EURASIP Journal on Applied Signal Processing 2002:2, 133–139
© 2002 Hindawi Publishing Corporation

Practical Challenges for Digital Watermarking Applications

Ravi K. Sharma

Digimarc Corporation, 19801 SW 72nd Avenue Suite 100, Tualatin, OR 97062, USA
Email: rksharma@digimarc.com

Steve Decker

Digimarc Corporation, 19801 SW 72nd Avenue Suite 100, Tualatin, OR 97062, USA
Email: sdecker@digimarc.com

Received 29 November 2001

The field of digital watermarking has recently seen numerous articles covering novel techniques, theoretical studies, attacks, and analysis. In this paper, we focus on an emerging application to highlight practical challenges for digital watermarking applications. Challenges include design considerations, requirements analysis, choice of watermarking techniques, speed, robustness, and the tradeoffs involved. We describe common attributes of watermarking systems and discuss the challenges in developing real world applications. Our application uses digital watermarking to connect ordinary toys to the digital world. The application captures important aspects of watermarking systems and illustrates some of the design issues faced.

Keywords and phrases: digital watermarking, spread spectrum watermarking, challenges for watermarking, watermarking tradeoffs, repetition code, smart toys, connected content.

1. INTRODUCTION

Digital watermarking provides a way to imperceptibly embed digital information into both digital (images, video, audio) and conventional (printed material) media content. Information contained within the watermark can be used to add value to a variety of applications [1] such as security, content protection, copy prevention, transaction monitoring, authentication [2], and so forth. A unique advantage of a digital watermark is that the information is imperceptibly bound to the original (cover or host) medium.

An emerging application of digital watermarking is that of connected content. In this application, traditional analog media such as printed content [3] are connected to the digital world using embedded digital watermarks. In this paper, we describe a novel connected content application, Smart Toy. In the Smart Toy concept, the play value of ordinary toys is enhanced using digital watermarks. The watermark transforms the toy into an extraordinary object. The watermark acts as an instrument to connect the toy (a real world object) to a digital entity (such as a computer or the Internet). Detecting the watermark is akin to recognizing the object. The digital entity can invoke a multitude of responses on recognizing the object.

The field of digital watermarking is characterized by active research, with numerous articles covering new techniques, theory, various attacks on watermarking techniques, robustness, and analysis. Given that the field is maturing rapidly, there needs to be at least an equal, if not greater, emphasis on the practical aspects of developing real-world watermarking applications. In this article, we focus on the Smart Toy application to highlight practical challenges for watermarking applications. Which watermarking technique to use? How to achieve a specific detection rate in limited time? How to balance watermark visibility with robustness? Our aim is to draw attention to these issues and the tradeoffs involved. To illustrate the challenges of practical applications and the tradeoffs, we use the Smart Toy application as a case study.

Each watermarking application has its own needs that determine the required attributes of the watermarking system and drive the choice of techniques used for embedding and detecting the watermark. Commonly discussed attributes of real world systems include: the many forms of robustness against distortion (either caused by commonplace processing operations or changes in geometry) and attack [4], visibility of the embedded mark [5], data capacity of the watermark, immunity of the detector to false alarms, and security. An attribute less commonly discussed, but very important for many real world applications is performance, that is, the speed of embedding and of detection of the watermark.

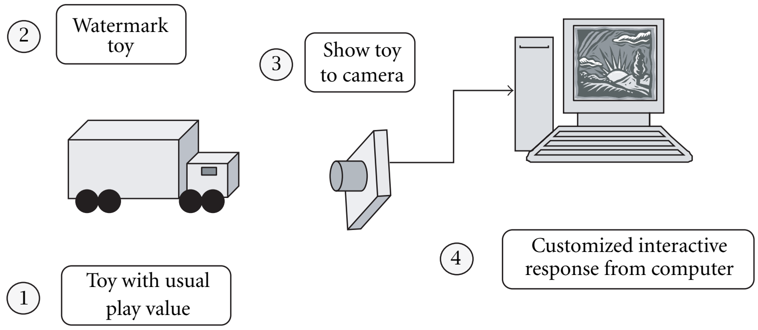
2
Watermark toy
3
Show toy to camera
4
Customized interactive response from computer
1
Toy with usual play value

Figure 1: Illustration of the Smart Toy concept.

There is an inherent tradeoff between many of these attributes that plays a critical role in a real-world application. At the detector, robustness, false positive rate and speed often compete with each other [6]. During detector design these attributes must be balanced, to meet the application requirements. For example, robustness to geometric distortions can be achieved at the cost of reduced speed. Application requirements also influence the mode of data acquisition at the detector. The data acquisition device often determines the choice of watermarking technology and its capabilities. In our application, a PC camera provides an easy interface for image capture; the user shows a toy to the camera.

At the embedder, the main tradeoffs are between visibility, capacity, robustness, and speed. The degree to which human intervention in the embedding process is permitted, impacts both speed of embedding and visibility. Capacity is always in tension with robustness. Watermark strength (energy of the embedded signal) also affects both visibility and robustness. The ability to automatically adapt visibility according to media characteristics [7] without sacrificing robustness (or some other set of attributes) is the foremost goal in embedder design.

The first task of application design is to determine the product requirements and use them to short-list and prioritize the various watermarking attributes that are necessary for this application. As illustrated by the Smart Toy application below, practical applications necessarily involve contradictory constraints and requirements that have to be traded against one another to achieve the intended goals and satisfy the customer's needs.

2. THE SMART TOY APPLICATION

The challenges and tradeoffs in developing watermarking applications are best understood by studying a real-world application. We describe a novel connected content application, Smart Toy, which touches upon several practical aspects of watermark application design and development.

A commercially successful watermarking application requires that the customer perceive value in the product, not in the technology. In Smart Toy, the watermark adds play value to a child's toy. This application aims to provide an interactive link between children's toys and the computer. Some aspect of the child's toy will contain a digital watermark that can inform the computer as to the nature of the object and to a lesser degree its location and orientation.

2.1. *The Smart Toy concept*

Figure 1 illustrates the concept of the Smart Toy application. The basic idea is to start with a child's toy such as a truck. The toy has its customary play value. Smart Toy involves digitally watermarking the toy. A simple way to achieve this is to affix a sticker that contains a digital watermark to a side of the toy. Now the ordinary toy is watermark-enabled. Once the toy is watermark-enabled it opens up infinite possibilities as far as what one can do with it. A child can take this watermarked toy and show it to a camera attached to a computer. The computer has watermark detection software running on it. The detector reads the watermark and based on the watermark takes some action. The action could be as simple as showing a video clip or it could involve a range of responses from the computer. These responses could be customized by the child or by parents and could even be interactive, involving responses from the child as well.

To summarize the above discussion, the Smart Toy concept consists of watermark-enabled toys, a computer with a camera that has a detector software running on it, and customizable actions that are associated with the detection of the watermark on the toy. Essentially the toy has become a physical object that the computer can respond to in an appropriate and customizable way. After understanding this concept, we can set out to state a set of requirements that will define a simple incarnation of the Smart Toy application.

3. REQUIREMENTS OF THE SMART TOY APPLICATION

We will now define a set of requirements for a simple embodiment of Smart Toy.

(1) The Smart Toy package will include a starter kit with software and one or two watermarked toys (such as vehicles). An add-on kit will contain more toys such as vehicles, buildings, stores, and other familiar neighborhood locales.

(2) The play action will consist of showing the toy to a camera connected to a PC. At the first instance of detecting the watermarked toy, the computer will retrieve a short video clip about the toy (either from a local database or the Internet).

Table 1: Requirements and their design implications.

Requirements	Implications
Play action consists of showing a watermarked toy to a camera	Robustness to geometric distortions, camera optics, lighting
Distinguish 100 toys from 50 manufacturers and age	Payload size
Should not affect frame-rate for unmarked objects	Detection speed
Less than 1 false positive per hour at 10 fps	False positive rate less than 1 in 100,000
Avoid one toy being detected as another	False reads less than 1 in 1000
Low cost	Small addition to base cost of toy

For example, the sounds of a fire engine and a short clip about firefighting would be played when a watermarked fire engine is first shown to the camera.

(3) One could customize the actions that would result from subsequent showings of the toy to the camera. For example, a different clip could be played each time. Or the action could be an interactive session with the computer chosen from among those supplied with the Smart Toy software. The action could also be programmed by the parents entering a desired response for the computer.

(4) The Smart Toy system needs to distinguish around 100 toys each from about 50 manufacturers and needs to carry information about the minimum age (3 to 7 years) for which it is intended.

(5) Smart Toy should not affect the frame-rate when an unmarked object is presented to the camera.

(6) When no watermarked object is presented to the camera, less than one false positive is permitted per hour of play.

(7) The probability of mistaking one toy as another should be low enough to give less than one mistake in every 16 hours of play.

(8) The cost of the Smart Toy enhancement to the toy should be low.

(9) There is no security requirement for this application.

(10) Camera and PC required are assumed to already exist at home.

(11) The watermark should not affect the aesthetic value of the toy.

3.1. Design implications of requirements

Table 1 states the design implications of some of the requirements described above. To satisfy the first requirement, the detector needs to be robust to geometric distortions as well as distortions caused by the camera optics and lighting variations. The requirement to distinctly identify toys and manufacturers and to carry the age information will influence the size of the payload in bits. The requirement that the detector should not affect the frame-rate for unmarked objects means that the speed of detection is crucial. At a frame-rate of 10 fps an unmarked frame must be rejected in 0.1 second or less. Note that there is no specific requirement regarding the detection speed for marked objects. However, for the response from the computer to seem natural, we will assume that a maximum detection time of 2 seconds could be taken for marked objects. It is desired that the false positive rate should be less than 1 in an hour of play. Assuming 10 fps, this translates into a false positive requirement of approximately 1 in 100,000. Another requirement states that the possibility of one toy being confused for another needs to be low. When one watermark is wrongly detected as another, we term it a false read. This requirement requires that the false reads be minimized. If we assume that, on average, 1 watermarked object will be shown to the camera per minute, then a target false read rate of 1 in 1000 will ensure that a false read will occur less than once every 16 hours of play. From the intended usage, it is clear that there is no security requirement for this application.

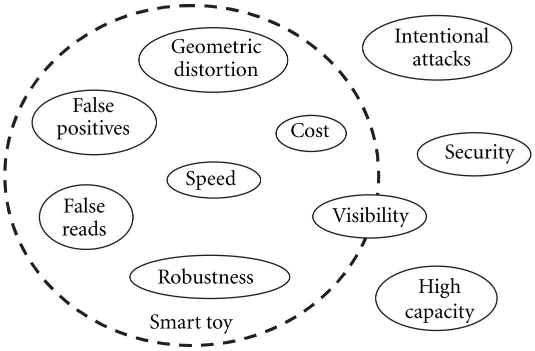

Figure 2: How Smart Toy requirements fit in the general space of watermarking requirements.

3.2. Smart Toy requirements as a subset of general watermarking requirements

Figure 2 is a schematic diagram showing how the requirements for Smart Toy relate to the space of all possible requirements for digital watermarking applications. Some of the most common important requirements for watermarking applications are represented in the diagram. The requirements for the Smart Toy application are enclosed within the dotted circle. The requirements outside the dotted circle are of no consequence to Smart Toy. For example, security and robustness to intentional attacks are important requirements for most watermarking systems. However, they are not essential for Smart Toy since it is an application where the watermark enables something rather than disabling or preventing something. Similarly, high capacity (or large payload size) is not required for Smart Toy. We just need to have a small number of bits that will satisfy the requirements. Notice that the visibility requirement is partially within the dotted circle and partially outside the dotted circle. This needs a little more elaboration. As in most watermarking applications, watermarking the toy should not degrade the aesthetic value of the toy. This dictates that the watermark be imperceptible. However, in this application we have some freedom to influence the artwork design (or host content) for watermarking. After all, these are toys that we are watermarking. One could easily choose or design artwork that would lend itself to holding a watermark without being perceptible, for example, artwork containing texture or busy regions.

3.3. Conflicts between requirements

As in any real world application, there are conflicts between some of these requirements. The emphasis on speed of detection directly conflicts with the robustness requirement. High robustness to distortions introduced by camera optics, geometry, and lighting variations, requires intensive signal processing operations to enhance and extract the watermark signal. Such operations are likely to be time (and memory) intensive. Similarly, the speed requirement also conflicts with the requirements to keep the false positive and false read rates low. Low rates for false positives and false reads can be achieved at the cost of additional processing of each frame that would reduce speed. The low false positive and false read targets also conflict with the robustness requirement. The detector must reject frames with low confidence of a watermark presence. As a result, the detector inadvertently discards frames that have weak, but recoverable, signal due to low embedding strength or large distortions. This would cause the detection rates to drop.

4. DESIGN CONSIDERATIONS

After understanding the requirements, the conflicts between them and their implications on the design, we can analyze how these requirements drive various design considerations.

4.1. Visibility of the watermark

One of the requirements of Smart Toy is that the watermark should not affect the aesthetic value of the toy. In other words, the watermark should be imperceptible. However, the fact that we are dealing with toys, and can influence the choice of graphics, can be used to advantage to reduce this imperceptibility requirement. Note that the graphic elements of a toy are the cover or host medium in this application. The contents of this host medium can be adapted to suit the watermark, thus minimizing the visibility impact of the watermark. For example, graphic elements could be modified to include textures or busy artwork that would help camouflage the watermark.

4.2. Data acquisition

To play the game, the child holds a watermarked toy up to a PC camera. This means that the detector has to deal with geometric distortions introduced because the location (rotation, scale, translation) and attitude (perspective distortion) of the image are not controlled. Also, a typical PC camera brings with it issues such as lens distortion, focus, compression, frame size, frame rate, sensor noise, and so on. In addition, there are problems caused by lighting variations and exposure. The detector has to be robust enough to deal with these issues.

The toy software can control camera settings such as frame rate, compression, exposure, and white balance as required. Given the characteristics of the installed base of PC cameras, the capture rate will be 5–8 frames per second to capture uncompressed images. A typical PC camera has a 640×480 pixel image sensor. Typical imagers have pixels about 9 μm on a side. At a typical focal length of 5 mm the pixels each subtend an angular distance of $\sim 2 \times 10^{-3}$ radians. This angular resolution sets the minimum meaningful size for a watermarking feature. At a working distance for the game of 20 cm, the minimum spatial extent of a resolvable feature is 4×10^{-2} cm. For robustness reasons, it may be advisable to oversample the watermarking information, leading to a larger minimum watermarking feature.

4.3. Robustness

As described in the considerations for data acquisition, the watermark should withstand distortions from camera capture, such as rotation, scaling, cropping, brightness adjustment, contrast enhancement, and lighting variations. Detection should be adaptive to camera-image distance. Detection should work on small watermarked areas on the toy. We will arbitrarily impose that the smallest watermark area that should be detected would be of size 4 cm by 4 cm. The watermark must be detectable under conditions that include soiling of the object.

4.4. Synchronization

Since presentation of the watermarked toy to the camera is not controlled, the detection process will require synchronization to align with the presented watermark signal. Ideally, synchronization should allow recovery of the payload from an image acquired at any angle of rotation about the camera's optic axis, for any distance within the focal zone of the camera, and with small pitch and yaw deviations from normality to the optic axis. A robust synchronization scheme is critical to the success and appeal of the toy. In practice the effective limits of the synchronization technique will dictate the exact style of play that Smart Toy will accommodate.

4.5. Payload, error correction, and spreading

The payload needs to have a sufficient number of bits to satisfy the requirements. In addition, we introduce additional bits to ensure future extensibility. The payload will contain the following fields to satisfy the requirements—toy ID (7 bits) that identifies the toy and the action, manufacturer ID (6 bits), intended minimum age (3 bits). Additionally, an open field (6 bits) is reserved for future use, giving a total of 22 payload bits.

The payload size and importance of the individual fields determine the error correction scheme and the amount of spread employed. We use a simple repetition code and spread spectrum processing to provide the error correction capability. There are two reasons for this strategy. First, the repetition code is very fast to decode and encode. Second, we anticipated that a repetition code in conjunction with spread spectrum processing would be able to provide reasonable robustness in the low signal-to-noise ratio environment caused by camera acquisition, geometric distortions, lighting variations, and the need to overcome soiling.

We will assume that all payload bits are equally important and repeat each bit 14 times to provide error correction. Each of the 14 repeats of each payload bit is further coded into 30 chips to give a total of 9240 bits. This process provides the spread spectrum robustness. To these 9240 bits, we append a 760-bit reference PN sequence derived from a key, to obtain a total of 10000 spread-spectrum bits.

4.6. Embedding the watermark signal

The watermarked image I' is obtained by embedding the watermark W in the original image $I, I' = f(I, g(I, W))$, where f is a function denoting the embedding operation, and g is a gain function that depends upon W and local and global image properties. These functions can either be linear or nonlinear. Embedding can be done either in a transform domain (e.g., frequency domain) or the spatial domain. For the Smart Toy application, we choose f to be an additive operation in the spatial domain. The watermark signal W consists of the spread spectrum bits combined with a synchronization signal, which can be a known pattern. Note that the spread spectrum signal itself can be designed to serve the dual purpose of a synchronization signal. The watermark signal is repeated in every $M \times N$ block of the image. A key determines the arrangement of the spread-spectrum and reference bits within the block. If the minimum camera resolvable feature has an extent of 4×10^{-2} cm, the minimum watermarked area of 4 cm by 4 cm would mean that 100 by 100 pixels are available in this area. Ideally, the spread-spectrum bits would be arranged such that each of them is uniformly available within this area to ensure desired robustness.

4.7. Watermark detection

Figure 3 shows the main steps taken during watermark detection. The detector has no knowledge of the original cover image. It obtains an estimate, $\hat{W}$, of the watermark signal from the watermarked image. The detector applies prediction techniques to estimate the original image from the watermarked one. The estimate $\hat{W}$ is then obtained by comparing the predicted image with the watermarked image. Now, $\hat{W}$ contains an estimate of the synchronization signal, an estimate of the spread spectrum payload, and remnants of the cover image. The detector then uses $\hat{W}$ and a knowledge of the synchronization signal to recover the geometry (rotation, scale, etc.) of the watermark. Before synchronization, the detector may apply pre-processing to suppress the unwanted components due to the image and the spread spectrum signal. Using the recovered synchronization, the detector proceeds to extract the reference bits and spread-spectrum payload.

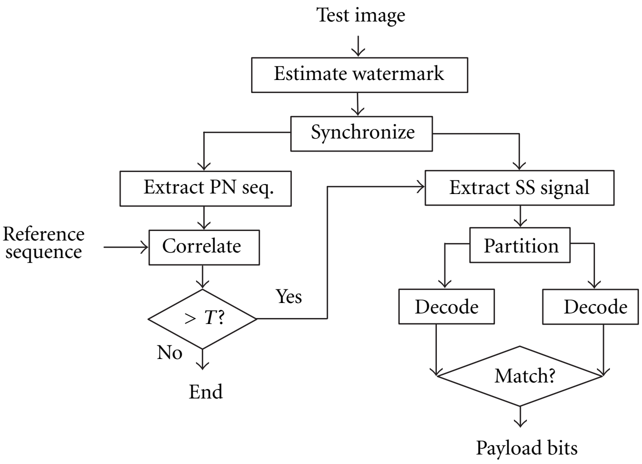

Figure 3: Overview of detector stages.

The estimated reference bits are used to determine if the recovered signal is a valid watermark by correlating with the reference PN sequence. Comparison of the estimated reference bits with the reference sequence can also be used to aid in robust recovery of the payload. The extracted spread-spectrum data is first de-spread and then partitioned into two equal sets. Each set is decoded independently to obtain the payload bits. A payload is declared valid when the payload bits from the two sets match identically. This process is described in greater detail in the discussion below on false positives. At the payload extraction stage too, the detector may pre-process $\hat{W}$ to further suppress components due to the image and the synchronization signal.

4.8. False positives and false reads

The first step in reducing the false positives is to correlate the estimated reference bits with the bits in the reference PN sequence. Frames with correlation less than a threshold T are rejected to reduce the number of false positives. The threshold T is chosen to give a false positive rate of less than 1 in 10^2. The reduction in false positives obtained using this method comes at the cost of a reduction in robustness. It is possible that some frames with a correlation less than T would have successfully extracted the payload. However, such frames must be rejected to meet the low false positive rate requirement.

Our technique for false read reduction is illustrated in Figure 4. A false read occurs when a payload is wrongly decoded as another. To reduce false reads, we partition the chips for the coded payload bits into two equal sets. These chips are de-spread and decoded independently. A read is declared valid only when the decoded bits from both sets match bit for bit. Assume that the probability of chip error is 0.4. If we assume that a 4 cm by 4 cm watermarked image will be presented to the detector, the above method for false read reduction gives a theoretical false read rate of approximately 1 in 10^3. Although partitioning helps to reduce false reads, the tradeoff here is that robustness is also reduced since fewer chips are available for each decode.

The method for rejecting false positives is independent

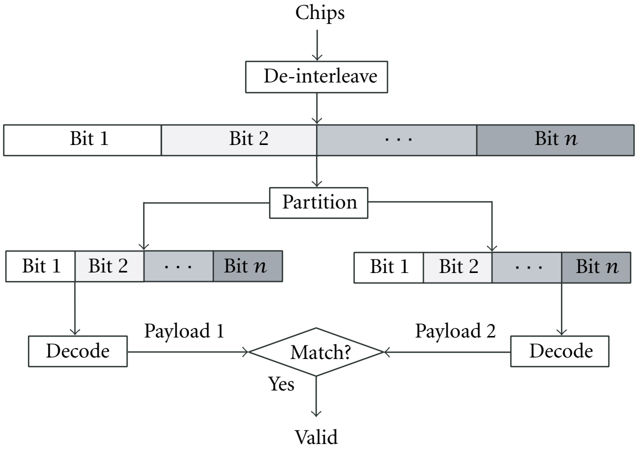

Figure 4: False read reduction.

of the method for rejecting false reads, since they are derived from independent signals. Consequently, we can combine the two rejection mechanisms to obtain the overall false positive rate for unmarked images. The combination gives an overall false positive rate of 1 in 10^5, which easily meets the requirement.

4.9. Detection speed

The detector gets a maximum of 100 ms to reject a frame that does not contain a watermark, implying that a fast decision must be made about the presence or absence of the watermark. When a watermark is present, more time is available for extracting the payload, since the Smart Toy system takes a dramatically different action upon reading the watermark. The requirement that a marked object be detectable with probability P_d within $T_{\max}$ seconds of presentation to the camera illustrates a key speed tradeoff in the design.

Let T_{avg} be the average time to detect a frame. The time T_{avg} can be determined by the frame rate of the camera and the speed of the detector over a distribution of capture conditions. The average number of frames, n, processed in $T_{\max}$ seconds, is given by $T_{\max}/T_{\text{avg}}$. Let P be the probability of detecting the watermark in any given frame. Then the speed tradeoff can be captured by the relationship $P_d = 1-(1-P)^n$. This relationship provides a tool to trade off the speed requirement with that of robustness. An interesting aspect of this relationship is that the required detection probability P_d can be achieved in two ways. One is to increase P by improving robustness at a cost of decreasing n. The second is to speed up the detector for a fixed P to ensure that n increases.

4.10. Back end and Internet connectivity considerations

The toy is playable using the local database of stored video clips provided with the toy software. Internet connectivity is not required. However, the play-value can be further enhanced by allowing a connection to the Internet for the download of additional sound and video clips, for registration of the toy, and for updating of the detection software. If Internet connectivity is available, then the sound and video clips the toy links to can be changed following the desires of the toy company. For example, for a small extra charge the toy could be enabled with a different response every week, or even upon each connection.

4.11. Cost

Cost is of paramount concern in the design of a toy or game. In this case, the cost increase over the base toy is small. The watermarking imposes no cost over the base toy, and the CD-ROM for the detection software is inexpensive. This should give a cost increase for the watermarking enhancement of less than US $1. By leveraging the existing PC and camera to implement the watermarking technology, the play-value of the toy is enhanced by far more than the cost.

5. TEST RESULTS

5.1. Digital images

The design discussed above has been implemented and tested against a total of 4430 marked (two embedding strengths each) and 10,000 unmarked images in digital form. The unmarked images gave no false positives. The marked digital images gave the results summarized in Table 2.

5.2. Images captured with PC camera

When the digital images were printed and detected through a PC camera, the detection rates were lower than those of the digital image sets. The detection rates on the camera-acquired frames were in the range of 60% to 75% depending upon the image. For frames acquired through the camera, detection rates were measured as the number of frames successfully detected as a percentage of the total number of frames presented to the camera. The reason for the low detection rates are twofold—the printing of the marked digital images routinely causes a reduction in the signal-to-noise ratio. Moreover, camera acquisition and geometric distortions further reduce the signal-to-noise ratio.

The false positive rates were much higher than expected. The observed false read rate was about 6.6×10^{-3}. Again, the reason was that the chip error from camera capture was higher than the estimated rate of 0.4 used in the design. For example, a chip error rate of 0.45 (which is just a 12.5% increase over the assumed chip error rate of 0.4) would cause the calculated false read rate to change from 1×10^{-3} to about 6.2×10^{-3} in our false read reduction technique. One way of improving the false read rates in situations with high chip error rates would be to compare the decoded payloads from each of the partitions to the payload obtained by performing an additional decoding using all available chips from both partitions. For a chip error of 0.45, this additional step would result in a false read rate of approximately 1×10^{-4}.

6. CONCLUSIONS

Many traditional applications of digital watermarking have revolved around security-centered issues (copy prevention,

Table 2: Detection results on marked digital images.

Image format	Number of images	Strength	Detection rate	False Reads	Avg. correlation of PN seq.	Avg. chip error
JPEG	3428	1	78.24%	0	0.36	0.32
JPEG	3428	2	91.10%	1	0.48	0.26
TIF	1002	1	83.03%	0	0.38	0.31
TIF	1002	2	92.81%	0	0.50	0.25

content protection, authentication, etc.). These applications have demanding requirements in terms of detection and visibility and require robustness to intentional attacks. Watermarking research has therefore attempted to address these requirements. As the field of watermarking matures, new applications are likely to emerge. We have presented one such application, Smart Toy, based on the concept of connected content. Physical objects, toys in this application, are watermarked acquiring a digital identity. A computer can detect the watermark on the physical object, thus allowing interaction with the digital world. The Smart Toy application is but one of many possible applications to exploit this connection, requiring high robustness but low concern for security.

Digital watermarking is a complex technology necessarily involving many conflicting requirements and tradeoffs. We have discussed the design process for a real-world watermarking application. The design process is dramatically simplified if the requirements of the intended product are carefully understood. A successful watermarking application only needs to satisfy the requirements of the product, not all possible requirements. Once realistic constraints on the system are established, the application may move ahead. As shown in the case study of the Smart Toy application, through careful design, product value can be enhanced by the judicious application of watermarking technology. In this case, a simple toy is dramatically enriched in play-value through the use of watermarking technology to link the physical and virtual worlds.

ACKNOWLEDGMENTS

The authors would like to thank Burt Perry, Brian MacIntosh, Joel Meyer, Ammon Gustafson, and Hugh Brunk of Digimarc Corporation for insightful suggestions.

In the original presentation of this material at MMSP01, the case study was accompanied by a demo. The authors would like to thank Megan Livermore, Dale Fowler, Dan Ramos, Eliot Rogers, Brett Bradley, Bryan Morris, and Matt Weaver for their help in pulling the demo together.

REFERENCES

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann, San Francisco, Calif, USA, 2001.

[2] B. Perry, S. Carr, and P. Patterson, "Digital watermarks as a security feature for identity documents," in *Proc. SPIE Optical Security and Counterfeit Deterrence Techniques III*, vol. 3973, pp. 80–87, January 2000.

[3] A. M. Alattar, "Smart images using Digimarc's watermarking technology," in *Proc. SPIE Security and Watermarking of Multimedia Contents II*, vol. 3971, pp. 264–273, San Jose, Calif, USA, January 2000.

[4] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *2nd International Information Hiding Workshop*, pp. 218–238, Portland, Ore, USA, 1998.

[5] R. B. Wolfgang, C. I. Podilchuck, and E. J. Delp, "Perceptual watermarks for digital images and video," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1108–1126, 1999.

[6] S. Decker, "Emerging considerations in commercial watermarking," *IEEE Communications*, vol. 39, no. 8, pp. 128–133, 2001.

[7] B. T. Hannigan, A. Reed, and B. A. Bradley, "Watermarking using improved human visual system model," in *Proc. SPIE Security and Watermarking of Multimedia Contents III*, vol. 4314, pp. 468–474, San Jose, Calif, USA, January 2001.

Ravi K. Sharma received his B.E. in Electrical Engineering from the University of Bombay, India, in 1992, and his M.S. and Ph.D. in Electrical Engineering from the Oregon Graduate Institute (OGI), in 1997 and 1999, respectively. At OGI his primary focus was on multi-sensor image and video fusion. Since 1998 he has been with Digimarc Corporation where he is involved in the research and development of digital watermarking techniques. His interests include image/video processing, signal processing in adverse environments, pattern recognition, and computer vision.

Steve Decker received his B.S. (physics) in 1969 from Auburn University, his M.S. (physics) and his Ph.D. (Applied Physics) from the California Institute of Technology in 1971 and 1975. After a number of years developing electronic imaging systems, he joined Digimarc as Director of R&D in 1999. Current research interests include digital watermarking, image, and signal processing.