Special Paper
방송공학회논문지 제24권 제7호, 2019년 12월 (JBE Vol. 24, No. 7, December 2019)
https://doi.org/10.5909/JBE.2019.24.7.1228
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

Analysis of Storage and Retrieval Results of Audio Sources and Signatures using Blockchain and Distributed Storage System

Kyoung-Sik Lee[a)] and Sang-Kyun Kim[a)‡]

Abstract

Recently, media platforms such as YouTube and Twitch provide services that can generate personal revenue by utilizing media content produced by individuals. In this regard, interest in the copyright of media content is increasing. In particular, in the case of an audio source, competition for securing audio source copyright is fierce because it is an essential element for almost all media content production. In this paper, we propose a method to store the audio source and its signature using a blockchain and distributed storage system to verify the copyright of music content. To identify the possibility of extracting the audio signature of the audio source and to include it as blockchain transaction data, we implement the audio source and its signature file upload system based on the proposed scheme. In addition, we show the effectiveness of the proposed method through experiments on uploading and retrieving audio files and identify future improvements.

Keyword : Blockchain, music copyright, music contents, IPFS, audio signature

Ⅰ. Introduction

Recently, it is possible to easily produce and consume media contents through a personal PC environment or mobile device. As a result, the commerciality and importance of personal media are increasing compared to conventional mass media such as TV and radio. In particular, YouTube, a representative platform of personal media, provides a service capable of generating personal revenue by inserting advertisements into content produced by individuals [1]. Through these services, YouTube accounted for 37% of the domestic advertising market in 2017, and its proportion increased to 40% in 2018 [2].

As a huge amount of media content is created and consumed through various platforms, copyright-related issues are emerging recently. In particular, in the case of an audio source, since it is used as an essential element for the production of almost all media contents including a video, the dispute related to the use of the audio source is more intense. To solve this problem, Twitch, an internet broad-

a) Myongji University
‡ Corresponding Author : Sang-Kyun Kim
E-mail: goldmunt@gmail.com
Tel: +82-2-300-0637
ORCID: https://orcid.org/0000-0002-2359-8709
※ This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2019 R1F1A1041882).
· Manuscript received October 22, 2019; Revised December 11, 2019; Accepted December 11, 2019.

casting platform, is conducting real-time monitoring on audio sources used for content production [3]. In YouTube, if the content is produced in violation of the copyright of the audio source, the revenue generated by this is prevented from being distributed to the content creator [4]. However, there is a need for a fundamental solution that allows the revenue generated from the distribution and use of content including the audio source to be allocated to the owners of the audio sources.

In this paper, we propose a basic method of implementing a system that can retrieve and verify the copyright of audio sources by extracting and storing signatures of audio sources through blockchain and distributed storage systems. Based on the proposed scheme, we implement an audio signature file uploading system and identify the efficiency and future improvements in the actual service provision through experiments on uploading and retrieving audio signature files.

This paper is organized as follows. In section 2, we examine trends related to digital signatures and blockchain applications and identify techniques related to audio signature extraction. In section 3, we propose the method of storing audio source and audio signature in blockchain and distributed storage systems. In section 4, we implement an audio signature file upload system based on the proposed method and measure the time required for uploading and retrieving signature files to identify the efficiency and future improvements in various environments. Section 5 presents the conclusion of this paper and the direction of future research.

Ⅱ. Related technologies

1. Ujo Music

Ujo Music is a blockchain-based music platform service that aims to return the rights and rewards of audio sources to creators. Creators can freely sell their creations and set their rights and prices as needed. Payment for the use of music is made through Ethereum's smart contracts, where revenue is distributed directly to all contributors involved in the creation at the rate specified in the smart contract [5].

Ujo Music's smart contract stores only the unique ID and transaction-related information, which were assigned when the music was initially registered and does not store metadata about the music information directly. All audio

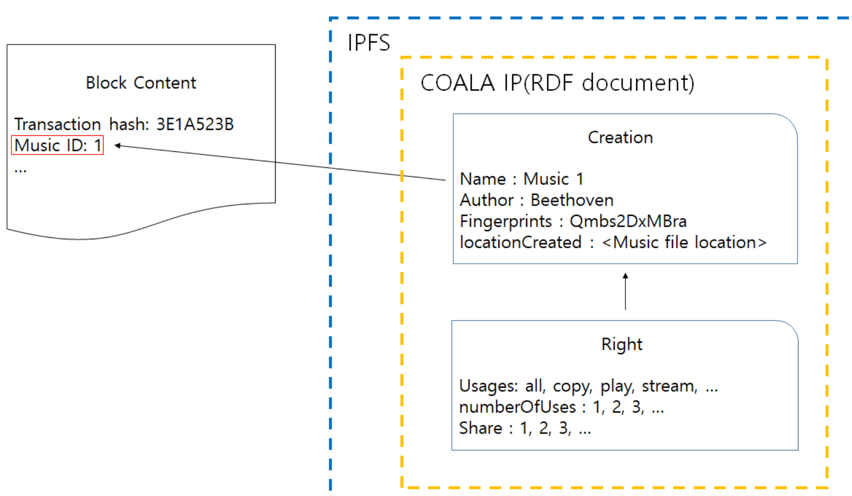

Fig. 1. Method to save music information through Ujo Music's COALA IP [6]

source-related metadata, including the copyright of the generated audio source, are created and managed in the RDF document compliant with the COALA IP protocol [6].

Figure 1 shows an example of Ujo Music's copyright management method. A unique ID of '1' is generated for the first generated audio source, and Ethereum's smart contract is registered with a unique ID '1' and transaction- related information. Basic information about the created audio source is managed through the Creation RDF document. It contains basic information including the name of the audio source, the author, and the fingerprint of the audio source. It also contains the location where the actual audio source is stored. Rights relating to such audio sources are described in the Right RDF document. The Right RDF document contains the scope of use for replication, simple streaming, and the number of uses and shares. In addition to Creation and Right described in the example of Figure 1, additional RDF documents such as User and Group can describe metadata related to the audio source.

2. JPEG-blockchain Framework

The Joint Photographic Experts Group (JPEG), which is leading the standardization of image file formats, has been actively discussing to apply blockchain. In particular, the ISO/IEC 19566-4 JPEG Privacy and Security project is in the process of standardizing blockchain-based image file management systems [7].

In the JPEG-blockchain framework, metadata of an im

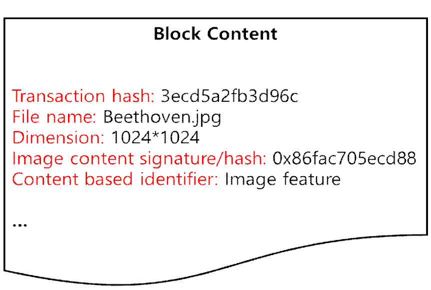

Fig. 2. JPEG-blockchain transaction data example [7]

age file is created as blockchain transaction data and included in a block. Image metadata includes basic information such as file name and file size, as well as information that can represent copyright information such as the signature hash value of the file [Figure 2].

Since there is a restriction on the size of data that can be recorded in a block of the blockchain, it is difficult to include a large file such as an original image as transaction data of the blockchain. To solve this problem, the JPEG-blockchain framework stores the original image file in a separate repository and includes the location information of the repository where the image file is stored in the block transaction data.

3. Dejavu Project

Dejavu is an open-source project developed in Python that processes a sequence of input sources and extracts audio signatures that are unique to each source. By comparing the audio signatures extracted for each audio source, it is possible to determine whether they are the same audio source [8].

The audio source input to déjà vu for feature extraction is transformed into a spectrogram in the form of a waveform through Fast Fourier transform. Next, the peak of each waveform is detected from the transformed spectrogram and represented as a pair of (time, frequency).

After listing the calculated peak values, you can apply the hash function to extract the audio signature of the audio source. In Dejavu, the binarization process is performed once more on the hash values thus obtained, to extract an audio signature that is finally reduced in size compared to the previous hash value.

Ⅲ. Audio source and signature data storage method through blockchain

1. Blockchain Utilization

Blocks in the blockchain are divided into header and transaction data. The transaction data stores all transactions for the generation of the current block, and the block header encrypts the data of the previous block and includes the result as a hash. Since the hash operation performed in the data encryption process of the previous block performs one-way operations only, it is designed to be difficult to decrypt the data of the previous block through the inversion of the encryption results. In addition, since all participants (nodes) of the blockchain share the same block data, it is possible to guarantee the integrity of the data, and it is impossible to tamper with the data by some specific node or blockchain administrator.

By utilizing the characteristics of the blockchain, if the audio signature extracting the characteristics of the audio source is included as transaction data of the blockchain block, it is possible to verify the copyright of a specific audio source through comparison with the audio source uploaded later. However, most blockchains currently limit the size of block transaction data that can be included. Bitcoin limits the block size to 1MB, and Ethereum also limits the size to a maximum of 1MB.

As a result of extracting the audio signature of the audio source and including it in the transaction of the blockchain, it was difficult to directly include the audio signature in the block transaction data [9]. Therefore, there is a need for a method of including the extracted audio signature as blockchain transaction data.

2. IPFS (Inter-Planetary File System)

IPFS is a P2P-based distributed file storage method such as Torrent, which manages one file into several pieces [11]. IPFS first undergoes a series of encryptions to prevent forgery and alteration of the original data file. The encrypted file is then divided into smaller shards and distributed across multiple nodes connected to IPFS.

Data distributed and stored in multiple nodes is managed through a structure called a distributed hash table (DHT). Each partitioned node has a structure of key and value pairs. A unique hash value identifies each node as a key. The corresponding value contains the location where the node is stored or information about the node that is related to it.

In the example shown in Figure 3, a 70MB file was split into three nodes using IPFS. Since the partitioned nodes are managed through the DHT, each has a key and a corresponding value. Each node adds a specific letter or number to the hash value (9627) of the original file (e.g. A9627, B9627, C9627) and uses it as a key. This means that each split node is split from the original file with a hash value of 9627. Each partitioned node has its storage location as a corresponding value, so it can be integrated or managed as an original file.

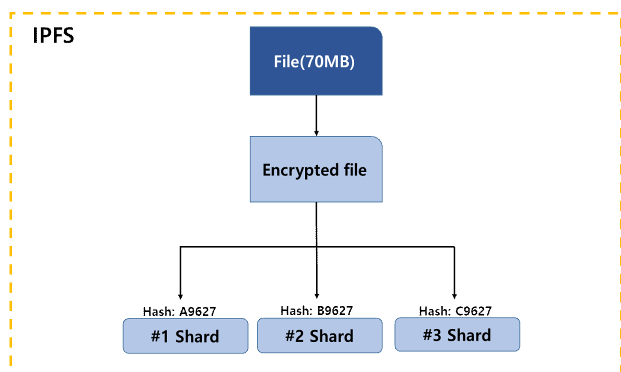

Fig. 3. Example of IPFS shard partitioning and DHT adoption [11]

3. How to store audio sources and signatures using blockchain and IPFS

In this paper, we propose a method for storing audio source and audio signature using blockchain and distributed storage systems.

First, the audio source file and the extracted audio signature file are stored through IPFS. The audio signature was extracted by the method provided by the Dejavu project. An audio source metadata file representing copyright information of the audio source is generated, including the IFPS hash value of the audio source file and signature file returned after storing. Again, the IPFS hash value of the audio source metadata file is generated as block transaction data, and the corresponding transaction is included in the block. The copyright metadata file of the audio source, the audio signature before hashing, and the audio source file are managed by being divided into separate nodes through IPFS. Because nodes of IPFS are managed through DHT structure, it is possible to access original data and signatures stored in IPFS by referring to IPFS hash value included in the sound copyright metadata files.

Figure 4 shows an example of the proposed method. After saving the audio signature and the original source file in IPFS, each IPFS hash value (e.g. 2FB9C7AA, C74DFA2B) is returned. The audio signature file and the original audio source file are divided and managed by IPFS. At this time, each hash value (e.g. 2FB9C7AA, C74DFA2B) includes a specific letter or number to form a key value of each node. In Figure 4, the audio signature file node has a hash value of a2FB9C7AA and b2FB9C7AA as a key, and the sound file node has a hash value of aC74DFA2B and bC74DFA2B as a key. This allows access to the audio signature file or the original audio source file later. The hash value of the audio signature and the original audio source file is included in a metadata file representing copyright information of the audio source. After storing the created metadata file through IPFS, the IPFS hash value of the metadata file is finally included in the block transaction data.

The block transaction data of the proposed scheme does not directly include the hash value of the audio source including the audio signature. Only the IPFS hash value that stores the copyright metadata file of the audio source is included in the block transaction data. The sound metadata file manages the original audio signature and the sound file.

Therefore, it is possible to access the audio source metadata file by referring to the hash included in the block transaction data, and to access the original audio signature by using the hash value in the metadata file. The restored original audio signature is compared with a specific audio source signature to search for a specific audio source.

Ⅳ. Experiment Results

In the case of music companies or related organizations, there is a need to upload a plurality of music-related files at once to store an existing music database to a blockchain

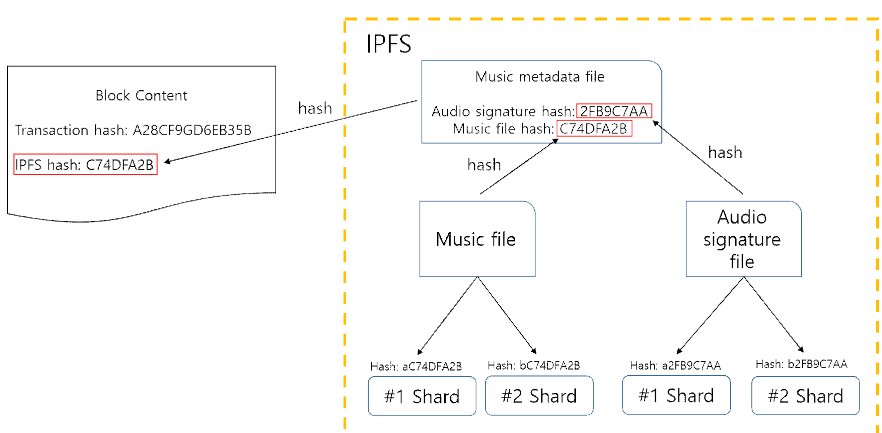

Fig. 4. Audio source and signature storage method using blockchain

database. On the contrary, in the case of the individual user, a search of audio sources stored in the blockchain database is required to register a new audio source.

Assuming this, the stability of file upload is evaluated by measuring the upload time of multiple audio signature files. In addition, we measure the search time of the blockchain database for query audio source signatures to grasp the efficiency of the search process and future improvements.

1. Configuring audio signature file upload system

The audio signature file upload system is implemented by NodeJS in Unix environment and utilizes the truffle framework to construct blockchain network in local environment. In addition, the Ethereum network is applied by using Ganache, and the Ethereum block is distributed through the Ropsten testnet.

2. Upload audio signature file

The audio source and the signature upload method for storing the audio database as a blockchain database are as follows:

1) Store audio signature files over IPFS networks;
2) Return IPFS hash of the saved audio signature file;
3) Generate a metadata file containing the IPFS hash of the audio signature file;
4) Save the metadata file on the IPFS network and return the IPFS hash of that metadata file;
5) Create a transaction that includes an IPFS hash of the metadata file.

In case of uploading several audio signature files at once, processes 1 ~ 4 are repeated as many as the number of audio signature files to upload. Then, the IPFS hash value of the generated metadata file is generated as a separate transaction and included in the block.

3. Upload time measurement test of the audio signature file

In order to grasp the upload stability of the audio signature file, it is assumed that one user uploads multiple audio signature files at once. By measuring the upload time according to the size and number of files to be uploaded, we investigate the processing efficiency of massive audio source data.

Table 1 shows the time required for uploading the audio signature file and the time required to save the IPFS compared to one audio signature file. The IPFS storage time is the time until the creation and storage of the metadata

Table 1. Duration of upload of audio signature file

Number of files	Block creation time (sec)	IPFS storage time (sec)	IPFS storage time per file (sec)
1	8.23	0.20	0.20
100	7.59	22.18	0.22
200	7.23	45.09	0.22
300	7.64	69.56	0.23
400	8.03	92.89	0.23
500	7.95	118.47	0.23
600	9.01	144.21	0.24
700	7.41	171.35	0.24
800	8.04	197.18	0.25
900	8.12	224.82	0.25
1000	7.98	248.86	0.25
Average	7.93	-	0.23

file including the IPFS hash value of the audio signature file. The block creation time refers to the time taken to generate a block by including the IPFS hash of the metadata file as transaction data.

The block creation time was 7.93 seconds on average from 1 file to 1000 files of 7MB on average, and the results were relatively uniform in all sections. This is the time that takes to create a block on the Ethereum Robsten Testnet. In general, the block generation speed of the Ethereum Robsten Testnet is 7-10 seconds, and the Ethereum Mainnet takes 15-20 seconds [12]. For reference, the block generation rate of Bitcoin is about 10 minutes on average [13], and the block generation of EOS takes about 3 seconds [14].

After dividing the IPFS storage time by the number of test files, we can get average storage time of 0.2 ~ 0.25 seconds per signature file in all intervals. The average IPFS storage time was 0.23 seconds. Also, as a result of measuring the IPFS hash size of the audio source metadata file included in the block transaction data, it was found to be 46 bytes per sound file.

The analysis showed that IPFS was storing at an average rate of 0.23 seconds per piece, regardless of the number of audio signature files to upload simultaneously. This means that for even a large number of file upload requests, an average upload speed of 0.23 seconds for each sound file is expected. In addition, since the IPFS hash size of the audio source metadata file is 46 bytes per audio source file, at least tens of thousands of audio metadata file hashes can be recorded in one block.

4. Search for Audio Signature File

The process of retrieving the query audio source file from the audio source database built on the blockchain is as follows:

1) Search for audio source transactions in blocks;
2) Access the audio signature file stored in IPFS by tracking the source metadata hash contained in the transaction data;
3) Comparison of query source signatures with audio signature files in the blockchain database.

The above process is repeated for all audio source transaction data until the same audio signature file is retrieved by the query of audio source file. If the same audio signature file is found, the search is successful. If the same audio signature file is not found even after searching for all blocks, the corresponding audio signature file is regarded as a signature file of a new audio source.

5. Experimental measurement of search time of audio signature files

The purpose of this study is to identify the effectiveness of the individual user's search of the audio source database on the blockchain in order to upload a specific audio source. The blockchain database that stores the audio signature files of a plurality of audio sources is searched, and the time taken to compare with the audio signature files of the query audio sources is measured. At this time, the audio sources of the blockchain database is generated and stores their respective blocks for each file.

Table 2. Audio signature file search time

File size	Time(sec)
1	0.38
100	30.87
200	69.17
300	120.56
400	127.90
500	186.32
600	212.49
700	258.41
800	269.37
900	348.14
1000	375.12

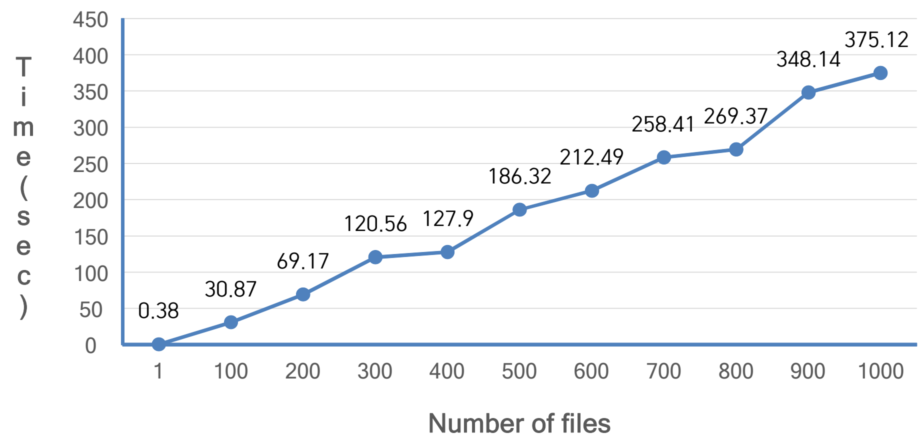

Fig. 5. Search time trend graph

Figure 5 shows the experimental results of Table 2 graphically.

Table 2 shows the time taken to search for the audio signature file. As shown in Figure 5, the larger the number of files stored in the blockchain database, the longer the search time for the query audio signature file. Looking at the change trend, the search time of the query file increases linearly in proportion to the number of stored audio signature files.

Ⅴ. Conclusion and future research

In this paper, we proposed a method to efficiently store audio signature files and original audio source files by applying blockchain and IPFS. Also, based on the proposed method, we implemented the audio signature file upload system to measure the time required for uploading and retrieving files to identify the efficiency and future improvements in various environments.

Through the experiment of uploading the audio signature file, it was confirmed that the IPFS storage speed per file stays constant regardless of the number of uploaded files. In the experiment of searching the audio signature file, it was found that the search time increases in proportion to the number of stored data. Therefore, it is required to introduce a more effective search method for a large number of audio source files and audio signature files. Therefore, further research is needed for a fast database search method through data indexing or audio fingerprint extraction. In addition, it is necessary to confirm the search performance of modulated query audio sources and to study robust audio signature extraction. In addition, the study will be expanded to the system implementation that can verify the copyright of the actual audio source by specifying the contents of the audio source metadata file that specifies the copyright of the audio source.

References

[1] Youtube advertising service, https://www.youtube.com/intl/ko/ads/?&subid=kr-ko-ha-yt-bk-c-plt!o3~CjwKCAjw0N3nBRBvEiwAHMwvNrnMfFqjLUOGS-OMsm8jSxcjPkAIx9Mon7z3BRJc_Ad98wAPoh5iTBoCBokQAvD_BwE~%7badgroup%7d~kwd-463486203081~1727016377~337034874168&gclid=CjwKCAjw0N3nBRBvEiwAHMwvNrnMfFqjLUOGS-OMsm8jSxcjPkAIx9Mon7z3BRJc_Ad98wAPoh5iTBoCBokQAvD_BwE&gclsrc=aw.ds (accessed May. 1, 2019).

[2] 2018 1st half sector analysis report, https://www.slideshare.net/MezzoMedia/2018-106564325 (accessed May. 1, 2019).

[3] Twitch music copyright guide, https://www.twitch.tv/p/ko-kr/legal/

community-guidelines/music/ (accessed May. 1, 2019).
[4] Youtube music copyright guide, https://creatoracademy.youtube.com/page/lesson/artist-copyright?hl=ko (accessed May. 1, 2019).
[5] Ujo music, https://www.ujomusic.com (accessed Jul. 30, 2019).
[6] COALA IP white paper, https://github.com/COALAIP/specs (accessed Jul. 30, 2019).
[7] D. Bhowmil, A. Natu, T. Ishikawa, T. Feng, C. Abhayaratne, "The Jpeg-Blockchain Framework For Glam Services", 2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), San Deigo, CA, USA, vol.1, pp. 1-6, 2018, DOI: https://doi.org/10.1109/ICMEW.2018.8551519.
[8] Dejavu project, https://github.com/worldveil/dejavu (accessed May. 5, 2019).
[9] K. Lee, S. Kim "Use of blockchain for music content copyright protection", Broadcast Engineering Summer Conference, Vol.19, No.6, pp.295-299, 2019.
[10] Bitcoin transcation guide, https://bitcoin.org/en/transactions-guide#introduction (accessed May. 5, 2019).
[11] IPFS Documentation, https://docs.ipfs.io/ (accessed May. 1, 2019).
[12] Ethereum White Paper, https://github.com/ethereum/wiki/wiki/White-Paper (accessed Oct. 1, 2019).
[13] Bitcoin White Paper, https://bitcoin.org/en/bitcoin-paper (accessed Oct. 1, 2019).
[14] EOS White Paper, https://github.com/EOSIO/Documentation (accessed Oct. 1, 2019).

저 자 소 개

Kyoung-Sik Lee

- 2011 ~ 2017 : Bachelor of Science in Computer Engineering, Myongji University
- 2017 ~ Current : Master's course in Computer Engineering, Myongji University
- ORCID : https://orcid.org/0000-0003-1308-2650
- Research interest : 4D media, audio contents, VR and Internet of Things, Blockchain

Sang-Kyun Kim

- 1997 : Degrees of Computer Science, University of Iowa. B.S(1991), M.S.(1995), Ph.D(1997)
- Mar. 1997 ~ Feb. 2007 : Senior researcher of Samsung Advanced Institute of Technology
- Mar. 2007 ~ Feb. 2016 : Professor of Computer Engineering, Myongji University
- Mar. 2016 ~ Current : Professor of the Undergraduate School of Convergence Software(Adjunct Professor of Computer Engineering), Myongji University
- ORCID : https://orcid.org/0000-0002-2359-8709
- Research interest : digital content(image, video and music) analysis and management, 4D media, Blockchain, VR, Internet of Things and multimedia standardization