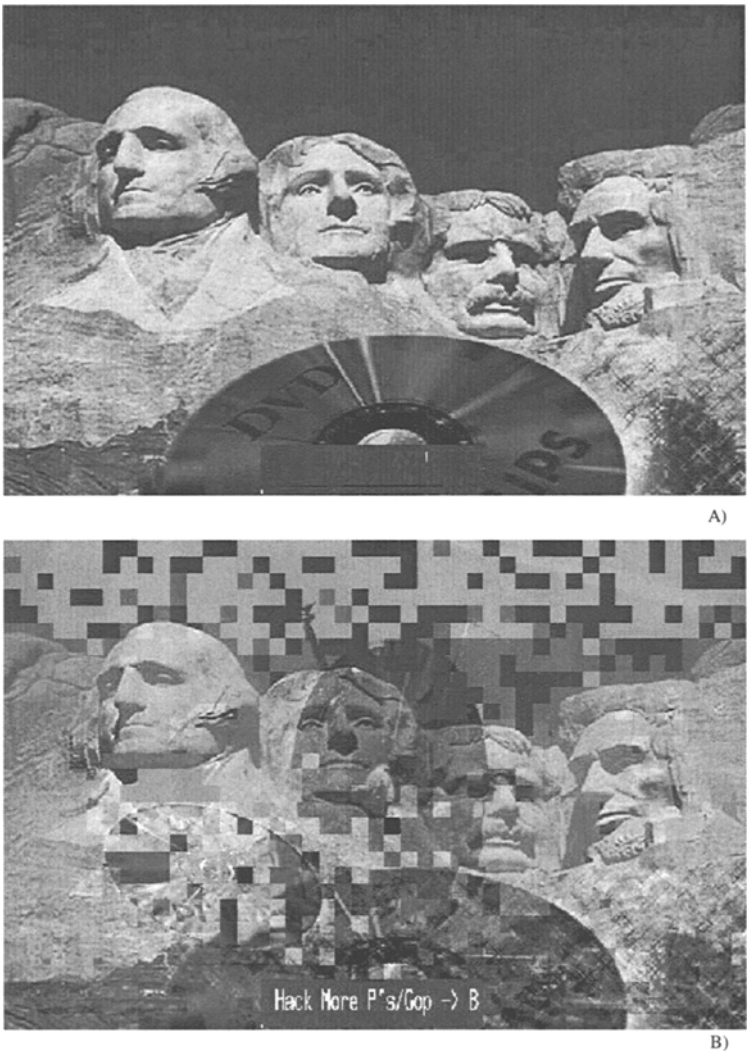

Fig. 4. A) Sample of an original video frame, and B) Artifacts caused by an intentionally modified picture type.

The choice of PTY-marks as a subliminal channel only makes sense if GOP-structures which have a "meaning" as a symbol transmitted across this channel are not being generated randomly by existing and currently envisioned MPEG-encoders. A limited survey of extant DVD-video material (see Appendix) and present MPEG encoding practices yielded the following:

1. With the maximal GOP length of 0.6 seconds, we have at least 15 GOPs/10 seconds, and therefore for a worst case 10 sec. slot every GOP should encode 6 bits on average.
2. Right now, and probably in the few years to come, the GOP-structure of choice for DVD but also commercial digital broadcast is: $IB\cdots BPB\cdots BPB\cdots \equiv IB^n\,(B^nP)^{m-1}$. Typically $n = 1, 2$ and $m = 4$ for professional equipment, and $n = 0$ for consumer-grade hard/software-encoders, and in either case usually fixed for the duration of a presentation. The conclusion that we draw from this is that the number and distribution of B-frames in a potential PTY-mark should not change too much with respect to a "normal" GOP to maintain coding complexity at a reasonable level[3], yet at the same time it should be descernible from that same standard GOP. The number of P-frames should stay approximately the same because a coded P-frame requires on average twice as many bits as a B-frame, and with fixed coding rate, extra P's decrease the SNR.
3. MPEG encoders may optimize the GOP structure a little further than the conventional sequences listed in item 1.: during scene changes, there is little temporal redundancy. To deal with this, an intelligent encoder encodes a $B-$frame which bears little resemblance to its neighbors as a $P-$frame. When this doesn't help, a new GOP is forced by coding as it an I frame. In such cases we see more esoteric GOP structures such as $IPPP$, $IBBPBBPBB\underline{PP}BB$, or consecutive single I's.
4. Occasionally we see GOPs with a more constant structure such as those containing $n > 4$ consecutive P's, representing freeze-frames without motion.

In the next sections we will give a particular implementation of a PTY-mark alphabet and discuss possible improvements.

3.1 Proposal for a PTY Alphabet

The material of this and the next section is the subject of current research at Philips NatLab as Philips' contribution [19] to the standardization subcommittee for DVD-video copy-protection through watermarking, the DHSG-CPTWG. The DataHiding SubGroup of the Copy Protection Technical Working Group, is an industry forum with participants drawn from content providers, consumer electronics and IT industries.

[3] Besides, if a string of B's becomes too long, the reference frame that they draw their motion estimation from, is too far in the past. Therefore correlation is bad and coding efficiency goes down dramatically.

If within a GOP we denote a P-frame as the bit "0" and a B-frame as the bit "1", every GOP has a one-to-one relationship to a binary sequence, e.g. $IBBPBBPBBPBB \equiv 11011011011$. Taking into account the requirements in the previous paragraphs, a PTY-alphabet was constructed as a Hamming-code with the following properties (numbers refer to the 4 items in the previous section):

- To accommodate item 1: ($\approx$ 6 bits/GOP), all *valid* PTY-marked GOPs should fall into 1 of $2^6 = 64$ groups.
- To accommodate item 2, the GOP-length is fixed to 12 (11 Ps and/or Bs), and the number of B-frames should be close 6, i.e. every group has a representative or "code-word" with six "1"s.
- To deal with item 3 (scene-changes and random GOPs), we impose that each group is created from its code-word by flipping *at most* 1 bit. and thus code-words must have Hamming distance 4.
- Regarding item 1: we have to eliminate all words with a Hamming distance $<$ 4 away from the "standard" GOPs "11011011011" and "10101010101", which emanate typically from standard encoders, and thus represent *unmarked* material.

This yields an alphabet of 62 code-words in all, see table below:[4]

Theoretically, imposing picture types may have a minor effect on signal-to-quantization-noise ratio of the MPEG encoded image. Experiments however showed that there is no deterioration of the image quality, neither observable by the human eye, nor measurable with statistical significance. We compressed video at predetermined rates and decompressed it. This result was subtracted from the original image and the rms error was computed. This measurement was made for marked and unmarked video, see fig. 3.1. The standard MPEG encoding method (diamonds) does not give significantly different SNR values than the method that embedded PTY marks (squares). The average difference in error is close to 0 dB (crosses). However, locally, i.e., in particular frames, the SNR may be different. This very much depends on the relative alignment of scene changes with B and P picture types. There is no systematic tendency of the marked GOPs being worse that normal MPEG encoded GOPs. In some frames, the marked sequence has better SNR that a typical sequence.

3.2 Improved PTY-alphabets

False Positives

From studying various commercially released movies it appears that in the vast majority of cases the GOP-structures that are being used in the code above do not appear. There are, however, a few false positives: i.e. GOPs which have

[4] Note that in the table code-word GOPs are represented by their so-called *coding*-order, not their *display*-order, because a PTY-detector would receive frames in the former fashion.

code-word	*char.*	*code-word*	*char.*	*code-word*	*char.*	*code-word*	*char.*
11011100010	1	01101110001	2	10111011000	3	01010111100	4
00101101110	5	00011010111	6	10000111011	7	11001001101	8
11100010110	9	01110001011	10	10110100101	11	10110010011	12
01100111010	13	10001011110	14	11110001100	15	11000110101	16
00010101111	17	00111110100	18	10101101001	19	01101000111	20
11010101001	21	00111001101	22	01000011111	23	01110100110	24
11101001010	25	10100111100	26	10001100111	27	00011111010	28
11011010100	29	11101100100	30	10010110110	31	10100001111	32
00111100011	33	01110010101	34	01011001110	35	00001111101	36
01111010010	37	01001101011	38	11010000111	39	10011110001	40
10110101010	41	00100110111	42	11100011001	43	00110111001	44
11100100011	45	11001111000	46	01011100101	47	10011101100	48
11111000001	49	01101011100	50	10101110010	51	01001110110	52
01100101101	53	11110110000	54	00110011110	55	00101011011	56
10111000110	57	01111101000	58	10010011101	59	01010110011	60
11000101110	61	01111111111	62				

Table 1. List of the code-words (PTY-marked GOPS) and the characters that they represent.

a meaning in the PTY-watermark sense, but belong to an unmarked piece of video, viz.: *IBBPBBPBBPP*. To avoid these type of harmful detections, we have a number of options:

- Put a higher layer of error detecting code on top of the subliminal PTY-channel e.g. by adding a CRC to its bit-content.
- From a copy-protection point of view, a watermark is only valid if its bit-content is *compatible* with the bit-content of another mark, viz. that of the physical medium that carries the MPEG-stream. E.g. for DVD-copy protection, various disk-marks have been proposed that don't travel along to a new disk when a copy is made: for instance extra subliminal bits are hidden in the redundancy in the error-correcting code layer on disk, or the EFM+ (Eight-To-Fourteen Plus) channel-coding, or allowing the spiral track to be modulated sinusoidally (so-called wobble). The purpose of this physical mark is to verify that copyrighted content is on *original* media, and not on (forged) ROM-disks or even RAM-disks.

False Negatives

Another issue is that of false negatives: i.e. a bit of content should be watermarked, but a detector cannot find PTY-marks in it as the MPEG-encoder was unable to put in PTY-marked GOPs. This may happen for instance:

- at scene changes. One practical comment might be that scene changes represent little commercial value and might therefore go unprotected by temporarily suspending PTY-marking.

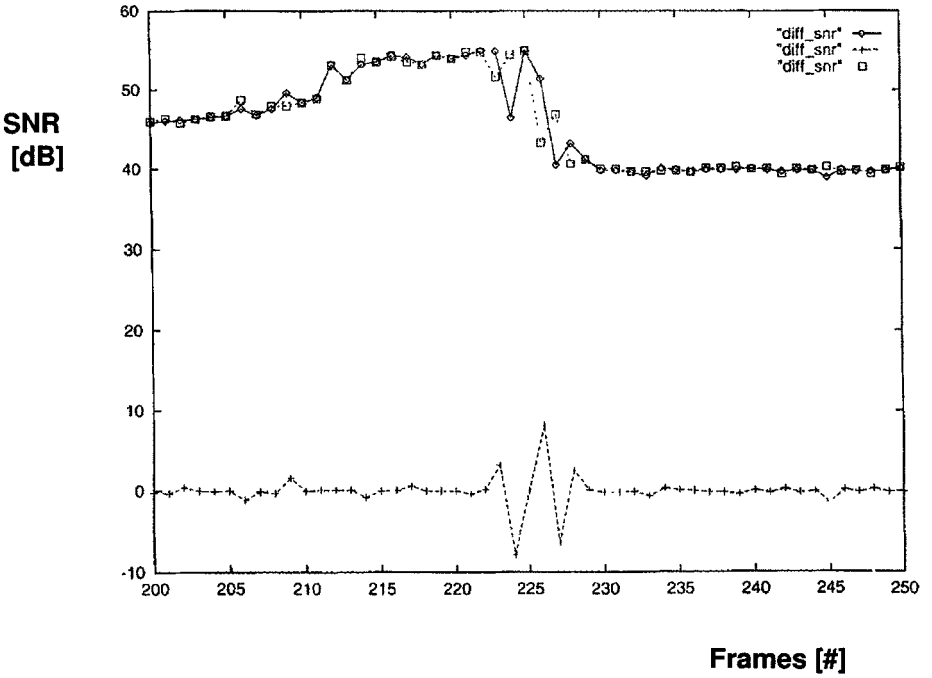

Fig. 5. Mean squared difference between the original video sequence and an MPEG compressed/decompressed sequence vs. frame-number for PTY-marked (squares) and "standard"-GOP MPEG material (diamonds). The crosses show the difference between the two.

- due to the fact that a PTY code-word in table above, may contain approximately "standard" numbers of P- and B-frames, but with sometimes very non-uniform distributions, e.g. long strings of Bs, which would might locally lead to bad compression efficiency.

To deal with the last issue: one way to stray not quite as far from standard GOPs is to indicate the presence of PTY watermarked content by an alternating sequence of GOPs: $I\,(B^{n-1}\,P^2)^{m'}\cdots I\,(B^nP)^{m'}\cdots$; for $n=2$, $m'=2$ we would get

$$\begin{aligned} &I\underline{BP}PBPP\cdots\\ &I\underline{BBP}BBP\cdots\\ &I\underline{BP}PBPP\cdots\\ &I\underline{BBP}BBP\cdots\\ &I\underline{BP}PBPP\cdots \end{aligned} \tag{1}$$

m and n would be chosen the same as for the case where no watermark is embedded. The "$\cdots$", e.g. $BBPBBP$ can be left up to the encoder to optimize.

The most obvious choice for "$\cdots$" seems to be "$\cdots$"$= (B^n P)^{m-m'}$, the "no-watermark" ending resulting in a GOP of the original length. A conservative choice would be $m' = m/2$, half of the original GOP is sacrificed to the sync-watermark. This degree of freedom at the end of the marked GOPs should allow the encoder to ensure a video bit-rate/quality within specs. The detection of this "Watermark-Present"-sequence by the detector yields synchronization on the bit level. To allow us to achieve the same on a potential symbol level, we let the synchronizing sequence "count-down". E.g. for $n = 2$, $m' = 2$ we run through the following sequence of 6 GOPs:

$$\begin{aligned} &IBBPBBP\cdots \\ &IBPBBPB\cdots \\ &IPBBPBB\cdots \\ &IBPPBPP\cdots \\ &IPBPPBP\cdots \\ &IPPBPPB\cdots \end{aligned} \tag{2}$$

Note that the encoding complexity of this sequence of GOPs is no different than the sequence in (1), and still avoids 3 or more consecutive Bs. Accumulating such a sync-sequence with a statistically relevant length (say a threshold of 10 syncGOPs in 10 sec) would then indicate a watermark present.

For a few movies ("Four Weddings and a Funeral" and "When We Were Kings" (see Appendix) we analyzed the GOP-structure. Approximately 9% of the GOPs in these movies deviate significantly from their encoder-default $IBBPBBP\cdots$ and might be difficult to watermark. However if we look at the distribution of these 'Bad' GOPs, we see that they tend to bunch together in clusters, such that in a 10 second window, they regularly achieve a density of 20-35% (see Figure 3.2). We define a " Bad" GOP for these movies to be one that doesn't start with the 7 frames $IBBPBBP$. In Figure 3.2 we have integrated this data to show the relative importance of 10 sec. slots with a given percentage of "Bad" (i.e. difficult to watermark) GOPs.

Beyond GOPs indicating the presence of the PTY watermark, we also would like to embed bit-content. In NTSC with on average 26 GOPs/10 seconds (possible low of 17 GOPs/10 sec) we would have to encode on average approximately 3 bits/GOP (but more if we want to introduce error correction). The following 3 options are currently under consideration:

- For embedders that typically produce GOPs with $m \geq 4$: embed 3 bits via GOP $\equiv IA_1A_2A_3BBP$; A_i is either BBP (binary 0) or BPB (binary 1). Generalization to other values of n are trivial.
- After the sync-sequence (1) or (2), the GOP length is allowed to vary by -1, 0, or +1 frames, by adding a P-frame or taking out a B from standard GOP $IBBPBBPBBPBB$. Valid GOPs are those having P-frames spaced not more than n frames apart (as usual). Assuming that the encoder default is $n = 3$, $m = 5$, we obtain 35 symbols, i.e. about 5 bits.

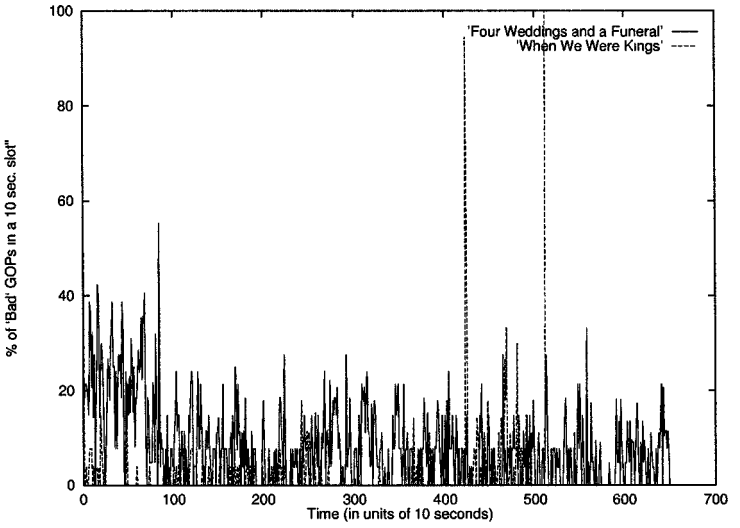

Fig. 6. The percentage of "Bad" GOPs in a 10 sec. slot as a function of the position of the slot in the movie. The data has been compiled from two movies (see Appendix). A "Bad" GOP is one that doesn't start with $IBBPBBP$. That sequence is the encoder-default for these two movies[6].

3.3 Complexity/SNR Analysis of Proposed GOPs

In this section we present data comparing the various non-orthodox GOP-structures on the issues of coding-difficulty and picture quality. Throughout this section the bitrate is held constant. We make this comparison based on the time it took various software MPEG encoders to encode pieces of test material with a particular "PTY-marked" GOP, and looking at the resulting SNR of the coded MPEG material. For video compression we used a public domain software MPEG-1 encoder compressing at predetermined rates from Berkeley [3]. The SNR was measured by decompressing the compressed video; this result was subtracted from the original image and the error was computed in the usual mean-squared sense and was normalized to 100 for the orthodox $IBBPBBP\cdots$-GOP structure. We took the encoding rate (in the terms of the number of processed frames per second of CPU-time) as a rough measure for the coding complexity introduced by forcing a particular GOP upon a video sequence. Comparison of the coding complexity and SNR of various GOP structures for a number of movie-clips and for various encoders can be found in table 3.3 below.

From this table it appears that PTY-watermarking does not create substantial visual artifacts to the image. Although theoretically, the method may have a minor effect on signal-to-quantization-noise ratio of the MPEG encoded image, the data do not bear this out. Also to the human eye, the difference between PTY-marked sequences and the canonical sequences are not observable.

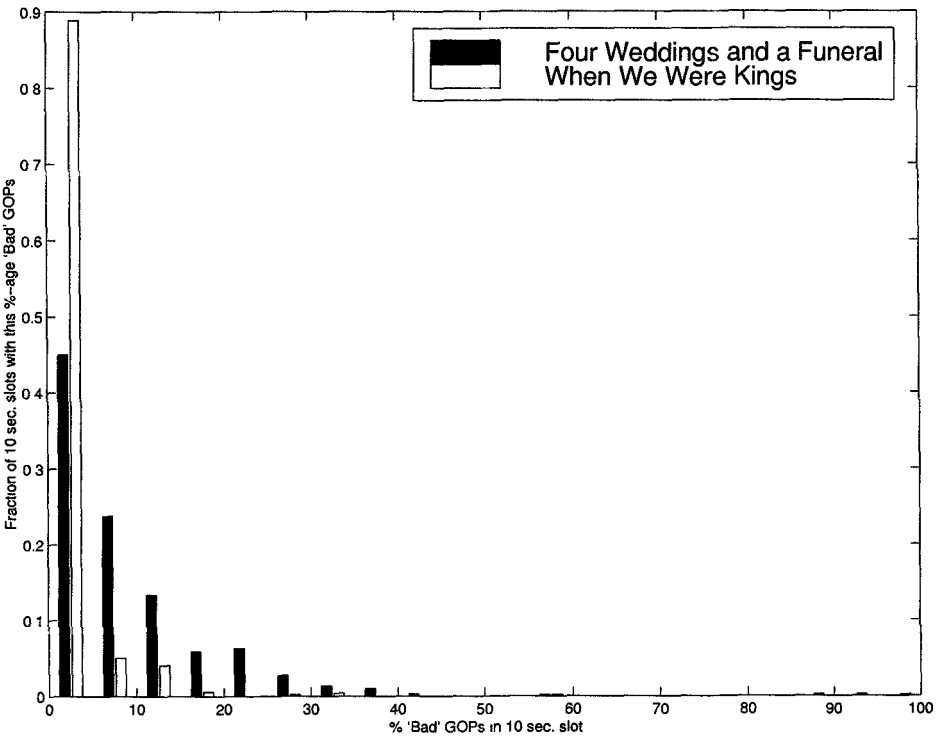

Fig. 7. The relative importance of 10 sec. time-slots with "Bad GOPs" as a function of the percentage of those Bad GOPs in the time slot. Based on the data in Figure 3.2.

4 Conclusion

PTY-Marks as advocated above, are based on the asymmetry in complexity between encoding a frame as a particular picture type vs. detecting that picture type. There have been attempts similar to ours to exploit this asymmetry for embedding information in particular, by carefully choosing the motion-estimation vectors [22].

Lacy et al. [23] have suggested a similar method that embeds a watermark in MPEG-*audio*. This mark can also only be removed by a complete decompression/compression cycle. According to their method a subliminal copyright messages is embedded in the LSB of the so-called "scale factors" of the scale-factor bands, in the MPEG-audio standard. The actual frequency components for which a scale factor determines the quantization accuracy, are compensated accordingly. This is done to ensure that when a scale-factor's LSB is altered from default, the change to the frequency components compensates this in a way such that the resulting decompressed signal differs from the unmanipulated one in

GOP Structure		*avg. SNR*	*peak SNR*	*Coding efficiency (frame/s, CPU-time)*
IBBPBBPBBPBB	(N=12 standard GOP)	100.0	158.1	0.381061
IBPPBPPBBPBB	(synch GOP)	100.2	155.8	0.442968
IBBPBBPBBPBBP	(GOP 1 longer)	100.4	159.4	0.395922
IBBPBBPBBPB	(GOP 1 shorter)	99.5	157.1	0.390396
IBPPBPPBPPBPP	(extreme synch GOP)	99.8	159.0	0.532198
IBBPBBPBBPBBPBB	(N=15 std GOP)	99.4	158.5	0.378430
IBBPBBPBBPBBPB	(GOP 1 shorter)	99.2	157.1	0.384966
IBBPBPBBPBBPBB	(GOP 1 shorter)	99.3	158.1	0.386660
IBBPBPBBPBBPBPBB	(GOP 1 longer)	99.7	159.0	0.392643
IBBPBBPBBPBBPBBPBB	(N=18 std GOP)	99.3	159.4	0.378108
IBBPBBPBBPBBPBBPB	(GOP 1 shorter)	99.2	159.0	0.383374
IBBPBPBBPBBPBBPBB	(GOP 1 shorter)	99.6	159.0	0.386100
IBBPPBBPBBBPBBPBPB	(random GOP)	99.0	159.0	0.393895

Table 2. PTY-marked GOPS, their coding complexity and SNR.

a manner that lies below the perceptual threshold[7]. This marking takes a very small toll on the bandwidth (order .2% – .4%).

Trying to remove this type of watermark by, say, randomizing the LSB of the scale factors, generates by definition quantization errors which lie *above* the perceptive threshold for hearing, unless the compression coefficients are manipulated along. This latter case for audio corresponds to a complete decompression/compression cycle, just like for PTY-mark removal (see Box). This method can be trivially extended to MPEG-video by manipulating the LSB of the scale-factors that set the quantization accuracy for DCT coefficients in a macroblock. Contrary to PTY-marks, to remove the subliminal message in the video, scale-factors only requires a *partial* MPEG encode/decode, as motion-estimation does not have to be performed again.

Another obvious extension of these PTY-marks can be made using MPEG-4: in that compression standard, temporal redundancy in video is not reduced on the level of macroblocks, but rather by dissecting frames into a collection of Video Objects (VOs, rather like "sprites" from the gone days of home computers) and coding their motion in front of a stationary background. These objects in turn can be built from various Graphics Primitives, or alternatively described in terms of an (MPEG-1 compressed) bitmap or mesh and even class objects such as facial expression. We might say that MPEG-1, -2 video is a special case of MPEG- 4 where the Graphics Primitive is always rectangular (viz. the 16 × 16 pel Macroblock). Obviously compared to MPEG-1 and -2 the MPEG-4 improved compression comes at the cost of an encoding procedure that will be quite a bit more complex as the "search-space" is vastly greater. This complexity may

[7] E.g. imagine a frequency component in a scale-factor band with scale-factor f has value D which after quantization becomes $\lfloor D/f \rfloor * f$. Then *after* marking f changes to f' (f, f' may only differ in LSB) and D is quantized to $\lfloor D/f' \rfloor * f'$.

again be exploited for the purpose of embedding a watermark. As an example, one might imagine tying a "never-copy" status to encoding VOs using a mesh or a particular subset of graphics primitives; non- copyrighted material would be encoded using primitives not from that subset.

In summary we showed how one might exploit the complexity of an MPEG encoder to embed a watermark for digital video. This watermark can be detected with very minimal means (e.g. in a drive) forestalling complicated hardware to set up a cryptographically secure link to MPEG decoding logic. An interesting aspect of this PTY-mark is that, to some extent, it has public-key properties. Detecting the watermark is trivially simple (and cheap in terms of hardware requirements), whereas embedding and modifying it requires a substantial effort, namely that of MPEG encoding. Of course, the disadvantage of PTY marks is their inability to survive analogue transmission. As the barrier of MPEG encoding may not sufficiently protect carriers of copyrighted video material (such as DVD) by itself, the scheme is used as an "accelerator" for recognizing copyrighted content. The method illustrates how public-key watermarking may in future be achieved if compression and representation standards anticipate and accommodate this feature.

Acknowledgement

The authors would like to thank Bas v/d Heuvel, Erwin Kragt, Joost Smolders, Ludo Tolhuizen and Emanuel Frimout for discussions, inspiration and help.

References

1. L. F. Turner, "Digital data security system." Patent IPN WO 89/08915, 1989.
2. J.T. Brassil, N.F. Maxemchuk, S. Low, and L. O'Gorman, "Electronic marking and identification techniques to discourage document copying" , Proceedings of IEEE Infocom 94. Toronto, June 1994, pp. 1278-1287
3. Caronni G.: "Assuring Ownership Rights for Digital Images", Proceedings of Reliable IT Systems, VIS '95, Vieweg Publishing Company, Germany, 1995
4. K. Tanaka, Y. Nakamura, and K. Matsui, "Embedding secret information into a dithered multi-level image", in Proc, 1990 IEEE Military Communications Conference, pp. 216-220, 1990.
5. E. Koch, J. Rindfrey, and J. Zhao: "Copyright protection for multimedia data", in Proc. of the Int. Conf. on Digital Media and Electronic Publishing, 1994.
6. E. Koch, J. Zhao: "Towards Robust and Hidden Image Copyright Labeling", Proceedings IEEE Workshop on Nonlinear Signal and Image Processing, Neos Marmaras, June, 1995, pp. 452-455.
7. M. D. Swanson, B. Zhu, and A. H. Tewfik, "Transparent robust image watermarking", in IEEE Int. Conf. onImage Processing, 1996
8. G. B. Rhoads, "Indentification/authentication coding method and apparatus.", World Intellectual Property Organization, WIPO WO 95/14289, 1995.
9. W. Bender, D. Gruhl, N. Morimoto, "Techniques for Data Hiding", Proceedings of the SPIE, 2420:40, San Jose CA, USA, February 1995

10. O. Paatelma and R. H. Borland, "Method and apparatus for manipulating digital data works", WIPO Patent WO 95/20291, 1995.
11. L. Holt, B. G. Maufe, and A. Wiener, "Encoded marking of a recording signal." UK Patent GB 2196167A, 1988.
12. I. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for images, audio and video", in IEEE Int. Conference on Image Processing, vol. 3, pp. 243-246, 1996.
13. I. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "A secure, robust watermark for multimedia", in Information Hiding: First Int. Workshop Proc., R. Anderson, ed., vol. 1174 of Lecture Notes in Computer Science, pp. 185- 206, Springer-Verlag, 1996
14. C. I. Podilchuk and W. Zeng, "Image-Adaptive Watermarking Using Visual Models", IEEE Trans. on Selected Areas of Communications vol 16, pp 525-539, 1998.
15. J. J. K. O. Ruanaidh, W. J. Dowling, and F. Boland, "Phase watermarking of digital images", in IEEE Int. Conf. on Image Processing, 1996.
16. R. D. Preuss, S. E. Roukos, A. W. F. Huggins, H. Gish, M. A. Bergamo, P. M. Peterson, and D. A. G, "Embedded signalling", US Patent 5,319,735, 1994.
J.P.M.G. Linnartz and A.C.C. Kalker and G.F. Depovere, "Modelling the false-alarm and missed detection rate for electronic watermarks", Proc. of Second International Information Hiding Workshop, Portland, OR, 1998.
17. T. Kalker, "Watermark estimation through detector observations", Proc. of the IEEE Benelux Signal Proc. Symp., Leuven, pp 119-122, 1995.
I. J. Cox and J.M.P.G. Linnartz: "Public watermarks and resistance to tampering", ICIP 97.
M. J. J. Maes, "Twin Peaks: the Histogram Attack to Fixed Depth Image Watermarks", Proc. of Second International Information Hiding Workshop, Portland, OR, 1998
J.-P. Linnartz and M. van Dijk, "Analysis of the Sensitivity Attack against Electronic Watermarks in Images", *ibid.*.
18. J. L. Mitchell, W. B. Pennebaker, C. E. Fogg and D. J. LeGall, "MPEG Video Compression Standard", Chapman and Hall, New York, NY, 1997.
19. Linnartz, J.-P., Kalker, T., Depovere, G.: "Response to Call for Proposals of the Data-Hiding SubGroup of the CPTWG" Sept. 1997.
20. Linnartz, J.-P., Heuvel, S.A.F.A v/d, Smolders, J., unpublished, 1997.
21. MPEG-1 Video Encoder in Software (V1.5), Rowe, L.A et al, Computer Science Division-EECS, UC Berkeley, 1995, http://bmrc.berkeley.edu/projects/mpeg.
22. Jordan, F., Kutter, M., Ebrahimi, T.: "Proposal of a Watermarking technique to Hide/Retrieve Copyright Data in Video", Technical Report M2281, ISO/IEC JTC1/SC29/WG11 MPEG-4 meeting, Stockholm, Sweden, July 1997.
23. J. Lacy, S. R. Quackenbush, A. Reibman and J. H. Snyder, "Intellectual Property Protection Systems and Digital Watermarking", Proc. of Second International Information Hiding Workshop, Portland, OR, 1998.

Appendix: Statistics of GOP Structure for Various DVD-Pictures

Four Weddings and a Funeral (Polygram)

When We Were Kings (Polygram)

GOP	*# of occurrences*
IBBBBPBBPBB	1
IBBPBBPBBPBBBB	1
IPPP	1
IPPPPPPP	1
IPPPPPPPPPPPPP	1
IBBPBBPBBPBBPBBPPP	2
IBBPBBPPBB	3
IPPPPPPPPPPBB	5
IBBPBBPBBPBBPBBPP	6
IPPPPPPPPPP	9
IBBPBBPBBPBBPBBP	12
IBBPBBPBBPBBPBBPPPBB	14
IBBPBBPBBPBBP	18
IBBPBBPBBPBBPBBPBB	18
IBBPBBPBBPBBPBBPPBB	26
IBBPBBPBBPBBPP	27
I	38
IBBPBBPBBPBBPPP	51
IBBPBBPBB	87
IBBPBBPPPBB	102
IBBPPP	104
IBBPPBB	108
IBBPBBPBBPBBPPPBB	110
IBBPBB	113
IBBPBBP	120
IBBPBBPP	122
IBBPBBPPP	122
IPPBB	123
IBBP	125
IBBPPPBB	125
IBBPBBPBBPBBPPBB	131
IBBPBBPBBPP	134
IPP	135
IBBPP	147
IBBPBBPBBPPP	150
IBBPBBPBBP	158
IPBB	200
IBBPBBPBBPPBB	213
IP	248
IBBPBBPBBPPPBB	765
IBBPBBPBBPBBPBB	1385
IBBPBBPBBPBB	11708

Table 3. "Four Weddings and a Funeral": GOPs Ordered by Frequency

GOP	*# of occurrences*
I	38
IP	248
IPP	135
IPPP	1
IPPPPPPP	1
IPPPPPPPPPPP	9
IPPPPPPPPPPPPP	1
IPPPPPPPPPPPBB	5
IPBB	200
IPPBB	123
IBBBBPBBPBB	1
IBBPPP	104
IBBPP	147
IBBP	125
IBBPPBB	108
IBBPPPBB	125
IBBPBB	113 ↑ 1482 "BAD" GOPs
IBBPBBPPPBB	102
IBBPBBPPP	122
IBBPBBPPBB	3
IBBPBBPP	122
IBBPBBP	120
IBBPBBPBB	87
IBBPBBPBBPPP	150
IBBPBBPBBPP	134
IBBPBBPBBP	158
IBBPBBPBBPPBB	213
IBBPBBPBBPPPBB	765
IBBPBBPBBPPBB	213
IBBPBBPBBPBBBB	1
IBBPBBPBBPBBPPBB	131
IBBPBBPBBPBBPPPBB	110
IBBPBBPBBPBBPPP	51
IBBPBBPBBPBBPP	27
IBBPBBPBBPBBP	18
IBBPBBPBBPBBPBBPPPBB	14
IBBPBBPBBPBBPBBPPP	2
IBBPBBPBBPBBPBBPP	6
IBBPBBPBBPBBPBBPPBB	26
IBBPBBPBBPBBPBBP	12
IBBPBBPBBPBBPBBPBB	18
IBBPBBPBBPBBPBB	1385
IBBPBBPBBPBB	11708
Total:	16969

Table 4. "Four Weddings and a Funeral": GOPs Ordered by deviation from typical GOPs with $n = 2$, $m = 4$

GOP	*# of occurrences*
IBBBBPBBPBB	1
IBBPBBPB	1
IBBPBBPBBP	1
IBBPBBPBBPB	3
I	24
IBPBPBPB	30
IBBPBBPBB	96
IBBPBB	106
IPPP	127
IBB	135
IBBPBBPBBPBB	12512

Table 5. "When we were Kings": GOPs ordered by frequency.

GOP	*# of occurrences*
I	24
IPPP	127
IBPBPBPB	30
IBBBBPBBPBB	1
IBB	135
IBBPBB	106 ↑ 423 "BAD" GOPs
IBBPBBPB	1
IBBPBBPBB	96
IBBPBBPBBP	1
IBBPBBPBBPB	3
IBBPBBPBBPBB	12512
Total:	13036

Table 6. "When we were Kings": Ordered by deviation from typical GOPs with $n = 2$, $m = 4$.