A Colorization Based Animation Broadcast System with Traitor Tracing Capability

Chih-Chieh Liu[1], Yu-Feng Kuo[2], Chun-Hsiang Huang[2] and Ja-Ling Wu[1,2]

[1]Graduate Institute of Networking and Multimedia,
National Taiwan University
[2]Department of Computer Science and Information Engineering,
National Taiwan University

{ja, pjacky, bh, wjl}@cmlab.csie.ntu.edu.tw

Abstract. Distributing video contents via broadcasting network mechanisms has become a promising business opportunity for the entertainment industry. However, since content piracy is always a serious problem, broadcasted contents must be adequately protected. Rather than implementing sophisticate key-management schemes for access control, an animation broadcast system based on colorization techniques is proposed. In the proposed system, gray-level animation video sequences are delivered via broadcast mechanisms, such as multicast, to reduce the overhead in server processing and network bandwidth. Moreover, color seeds labeled with fingerprint codes are delivered to each client through low-bandwidth auxiliary connections and then used to generate high-quality full-color animations with slight differences between versions received by each client-side device. When a user illegally duplicates and distributes the received video, his identity can be easily found out by examining features extracted from the pirated video. The proposed scheme also shows good resistance to collusion attacks where two or more users cooperate to generate an illegal copy in expectation of getting rid of legal responsibility. The proposed scheme exhibits advantages in network bandwidth, system performance and content security.

Keywords: animation broadcasting, traitor tracing, colorization

1 Introduction

With the establishment of broadband-network infrastructures and the proliferation of network usages, the entertainment industry is exploiting business opportunities related to delivering digital contents over network connections. Among all proposals, on-demand audio/video entertainments [1, 2] may be the most welcomed solution. Digital contents customized according to user preferences and DRM (digital rights management) regulations can be delivered to each subscriber via virtual or physical leased lines immediately after the service provider receives corresponding requests. However, the on-demand approach undoubtedly imposes heavy traffic on backbone networks when appealing contents are provided. Furthermore, the server will be occupied in order to generate many customized versions of the same content. Conventional traitor-tracing schemes [3, 4] where consumer-specific information should be embedded into each piece of content in the server side naturally fits this content-delivery model but also adds inevitable overhead to workload of server.

On the contrary, delivering audio/video entertainments over broadcast channel is another feasible alternative. At the cost of convenience provided by on-demand delivery, content broadcast can greatly alleviate the overhead on both network bandwidth and server performance. However, content broadcasting also introduce new problems. For example, protecting intellectual property rights of content owners is not easy. Currently, broadcast encryption schemes based on complicated key management mechanisms have been proposed, such as [5–7]. However, broadcast encryption schemes often introduce considerable overheads in either network traffic or client storage. Furthermore, once decrypted, received content becomes unprotected and may be distributed arbitrary since the broadcasted content is not customized for individual client at all. Though advanced broadcast encryption schemes can guard the rights of content owners by revoking keys of pirated devices, this type of protection is based on the elimination of pirated devices, rather than protecting each piece of content.

After taking the trade-offs between on-demand content delivery and content broadcast into consideration, an animation broadcast system based on colorization techniques is proposed. In this scheme, gray-level animation video sequences are delivered via broadcast network mechanisms like multicast in the IP network, and color seeds used for rendering full-color videos are transmitted over a low-bandwidth auxiliary channel. In other words, versions of full-color animations possessing features that can be used to identify the illegal user are generated by individual client-side rendering device. The proposed scheme can even resist the collusion attacks that more-than-one users maliciously produce a pirated copy out of their own video sequence.

This paper is organized as follows. Section 2 illustrates the architecture of the proposed animation broadcast system and introduces major modules, including color seeds generation, content delivery via the broadcast channel and the auxiliary channel, rendering full-color animations in the client side and the corresponding traitor tracing mechanism. Section 3 shows experimental results and system performances of the proposed scheme. Section 4 gives some discussions about security issues. Conclusions and future directions of our research are given in Section 5.

2 The Colorization-Based Animation Broadcast System

2.1 Colorization Techniques for Images and Videos

Colorization is a computer-assisted process that adds colors to grayscale images or movies. The work of colorization needs two inputs: one is a grayscale image or video that needs to be colorized; the other is the chrominance side-information. The chrominance information may be interactively provided by user scribbling [8], as well as extracted from images or video with similar color layouts [9, 10]. In conventional colorization techniques targeting on general images or video, the process of searching for good color seeds is usually modeled as an optimization problems so that minimally distorted images can be reproduced, such as introduced in [8].

Furthermore, edge detection also plays a very important role since it is designate to identify the intensity discontinuity and mark out the object edges. In [11], adaptive edge detection schemes and elaborated color-seed searching algorithms together result in better visual quality of generated images.

In our applications, color seeds are obtained by analyzing the full-color version of frames in video sequences, rather than relying on information from user interventions or analysis based on other colorful images. Since the focus of this paper is to demonstrate how the traitor-tracing capability for broadcast video can be readily implemented based on colorization schemes and without loss of generality, frames in animation video sequences are taken as our test contents. Because frames in animations often consist of clear edges and less gradient areas, full-color video with satisfying visual quality can be rendered by simple colorization schemes. Consequently, the modules corresponding to edge detection and color-seeds finding in general video colorization schemes are simplified for the ease of implementation. Furthermore, each frame in an animation sequence is assumed to be compressed and delivered independently.

2.2 Generating the Gray-level Frame and Color Seeds

Fig. 1 shows the server-side operations of the animation broadcast system. A full-color animation frame is firstly converted to its YUV representation. Then, the intensity value of each pixel in the Y component of this frame is uniformly quantized to reduce the number of consisting intensities (i.e. number of bins in the color histogram of the quantized gray-level image), as described by:

$$I'(x,y)=[I(x,y)/Q],\ 0 \leqq x \leqq M-1,\ 0 \leqq y \leqq N-1 \quad (\mathbf{1})$$

where $I'(x,y)$ and $I(x,y)$ are intensity values of the pixel with coordinates (x,y) in the Y component of an M by N frame before and after performing an uniform quantization. Q is the employed quantization step, which is empirically set as 8 in our experiments. Instead of performing complicated edge-detection operations, neighboring pixels in the quantized gray-level image are labeled as connected components based on whether their quantized intensity values are the same. In other words, neighboring pixels sharing similar intensities in the original frame will be clustered into the same connected component. Note that there may be many small connected components in the resulting labeled image. To reduce the number of connected components (so as to reduce the transmission overhead), tiny connected components consisting of less than K pixels will be incorporated into nearby connected component if the difference between their quantized intensity values is less than a threshold value T_1。In the experiments of this paper, K and T_1 are empirically set as 16 and 32, respectively. Then, assume that there are finally C connected components in this frame. The initial color seeds can be calculated as:

$$S=\{(u_1, v_1),\ldots,(u_C, v_C)\} \quad (\mathbf{2})$$

where u_i and v_i are the mean values of consisting pixels in the i-th connected component in the U and the V color domains. When there are totally L users, color seeds required by the frame that will be delivered to user i can be constructed by:

$$S_i = S + a \cdot P_i \quad 0 \leqq i \leqq L-1 \tag{3}$$

where P_i is a vector sequence consisting of C pairs of pseudo-random binary values, and a is simply a weighting factor.

As shown in Fig. 1, a corresponding gray-level frame and L color seeds corresponding to each client-side user will be generated. Each sequence of color seeds is slightly different from other sequences but all of them can be used to produce animation frames of similar visual quality.

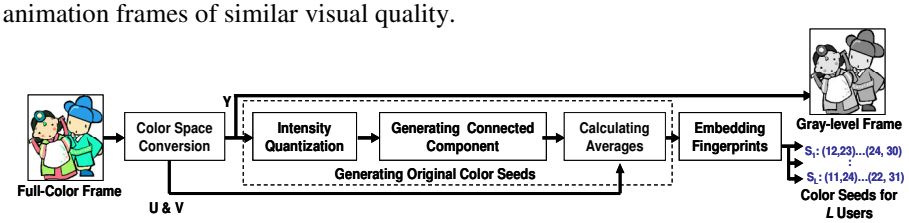


Fig. 1. Generating the gray-level frame and color seeds in the server side

2.3 Animation Broadcasting

The gray-level frame will be delivered to all the users with a lightweight broadcast channel to reduce the overhead on network traffic. In modern network implementations, broadcast functionality is readily provided, e.g. the multicast in IP networks. In this paper, gray-level animations are assumed to be of no commercial values for content pirates, thus no specialized protections are provided. Nevertheless, security mechanisms like access control schemes or digital watermarking can be easily incorporated into the proposed architecture to protect the rights of the gray-level version of animation.

To render the full-color animation video in the receiving end, color seeds for each user are delivered via a low-bandwidth auxiliary channel. Animations displayed on each user's display will share similar visual quality, but the rendered frames are in fact of minute difference in colors.

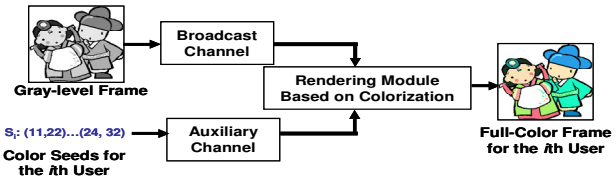


Fig. 2. Delivering the gray-level frame and color seeds and rendering for individual users

2.4 Tracing Traitors

When pirated full-color animation videos are discovered, the traitor tracing mechanism will be invoked to find out the responsible pirate. Identifying features will

be extracted from the discovered frames and compared with relevant data reserved by the content provider. The similarity between a suspect frame and a reference frame previously delivered to a certain user is calculated by:

$$Similarity = \frac{\sum_{i=1}^{C} f^i(\sum_{(x,y)\in CC_i} D^U(x,y), \sum_{(x,y)\in CC_i} D^V(x,y))}{C} \quad (4)$$

$$f^i(a,b) = \begin{cases} 1, & if \quad a > N_i/2 \quad or \quad b > N_i/2 \\ 0, & otherwise \end{cases} \quad (5)$$

$$D^U(x,y) = \begin{cases} 1, & if \quad |U_s(x,y) - U_r(x,y)| < T_2 \\ 0, & otherwise \end{cases} \quad (6)$$

$$D^V(x,y) = \begin{cases} 1, & if \quad |V_s(x,y) - V_r(x,y)| < T_2 \\ 0, & otherwise \end{cases} \quad (7)$$

where $U_s(x,y)$ and $U_r(x,y)$ stand for the values of pixel (x, y) in the U color space of the suspected frame and the reference frame. $V_s(x,y)$ and $V_r(x,y)$ are the counterparts of $U_s(x,y)$ and $U_r(x,y)$ in the V color space. T_2 is a threshold value determining whether two chrominance colors will be regarded as similar and is empirically set as 2. CC_i stands for the i-th connected component and N_i is the total number of pixels in CC_i. If more than one half pixels in a connected component of a suspected frame are regarded as similar to their counterparts in the reference frame, the similarity value will be increase by $1/C$. Therefore, according to the similarity measure, the traitor who illegally distributed the received video can be unambiguously identified according to the distribution of similarity values.

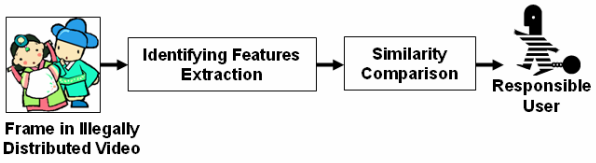


Fig. 3. The traitor tracing mechanism of the proposed broadcast system

3. Experimental Results

In the following experiments, frames adopted from animation movies are used to evaluate the effectiveness of the proposed scheme. Fig. 4 shows all the original frames. In the beginning, the "Chicken Little" frame in Fig. 4(a) is used to

demonstrate the visual quality and the compression ratio of the colorization scheme. Table 1 shows the file sizes and PSNR values when JPEG compression of different quality settings are performed. Table II are corresponding performance using the proposed colorization scheme. Note that the Y component is compressed with JPEG compression quality 100 and the color seeds are compressed by simple VLC coding. In the colorization-based scheme, since the Y component is delivered to users by broadcast channels, this overhead can be neglected when the number of receiving users is large. According to Table I and Table 2, it is clear that, under the condition that similar visual quality is achieved, a user of the traitor-tracing enabled scheme based on colorization receives less amounts of side information as compared with sending each user a compressed and customized version.

Fig. 4. Test animation frames (a) Chicken Little, (b)My Neighborhood Totoro, (c) Shrek and (d) The Incredibles

Table 1. File sizes and visual qualities of JPEG-compressed "Chicken Little" frames

Frame Type	JPEG Quality	File Size (Bytes)	PSNR (dB)
Uncompressed.	N/A	1,843,254	N/A
Full-Color with JPEG Compression	15	20,076	39.09
	20	20,366	39.36
	72	42,225	43.09
	75	45,967	43.53
	100	247,791	47.23

Table 2. File sizes and visual qualities of "Chicken Little" frames based on colorization

Frame Type	# of Color-Seed Pairs	Data Size (Bytes)	PSNR (dB)
Y-Only & JPEG	N/A	66,044	N/A
Color Seeds	1,940	3,119	39.09
	2,737	4,404	39.40
	9,852	12,759	43.07
	11,953	15,684	43.51

Fig. 5 depicts the frames reconstructed by the proposed colorization scheme. Configurations and results of this experiment are listed in Table 3. Though minute rendering differences from the original frame do exist, they are imperceptible under normal viewing conditions.

Fig. 5. Reconstructed animation frames

Table 3. Performance of the colorization scheme using different animation frames

Animations	Frame Size	# of Color-Seed Pairs	PSNR of Recs. Frame (dB)
Chicken Little	1024x600	1,940	39.09
Totoro	480x360	2,290	30.48
Shrek	560x416	1,759	38.27
The Incredibles	608x256	1,055	37.50

Figure 6 shows the visual quality of an animation clip consisting of 90 animation frames, and the frame rate is 30 frames per second. It clearly shows that the fingerprinted color seeds still can be used to generate video of good visual quality.

The differences of PSNR values between the animation frames generated using fingerprinted color seeds and those produced using original color seeds are less than 2 dB.

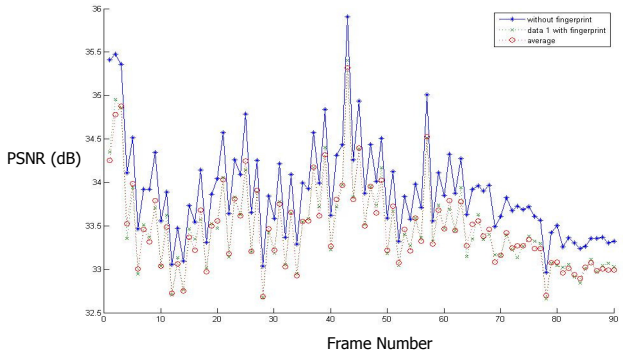


Fig. 6. The visual quality of a animation clip based on color seeds and fingerprinted color seeds

To test the traitor tracing capability of the proposed scheme, 128 frames are rendered based on the same gray-level frame and 128 sequences of color-seed pairs generated from Fig. 4(b), the "My Neighborhood Totoro". Assume that a pirated copy provided by the 10^{th} user is found. Fig. 7 shows the similarity measures, and it clearly identifies the user who is to be blamed as the source of pirate copies.

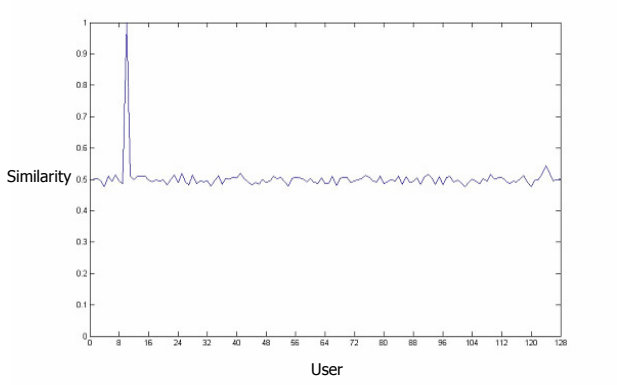
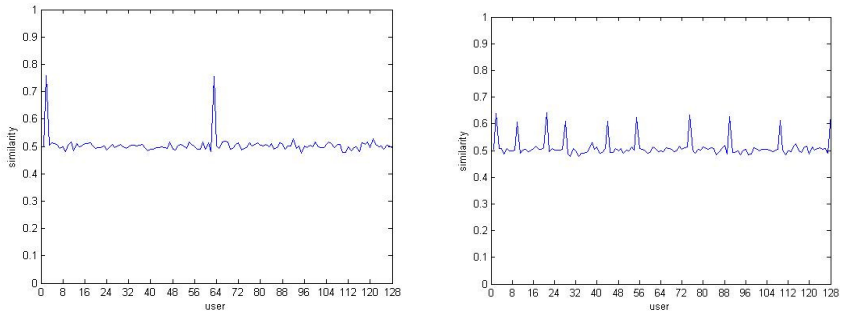


Fig. 7. The proposed scheme unambiguously detects the 10th user who illegally distribute the received animation video

In the literature of fingerprinting, robustness against the collusion attack is another important evaluation criterion. Fig. 8 (a) shows the experimental result that the pirated copy is obtained by averaging the video received by two colluders (the 2^{nd} and 64^{th} users). Fig. 8(b) shows the similarity measures when 10 users colluded together to

generate a pirated copy. In both cases, the proposed traitor-tracing scheme can clearly identify the involved users.



(a) (b)

Fig. 8. The proposed scheme can successfully detect (a) 2 users and (b) 10 users who colluded together to generate the pirated version of animation video.

4. Security Issues

In the proposed architecture of colorization-based animation broadcast system, the security lies in the availability of good color seeds. In this paper, a simplified colorization scheme for generating animation videos of acceptable quality is demonstrated for the purposes of easy implementation and proving of concepts. In fact, color seeds for high-quality entertainment animations must be computed by computers possessing great computational resources for significant computation time. Therefore, the cost of generating color seeds from a full-color animation movie by the pirate is much larger than buying a legal copy. Furthermore, the auxiliary channel must be adequately protected by security measures to avoid eavesdropping or interception of color seeds. Nevertheless, due to the low rate of data delivered via this channel, the overhead is relatively smaller than protecting on-demand video or conventional broadcast video. Finally, the rendering device in the client side shall be temper-proving to prevent the pirate from directly obtaining the color seeds.

5. Conclusions and Future Works

In this paper, a traitor-tracing enabled animation broadcast scheme based on colorization techniques is proposed. The proposed system can reduce the overhead in network traffic and server load as compared with on-demand video delivery and surpass the conventional video broadcast system in that only lightweight rendering module based on colorization is required in the client side. Our future works will be colorization-based video broadcast systems for general high-quality videos.

Furthermore, colorization architectures compatible with videos compressed with important video standards are to be devised.

References

1. Sincoskie, W. D.: System Architecture for a Large Scale Video on Demand Service. Computer networks and ISDN systems, Vol. 22, Issue 2 (1991)
2. Viswanathan S. and Imielinski, T.: Metropolitan Area Video-on-Demand Service Using Pyramid Broadcasting, Multimedia Systems. Vol. 4, NO. 4 (1996)
3. Wu, M., Trappe, W., Wang, Z. J. and Liu, K. J. R.: Collusion Resistant Fingerprint for Multimedia. IEEE Signal Process. Mag., Vol. 21, No. 2 (2004).
4. Zhao H. V. and Liu K. J. R.: Fingerprint Multicast in Secure Video Streaming. IEEE Trans. On Image Processing, Vol.15, Issue 1(2006)
5. Fiat, A. and Naor, M.:Broadcast Encryption. Advances in Cryptography – CRYPTO 93' Proceeding, LNCS, Vol. 773 (1994)
6. Halevy D. and Shamir A.: The LSD Broadcast Encryption Scheme. Advances in Cryptology –CRYPTO '02, LNCS, Vol. 2442, (2002)
7. Lotspiech, J., Nusser S. and Pestoni F.:Anonymous Trust: Digital Rights Management Using Broadcast Encryption. Proceedings of the IEEE, Vol. 92, No. 6 (2004)
8. Anat, L., Dani,L., and Yair, W.: Colorization Using Optimization, Proc. of SIGGRAPH, (2004) pp.689-693
9. Erik R., Michael, A., Bruce G. and Peter S.: Color Transfer between Images. IEEE Computer Graphics and Applications (2001) pp. 34-41,
10. Welsh, T., Ashikhmin, M. and Mueller, K.: Transferring Color to Greyscale Images. ACM Transactions on Graphics (2002)
11. Huang, Y. C., Tung, Y. S., Chen, J. C., Wang S. W. and Wu, J. L.: An Adaptive Edge Detection Based Colorization Algorithm and Its Applications. Proceedings of the 13th annual ACM international conference on Multimedia (2005)