Applied Computer Science, vol. 20, no. 4, pp. 63–76
doi: https://doi.org/10.35784/acs-2024-40

Submitted: 2024-10-19 | Revised: 2024-11-22 | Accepted: 2024-11-28

Keywords: steganography, video steganography, human perception, undetectability, information hiding

Marcin PERY [0009-0005-1272-2048]*, *Robert WASZKOWSKI* [0000-0002-0170-227X]*

COMPUTATIONAL SYSTEM FOR EVALUATING HUMAN PERCEPTION IN VIDEO STEGANOGRAPHY

Abstract

This paper presents a comprehensive computational system designed to evaluate the undetectability of video steganography from human perspective. The system assesses the perceptibility of steganographic modifications to the human eye while simultaneously determining the minimum encoding level required for successful automated decoding of hidden messages. The proposed architecture comprises four subsystems: steganogram database preparation, human evaluation, automated decoding, and comparative analysis. The system was tested using example steganographic techniques applied to a dataset of video files. Experimental results revealed the thresholds of human-level undetectability and automated decoding for each technique, enabling the identification of critical differences between human and algorithmic detection capabilities. This research contributes to the field of steganography by offering a novel framework for evaluating the trade-offs between human perception and automated decoding in video-based information hiding. The system serves as a tool for advancing the development of more secure and reliable video steganographic techniques.

1. INTRODUCTION

Steganography, a discipline dedicated to concealing information within various media, traces its theoretical roots to Shannon's information theory introduced in 1948 (Shannon, 1948). The field was later established as a distinct area of research through Kahn's seminal work in 1967 (Kahn, 1967). The primary goal of steganography is to embed data into digital or analog covers while ensuring that the hidden information remains imperceptible to unintended recipients (Johnson & Jajodia, 1998).

The effectiveness of steganographic techniques is determined by their ability to balance three critical attributes: undetectability, capacity, and robustness (Fridrich, 2009). These characteristics are interdependent, requiring careful trade-offs during the design process to meet specific application requirements and address potential security threats. Undetectability refers to the capacity of a steganographic method to conceal information in such a way that it evades detection, either by human perception or statistical analysis. Capacity measures the

* Military University of Technology, Faculty of Cybernetics, Institute of Computer and Information Systems, Poland, Warsaw, marcin.pery@wat.edu.pl, robert.waszkowski@wat.edu.pl

amount of information that can be embedded without compromising the integrity of the cover medium. Robustness reflects the technique's ability to maintain the hidden information's integrity despite transformations such as compression, filtering, or format conversion.

Steganographic methods are employed across a variety of media, including text, images, audio, video, and network data (Petitcolas et al., 1999). The selection of an appropriate technique depends on the characteristics of the cover medium, the desired payload size, and the required levels of robustness and imperceptibility (Sencar et al., 2004). Ultimately, the success of a steganographic system relies on achieving an optimal balance between these properties while preserving the cover medium's integrity and evading detection (Cox et al., 2007).

1.1. Video steganography

Video steganography garners significant research interest because video files inherently offer large data capacities and complex temporal relationships, making them highly suitable carriers for covert communication. Unlike static images, videos enable more sophisticated embedding strategies that exploit frame-to-frame variations, motion vectors, and predictive coding, thereby increasing the difficulty of detection. Additionally, the dynamic nature of video data introduces more nuanced perceptual thresholds, allowing steganographic modifications to remain imperceptible not only to automated detection systems but also to human observers under various viewing conditions. As a result, video steganography presents both a robust framework for information hiding and a rich domain for exploring new techniques that balance imperceptibility, capacity, and computational efficiency (Majeed et al., 2024).

Video steganography techniques leverage the unique attributes of video data, such as its temporal dimension, to embed information in a manner that is imperceptible to both human and automated detection systems. Early methods in video steganography often adapted techniques from image and audio steganography, using individual frames as static covers for embedding data. Techniques such as Least Significant Bit (LSB) substitution (Chan & Cheng, 2004) and Discrete Cosine Transform (DCT) were among the first to be utilized (Kadhim et al., 2019).

More advanced approaches have since been developed, exploiting motion vectors, inter-frame prediction mechanisms, and frame-to-frame differences to encode information. These techniques enhance undetectability by leveraging the temporal relationships inherent in video sequences. The dynamic nature of video content and its inherent complexity provide additional layers of security, making video steganography a robust choice for covert communication (Kunhoth et al., 2023).

1.2. The problem of undetectability in video steganography

Undetectability in video steganography can be assessed from two perspectives: resistance to automated steganoanalysis and imperceptibility to the human eye (Huynh-Thu & Ghanbari, 2008; Wang et al., 2004). While much of the existing research has focused on developing methods that evade automated detection systems (Hossain et al., 2024), this study emphasizes human imperceptibility, particularly in scenarios where analog distortions

or transformations render automated systems less effective (Anderson & Petitcolas, 1998; Katzenbeisser & Petitcolas, 2000).

This work introduces a novel computational framework designed to evaluate both human perception and automated decoding of hidden messages. The system simultaneously assesses two critical performance metrics: the threshold at which hidden information becomes perceptible to the human eye and the effectiveness of automated steganoanalysis tools in detecting hidden data. By comparing these metrics, the system provides a comprehensive evaluation of the steganographic technique's undetectability from both perspectives.

This dual analysis offers valuable insights into the effectiveness of various video steganographic techniques. Specifically, it determines whether a given method achieves sufficient undetectability for human observers while maintaining baseline robustness against automated detection. By focusing on human perception as the primary criterion for evaluating undetectability, this study advances the development of steganographic techniques optimized for real-world scenarios.

The proposed framework not only enhances the understanding of human perceptibility thresholds in video steganography but also provides a foundation for creating more resilient and imperceptible methods.

1.3. Research gap in video steganography

Despite substantial progress in developing increasingly sophisticated video steganographic methods, a notable gap persists in balancing human-level imperceptibility with automated detection resilience. Previous research has predominantly emphasized evading statistical or machine-driven steganoanalysis (Jangid & Sharma, 2017), dedicating less attention to the threshold at which steganographic modifications become perceptible to the human eye. In practice, this gap is especially pronounced in scenarios where video content undergoes analog transformations-such as screen capture or re-recording-which can degrade the statistical features exploited by automated systems but may introduce perceptual artifacts detectable by human observers.

Moreover, although various techniques leverage advanced embedding strategies (e.g., motion vectors, inter-frame differentials, and predictive coding), they often lack a rigorous, empirical framework that explicitly quantifies the interplay between automated detection rates and human perceptibility thresholds. The absence of such a framework makes it challenging to evaluate how effectively a given steganographic method maintains imperceptibility in dynamic, real-world contexts, where lighting conditions, display characteristics, and other environmental factors can amplify perceptual cues.

To address these shortcomings, the present work introduces an integrated computational system capable of systematically measuring the human perceptibility threshold (HumanLevel) alongside the minimum embedding level required for successful automated decoding (AutomaticLevel). By providing parallel assessments of both metrics, this approach not only furnishes a more complete understanding of undetectability but also enables targeted refinements of steganographic algorithms to achieve optimal trade-offs between imperceptibility and decode robustness. Through this dual-focus analysis, the study advances the field toward practical, real-world-ready video steganography that effectively balances the demands of both human and machine adversaries.

2. THE PROPOSED SYSTEM

2.1. Assumptions

The proposed system is designed to operate under the assumptions shown in Tab. 1.

Tab. 1. Assumptions of the proposed system

Input data. The system receives the following inputs:	
1.	A steganogram in the form of a video file, identified by a unique file identifier.
2.	The identifier of the steganographic technique used for encoding.
3.	A numerical parameter ranging from 0 to 5 that specifies the encoding level of the steganogram (0 indicates the original, unmodified cover - no steganographic encoding applied, 1 to 5 represent increasing levels of encoding, where higher levels introduce more significant changes to the cover).
4.	A unique identifier for each participant evaluating the steganogram's quality (i.e., the visibility or invisibility of steganographic modifications).
Output data. The system produces the following raw output:	
1.	The identifier of the evaluated steganogram, including its encoding level.
2.	The information on whether the participant identified the video as steganographic or not.
3.	An indicator of whether the hidden message in the steganogram can be automatically decoded at the given encoding level.
Statistical analysis. The system provides aggregated results, including:	
1.	The maximum encoding level imperceptible to the human eye for a given steganogram, calculated based on the specified confidence percentile.
2.	The minimum encoding level required for successful automatic decoding of the hidden message.

The system requires encoding and decoding functions specific to each steganographic technique under investigation. The encoding function embeds a secret message into the source cover retrieved from a database, while the decoding function is executed whenever a participant evaluates a steganogram.

The system is agnostic to the underlying mechanisms of the steganographic functions and does not compare their operational specifics. Its primary purpose is to determine the encoding level at which modifications become perceptible to human observers and the level necessary for successful automated decoding of the hidden message.

2.2. System architecture

The system is composed of four primary subsystems:

1. Steganogram Database Preparation Subsystem - creates a database of steganograms from source covers,
2. Human Evaluation Subsystem - conducts experiments with human participants,
3. Automated Decoding Subsystem - tests the automatic decoding of hidden messages from the steganograms,
4. Comparative Analysis Subsystem - compares results from human evaluations and automated decoding to derive insights.

A schematic representation of the system architecture is provided in Fig. 1.

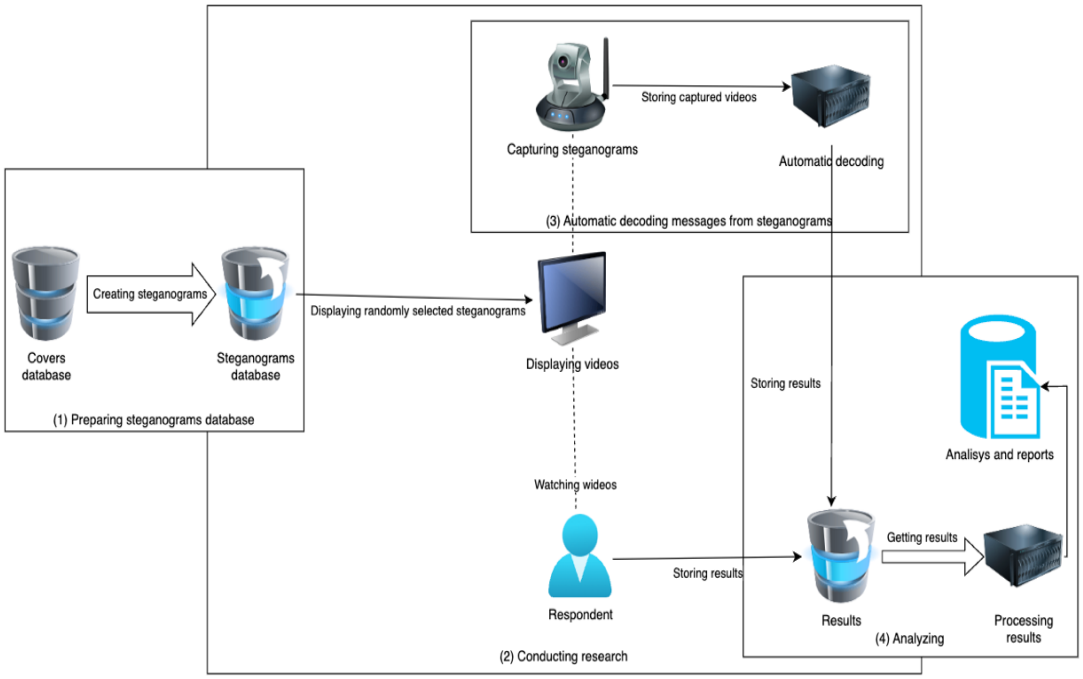

Fig. 1. System architecture

2.3. Creation of the steganogram database

In the research experiment, the preprocessing phase began with standardizing the source video files to ensure consistent resolution, frame rate, and format - an essential step for achieving reliable steganographic performance. All videos were transcoded into MP4 format using the H.264 codec with uniform frame rates and resolutions, thereby minimizing any inadvertent variability introduced by differing compression schemes or container formats. The steganographic embedding process itself was carried out using two distinct methods: one based on the Least Significant Bit (LSB) technique and the other employing a Discrete Cosine Transform (DCT) approach. These methods were implemented through custom Python scripts (see Fig. 5 and 6) leveraging libraries such as OpenCV and FFmpeg for frame-level manipulation and re-encoding. During the encoding phase, the predefined message was embedded at levels 1 through 5, where each incrementally higher level introduced more pronounced modifications to the video content. In contrast, level 0 was reserved for the unaltered source file. As a result of this procedure, the final steganogram database contained six times the original number of video files, each assigned a unique identifier of the form "CoverId_LevelId.mp4." This systematic preprocessing and encoding pipeline ensured that all generated steganograms were both comparable and reproducible, facilitating subsequent analyses of their imperceptibility and automated decodability. An example directory structure for the source and steganogram databases is shown in Fig. 2.

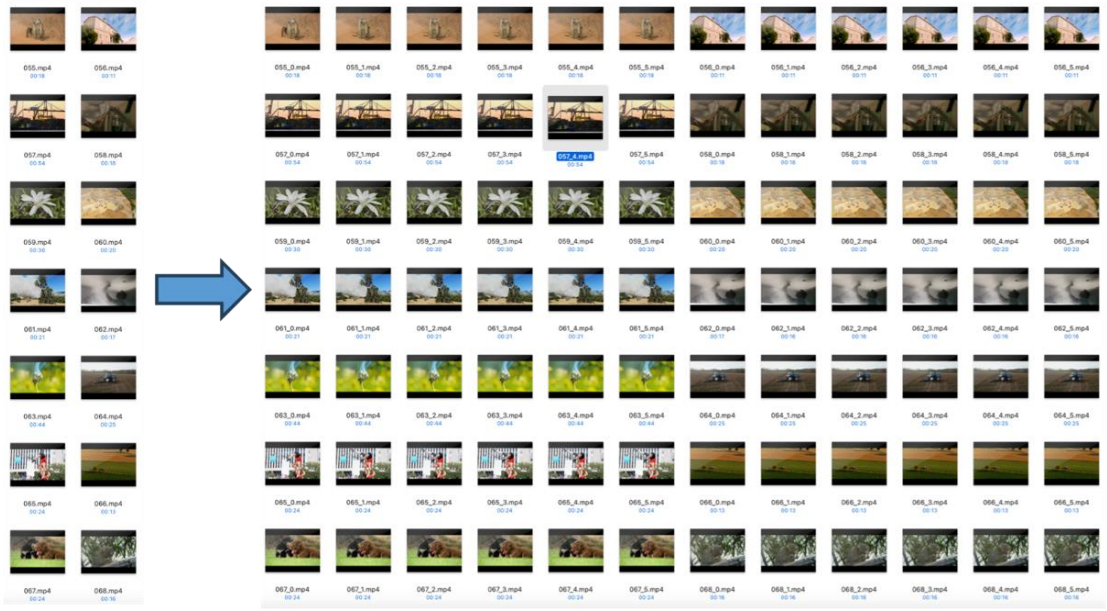

Fig. 2. Sample cover database and steganogram database

2.4. Experimental process

The experimental process operates as follows:

1. Parameters. Researchers specify:
   - the number of video files to be displayed,
   - whether to display only unique videos or allow repeated presentations of the same cover with different encoding levels.

2. Video Selection. The system randomly selects a sequence of videos from the steganogram database for each participant.

3. Video Display. Videos are displayed sequentially. After each video, participants are prompted to decide whether the video has been modified steganographically. The interface for participant input is shown in Fig. 3.

Each evaluation is recorded in the results database, including the video identifier, participant identifier, and the participant's decision (binary: modified or unmodified).

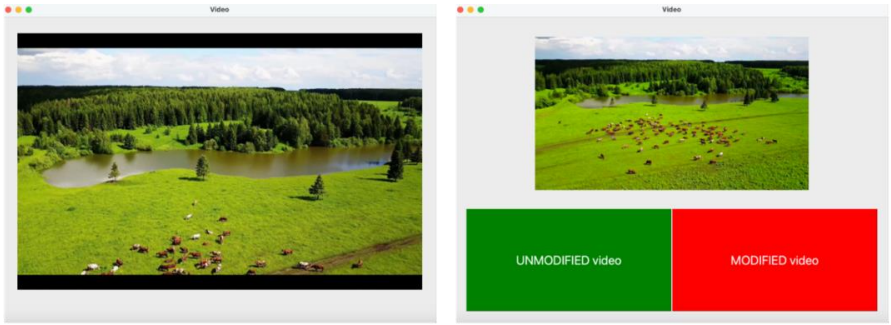

Fig. 3. Sample screens from interface of user application

2.5. Automated decoding process

Simultaneously with participant evaluations, a smartphone camera records the displayed videos and transmits the footage to a server. The server attempts to decode the hidden message using the decoding function of the steganographic technique under investigation. The decoding result (binary: success or failure) is stored alongside the participant evaluation data in the results database.

An example of the decoding process is illustrated in Fig. 4, where a QR code embedded in the video gradually becomes recognizable as more frames are analyzed.

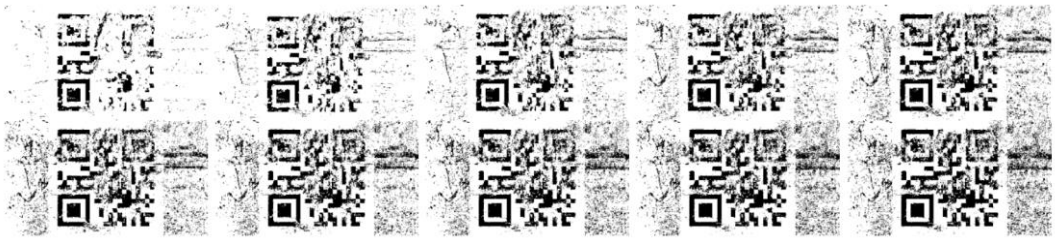

Fig. 4. Example of incremental increasing information in decoded message

2.6. Data analysis and results

After data collection, the system performs statistical analysis based on the following metrics:

1. Human-Level Undetectability (HumanLevel). Defined as the minimum encoding level at which more than 10% of participants correctly identify the video as steganographically modified. This threshold can be adjusted for different studies.
2. Automatic Decoding Threshold (AutomaticLevel). Defined as the minimum encoding level required for the decoding function to successfully extract the hidden message.

For each cover video, the system calculates HumanLevel and AutomaticLevel, storing the results in the output database. An example structure of database tuple is shown as follows: (PersonId, TechniqueId, CoverId, Level, HumanLevel, AutomaticLevel).

3. EXAMPLE OF SYSTEM OPERATION

The system was implemented and tested through a research experiment with the following setup:

- A source video database of 20 publicly available files from Pixabay.com was created, ensuring compliance with non-commercial use licenses (Pixabay.com, 2024). For this purpose, over 600 random files were downloaded from Pixabay.com, from various thematic categories and characterized by different dynamics and colors. Then, 20 files were randomly selected from the downloaded files, which constituted the basis for testing.
- Ten participants evaluated the videos to determine the visibility of steganographic modifications.

– The system was configured with two simple steganographic techniques: Technique 1 - LSB-based encoding in the spatial domain (the pseudocode is presented in Fig. 5) and Technique 2 - DCT-based encoding in the transform domain (the pseudocode is presented in Fig. 6).

We emphasize that very basic implementations of both techniques for encoding messages in individual video frames were used solely to verify the functionality of the proposed system, not to analyze their specific properties. Therefore, drawing substantive conclusions about the characteristics of these techniques based on the results presented in this study is unwarranted.

```
FUNCTION LSB_code(cover_image, message):
    INPUT:
        cover_image - Array representing the cover image
        message - Array representing the message to encode
    OUTPUT:
        steganogram - Array representing the encoded image

    # Step 1: Create a copy of the cover image to avoid
       modifying the original
    steganogram = Copy of cover_image

    # Step 2: Calculate the capacity of the steganogram
    capacity = Total number of elements in steganogram

    # Step 3: Convert the message into bits
    bits = Convert message to a bit array using unpackbits

    # Step 4: Check if the message fits within the image
       capacity
    IF length of bits <= capacity THEN:
        # Step 5: Pad the bit array with zeros to match the
       capacity
        padded = Pad bits with zeros to reach length equal to
       capacity

        # Step 6: Create a mask to clear the least
       significant bits of the image
        mask = Array of ones, same size as steganogram, with
       LSB set to 0 (0b11111110)

        # Step 7: Replace the LSB of each pixel in the image
       with the bits from the message
        steganogram = (steganogram AND mask) OR reshaped
       padded bits

    RETURN steganogram
```

```
FUNCTION LSB_decode(steganogram_image, size):
    INPUT:
        steganogram_image - Array representing the
       encoded image (steganogram)
        size - Size of the message to decode (in
       bytes)
    OUTPUT:
        message - Decoded message as an array of
       bytes

    # Step 1: Calculate the number of bits to
       decode
    num_bits = size * 8

    # Step 2: Extract the least significant bits
       from the steganogram image
    bits = Perform bitwise AND operation between
       steganogram_image and 1

    # Step 3: Flatten the bit array and select
       only the required number of bits
    message_bits = Flatten the bits array and take
       the first num_bits elements

    # Step 4: Convert the selected bits back into
       bytes
    message_bytes = Use packbits to convert
       message_bits to bytes

    # Step 5: Return the decoded message
    RETURN message_bytes
```

Fig. 5. Technique 1 pseudocode example

```
FUNCTION DCT_code(cover_image, message):
    INPUT:
        cover_image - The cover image
        message - The message to encode
    OUTPUT:
        steganogram_image - The steganographic image with the
       message encoded

    # Step 1: Prepare the dimensions of the image to be
       multiples of 8
    height, width = Dimensions of cover_image
    WHILE height is not a multiple of 8:
        Increment height
    WHILE width is not a multiple of 8:
        Increment width

    # Step 2: Resize and convert the image to YCrCb space
    cover_image = Resize to (width, height) and convert
       cover_image to YCrCb color space

    # Step 3: Convert the message to a bit array
    bits = Convert message to bits using unpackbits
    bit_count = 0
    data_len = Length of bits

    # Step 4: Initialize an empty array for the steganogram
    steganogram = Empty array with the same dimensions as
       cover_image

    # Step 5: Process each channel (Y, Cr, Cb) separately
    FOR each channel_index in [0, 1, 2]:
        channel = Split the channel into 8x8 blocks

        # Step 6: Perform DCT and quantization on each block
        dct_blocks = Perform DCT on each block
        dct_quants = Quantize each DCT block using the
       quantization matrix

        # Step 7: Perform zigzag scanning on quantized blocks
        coefficients = Zigzag scan each quantized block

        IF channel_index == 0:
            # Step 8: Encode message bits into DCT coeff
            FOR each block in coeff:
                FOR i = 1 to Length of block:
                    IF bit_count >= data_len:
                        BREAK
                    IF the coeff is > 1:
                        IF the current bit in bits is 1:
                            Set LSB of the coeff to 1
                        ELSE:
                            Set LSB of the coeff to 0
                        Increment bit_count
            IF bit_count < data_len:
                Error

        # Step 9: Perform inverse zigzag scanning
        desorted_coeff = Inverse zigzag scan each block

        # Step 10: Dequantize and perform IDCT on blocks
        dct_dequants = Dequantize each block using the
       quantization matrix
        idct_blocks = Perform IDCT on each block

        # Step 11: Combine the blocks back into a single
       channel
        steganogram[:, :, channel_index] = Combine the IDCT
       blocks into a channel

    # Step 12: Convert the image back to BGR color space
    steganogram = Convert steganogram from YCrCb to RGB color
       space
    steganogram_image = Clip the values of steganogram to the
       range [0, 255] and convert to uint8

    RETURN steganogram_image
```

```
FUNCTION DCT_decode(steganogram_image, size):
    INPUT:
        steganogram_image - The steganographic
       image (as a NumPy array)
        size - Size of the encoded message to
       decode (in bytes)
    OUTPUT:
        message - The decoded message (as a NumPy
       array)

    # Step 1: Convert the image to YCrCb color
       space
    steganogram_image = Convert steganogram_image
       to YCrCb color space

    # Step 2: Split the luminance channel (Y) into
       8x8 blocks
    channel = Split the Y channel into non-
       overlapping 8x8 blocks

    # Step 3: Perform DCT on each block
    dct_blocks = Perform DCT on each block in the
       channel

    # Step 4: Quantize the DCT coefficients
    dct_quants = Quantize each DCT block using the
       quantization matrix

    # Step 5: Perform zigzag scanning on quantized
       blocks
    coefficients = Zigzag scan each quantized
       block

    # Step 6: Extract the least significant bits
       (LSBs) from DCT coefficients
    bits = Extract LSBs from coefficients starting
       from the second element (skipping DC term)
           ONLY IF the coefficient is greater than
       1

    # Step 7: Assemble bits into a message
    message = Convert the first (size * 8) bits
       into bytes using packbits

    RETURN message
```

Fig. 6. Technique 2 pseudocode example

The hidden message was a QR code (DensoWave, 2024) containing the text “Grom,” encoded using OpenCV. Example QR code specifications are shown in Fig. 7.

Fig. 7. Hidden message coded in steganograms as a QR code

3.1. Experimental results

Results for Technique 1 are illustrated in Fig. 8, while results for Technique 2 are presented in Fig. 9.

CoverId	AutomaticLevel	HumanLevel	Diff
1	5	2	-3
2	5	2	-3
3	4	1	-3
4	4	1	-3
5	5	2	-3
6	5	3	-2
7	4	4	0
8	5	3	-2
9	5	3	-2
10	5	3	-2
11	5	3	-2
12	4	3	-1
13	4	3	-1
14	4	3	-1
15	4	3	-1
16	3	4	1
17	5	2	-3
18	5	3	-2
19	5	3	-2
20	4	3	-1

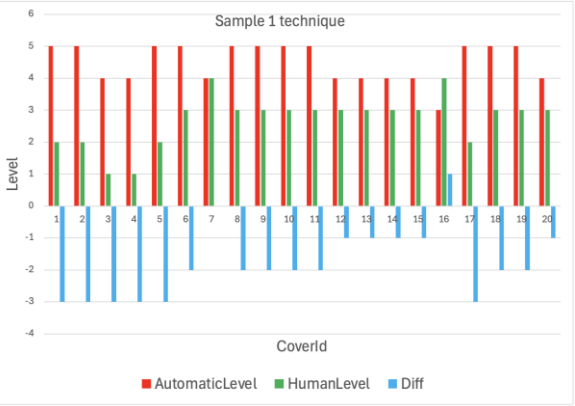

Fig. 8. Results for technique 1

CoverId	AutomaticLevel	HumanLevel	Diff
1	4	5	1
2	4	4	0
3	5	4	-1
4	5	4	-1
5	5	6	1
6	5	5	0
7	5	5	0
8	5	4	-1
9	5	4	-1
10	4	4	0
11	4	4	0
12	5	5	0
13	5	6	1
14	5	5	0
15	5	5	0
16	5	5	0
17	4	5	1
18	4	5	1
19	4	5	1
20	4	2	-2

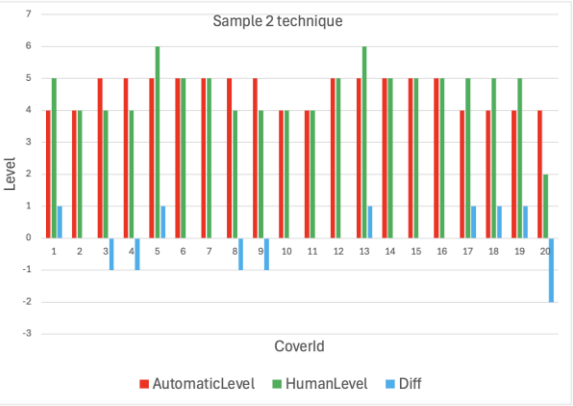

Fig. 9. Results for technique 2

3.2. Discussion

The experimental results provide a multifaceted view of undetectability in video steganography, underscoring the efficacy of our proposed system in revealing both the human perceptibility threshold (HumanLevel) and the minimum encoding level required for automatic decoding (AutomaticLevel). By systematically quantifying these two parameters, we have derived a data-driven means of assessing the trade-off between robustness against automated detection and imperceptibility to human observers. The resultant Diff (HumanLevel – AutomaticLevel) serves as a valuable indicator of how effectively a steganographic method balances these often-competing objectives.

First and foremost, the study confirms that conventional assumptions about steganographic performance can be overly narrow if they rely solely on automated steganoanalysis. While these automated techniques typically focus on statistical anomalies in the spatiotemporal or transform domains, they might fail to capture subtle perceptual cues that are evident to human participants. This outcome is particularly salient in scenarios where steganograms undergo analog transformations (e.g., screen capture by a smartphone camera) that degrade or alter the statistical artifacts commonly exploited by automated methods). As a result, the human visual system may become the more sensitive "detector" in real-world usage contexts, stressing the importance of foregrounding human-level perceptibility in steganographic evaluations.

Moreover, the fact that negative or zero Diff values surfaced in some trials indicates that under certain encoding levels, participants were able to detect modifications before or at the same threshold required for successful automated decoding. This finding underscores a potential vulnerability in the steganographic methods tested, as it implies that even a modestly trained or attentive human observer could accurately flag a video containing hidden information at lower embedding intensities than those necessary for reliable message recovery. Such results offer direct feedback to steganographers aiming to optimize their techniques. Specifically, they highlight a pressing need to refine embedding algorithms to remain imperceptible to human vision while simultaneously ensuring that the hidden message can be robustly retrieved.

From a methodological standpoint, the proposed system's architecture - encompassing database preparation, human evaluation, automated decoding, and comparative analysis - demonstrates strong adaptability and scalability. By decoupling the steganographic technique from the core data collection and analysis framework, researchers can substitute or integrate newer, more advanced methods without substantially altering the experimental pipeline. As innovation in the domain of neural-network-based steganography and deep-learning-based steganalysis accelerates, it will be relatively straightforward to incorporate these novel approaches into the current system architecture. This flexibility is crucial given the rapidly evolving nature of digital video formats and transmission infrastructures.

Despite these positive outcomes, it is essential to acknowledge certain limitations of the present study. First, the experimental dataset comprised only 20 videos, which may not fully reflect the diversity of video content, compression artifacts, and motion characteristics encountered in practical usage. The use of simple steganographic methods - limited to LSB and basic DCT-based embedding - further restricts the generalizability of the findings. Although these techniques suffice for demonstrating the concept and the system's functionality, the insights gleaned would likely evolve once more sophisticated embedding

approaches (e.g., advanced transform-domain methods, adaptive or key-based embedding) are tested. Additionally, the participant pool of ten individuals might not capture the full variability in human visual sensitivity; expanding the demographic and size of the participant group would more robustly validate the system's findings.

Looking ahead, the results open several exciting avenues for future research. One promising direction lies in refining the human evaluation protocol to include psychophysical experiments that quantify just-noticeable differences (JND) (Stern & Johnson, 2010) for various types of artifacts introduced by steganography. Controlled laboratory settings that account for viewing conditions such as screen brightness, resolution, and motion patterns can offer deeper insights into why certain participants detect specific distortions while others do not. Furthermore, incorporating eye-tracking technologies could elucidate which regions of the video draw the most scrutiny and whether steganographic content in these regions is more easily identifiable.

Another critical extension involves systematically exploring the resilience of steganograms to lossy transformations beyond the smartphone camera capture scenario tested in this study. Real-world video content frequently undergoes compression, transcoding, and partial frame drops during transmission. By exposing steganograms to an expanded set of distortions - including heavy compression, random bit errors, or resolution scaling-researchers could observe whether these operations reduce or elevate the risk of human perception, as well as their implications for successful automatic decoding. Such investigations would help refine the design of robust, real-world-ready steganographic solutions.

Overall, the present work underscores the fundamental importance of balancing human imperceptibility and automated decodability. Video steganography systems can no longer prioritize automated detection resistance alone without accounting for the innate perceptual abilities of human observers. By introducing a framework that systematically captures and compares HumanLevel and AutomaticLevel for each video, this study advances a more holistic approach to evaluating steganographic performance. We anticipate that future research efforts will refine this evaluation protocol, expand the repertoire of tested steganographic techniques, and incorporate additional metrics (e.g., embedding capacity, computational overhead, resilience to signal processing operations) to build ever more imperceptible and robust video steganography methods.

4. CONCLUSIONS

This study introduced a novel computational system for evaluating the imperceptibility of steganographic modifications in video files by integrating human perception tests with automated decoding processes. Through the dual assessment of HumanLevel (the encoding level detectable by human observers) and AutomaticLevel (the encoding level required for successful machine-based decoding), the system offers a comprehensive framework for quantifying the balance between undetectability and functional robustness. The experimental findings underscore the significance of identifying potential discrepancies between human perception and automated decoding, particularly in scenarios where human observers detect steganographic modifications more readily than machines can decode them.

Beyond confirming the need to consider both human and automated adversarial perspectives, this work highlights several promising research directions. One avenue is the refinement of experimental protocols to include psychophysical methodologies that measure just-noticeable differences and isolate specific conditions (brightness, resolution, and motion patterns) that influence detectability. Incorporating eye-tracking technologies could further elucidate how visual attention is distributed across video frames, offering deeper insights into where and why certain steganographic artifacts become perceptible.

Additionally, broadening the scope of evaluation to encompass real-world distortions-such as heavy compression, transcoding, random bit errors, and resolution scaling-will yield more realistic assessments of a technique's robustness in typical transmission and processing workflows. By simulating conditions encountered in practical settings, including video streaming platforms or mobile networks, researchers can systematically probe the vulnerability of steganographic systems to artifacts introduced by variable bandwidth, intermittent signal interference, and cross-platform transcoding. These investigations help refine methods to ensure reliable decoding while maintaining minimal visibility of embedded information, ultimately enabling more secure and imperceptible embedding of sensitive data across diverse real-world deployment scenarios.

The modular and adaptable architecture of the proposed system supports iterative testing of a wide spectrum of steganographic techniques under diverse settings. This flexibility not only facilitates the optimization of encoding strategies but also enables the exploration of emerging methods for information hiding. As the field progresses, incorporating additional metrics - such as embedding capacity, computational overhead, and resilience to signal processing - will be critical for advancing the practical adoption of video steganography.

Ultimately, the findings presented here underscore that robust and truly imperceptible video steganography must address both human perceptual thresholds and automated detection capabilities. By systematically comparing these two facets, the proposed system establishes a foundational platform for further research and development. Ongoing refinement of this framework will guide the creation of next generation steganographic methods that more effectively balance security with invisibility, thereby enhancing their suitability for real-world scenarios.

Author Contributions

The results of the work were obtained mainly by the first author under the scientific supervision of the second one.

Acknowledgments

The work was partially financed by the Military University of Technology in Warsaw, Poland as part of the project No. UGB 22-701.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

REFERENCES

Anderson, R., & Petitcolas, F. (1998). On the limits of steganography. *IEEE Journal on Selected Areas in Communications, 16*(4), 474-481. https://doi.org/10.1109/49.668971

Chan, C.-K., & Cheng, L. (2004). Hiding data in images by simple LSB substitution. *Pattern Recognition, 37*(3), 469-474. https://doi.org/10.1016/j.patcog.2003.08.007

Cox, I., Miller, M., Bloom, J., Fridrich, J., & Kalker, T. (2007). *Digital watermarking and steganography*. Morgan Kaufmann Publishers.

DensoWave. (2024). *QR code*. https://www.qrcode.com/

Fridrich, J. (2009). Steganography in digital media: Principles, algorithms, and applications. *Cambridge University Press*, 277-292. https://doi.org/10.1017/CBO9781139192903.014

Hossain, M. A., Noor, N. U., Ullah, A., Noman, S. H., Pranta, S. D., Bristy, L. M., & Bashar, M. (2024) Enhancing video steganography techniques, using hybrid algorithms. *Open Access Library Journal*, *11*, e11489. https://doi.org/10.4236/oalib.1111489

Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, *44*(13), 800-801. https://doi.org/10.1049/el:20080522

Jangid, S., & Sharma, S. (2017). High PSNR based video steganography by MLC(multi-level clustering) algorithm. *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 589-594). IEEE. https://doi.org/10.1109/ICCONS.2017.8250530

Johnson, N. F., & Jajodia, S. (1998). Exploring steganography, seeing the unseen. *IEEE Computer, 31*(2), 26-34. https://doi.org/10.1109/MC.1998.4655281

Kadhim, I. J., Premaratne, P., Vial, P. J., & Halloran, B. (2019). Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research. *Neurocomputing, 335*, 299-326. https://doi.org/10.1016/j.neucom.2018.06.075

Kahn, D. (1967). *The codebreakers: The story of secret writing*. Macmillan.

Katzenbeisser, S. & Petitcolas, F. (2000). Information hiding techniques for steganography and digital watermarking. *Artech House, 28*(6), 1-2. https://doi.org/10.1201/1079/43263.28.6.20001201/30373.5

Kunhoth, J., Subramanian, N., Al-Maadeed, S., & Bouridane, A. (2023). Video steganography: Recent advances and challenges. *Springer - Multimedia Tools and Applications, 82,* 41943-41985. https://doi.org/10.1007/s11042-023-14844-w

Majeed, N. D., Al-Askery, A., & Hasan, F. S. (2024). Hybrid video steganography and cryptography techniques: Review paper. *The Fifth Scientific Conference for Electrical Engineering Techniques Research (EETR2024)* (020013). AIP Publishing. https://doi.org/10.1063/5.0236185

Petitcolas, F. A. P., Anderson, R. J., & Kuhn, M. G. (1999). Information hiding - a survey. *Proceedings of the IEEE, 87*(7), 1062-1078. https://doi.org/10.1109/5.771065

Pixabay.com. (2024). *Pixabay License*. https://pixabay.com/service/license-summary/

Sencar, H. T., Ramkumar, M., & Akansu, A. N. (2004). *Data hiding fundamentals and applications*. Elsevier Academic Press.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal, 27*(3), 379-423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Stern, M. K., & Johnson, J. H. (2010). Just noticeable difference. In I. B. Weiner & W. E. Craighead (Eds.), *The Corsini Encyclopedia of Psychology* (1st ed., pp. 1-2). Wiley. https://doi.org/10.1002/9780470479216.corpsy0481

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600-612. https://doi.org/10.1109/TIP.2003.819861