symmetry

MDPI

Article

Authentication Framework for Augmented Reality with Data-Hiding Technique

Chia-Chen Lin [1,*], Aristophane Nshimiyimana [1], Morteza SaberiKamarposhti [2] and Ersin Elbasi [3,*]

[1] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan; aristophanenshimiyimana@gmail.com
[2] Faculty of computing and informatics (FCI), Multimedia University (MMU), Cyberjaya 63100, Malaysia; msaberyk@ieee.org
[3] College of Engineering and Technology, American University of the Middle East, Egaila 54200, Kuwait
* Correspondence: ally.cclin@ncut.edu.tw (C.-C.L.); ersin.elbasi@aum.edu.kw (E.E.); Tel.: +886-423924505 (ext. 8730) (C.-C.L.); +965-2225-1400 (ext. 2172) (E.E.)

Abstract: Currently, most existing research on the security of augmented reality (AR) primarily focuses on user-to-user or user-to-infrastructure authentication, aiming to establish secure communication and prevent potential attacks, such as man-in-the-middle attacks, among others. AR techniques enable users to interact with virtual information and objects within the real environment. However, creating virtual objects can be time-consuming, and unfortunately, malicious individuals may unlawfully copy these objects and integrate them into their own AR scenarios. It is essential to authenticate whether AR objects and the associated content they interact with are legitimately integrated. This paper proposes a novel method for authenticating AR-delivered content using data-hiding techniques with the symmetric property. The proposed method utilizes data hiding to embed data within the content that AR objects interact with, enabling the identification of illegal and tampered AR objects. A scenario involving an AR e-book is defined, and two data-hiding methods are applied to this scenario. The experimental results demonstrate that both data-hiding methods effectively identify the tampered image page of an AR e-book under different tone versions, new trigger image(s) added, or new image(s) replacement.

Keywords: data hiding; augmented reality; authentication; data integrity

check for updates

Citation: Lin, C.-C.; Nshimiyimana, A.; SaberiKamarposhti, M.; Elbasi, E. Authentication Framework for Augmented Reality with Data-Hiding Technique. *Symmetry* **2024**, *16*, 1253. https://doi.org/10.3390/sym16101253

Academic Editor: Sergei D. Odintsov

Received: 15 July 2024
Revised: 3 September 2024
Accepted: 6 September 2024
Published: 24 September 2024

1. Introduction

Augmented reality (AR) [1] is an interactive technology that enhances the human view of the real world by overlaying what the human eye sees with computer-generated content. It combines the real world and computer-generated sensory information, including visual, auditory, haptic, and olfactory. AR is prevalent since it uses computer vision and object recognition, and incorporates AR cameras into smartphone applications. By taking the image from the camera and processing it from computer vision in real-time, the surrounding environments of the user in the real world become interactive and digitally manipulated. The computer-generated content that can be the three-dimensional (3D) components of virtual and real objects delivers contextual information, such as GPS navigation, games, and social experiences. A growing number of AR applications have been used in various industries, such as retail, healthcare, and education, since AR is an increasingly popular interactive experience of online gaming and social media.

AR serves diverse purposes, including personal creation, education, and entertainment. Programmers can craft digital 3D components within AR, while end-users can also generate content within AR interactive environments. However, the widespread applicability of AR technology and the inherently digital nature of AR creations make them susceptible to copying and dissemination. Additionally, discussions on adequate digital content protection measures for AR works are lacking. Consequently, creators employing AR technology in their digital works face a heightened risk of copyright infringement.

Recently, the security of AR issues has attracted scholars' attention. In 2016, Gaebel et al. introduced the Looks Good To Me (LGTM) authentication protocol. LGTM utilizes the unique hardware and contextual capabilities of AR headsets to integrate natural human trust mechanisms into digital authentication, aiming for both usability and security. It achieves this by combining localization-based wireless communication with facial recognition, allowing users to authenticate each other effectively [2]. In 2020, Wazir et al. developed an innovative authentication system that uses AR to create graphical doodle passwords in a 3D space. Users can draw their passwords in 3D by interacting with their smartphone screens. Authentication is achieved by comparing the size and coordinates of the last five doodles to verify the user's identity [3]. The following year, Bhalla et al. gathered data on spatial and behavioral patterns from AR headset users who were wearing AR headset. They proposed several advanced models along with machine learning techniques, including k-Nearest Neighbors, Random Forest, Support Vector Machines, and Bag of Symbolic Fourier Approximation Symbols. These models analyze user interactions with the AR environment to create unique user signatures, with an accuracy rate of approximately 92.675% [4]. In 2022, Stephenson et al. assessed the existing authentication mechanisms for AR/VR devices by reviewing research and practical implementations. They identified key factors for effective authentication on AR and VR devices, including usability, deployability, accessibility, and security, based on user experiences [5].

While researchers have developed user authentication systems for AR interactive environments [2–5], there is a noticeable absence of literature exploring the design of content copyright protection mechanisms and security concerns for AR objects within these AR interaction environments capable of responding to user actions. In this paper, we focus on potential solutions for the copyright protection of AR content. Unlike previous research [2–5] that has concentrated on user identity verification in AR protection issues, our study stands out as the first to address AR digital content protection specifically. We hope that through this research, we can contribute to a more comprehensive understanding of AR protection issues. The main contributions and novelties of this paper are summarized as follows:

(1) We propose a data-hiding-based authentication framework for AR content, and the integrity of AR content can be verified, which has not been discussed in other state-of-the-art schemes.

(2) The proposed authentication framework works with two different data-hiding strategies, and the experimental results confirm the usability and practicality of the proposed authentication framework.

(3) The proposed authentication framework withstands the luminance attack, color saturation attack, and object replacement attack.

To offer readers a clear picture regarding the AR scope which we focus on in this paper and make this paper self-contained, Section 2 first introduces the content structure of an AR e-book, and then introduces the copyright protection techniques that can be applied to verify the content of an AR e-book. Section 3 contains a detailed exploration of the proposed embedding and detection methods. The experimental results and their analyses appear in Section 4. Finally, we conclude the paper in Section 5.

2. Related Work

2.1. Content Structure of AR e-Book

Currently, there are lots of AR applications that have been designed, such as entertainment/gaming, education/training, retail/E-commerce, healthcare, navigation and wayfinding, and so on. For different AR applications, the copyright protections of AR content could be different. In this paper, we mainly focus on AR e-books. In order to study the digital content protection of AR objects, in this paper, we define the AR interactive environment as an e-book containing AR objects. In an AR e-book, each page can be in PDF format, image format (e.g., JPEG and PNG), or HTML format. Each page may contain titles or subtitles, as demonstrated in Figure 1. Under a subtitle, different objects such as text, image, audio, or animation will be presented.

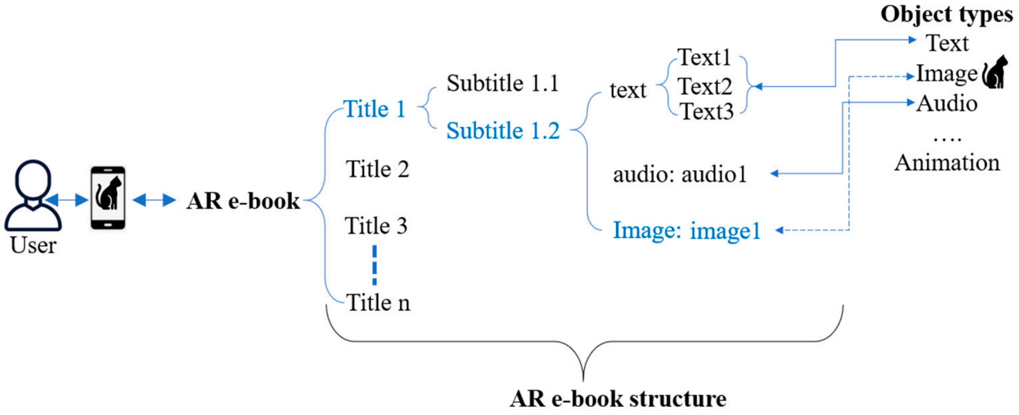


Figure 1. AR e-book structure and AR triggering example.

In an AR e-book, a specific area called a trigger image will drive the AR object. When the trigger image is scanned by a hardware device such as a mobile phone or camera, the AR app will recognize the trigger image and activate the corresponding AR object or AR content, such as 2D image, 3D image, audio, or even animation.

2.2. Copyright Protection Techniques

In general, conventional digital signature techniques [6] can employ asymmetric cryptography to verify the authenticity of digital images, and they also can provide non-repudiation. However, digital signatures require extra bandwidth and storage to attach the signature. Moreover, they cannot localize the tampered areas if the image is deemed unauthentic. Data hiding [7] provides the proper solutions to address these issues of copyright and authorship.

Data hiding, a science of concealed communication, embeds data into digital media for the purpose of secret communication, authentication, annotation, and copyright protection. Data hiding embeds the authentication data into the image, making it capable of authenticating itself without accessing the original image. The authentication data embedded and lumped in the image is sensitive to any modifications of the images and thus can localize the modifications as well as verify the content integrity of the image. Compared to digital signatures, the advantage of data hiding is obvious since the authentication data are embedded into the image rather than appended to the image. Note that hiding data in a digital image inevitably destroys the image content. However, images provide satisfactory properties for data hiding since the visual redundancy of digital images makes it possible to embed the imperceptible data in the image. The existence of embedded data can be concealed by using data-hiding algorithms. Therefore, it is intuitively clear that the resulting image (stego image) is perceptually equal to the original image.

Lin et al. [8] proposed an efficient data-hiding method for image tamper detection and recovery. They embedded the 2-bit authentication data and 6-bit recovery data into each 2 × 2 image block. A hierarchical detection structure was used to ensure the accuracy of tamper localization. Their method is effective in resisting the collage attack and VQ attack. In 2011, Zhang et al. [9] proposed two self-embedding data-hiding schemes by using a reference-sharing mechanism. The embedded data are a reference derived from the original principal content in different regions. After verifying the tampered blocks, both the embedded data and the original image content in the reserved area are used to recover the tampered blocks. In 2013, Chang and Tai [10] proposed a block-based data-hiding scheme for image tamper detection and recovery. For each 2 × 2 image block, the data needed to be embedded is composed of the 6-bit recovery data and the 2-bit authentication data. They used the 2-LSB embedding scheme to embed data into the image. Their method can resist collage attacks, VQ attacks, and constant-average attacks [11]. The accuracy of tamper localization needs to be improved since the authentication data are only 2 bits for each block. Sarreshtedari and Akhaee [12] presented a data-hiding method for the purposes of detecting the tampered area and recovering the lost content in the tampered regions. They used Reed–Solomon codes to encode the embedded data in order to help the RS-code decoder retrieve the original source encoded image for self-recovery. The ability of self-recovery depends on the correcting ability of this error correction code.

In 2016, Qin et al. [13] proposed a self-embedding data-hiding scheme for tampering recovery. They used a reference-data interleaving mechanism and the adaptive LSB layer selection for data hiding. The reference data needed to be embedded are derived from the interleaved and scrambled MSB bits of original image pixels. They also provided detailed analyses for the theoretical values of watermarked image quality and perfect recovery probability. To improve the embedding efficiency, Lin et al. [14] proposed a hybrid image authentication scheme with the help of absolute moment block truncation coding (AMBTC). They used different embedding methods for smooth and complex blocks. The authentication data are embedded into the AMBTC compressed codes, and thus their method gives a significantly improved embedding capacity. Lin et al. [15] presented a reversible data-hiding method based on AMBTC. The authentication is reversible in the sense that the hiding distortion can be removed after image authentication. In 2018, Tai and Liao [16] proposed a self-embedding watermarking method for image authentication. The authentication data and recovery data are generated from the wavelet transform. Therefore, the tampered regions can be effectively recovered with a high image contrast.

In 2020, Yu et al. [17] proposed an adaptive bit-plane data-hiding method by using a parity check matrix based on matrix coding to enhance the hiding performance. In 2023, Nazir et al. [18] proposed a blind watermarking scheme for image authentication and protection. Data hiding on color RGB components is centered on singular value decomposition (SVD) and discrete wavelet transform (DWT). They encrypted SVD components and used a logistic map along with the hyperchaotic system to achieve confidentiality and avoid false positive problems. Li et al. [19] provided an ownership verification technique for deep neural networks (DNNs). They proposed a new framework that not only can efficiently resist linear functionality equivalence attacks but also helps the existing white-box watermarking schemes to enhance their robustness. Tang et al. [20] presented a two-stage reversible data-hiding method that embeds a robust watermark into the selected Pseudo-Zernike moments. The proposed adaptive normalization method achieves an invariance to pixel amplitude variation and gives a balance between robustness and imperceptibility. Anand and Singh [21] proposed a dual watermarking scheme for the CT scan images of COVID-19 patients. They used the Electronic Patient Record (EPR) text and medical image as multiple watermarks to ensure a high level of authentication. In order to obtain better image quality and higher hiding capacity, Chang et al. [22] proposed a data-hiding scheme based on the turtle shell. In their method, a secret digit is embedded into each cover pixel pair by looking for the turtle shell. Chen et al. [23] embedded the authentication data into AMBTC compressed codes by using the turtle shell data-hiding scheme.

For AR applications, Lee et al. [24] used the data-hiding technique to propose a tracking system based on invisible markers. The marker is embedded into the AR contents. It can be extracted from the scene image by using a general camera. Although the invisible watermarking technique is adopted, it is used to serve as the trigger image instead of copyright protection. In 2019, Li et al. [25] presented an AR-based data-hiding architecture, which combines data hiding and deep learning for secret communication. The secret data can be transmitted in various formats with a higher embedding rate due to the property of AR. Since they used AR as a cover media to transmit the confidential message(s) via the pre-shared secret keys and reveal model, there is no copyright protection issue involved. In 2021, Bhattacharya et al. mentioned blockchain is able to prevent assets from being replicated for IoT-based smart factories because blockchain can help create and provide the identity of assets [26]. However, their research scope is defined as an IoT-based smart factory and a private blockchain platform can be set up to manage all assets within an IoT-based smart factory or multiple IoT-based smart factories belonging to the same company. Although blockchain technology has been recognized as a solution for asset identification issues within enterprises [26], how to utilize blockchain technology to address the problem of illegal misuse in AR electronic publishing (i.e., AR e-books) remains to be explored. This is because applying blockchain to AR e-publishing requires an authority to pre-assign identification numbers to each AR object within an AR e-book. Without such pre-assigned identification, the blockchain framework, which relies on object numbering for identification, will not be able to effectively recognize AR objects.

Considering the potential actions of malicious attackers, it is important to recognize that they might attempt to tamper with specific content within an AR e-book. For instance, on a page featuring two trigger images to activate two AR objects, a malicious attacker could overlay a new image to conceal one trigger image, thereby deactivating the corresponding AR object. Subsequently, they could falsely assert the ownership of the altered AR e-book. Alternatively, a malicious attacker might retain all the trigger images but alter the remaining content of an AR e-book, still asserting ownership. Considering the attack scenarios described above, individuals with malicious intent can easily plagiarize AR works with minimal effort, leading to unjust gains. To counteract such attacks, this paper proposes an AR content authentication framework along with two data-hiding strategies to verify AR content. The authentication data embedded into AR content makes it possible to verify the integrity of AR content and provide the ability to localize the tampered regions. The AR authentication can be easily performed through AR cameras.

3. The Proposed AR Content Authentication Framework

AR integrates computer vision, object recognition, and AR cameras into mobile applications. The proposed AR content authentication framework aims to verify AR content directly without requiring access to the original AR object file through AR cameras. It consists of two phases, one is the data-hiding phase depicted in Figure 2, and the other is the data extraction and tamper detection phase depicted in Figure 3. Data extraction is the inverse operation of data embedding, and tamper detection is the verification based on the extracted authentication code. The details regarding the proposed AR content authentication framework are outlined below. Based on our proposed authentication framework, two different data-hiding strategies are proposed: one is an LSB-based relative reference hiding strategy and the other is a DWT-based hiding strategy. The detailed descriptions of data embedding, and data extraction and tamper detection for LSB-based relative reference hiding strategy are introduced in Sections 3.1 and 3.2. The detailed descriptions of data embedding, and data extraction and tamper detection for DWT-based hiding strategy are introduced in Sections 3.3 and 3.4. It is noted that in the following paragraphs, the trigger image is denoted as regions of interest (ROIs), and the rest area on a page of an AR e-book is denoted as the region of non-interest (RONI) as shown in Figure 4.

Figure 4 demonstrates a page of the AR e-book for example; there are two regions of interest (ROIs), and the rest area is the region of non-interest (RONI). The ROI can be used to trigger the AR object, and the ROIs can be established in sub-sequential with corresponding 3D contents or animation files according to the script defined by the author.

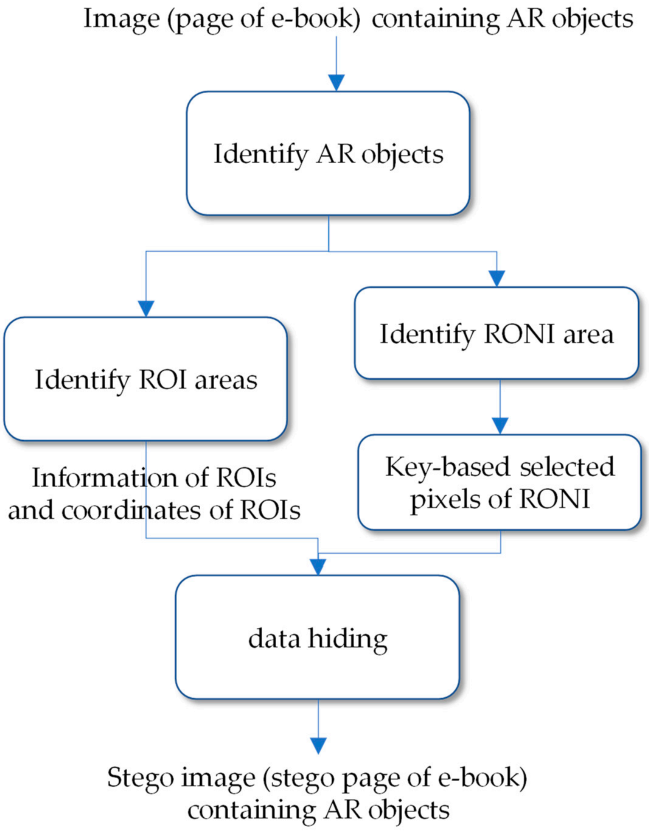


Image (page of e-book) containing AR objects
Identify AR objects
Identify ROI areas
Identify RONI area
Information of ROIs and coordinates of ROIs
Key-based selected pixels of RONI
data hiding
Stego image (stego page of e-book) containing AR objects

Figure 2. Flowchart of the proposed data-embedding phase for AR content.

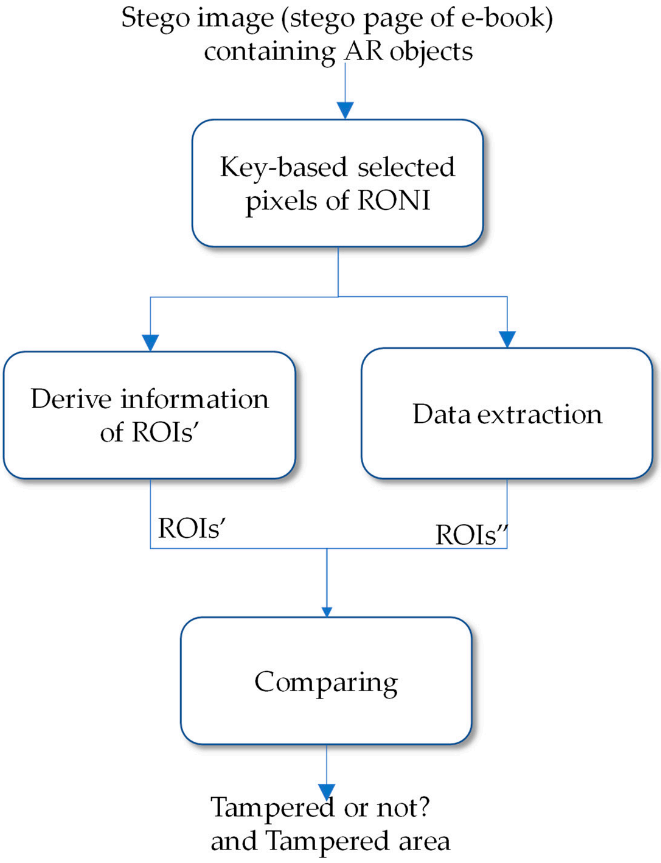

Stego image (stego page of e-book) containing AR objects
Key-based selected pixels of RONI
Derive information of ROIs′
Data extraction
ROIs′
ROIs″
Comparing
Tampered or not? and Tampered area

Figure 3. Flowchart of the data extraction and tamper detection phase.

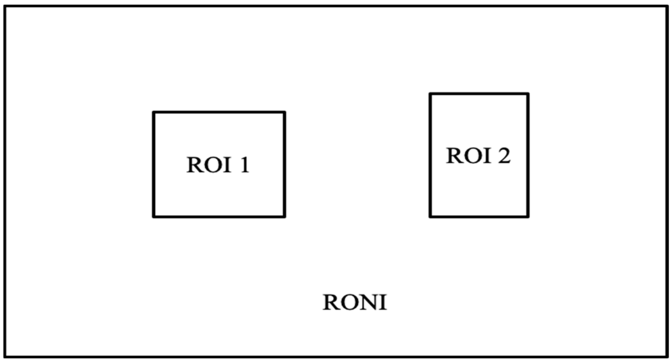


Figure 4. Example of the page of the AR e-book containing ROIs and RONI.

3.1. Data-Embedding Phase-LSB-Based Relative Reference Hiding Strategy

As mentioned earlier in Section 3, ROI represents the trigger image, and a page of the AR e-book consists of one or more ROIs, with the remaining area termed RONI. For clarity in the subsequent discussions, we will abbreviate the term "cover image" to CI when referring to the pages of the AR e-book. Similarly, we will use the abbreviation TI to denote the "trigger image," which activates the AR object and is situated within the ROI. Once the data-hiding phase is completed, the stego page of the AR e-book is obtained and it is noted as "stego image" and SI for short.

In the first data-hiding strategy, we proposed the relative reference strategy to conceal TI into the RONI of CI. The relative reference-based data-hiding algorithm is described as follows:

Input: Cover image (CI) and Trigger image (TI)
Output: Stego Image (SI)

Step 1: Identify the TI from CI and denote its location area as ROI.
Step 2: Calculate the Global Average (GA) for the Green channel and denote as $\mathrm{GA_G}$ of pixel values in CI according to Equation (1)

$$GA_G = \sum\nolimits_{i=0,j=0}^{i=w,j=h} Gp_{ij} \Big/ w \times h \tag{1}$$

where $Gp_{i,j}$ indicates the pixels located at the Green channel in the CI. w is the width and h is the heigh of the CI.

Step 3: Generate an Authentication Bitstream (ABS) for the Green channel of TI by comparing each pixel value located at the Green channel of the OI with the corresponding $\mathrm{GA_G}$ according to Equation (2).

$$ABS_G = \begin{cases} 1, \ if \ \ GA_G \leq Gp'_{ij} \\ 0, \quad otherwise \end{cases}, \tag{2}$$

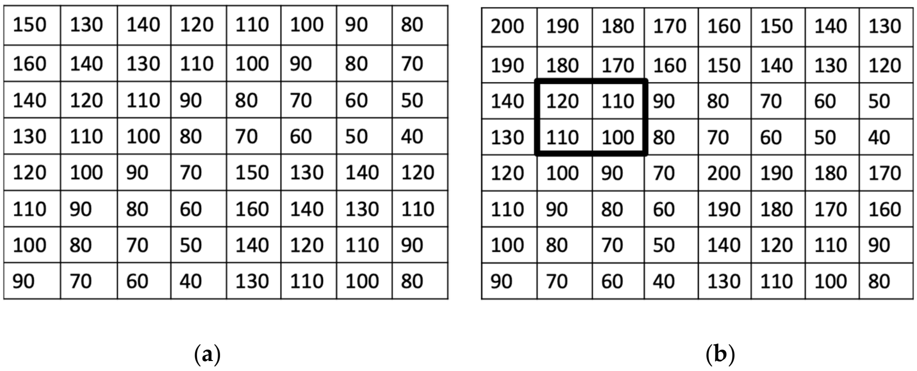
where Gp'_{ij} indicates the pixel value located at the Green channel of the TI. i is ranged between 1 and w', j is ranged between 1 and h', w' is the width, and h' is the height of the TI.

Step 4: Convert the coordinate of TI into binary stream and denote as coTI. Concatenate coTI and $\mathrm{ABS_G}$ to obtain secret stream $\mathrm{S_G}$ for the channel Green in CI, respectively.
Step 5: Generate a key to select pixels from the NROI of CI to conceal n copies of the size of secret stream S with LSB substitution.

Once Steps 1–5 are completed, the stego image which contains the ROI coordinate and authentication data is obtained.

Example of Data-Embedding Phase-LSB-Based Relative Reference Hiding Strategy

Here, an example is demonstrated to explain Section 2.1 for data-hiding procedures with a relative reference strategy. In this example, we assume that CI is sized 8 × 8 pixels as shown in Figure 5 and the pixels located in Red, Green, and Blue channels map to TI that is sized as 2 × 2 pixels shown within the black bold border in Figure 5b are shown in Figure 6.



150	130	140	120	110	100	90	80
160	140	130	110	100	90	80	70
140	120	110	90	80	70	60	50
130	110	100	80	70	60	50	40
120	100	90	70	150	130	140	120
110	90	80	60	160	140	130	110
100	80	70	50	140	120	110	90
90	70	60	40	130	110	100	80

(**a**)

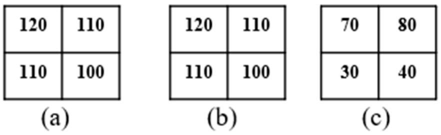
200	190	180	170	160	150	140	130
190	180	170	160	150	140	130	120
140	120	110	90	80	70	60	50
130	110	100	80	70	60	50	40
120	100	90	70	200	190	180	170
110	90	80	60	190	180	170	160
100	80	70	50	140	120	110	90
90	70	60	40	130	110	100	80

(**b**)

100	120	110	130	140	150	160	170
110	130	120	140	150	160	170	180
200	70	80	170	160	150	140	130
110	30	40	140	150	160	170	180
100	120	110	130	140	150	160	170
110	130	120	140	150	160	170	180
200	190	180	170	160	150	140	130
110	130	120	140	150	160	170	180

(**c**)

Figure 5. (**a**) The Red channel pixel values in CI, (**b**) the Green channel pixel values in CI, and (**c**) the Blue channel pixel values in CI.



120	110
110	100

(a)

120	110
110	100

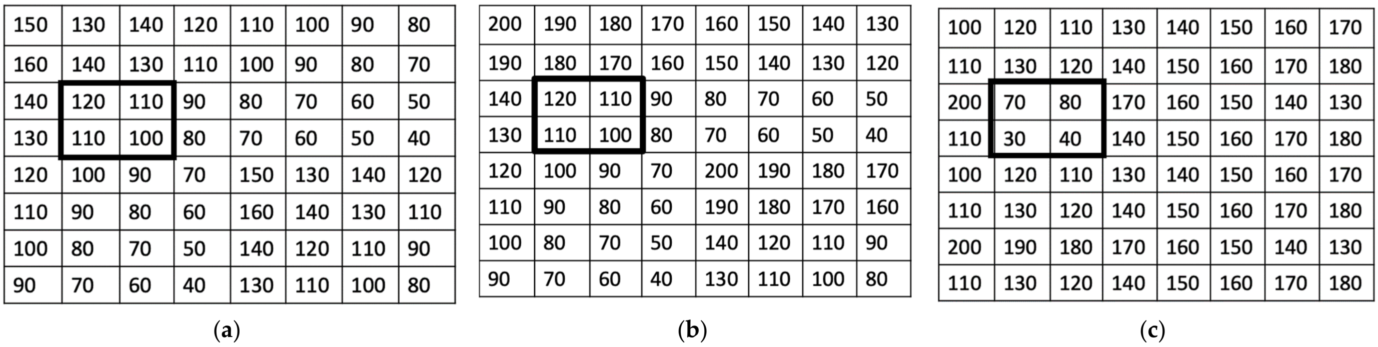
(b)

70	80
30	40

(c)

Figure 6. (**a**) Red channel pixel values in TI, (**b**) Green channel pixel values in TI, and (**c**) Blue channel pixel values in TI.

Moreover, we assume the 2 × 2 TI shown in Figure 6b is located at the ROI framed with the dark line shown in Figure 5b. Figure 7 demonstrates three TI's locations in three channels.



150	130	140	120	110	100	90	80
160	140	130	110	100	90	80	70
140	120	110	90	80	70	60	50
130	110	100	80	70	60	50	40
120	100	90	70	150	130	140	120
110	90	80	60	160	140	130	110
100	80	70	50	140	120	110	90
90	70	60	40	130	110	100	80

(**a**)

200	190	180	170	160	150	140	130
190	180	170	160	150	140	130	120
140	120	110	90	80	70	60	50
130	110	100	80	70	60	50	40
120	100	90	70	200	190	180	170
110	90	80	60	190	180	170	160
100	80	70	50	140	120	110	90
90	70	60	40	130	110	100	80

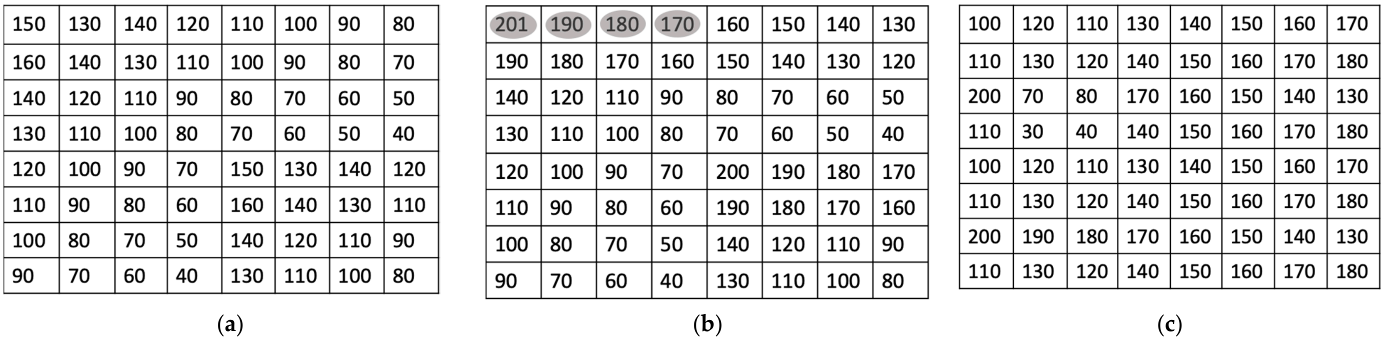
(**b**)

100	120	110	130	140	150	160	170
110	130	120	140	150	160	170	180
200	70	80	170	160	150	140	130
110	30	40	140	150	160	170	180
100	120	110	130	140	150	160	170
110	130	120	140	150	160	170	180
200	190	180	170	160	150	140	130
110	130	120	140	150	160	170	180

(**c**)

Figure 7. (**a**) TI in CI Red channel, (**b**) TI in CI Green channel, and (**c**) TI in CI Blue channel.

Take CI Green channel shown in Figure 5b for example, the first three pixels in the Green channel of CI are 200, 190, 180 and 170. Finally, GA_G is obtained as 118 according to Equations (1) and (2); and ABS_G is obtained as "1000" because four Green channel pixel values in TI are 120, 110, 110, and 100 as shown in Figure 7b, respectively, following a zigzag scan; and there are one pixel (120) is larger than GA_R = 118 but there are three pixels (110, 110, 100) are less than GA_R = 118. After pending ABS_G as "1000" to the LSB of the pixels in the Green channel in CI with LSB substitution without considering the coordinate of TI, the first pixels of the stego CI in the Green channel are denoted as 201, 190, 180, and 170 as shown in Figure 8b, but the rest two channels (Red and Blue) without modification is shown in Figure 8b,c. Comparing Figures 7b and 8b, the LSB substitution can be easily followed.



150	130	140	120	110	100	90	80
160	140	130	110	100	90	80	70
140	120	110	90	80	70	60	50
130	110	100	80	70	60	50	40
120	100	90	70	150	130	140	120
110	90	80	60	160	140	130	110
100	80	70	50	140	120	110	90
90	70	60	40	130	110	100	80

(**a**)

201	190	180	170	160	150	140	130
190	180	170	160	150	140	130	120
140	120	110	90	80	70	60	50
130	110	100	80	70	60	50	40
120	100	90	70	200	190	180	170
110	90	80	60	190	180	170	160
100	80	70	50	140	120	110	90
90	70	60	40	130	110	100	80

(**b**)

100	120	110	130	140	150	160	170
110	130	120	140	150	160	170	180
200	70	80	170	160	150	140	130
110	30	40	140	150	160	170	180
100	120	110	130	140	150	160	170
110	130	120	140	150	160	170	180
200	190	180	170	160	150	140	130
110	130	120	140	150	160	170	180

(**c**)

Figure 8. (**a**) Red channel pixel values in SI, (**b**) Green channel pixel values in SI, and (**c**) Blue channel pixel values in SI.

3.2. Data Extraction and Tamper Detection Phase-LSB-Based Relative Reference Hiding Strategy

The data extraction and tamper detection phase is very similar to the data-embedding phase after the user uses an AR camera or smartphone to capture the image called SI′ from the SI and then conduct the data extraction and taper detection. Once the SI′ is obtained, the user needs to use the secret key to find out the selected pixels from the Green channel in the SI′ that carry the coordinates of TI during the data-hiding phase. Once the ROI is located, the hidden $\mathrm{ABS_G}$ is extracted. The authentication data can be recalculated according to Equations (1) and (2) with the pixel values in TI and SI′. If the recalculated $\mathrm{ABS_G}'$ = $\mathrm{ABS_G}$, respectively, the SI′ is authenticated, and the AR object is retrieved and pops up in front of the user through the AR camera. Otherwise, the pixels in the ROI will be indicated as tampered.

3.3. Data-Embedding Phase-DWT-Based Hiding Strategy

Besides using relative reference hiding to generate the authentication data and embedding the coordinate of ROI and authentication code into selected pixels located in RNOI with LSB substitution, the second method consisting of data hiding, data extraction, and tamper detection based on DWT is described in Sections 3.3 and 3.4, respectively. In the DWT-based data-hiding phase, the CI first conducts Level-1 DWT transformation. The coefficients of the Green channels located at the LL of CI without covering the coefficients belonging to the ROI are denoted as GLL'_{CI}, respectively. Next, $\mathrm{ABS_G}$ denotes the Green channel pixel values in TI that are located at the ROI. Concatenate the coordinates of TI and $\mathrm{ABS_G}$ to obtain secret stream $\mathrm{S_G}$ for the Red coefficients in GLL'_{CI}. Finally, calculate the stego coefficients in RLL'_{CI}, GLL'_{CI}, and BLL'_{CI} according to Equation (3), Equation (4), and Equation (5), respectively.

$$Rc'_{ij} = \alpha \times \mathrm{S}_R + (1-\alpha) \times Rc_{ij} \tag{3}$$

$$Gc'_{ij} = \alpha \times \mathrm{S}_G + (1-\alpha) \times Gc_{ij} \tag{4}$$

$$Bc'_{ij} = \alpha \times \mathrm{S}_B + (1-\alpha) \times Bc_{ij} \tag{5}$$

where α is the weight to justify the robustness of S_R, S_G, and S_B to the coefficients in RLL'_{CI}, GLL'_{CI}, and BLL'_{CI}, respectively. α is ranged between 0 and 1. It is noted that the coefficients in GLL'_{CI} which are selected by a secret key from the right-bottom of the LL of CI without covering the coefficients belonging to ROI. However, the number of selected coefficients that are located at the right-down and left-up of the LL of CI are the same to ensure the symmetric property during data hiding. After the stego coefficient generation, the inverse DWT operation is conducted to obtain the SI. It is noted that the parameter, such as α in the DWT-based method, is ranged between 0 and 1 and it can be set as an initial value in the authentication framework; therefore, no extra data transmission is required.

3.4. Data Extraction and Tamper Detection Phase-DWT-Based Hiding Strategy

When DWT data hiding is conducted to conceal the coordinates of ROI and authentication code, only LL coefficients that are located at the right-bottom will be selected by a secret key. Because the LL coefficients mapped to ROI are not involved in the data-hiding phase, the pixel values of TI in the SI remain the same as those in the CI. Therefore, the data extraction is performed by conducting Level-1 DWT and then simply extracting the hidden data from the selected coefficients. Next, perform Equation (3), Equation (4), and Equation (5) with α pre-determined in the data-hiding phase to obtain the hidden S'_G. The extracted coordinates are used to extract the pixel values of the Green channel S_G in the ROI in the SI. The comparisons between the S'_G and S_G are performed. If $S'_G = S_G$, it means that the SI is authenticated. Otherwise, the coordinates are identified as tampered in the SI.

4. Prototype System

To prove our AR authentication framework is practical and can work well in the AR e-book scenario, the prototype system is implemented by using Python 3.9 and Unity hub with Vuforia and OpenCV. This prototype platform runs on a Mac OS-based PC equipped with a 2.9 GHz Dual-Core Intel® CoreTM i5 CPU and is enhanced by an Intel Iris GPU for accelerated performance. In addition, the AR e-book with our proposed authentication function is deployed on the Android device, which serves as the AR device, with a CPU of Octa-core 4 × 1.8 GHz Kryo 260 Gold and Adreno 509's GPU with Android version 10.

Following the flowcharts demonstrated in Figure 9, the user can use their AR reader installed in their smartphone to view the AR e-book; the smartphone's camera will capture the targeted image as shown in Figure 10; and then the authentication/verification and the tamper detection operation are conducted. It is noted that the detailed description of Step 3, as shown in Figure 9, corresponds to the data extraction and tamper detection phases discussed in Sections 3.2 and 3.3 , respectively, for various data-hiding strategies.

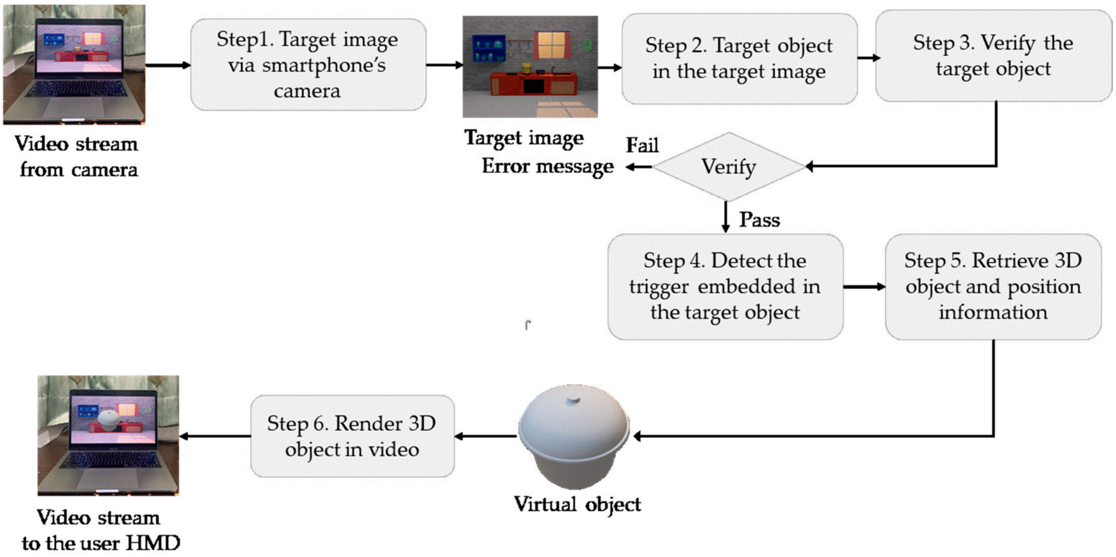

Video stream
from camera
Step1. Target image via smartphone's camera
Target image
Step 2. Target object in the target image
Step 3. Verify the target object
Fail
Error message
Verify
Pass
Step 4. Detect the trigger embedded in the target object
Step 5. Retrieve 3D object and position information
Step 6. Render 3D object in video
Virtual object
Video stream
to the user HMD

Figure 9. Authentication/verification and tamper detection model.

With our proposed hidden authentication code extracted from the RONI, if it is identical to that recalculated authentication code, a verification result such as "Verified" will appear. Meanwhile, the AR object pops up as shown in Figure 11. If the verification fails, a verification result such as "Unverified" will appear as shown in Figure 12a,b. It is noted that Figure 12a,b demonstrate two examples in which the verifications fail and the verification result is "Unverified".

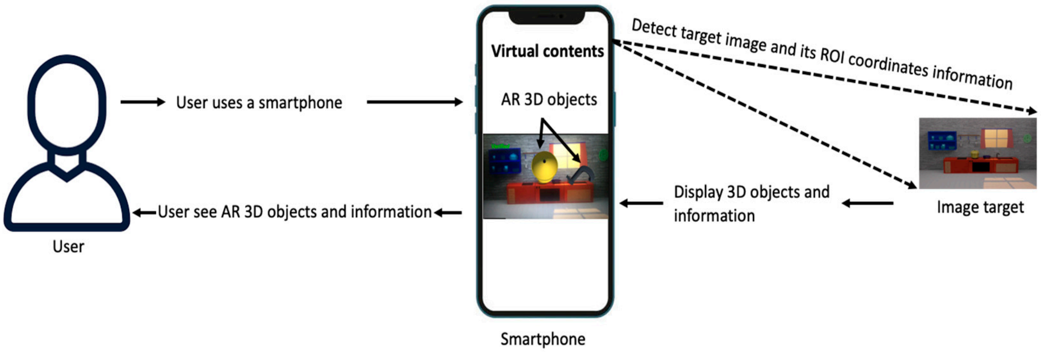

User
User uses a smartphone
User see AR 3D objects and information
Virtual contents
AR 3D objects
Smartphone
Detect target image and its ROI coordinates information
Display 3D objects and information
Image target

Figure 10. AR 3D object detection procedure.

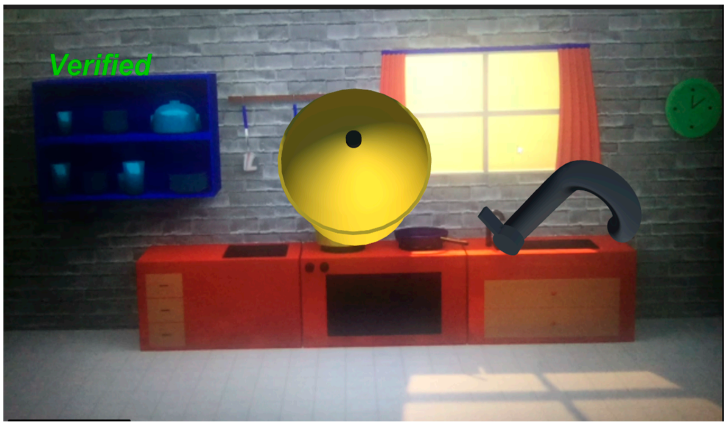


Figure 11. Verified stego image (SI).

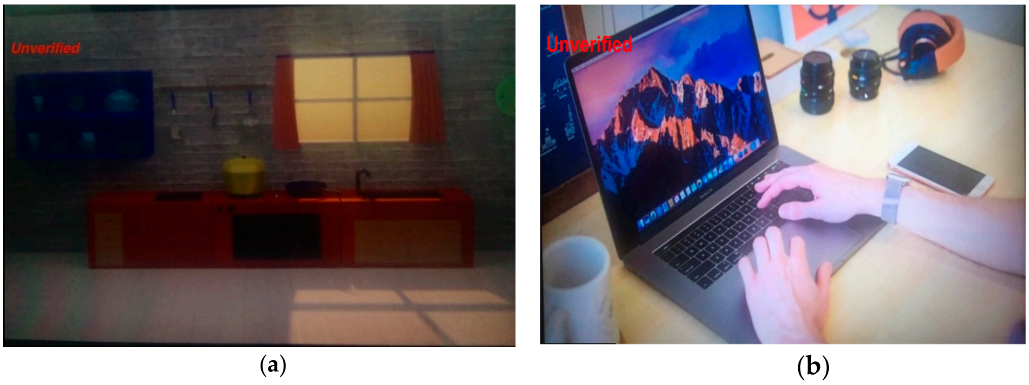


Figure 12. (**a**) Attacked stego image and the "Unverified" message appears, and (**b**) attacked stego image and the "Unverified" message appears.

5. Experimental Results

Based on our proposed AR content authentication framework, our hiding methods are designed in Section 2—one is the relative reference strategy with LSB substitution and the other is the DWT-based data-hiding strategy. To test the detection performance of the trigger image by using our AR content authentication with different data-hiding strategies, one image page of the AR e-book and the corresponding versions with different parameters are listed in Figure 13. It is noted that in Figure 13, we test five tampered images with different combinations of brightness (B) and contrast (C). With the test image page of the AR e-book shown in Figure 13a, several experiments are conducted in the following paragraphs by using the prototype system described in Section 3. In general, the visual quality of the stego image carrying the coordinates of ROI and authentication data is PSNR, which is defined in Equations (6) and (7).

$$MSE = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}\left(SI_{i,j} - CI_{i,j}\right)^2, \tag{6}$$

Peak signal–noise ratio (PSNR) is defined by the following (10):

$$PSNR = 10log_{10}\frac{(255)^2}{MSE}\ , \tag{7}$$

where $W \times H$-pixels indicate the size of CI and SI, respectively. To further evaluate the performance of our proposed authentication framework with two designed data-hiding strategies, the two images shown in Figure 13 carrying the trigger images such as obj image 1 and obj image 2 served as the test images.

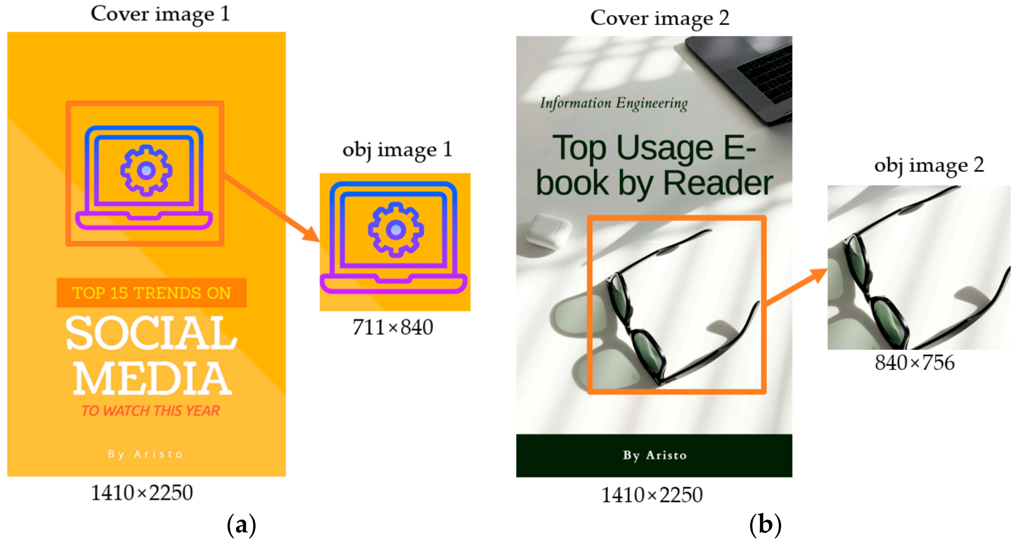


Figure 13. Two test cover images are sized as 1410 × 2250 pixels and (**a**) the trigger images (obj image1 and obj image2) are sized as 711 × 840 pixels and (**b**) the trigger images are sized as 840 × 756 pixels.

Due to potential variations in light sources and intensities when users review AR e-books, as well as potential color differences produced by AR devices during the review, we conducted experiments to evaluate our authentication scheme's performance across different brightness levels and color systems. The relevant experimental findings are presented in Tables 1–4. Note that Tables 1 and 2 present the image quality of the stego images and the image quality of the extracted object images with the relative reference hiding strategy described in Sections 2.1 and 2.2.

Table 1. Detection results and image quality of the stego images and tampered stego images using the relative reference strategy with the relative reference hiding strategy and LSB substitution (B: brightness; C: contrast).

Stego Image		Parameters	Similarity (%)	PSNR (dB)
Stego image 1	Untampered	B: 0 C: 0	100	59.87
	Tampered 1	B: −40 C: −40	64	31.74
	Tampered 2	B: −20 C: −40	57	34.94
	Tampered 3	B: 0 C: −20	68	36.71
Stego image 2	Untampered	B: 0 C: 0	100	59.58
	Tampered 1	B: −40 C: −40	58	28.40
	Tampered 2	B: −20 C: −40	66	29.49
	Tampered 3	B: 0 C: −20	66	37.82

Table 2. The PSNRs and MSEs of the extracted trigger image (obj image1 and obj image2) from the stego image under different parameters using the relative reference strategy and LSB substitution (B: brightness; C: contrast).

Extracted Obj Image		Parameters	PSNR (dB)	MSE
Stego image 1	Tampered 1	B: −40 C: −40	52.15	0.39
	Tampered 2	B: −20 C: −40	50.85	0.53
	Tampered 3	B: 0 C: −20	49.47	0.73
Stego image 2	Tampered 1	B: −40 C: −40	48.13	0.99
	Tampered 2	B: −20 C: −40	48.13	1.0
	Tampered 3	B: 0 C: −20	45.89	1.67

Table 3. Detection results and image qualities of stego images and tampered stego images using DWT-based data hiding (B: brightness; C: contrast, α = 0.01).

Extracted Obj Images		Parameters	Similarity (%)	PSNR (dB)
Stego image 1	Untampered	B: 0 C: 0	100.0	62.91
	Tampered 1	B: −40 C: −40	70.55	55.00
	Tampered 2	B: −20 C: −40	81.09	57.93
	Tampered 3	B: 0 C: −20	83.00	58.00
Stego image 2	Untampered	B: 0 C: 0	100.0	57.95
	Tampered 1	B: −40 C: −40	71.73	50.00
	Tampered 2	B: −20 C: −40	72.64	52.00
	Tampered 3	B: 0 C: −20	75.58	54.00

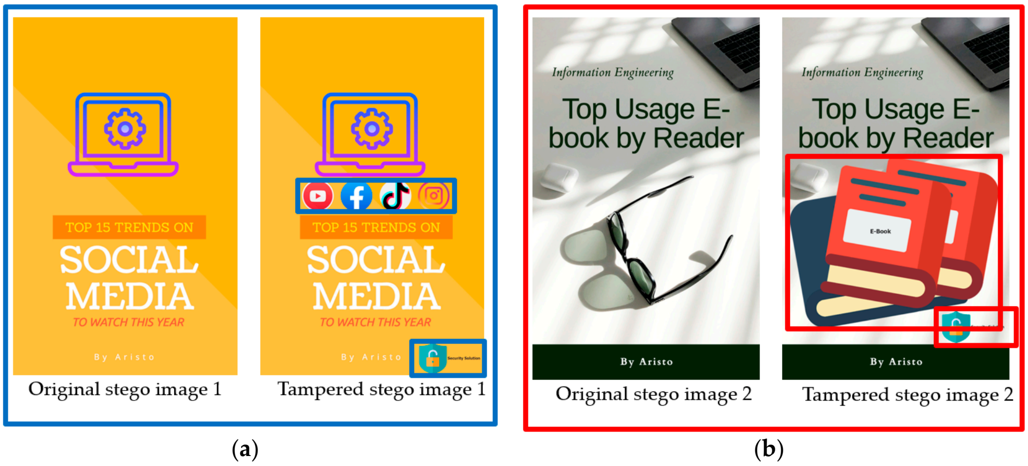
Table 4. The PSNRs and MSEs of the extracted trigger image (obj image1 and obj image2) from the stego images and tampered stego images under different parameters using DWT-based data hiding (B: brightness; C: contrast).

Extracted Obj Images		Parameters	PSNR (dB)	MSE
Stego image 1	Untampered	B: 0 C: 0	53.68	0.27
	Tampered 1	B: −40 C: −40	50.00	0.35
	Tampered 2	B: −20 C: −40	51.00	0.30
	Tampered 3	B: 0 C: −20	52.00	0.28
Stego image 2	Untampered	B: 0 C: 0	49.66	0.70
	Tampered 1	B: −40 C: −40	46.00	0.75
	Tampered 2	B: −20 C: −40	47.00	0.73
	Tampered 3	B: 0 C: −20	48.00	0.72

Table 1 shows that the similarity rate of the trigger image (obj image1 or obj image2) with the relative reference strategy with the relative reference hiding strategy remains 100% and the visual quality of the stego image retains 59.87 dB and 58.58 dB for stego image 1 and stego image 2, respectively. However, when the brightness and contrast are justified accordingly, the detection performance is below 70% and the PSNRs are below 38 dB. However, even though the brightness and contrast of the stego images have been modified, Table 2 confirms that the image quality of the two extracted trigger images (obj image 1 and obj image2) still ranged between 45.89 dB and 52.15 dB. According to general image quality evaluation standards [14], if the PSNR exceeds 30 dB, the human visual system has difficulty distinguishing between the original image and the stego image. For example, as shown in Tables 1 and 2, the image quality of the two extracted trigger images (obj image 1 and obj image 2) ranges from 57 dB to 62 dB regardless of whether the LSB-based or DWT-based data-hiding strategy is used. This is well above the 30 dB threshold. Therefore, it can be concluded that both proposed data-hiding strategies effectively maintain image quality.

Table 3 demonstrates that the similarity rate of the trigger image (obj image1 or obj image2) with the relative reference strategy with DWT-based data hiding remains 100% and the visual quality of the stego image retains 62.91 dB and 57.95 dB for stego image 1 and stego image 2, respectively. However, when the brightness and contrast are justified accordingly, the detection performance will remain above 70% and the PSNRs are above 50 dB. The similarity rate becomes less than 100%; this is because the image has been modified, therefore, the extracted results can not be the same as the original one. However, comparing the results of the two proposed data-hiding strategies shown in Tables 1 and 3, and Tables 2 and 4, it is noted that the average visual quality of the stego images with DWT is always higher than that using the relative reference strategy and LSB substitution. Moreover, the average PSNRS of the extracted trigger images (obj image1 and obj image2) is almost above 46 dB, which is higher than that with the relative reference strategy and LSB substitution.

To test the authentication performance and tamper detection performance of our AR content authentication with different data-hiding strategies, some modifications are made to the stego image of Figure 13a,b. Take Figure 14a for example—the original content of the stego image remained the same but two extra objects were illegally added and each tamper area is framed with the blue line. By contrast, in Figure 14b, one object of the stego image is replaced with a new object, and an extra object is also added to the stego image. Each tampered area of Figure 13b is framed with a red line as shown in Figure 14b.



Original stego image 1 Tampered stego image 1 Original stego image 2 Tampered stego image 2

(a) (b)

Figure 14. Examples of two tampered stego images. (**a**) Original stego image 1 vs. tempered stego image 1; (**b**) Original stego image 2 vs. tempered stego image 2.

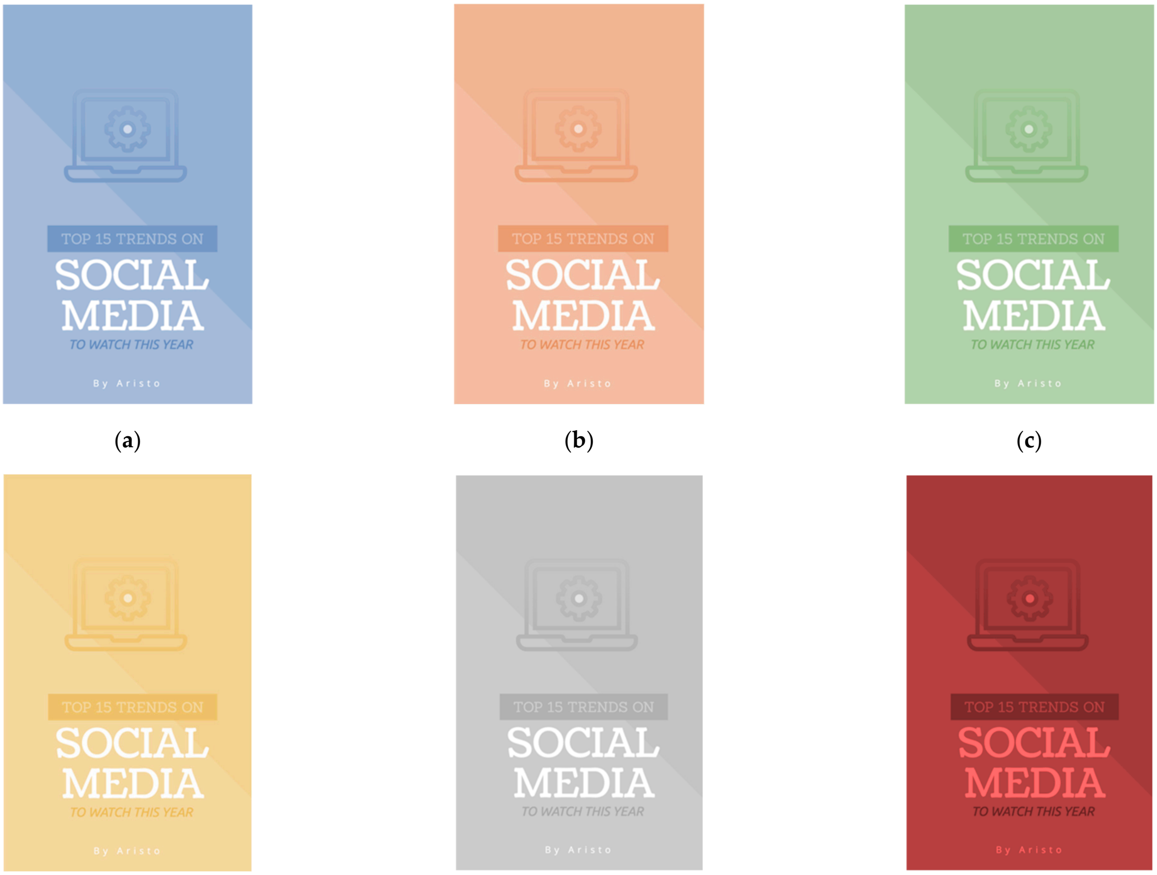
From Tables 5 and 6, we can see that if there is no attack, the hidden authentication data can be 100% extracted. However, when the extra object image is added to the stego image or the original trigger image is replaced with a new image, the data extraction rate will be changed and cannot remain 100%. In this case, the stego image will be determined as unverified. Take into account that those who plagiarize might attempt to alter the hue of an image page of an AR e-book, such as converting it into a blue-toned version, in order to avoid detection by the proposed AR content authentication scheme. Hence, we modified stego image1 and stego image2 to consist of only six individual colors each shown in Figures 15 and 16, aiming to test the efficacy of our proposed content authentication scheme in detecting such manipulations. The outcomes of our experiments are detailed in Tables 7 and 8. These findings indicate that the embedded authentication data can only be reliably extracted within a range of 40% to 80%, indirectly implying the potential manipulation of the image data.

Table 5. Data extraction rates and PSNRs of stego images and tampered stego images with the relative reference strategy and LSB substitution.

Extracted Obj Images		**Similarity (%)**	**PSNR (dB)**
Stego image 1	Untampered	100	59.87
	Tampered	74	39.84
Stego image 2	Untampered	100	59.58
	Tampered	52	32.37

Table 6. Data extraction rates and PSNRs of stego images and tampered stego images with DWT-based data hiding ($\alpha = 0.01$).

Extracted Obj Images		**Similarity (%)**	**PSNR (dB)**
Stego image 1	Untampered	100	62.91
	Tampered	94	55.00
Stego image 2	Untampered	100	57.95
	Tampered	70	50.00



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 15. Examples of converting the original stego image1 to monochrome with tone-mapping: (**a**) tone-mapping to blue color, (**b**) tone-mapping to orange color, (**c**) tone-mapping to green color, (**d**) tone-mapping to yellow color, (**e**) tone-mapping to gray color, and (**f**) tone-mapping to red color.

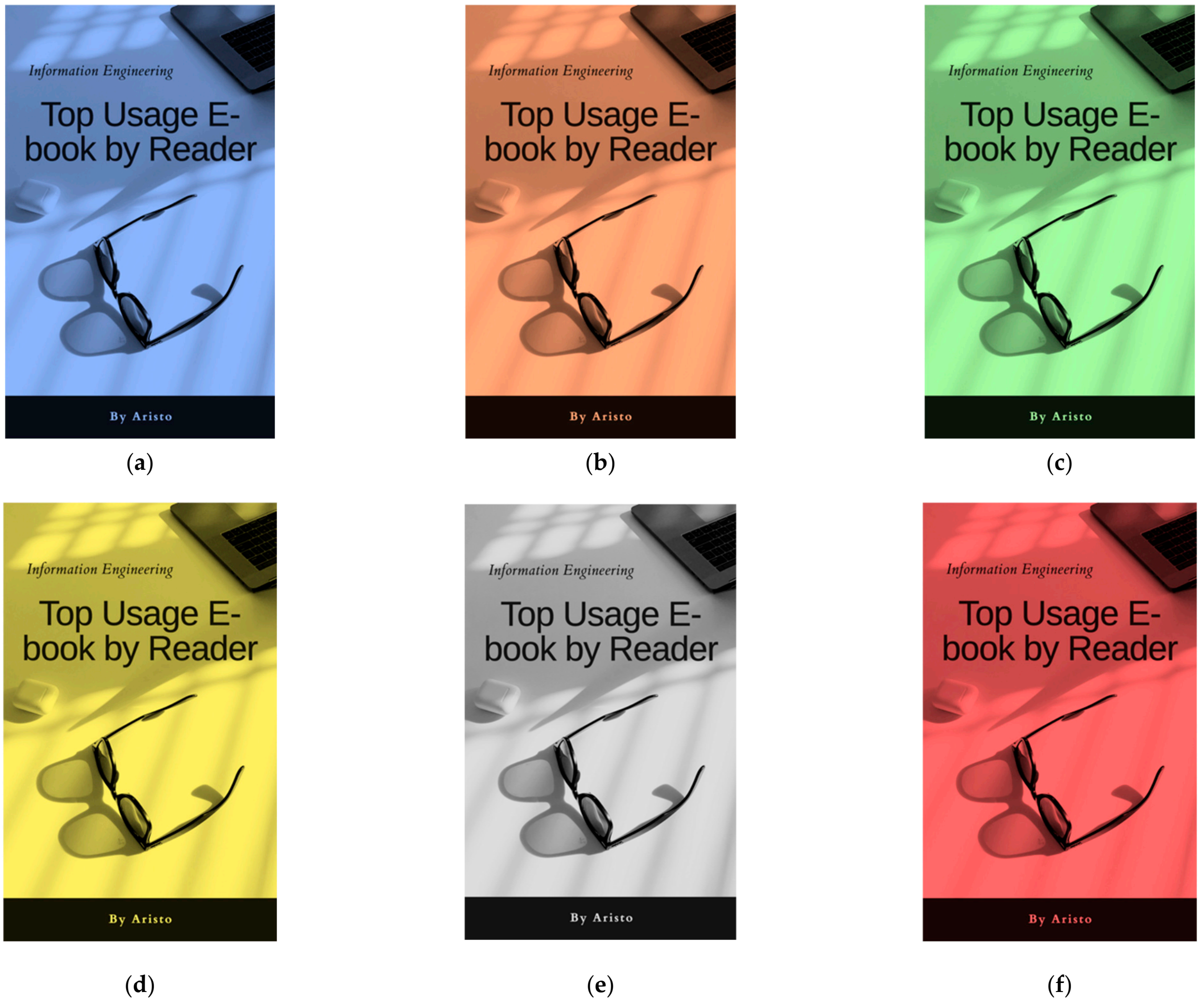


Figure 16. Examples of converting the original stego image2 to monochrome with tone-mapping: (**a**) tone-mapping to blue color, (**b**) tone-mapping to orange color, (**c**) tone-mapping to green color, (**d**) tone-mapping to yellow color, (**e**) tone-mapping to gray color, and (**f**) tone-mapping to red color.

Table 7. Detection results of tempered SI under six different tones (color tones).

Image	Tone-Mapping	Similarity (%)
Tempered stego image 1	Blue	72.00
	Orange	57.04
	Green	57.82
	Yellow	40.00
	Gray	55.12
	Red	45.29

In addition to showing detection rates (similarity) with our proposed authentication framework and two data-hiding strategies, Tables 9 and 10 present the execution times for the embedding phase and the data extraction and tamper detection phase, with attacks depicted in Figures 14 and 15. The average execution time for both embedding and data extraction/tamper detection phases is approximately 22 s. The extraction and tamper detection phase takes slightly longer than the embedding phase because, after extracting the hidden authentication code, tamper detection must be performed to verify the SI.

Table 8. Detection results of tempered SI under six different tones (color tones).

Image	Tone-Mapping	Similarity (%)
Tempered stego image 2	Blue	47.39
	Orange	55.28
	Green	76.30
	Yellow	79.78
	Gray	69.52
	Red	64.23

Table 9. The execution time of the two data-hiding strategies for attacks demonstrated in Figure 14 (unit: seconds).

Data-Hiding Strategies	Image	Tamper	Embedding Phase	Data Extraction and Tamper Detection Phase
The relative reference strategy and LSB substitution	Stego image 1	Untampered	19.32	19.554
		Tampered	19.32	19.555
	Stego image 2	Untampered	20.10	20.331
		Tampered	20.10	20.373
DWT data hiding	Stego image 1	Untampered	21.30	21.765
		Tampered	21.30	21.768
	Stego image 2	Untampered	26.13	26.571
		Tampered	26.13	26.604

Table 10. The execution time of the two data-hiding strategies for attacks demonstrated in Figure 15 (unit: seconds).

Data-Hiding Strategies	Image	Tamper	Embedding Phase	Data Extraction and Tamper Detection Phase
The relative reference strategy and LSB substitution	Stego image 1	Untampered	19.32	19.554
		Tampered	19.32	19.619
	Stego image 2	Untampered	20.10	20.303
		Tampered	20.10	20.317
DWT data hiding	Stego image 1	Untampered	22.20	22.599
		Tampered	22.20	22.602
	Stego image 2	Untampered	26.40	26.796
		Tampered	26.40	26.799

At present, any mobile device with an AR app and camera can read the AR e-book and implement the authentication framework proposed in this study. To ensure the widespread applicability of AR applications, the two authentication code embedding and verification methods designed in this paper do not require high computational power. Tables 9 and 10 confirm that both our data-hiding strategies maintain an average computational time of around 22 s, making them well-suited for real-time AR e-book applications.

6. Conclusions

Augmented reality (AR) is an excellent way to bring people to a truly interactive experience with computer-generated components adding on people's view of the real world. AR not only visually changes natural environments but also provides additional information to users. Establishing the authenticity and integrity of AR content is thus very essential when problems are raised about the origin of AR content. In this paper, an AR content authentication scheme is proposed with two data-hiding strategies: the relative reference strategy with LSB substitution and DWT-based data hiding. The experimental results confirm that DWT-based data hiding offers more robustness of the hidden authentication data under either different tone-converting attacks or extra object insertion or image replacement. With the extraction rate of the hidden authentication, the image page of an AR e-book can be judged as verified or unverified. Once the image page is authenticated, the retrieved 3D objects pop up according to the extracted coordinates of the ROI. The experimental results show that the proposed method can verify the integrity of AR contents while sustaining superior tamper localization accuracy.

In this paper, we do not consider external lighting conditions when verifying the image pages of AR e-books, so our proposed method can effectively detect content attacks targeting hue adjustment. However, external lighting does affect the image tone during image capture, and thieves can manipulate the tone of the image page to evade content detection attacks. Balancing these conflicting concerns will be our next challenge. Furthermore, in the design of our current approach, we initially employed re-rotation techniques to address the non-forward image acquisition problem. Identifying more adaptive approaches in the future is another challenge that needs to be addressed.

Author Contributions: Conceptualization, C.-C.L.; software, A.N.; validation, C.-C.L., M.S. and E.E.; writing—original draft preparation, A.N.; writing—review and editing, C.-C.L.; visualization, A.N.; supervision, C.-C.L.; project administration, E.E.; C.-C.L. reviewed and edited and analyzed the data, performed the experiments, and wrote the paper; M.S. reviewed the paper; and E.E. supported the funding. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Science and Technology Council NSC112-2634-F-005-001-MBK, 111-2410-H-167-005-MY2.

Data Availability Statement: Data available on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Feiner, S.; Macintyre, B.; Seligmann, D. Knowledge-based augmented reality. *Commun. ACM* **1993**, *36*, 53–62. [CrossRef]
2. Gaebel, E.; Zhang, N.; Lou, W.; Hou, Y.T. Looks good to me: Authentication for augmented reality. In Proceedings of the 6th International Workshop on Trustworthy Embedded Devices, Vienna, Austria, 28 October 2016; pp. 57–67. [CrossRef]
3. Wazir, W.; Khattak, H.A.; Almogren, A.; Khan, M.A.; Din, I.U. Doodle-Based Authentication Technique Using Augmented Reality. *IEEE Access* **2020**, *8*, 4022–4034. [CrossRef]
4. Bhalla, A.; Sluganovic, I.; Krawiecka, K.; Martinovic, I. MoveAR: Continuous biometric authentication for augmented reality headsets. In Proceedings of the CPSS '21: Proceedings of the 7th ACM on Cyber-Physical System Security Workshop, Virtual, 7 June 2021; pp. 41–52. [CrossRef]
5. Stephenson, S.; Pal, B.; Fan, S.; Fernandes, E.; Zhao, Y.; Chatterjee, R. Sok: Authentication in augmented and virtual reality. In Proceedings of the IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 22–26 May 2022. [CrossRef]
6. Schneier, B. *Applied Cryptography: Protocols, Algorithms and Source Code in C*; Wiley: Hoboken, NJ, USA, 2015.
7. Wu, M.; Liu, B. Data hiding in image and video. I. Fundamental issues and solutions. *IEEE Trans. Image Process.* **2003**, *12*, 685–695. [PubMed]
8. Lin, P.L.; Hsieh, C.-K.; Huang, P.-W. A hierarchical digital watermarking method for image tamper detection and recovery. *Pattern Recognit.* **2005**, *38*, 2519–2529. [CrossRef]
9. Zhang, X.; Wang, S.; Qian, Z.; Feng, G. Reference Sharing Mechanism for Watermark Self-Embedding. *IEEE Trans. Image Process.* **2011**, *20*, 485–495. [CrossRef] [PubMed]
10. Chang, Y.; Tai, W. A block-based watermarking scheme for image tamper detection and self-recovery. *Opto-Electron. Rev.* **2013**, *21*, 182–190. [CrossRef]

11. Chang, C.-C.; Fan, Y.-H.; Tai, W.-L. Four-scanning attack on hierarchical digital watermarking method for image tamper detection and recovery. *Pattern Recognit.* **2008**, *41*, 654–661. [CrossRef]
12. Sarreshtedari, S.; Akhaee, M.A. A Source-Channel Coding Approach to Digital Image Protection and Self-Recovery. *IEEE Trans. Image Process.* **2015**, *24*, 2266–2277. [CrossRef] [PubMed]
13. Qin, C.; Wang, H.; Zhang, X.; Sun, X. Self-embedding fragile watermarking based on reference-data interleaving and adaptive selection of embedding mode. *Inf. Sci.* **2016**, *373*, 233–250. [CrossRef]
14. Lin, C.-C.; Huang, Y.; Tai, W.-L. A novel hybrid image authentication scheme based on absolute moment block truncation coding. *Multimedia Tools Appl.* **2017**, *76*, 463–488. [CrossRef]
15. Lin, C.-C.; Liu, X.-L.; Tai, W.-L.; Yuan, S.-M. A novel reversible data hiding scheme based on AMBTC compression technique. *Multimedia Tools Appl.* **2015**, *74*, 3823–3842. [CrossRef]
16. Tai, W.-L.; Liao, Z.-J. Image self-recovery with watermark self-embedding. *Signal Process. Image Commun.* **2018**, *65*, 11–25. [CrossRef]
17. Yu, Z.; Lin, C.-C.; Chang, C.-C. ABMC-DH: An Adaptive Bit-Plane Data Hiding Method Based on Matrix Coding. *IEEE Access* **2020**, *8*, 27634–27648. [CrossRef]
18. Nazir, H.; Ullah, M.S.; Qadri, S.S.; Arshad, H.; Husnain, M.; Razzaq, A.; Nawaz, S.A. Protection-Enhanced Watermarking Scheme Combined With Non-Linear Systems. *IEEE Access* **2023**, *11*, 33725–33740. [CrossRef]
19. Li, F.-Q.; Wang, S.-L.; Liew, A.W.-C. Linear Functionality Equivalence Attack Against Deep Neural Network Watermarks and a Defense Method by Neuron Mapping. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 1963–1977. [CrossRef]
20. Tang, Y.; Wang, S.; Wang, C.; Xiang, S.; Cheung, Y.-M. A Highly Robust Reversible Watermarking Scheme Using Embedding Optimization and Rounded Error Compensation. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 1593–1609. [CrossRef]
21. Anand, A.; Singh, A.K. Dual Watermarking for Security of COVID-19 Patient Record. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 859–866. [CrossRef]
22. Chang, C.C.; Liu, Y.; Nguyen, T.S. A novel turtle shell based scheme for data hiding. In Proceedings of the 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), Kitakyushu, Japan, 27–29 August 2014; pp. 89–93.
23. Chen, C.-C.; Chang, C.-C.; Lin, C.-C.; Su, G.-D. TSIA: A Novel Image Authentication Scheme for AMBTC-Based Compressed Images Using Turtle Shell Based Reference Matrix. *IEEE Access* **2019**, *7*, 149515–149526. [CrossRef]
24. Lee, H.R.; Shin, J.S.; Hwang, C.J. Invisible marker tracking system using image watermarking for audgmented reality. In Proceedings of the 2007 Digest of Technical Papers International Conference on Consumer Electronics, Las Vegas, NV, USA, 10–14 January 2007.
25. Li, C.; Sun, X.; Li, Y. Information hiding based on Augmented Reality. *Math. Biosci. Eng.* **2019**, *16*, 4777–4787. [CrossRef] [PubMed]
26. Bhattacharya, P.; Saraswat, D.; Dave, A.; Acharya, M.; Tanwar, S.; Sharma, G.; Davidson, I.E. Coalition of 6G and Blockchain in AR/VR Space: Challenges and Future Directions. *IEEE Access* **2021**, *9*, 168455–168484. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.