

논문-02-07-4-08

# MPEG-2 기반영상 GOP 단위 편집을 위한 PES 시스템 영역에서의 스트림 접합 방법

김 동 준\*, 최 윤 식\*

## Splicing Techniques in the PES System Layer for the Editing at a GOP Module of The MPEG-2 based Images

Dong-joon Kim\* and Yoon-sik Choe\*

### 요 약

본 논문은 MPEG-2 시스템 스트림 영역에서의 접합(splicing)이 MPEG-2 비디오 화면의 손상 없는 재생을 보장할 수 있는 방법을 연구한다. 기존의 접합은 MPEG-2 시스템의 TS 영역에서 수행하고 있는데, 비디오 문제로 한정시킨 영상편집의 관점에서 보면, 화면의 깨짐이 없는 임의의 픽처에서의 접합을 보장할 수 없다. 본 논문에서는 MPEG-2 시스템의 PES 영역에서의 비디오 화면을 고려한 접합을 다룸으로서 상위의 TS와 PS에서 동일한 목적의 접합을 수행할 수 있도록 하는 연구의 기반을 마련하고자 하였다. 그러기 위해서 먼저, 각각 다른 두 PES 스트림의 접합 시 발생하는 문제와 이로 인해 비디오 화면에 미치게 되는 영향을 비교, 분석하였고, 이를 토대로, PES 영역에서 직접 비디오 화면의 재생시에 일어날 수 있는 문제의 원인을 해결하는 방안을 모색하였다. 그 결과, PES 패킷의 헤더 정보 중 PES 패킷 길이 필드 값을 이용하면, PES 패킷의 상태와 관계없이 GOP 단위의 접합을 보장할 수 있음을 확인하였다.

### Abstract

In this paper, several methods which guarantee the seamless editing when the splicing is applied to the MPEG-2 system streams are described. In the MPEG-2 system standard, it is recommended that the splicing technique is applied in the Transport Stream(TS) domain. However, in view of the video editing, the splicing at an arbitrary picture unit can not be guaranteed. The splicing in the PES domain is only considered in this paper because the PES is one of the MPEG-2 system streams and this result could be directly extended to the video splicing techniques in the TS or PS domain. Thus, the problems that might occur when different two PES streams are spliced and its effects to the video quality are compared and analyzed. Based on this analysis, several methods that can resolve these problems directly in the PES domain are proposed. Consequently, the computer simulation demonstrates that the splicing at a GOP module is guaranteed to have good video quality even without considering the state of the PES packet only if the PES\_packet\_length field of the PES packet header is used.

### I. 서 론

디지털 방송에서는 방송중간에 다른 지역 방송국의 프로

그램이나 상업광고의 삽입과 같이, 기존의 방송 스트림에 다른 방송 스트림이 삽입되는 형태의 접합이 필요할 때가 있다. MPEG-2 시스템 표준에서는 TS 상에서 이러한 접합이 이루어지도록 하고 있는데, 이렇게 함으로서 전송중 실시간 접합이 가능하게 된다<sup>[1]</sup>. 하지만, 이 접합은 비디오나 오디오 영역보다 상위인 시스템 영역에서 이루어지는 것이

\* 연세대학교 전기전자공학과  
Department of Electric and Electronic Engineering, Yonsei University  
※ 마지막으로, 이 논문은 ITRC 사업의 지원을 받았음을 밝힌다.

기 때문에, 비디오 문제로 한정시킨 영상편집의 관점에서 보면, 화면의 깨짐이 없는 임의의 영상에서의 접합을 보장할 수 없다<sup>[1][2]</sup>. 따라서, 실시간에서 임의의 접합이 가능하도록 시스템 영역에서의 비디오 접합에 관한 연구가 필요하게 되었다.

본 논문에서는 우선, TS와 PS의 유료부하가 되는 PES 영역에서의 접합에 초점을 맞추었는데, 먼저 두 개의 다른 PES 스트림을 접합시킬 때, 앞뒤의 스트림이 접합하게 되는 형태를 기초 스트림(비디오 데이터)의 관점에서, 그리고 PES 스트림 자체의 관점에서 고찰한다. 또한 그 접합의 결과가 비디오 화면에 미치는 영향을 비교, 분석한다. 그리고 GOP 단위의 편집을 PES 영역에서 수행할 때, PES 헤더 정보가 가지고 있는 PES 패킷 길이 필드 값을 이용하는 방법을 제안한다.

## II. MPEG-2 시스템

### 1. MPEG-2 시스템 스트림의 구조

MPEG-2 시스템 파트는 시스템 코딩에 관한 전반적인 분야의 권고안이다. 시스템 스트림이 형성되는 과정을 도시한 블록 다이어그램은 다음과 같다<sup>[2]</sup>.

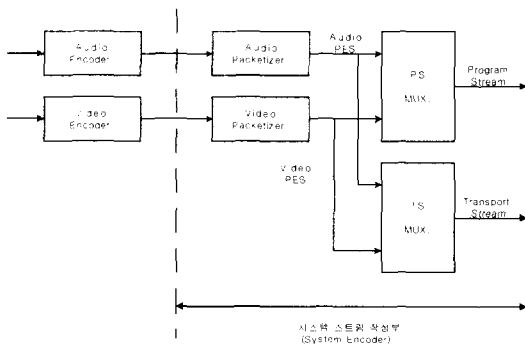


그림 1. MPEG-2 시스템 부호화기의 개요  
Fig. 1. The block diagram of MPEG-2 system encoder

시스템 부호화기의 입력 스트림들은 각각 패킷화기에 의해 PES 패킷이라는 형태로 패킷화 되는데, 패킷은 보통 헤더와 유료부하로 이루어지며, 유료부하 부분에는 기초 스트림(elementary stream)이 삽입되고, 헤더 부분에는 기타의 유용한 정보(시간정보, 주소나 꼬리표 등)가 들어가게 된다. 그리고 이렇게 패킷화된 각각의 PES 패킷들은 사용목적에

따라 각각 다른 과정으로 다중화되어 TS 스트림과 PS 스트림이라는 두 가지 형태의 시스템 스트림으로 복호화 된다.

### 2. Packetized Elementary Stream(PES)

PES 패킷의 계층적 구조는 다음 그림과 같다<sup>[2]</sup>.

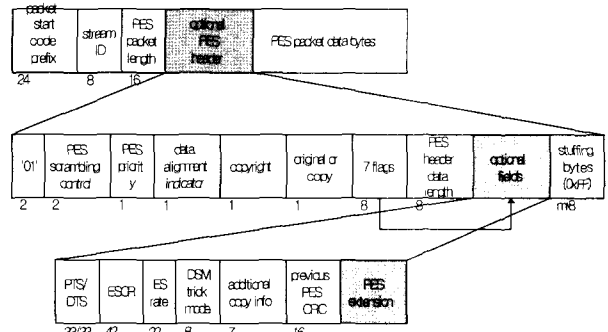


그림 2. PES 패킷의 구조  
Fig. 2. PES packet syntax diagram

PES 패킷은 헤더와 유료부하로 구성되며 헤더에는 현재 패킷의 유료부하에 있는 데이터의 특성 및 전체 스트림과 현재 패킷과의 관계를 나타내는 정보 및 시간 정보 또 유료부하의 길이 정보 등이 있다.

그리고 유료부하는 보통 기초 스트림으로 채워지게 되는데, 한 PES 패킷의 유료부하에는 여러 개의 액세스 단위가 있을 수 있으며, 또 한 개의 액세스 단위가 여러 개의 PES 패킷으로 나뉘어져 들어갈 수도 있다. 보통, 비디오의 액세스 단위는 그 크기가 크기 때문에 여러 개의 PES 패킷으로 나뉘어진다<sup>[1]</sup>.

### III. PES 시스템 영역에서의 비디오 스트림 접합

PES 시스템 영역에서 비디오 스트림을 고려하여 접합을 하려고 할 때 부딪히는 문제들은 보통 두 가지의 관점에서 생각해 볼 수 있다. 그 중 하나는 PES 패킷 자체의 복호화에서 발생하는 문제로 볼 수 있고, 또 다른 하나는 PES 패킷들의 유료부하인 픽처들 사이의 관계에 의한 문제. 즉, 기초 스트림 상에서 발생하는 문제로 볼 수 있다. 하지만, 여기서는 PES 시스템 영역에서의 접합 문제를 다루게 되므로, PES 패킷 자체의 복호화에서 발생하는 문제를 해결하는데 초점을 둔다.

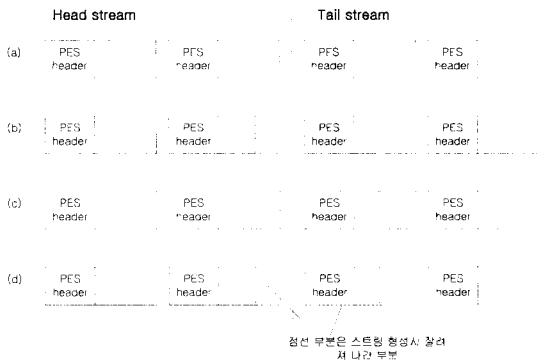


그림 3. PES 영역에서의 접합 형태 (a)완전한 PES 패킷 (b)Head 스트림 손상 (c)Tail 스트림 손상 (d)Head, Tail 스트림 손상  
 Fig. 3. Forms of junction in PES domain (a)Perfect PES packet (b)injury of head stream (c)injury of tail stream (d)injury of head and tail stream

그림 3에서 보는 것처럼 PES 접합에서 접합의 형태는 4 가지로 나타날 수 있는데, 여기서는 PES 헤더 중간에 접합이 일어나는 경우는 제외하였다.

1. GOP 단위에서의 PES 스트림 접합

GOP 단위에서의 접합을 다룰 때, GOP 단위는 꼬리 스트림(tail stream)에서만 만족시키면 된다. 다만, 꼬리 스트림의 첫 번째 GOP가 개방된 GOP인 경우에 GOP 헤더에서 broken\_link를 1로 세팅시켜주어 GOP 첫부분의 B 픽처들이 잘못된 참조 픽처를 가지고 있음을 나타내 주어야 한다.<sup>[3]</sup>

1.1. 꼬리 스트림이 PES 헤더로 시작되는 경우

이 경우에는 머리 스트림(head stream)의 마지막 부분이 완전한 PES 패킷의 형태로 끝났을 때, 접합에서 거의 문제가 발생하지 않는다. 하지만, 머리 스트림의 마지막 부분이 완전한 PES 패킷을 이루지 못하고, PES 패킷의 유료부하 중간에서 잘려졌다면 PES 복호화기에서의 복호화시 심각한 문제가 발생할 수 있다. 그리고 이것은 재생된 비디오 화면에 직접적인 영향을 미치게 된다.

이러한 문제가 발생하는 이유는, PES 복호화기에서 PES 패킷의 PES 패킷 길이 필드의 값을 확인하고, 그 정보를 통해 패킷의 길이를 가정하기 때문이다. 만약 한 PES 패킷(패킷 1이라고 가정)이 유료부하 중간에서 잘려나가고 그 뒤에 새로운 PES 패킷(패킷 2라고 가정)이 오게 된다면, 패킷 1의 PES 패킷 길이 필드의 값에 의해 패킷 2의 시작코드와는 상관없이 패킷 1에서 잘려나간 길이만큼 패

킷 2의 앞부분이 패킷 1의 유료부하 정보로 인식된다. 그리고, 패킷 2에서, 패킷 1의 유료부하 정보로 인식되지 않은 나머지 부분은 PES 복호화기에서 삭제된다. 패킷 2의 헤더가 손상된 것으로 인식되기 때문이다. 이러한 PES 복호화기의 동작원리 때문에 접합된 PES 스트림이 PES 복호화기를 거치면서 비디오 영상에 관련된 중요한 유료부하 정보를 잃어버리게 될 수 있다.

이러한 문제를 해결하기 위해서는 PES 복호화기에서 정보의 손실 없이, 즉 패킷에서 삭제가 이루어지는 부분 없이 복호화가 이루어지도록 할 필요가 있다. 이를 위해서 본 논문에서는 PES 패킷 헤더의 PES 패킷 길이 필드 값을 조절해주는 방법을 제안한다.

이를 위해서 접합이 일어나는 부분인 머리 스트림의 마지막 PES 패킷과 꼬리 스트림의 첫 번째 PES 패킷이 복호화기에서 각각 별개의 패킷으로 인식될 수 있도록 머리 스트림의 마지막 PES 패킷의 PES 패킷 길이 필드 값을 조절해준다. 이 때, 새로운 값 PES\_packet\_length'은 다음과 같이 계산한다.

$$\text{PES\_packet\_length}' = \text{기존의 PES 패킷 길이 필드의 값} - \text{머리 스트림의 마지막 패킷에서 잘려나간 부분의 바이트 수} \quad (1)$$

1.2. 꼬리 스트림이 유료부하로 시작되는 경우

이 경우는 머리 스트림의 마지막 PES 패킷이 온전하더라도, 꼬리 스트림의 첫 번째 PES 패킷의 헤더가 손상되어 그 패킷 전체가 인식되지 않기 때문에 문제가 발생한다. 다시 말해서, 꼬리 스트림의 첫 번째 PES 패킷의 모든 내용은 삭제되고, 이어지는 PES 패킷의 헤더가 나타나는 시점부터 복호화되기 때문에 한 GOP의 중요한 정보들을 전부 혹은 부분적으로 잃어버릴 수 있다.

만약, 머리 스트림의 마지막 PES 패킷도 유료부하 중간부분에서 잘려진 경우라면 오히려 머리 스트림의 PES 패킷 길이 필드 값이 남기 때문에 머리 스트림에서 잘려나간 부분의 길이만큼 꼬리 스트림의 중요 데이터가 어느 정도 보존될 수는 있다. 하지만, 이러한 상황이 일반적이지는 않다.

이 문제를 해결하기 위해서는 1.1절에서와 마찬가지로 PES 복호화기에서 정보의 손실 없이 PES 패킷의 복호화가 이루어지도록 할 필요가 있다.

그렇게 하기 위해서, 꼬리 스트림만 손상된 경우에는 머리 스트림의 마지막 PES 패킷 헤더의 PES 패킷 길이 필드 값을 증가시켜주면 된다. 그렇게 함으로서 PES 복호화기가 꼬

리 스트림의 첫 번째 PES 패키지의 유료부하를 머리 스트림의 마지막 PES 패키지의 유료부하로써 인식하게 만드는 것이다.

이 때, 조절해주어야 할 머리 스트림의 PES 패키지 길이 필드의 값은 다음과 같이 계산된다.

$$PES\_packet\_length' = \text{기존의 PES 패키지 길이 필드의 값} + \text{꼬리 스트림의 첫 번째 패키지에서 남아있는 부분의 바이트 수} \quad (2)$$

머리 스트림과 꼬리 스트림이 모두 손상된 복합적인 경우의 접합의 문제를 해결해주기 위해서는 머리 스트림의 PES 패키지 길이 필드 값을 다음과 같이 계산하여 조절해주어야 한다.

$$PES\_packet\_length' = \text{기존의 PES 패키지 길이 필드의 값} - \text{머리 스트림의 마지막 패키지에서 잘려져 나간 부분의 바이트 수} + \text{꼬리 스트림의 첫 번째 패키지에서 남아있는 부분의 바이트 수} \quad (3)$$

### IV. 실험

#### 1. 실험 방법

그림 4에서 점선으로 표시된 부분이 접합을 수행하는 영역이고, 접합된 최종 스트림을 복호화하는 복호화기는 윈도우미디어 플레이어 7.0과 Elecard MPEG2 플레이어 버전 1.32를 사용하였다. 그리고 실험에 사용된 스트림은 상업광고 동영상으로서 유료부하가 비디오 스트림만으로 이루어져 있고, 일정한 길이로 만들어진 PES 형태의 스트림이다.

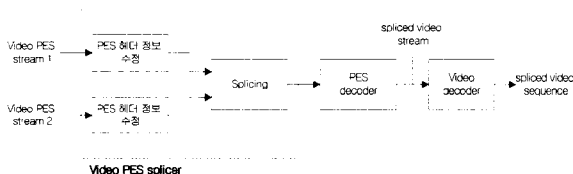


그림 4. PES 영역 접합의 블록 다이어그램  
Fig. 4. Block diagram of PES domain splicing

표 1. 실험 스트림의 유료부하 내용

Table 1. Details of payload of experimental stream

	영상의 크기	GOP 개수	프레임 수	PES 패키지 개수	ES 크기 (Bytes)
머리 스트림	720x480	2	13/15	301	609335
꼬리 스트림	720x480	2	15/15	319	645723

표 2. 접합 부분에서 실험 스트림의 PES 패키지 규격

Table 2. The PES packet spec. of the experimental stream at the splicing region

	전체 크기 (Bytes)	PES_packet_length	헤더 크기 (Bytes)	유료부하 크기 (Bytes)
머리 스트림의 마지막 패키지	2034	2028	9	2025
꼬리 스트림의 첫 번째 패키지	2034	2028	19	2015

다음 그림 5는 실험에 사용된 두 비디오 스트림의 영상이다. 5(a)는 머리 스트림의 끝부분 영상이고, 5(b)는 꼬리 스트림의 시작부분 영상이다.

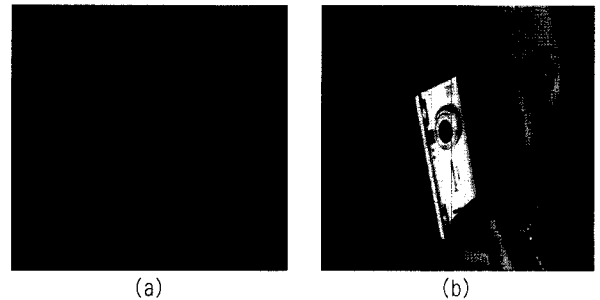


그림 5. 실험에 사용된 영상 (a)머리 스트림 영상 (b)꼬리 스트림 영상  
Fig. 5. Experiment images (a)Head stream image (b)Tail stream image

#### 2. 실험 결과 및 고찰

##### 2.1. GOP 단위에서의 PES 스트림 접합

꼬리 스트림의 패키지는 온전하고, 머리 스트림의 마지막 패키지 끝부분이 잘려진 형태의 접합 결과를 PSNR로 나타내면 그림 6과 같다.

그림 6에서 헤더 정보를 수정해주지 않았을 때의 그래프를 먼저 살펴보면, 27번째 프레임부터 급격히 PSNR이 감소하여 한 GOP에 해당하는 프레임들이 모두 낮은 PSNR

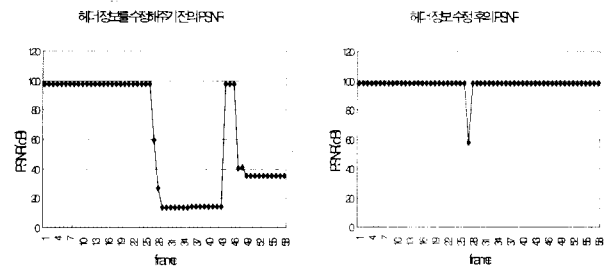


그림 6. PSNR 비교 (머리 스트림 손상)

Fig. 6. Comparison of PSNR (injured head stream)

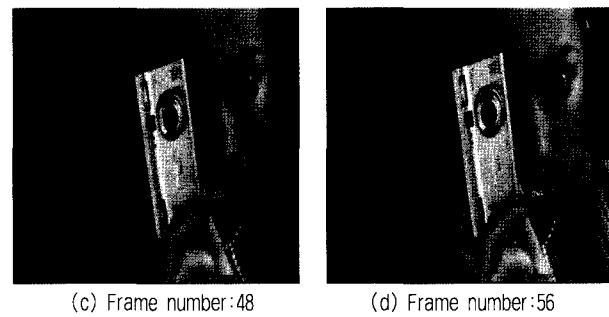
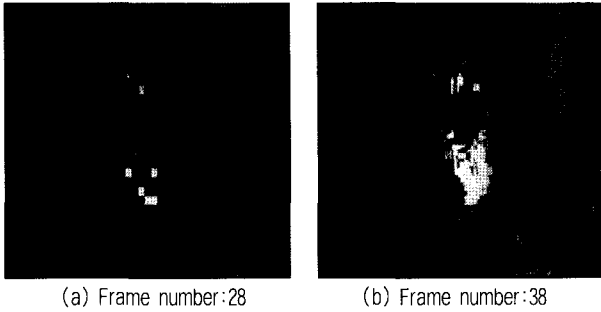


그림 7. 헤더 정보 수정 전의 영상  
Fig. 7. Images before modification

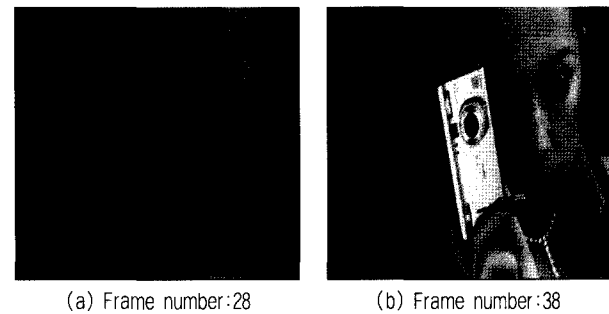


그림 8. 헤더 정보 수정 후의 영상  
Fig. 8. Images after modification

값을 나타내고 있음을 확인할 수 있다. 이로써, PES 패킷의 손상으로 인한 오류가 한 GOP에 전반적으로 미치는 것을 알 수 있다. 그리고 꼬리 스트림의 두 번째 GOP에서도 심각하진 않지만, 그림 7의 (c), (d)에서 볼 수 있는 바와 같이 특정부분(화면의 맨 아래쪽)에 오류가 발생하는 것을 볼 수 있다.

반면, 헤더 정보를 수정해 주었을 때의 PSNR 그래프를 보면, 하나의 프레임만 제외한 모든 프레임이 완전히 재생되었음을 알 수 있다. 여기서, PSNR 98.13이라는 값은 실제로는 무한대, 즉 원본 영상과의 완전한 일치란 의미한다.

머리 스트림의 패킷은 온전하고, 꼬리 스트림의 첫 번째

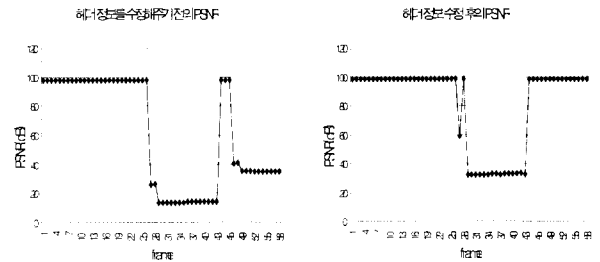


그림 9. PSNR 비교 (꼬리 스트림 손상)  
Fig. 9. Comparison of PSNR (injured tail stream)

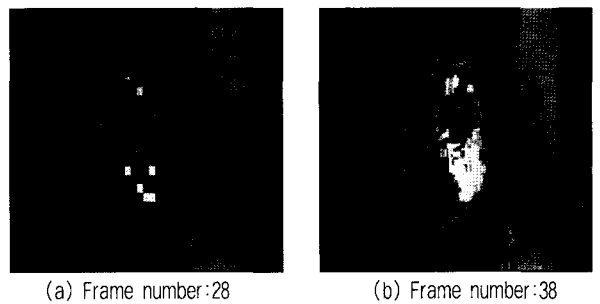


그림 10. 헤더 정보 수정 전의 영상  
Fig. 10. Images before modification

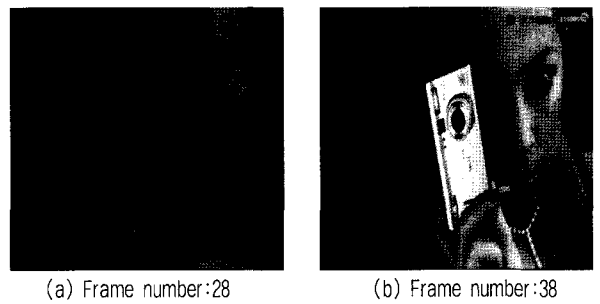


그림 11. 헤더 정보 수정 후의 영상  
Fig. 11. Images after modification

패킷의 헤더 부분이 잘려진 형태의 접합 결과는 그림 9, 10, 11과 같다.

그림 6에서와 마찬가지로 헤더 정보를 수정해 주기 전의 그래프는 접합된 부분 이후의 프레임들이 낮은 PSNR 값을 가지는 것을 볼 수 있다. 그리고 그림 10의 영상을 통해서도 그 영향을 확인할 수 있다.

그런데, 그림 9와 그림 11에서 헤더 정보 수정 후의 상태를 보면, 접합 이후 첫 GOP에 해당하는 프레임들이 헤더 정보 수정 전보다 많이 개선되기는 했지만 약간의 오류가 남아있음을 알 수 있다. 이는 꼬리 스트림의 헤더 정보

에서 손상되지 않고 남아있던 부분이 유료부하로서 여전히 잔류하게 되었을 때 나타나는 현상으로 추측하고 있지만, 이 부분에 대해서는 앞으로의 연구를 통해 논거를 확립하고 개선안을 구해야 할 것이다.

머리 스트림의 패킷과 꼬리 스트림의 패킷이 모두 손상된 경우의 접합 결과는 그림 12, 13, 14와 같다. 이 경우 문제 해결 방식이 꼬리 스트림만 손상된 경우와 유사하므로, 나타나는 결과는 그림 9, 11의 경우와 크게 다르지 않다.

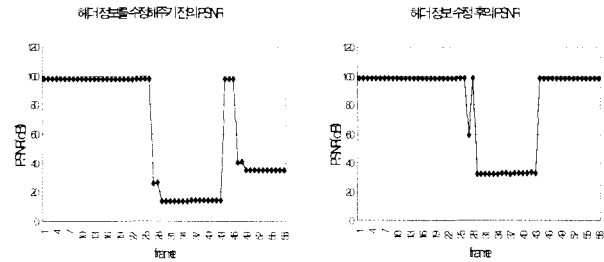


그림 12. PSNR 비교 (머리, 꼬리 스트림 손상)  
Fig. 12. Comparison of PSNR (injured head and tail stream)



(a) Frame number: 28  
(b) Frame number: 38  
그림 13. 헤더 정보 수정 전의 영상  
Fig. 13. Images before modification



(a) Frame number: 28  
(b) Frame number: 38  
그림 14. 헤더 정보 수정 후의 영상  
Fig. 14. Images after modification

## 2. 그 밖의 실험 결과 및 종합

표 3. 접합 결과 (머리 스트림 손상)  
Table 3. Result of splicing (injured head stream)

머리 스트림 손상된 정도 (Bytes)	헤더 정보 수정 전			헤더 정보 수정 후		
	프레임 수	PES 패킷 수	ES 크기 (Bytes)	프레임 수	PES 패킷 수	ES 크기 (Bytes)
4	57	619	1253043	58	620	1255054
15	57	619	1253043	58	620	1255043
150	57	619	1253043	58	620	1254908
1500	57	619	1253043	58	620	1253558

표 4. 접합 결과 (꼬리 스트림 손상)  
Table 4. Result of splicing (injured tail stream)

꼬리 스트림 손상된 정도	헤더 정보 수정 전			헤더 정보 수정 후		
	프레임 수	PES 패킷 수	ES 크기 (Bytes)	프레임 수	PES 패킷 수	ES 크기 (Bytes)
헤더 일부분	57	619	1253043	58	620	1255058
헤더 전체	57	619	1253043	58	620	1255058
GOP 헤더 이전까지	57	619	1253043	58	620	1254773

## V. 결론

본 논문에서는 MPEG-2 시스템 영역 중 PES 스트림 상에서 비디오의 이음새 없는 접합을 구현하는 방법에 대해 고찰해 보았다. 그리고, 그 가능성과 한계를 확인할 수 있었다.

PES 패킷의 헤더 정보 중 PES 패킷 길이 필드 값을 이용하면, PES 패킷이 어느 정도 손상된 상태에서도 꼬리 스트림 유료부하의 GOP 헤더부터 온전하다면 GOP 단위의 비디오 스트림 접합을 거의 오류 없이 보장할 수 있음을 확인하였다.

향후 연구의 방향은 먼저 IV의 2에서 언급했던 헤더 정보 수정 후에 남아있는 오류의 원인 발견과 해결이다. 그리고, 이후 보다 상위의 시스템 영역인 TS나 PS에서의 비디오 스트림 접합으로 이어져야 한다. PES만을 고려하는 시스템 영역에서는 타임 스탬프가 비디오 기초 스트림의 복호화와 재생에 영향을 미치지 않지만, TS나 PS에서는 시간 참조값과 함께 기초 스트림의 복호화와 재생에 중요한 역할을 하게 될 것이다. 따라서 TS, PS 영역에서의 비디오 스트림 접합에서는 이 타임 스탬프 문제가 해결, 혹은 이용해야 할 중요한 주제가 될 것이다. 그리고 한 시스템 스트

림의 전체적인 정보를 담고 있는 PSI(Program Specific Information) 역시 서로 다른 종류의 스트림이 만나는 접합의 문제에서 고려해야 할 대상이다.

### 참 고 문 헌

[1] 유시룡, 장규환, 이병욱, 김종일, 정해복, "MPEG 시스템", 대영사, 2000, 2, 10.

[2] "Information Technology - Generic Coding of Moving Pictures and Associated Audio Information : Systems," *ITU-T Recommendation H.222.0*, Jul, 1995.

[3] 윤범식, "MPEG-2 비디오 스트림 편집 과정에서의 비트율 제어에 관한 연구", *연세대학교 전기컴퓨터 공학과 학위논문[석사]*, 1999.

[4] "Information Technology - Generic Coding of Moving Pictures and Associated Audio Information : Video," *ITU-T Recommendation H.262*, Jul, 1995.

[5] Janusz Konrad, "Visual Communication of Tomorrow : Natural, Efficient and Flexible," *IEEE communications Magazine*, Vol. 31, No. 1, pp. 126-133, Jan, 2001.

[6] P. A. Sarginson, "MPEG-2 : A Tutorial Introduction to the Systems Layer," *The Institution of Electrical Engineers*, 1995.

[7] Kamlesh Talreja and P.Venkat Rangan, "Editing Techniques for MPEG Multiplexed Streams," *IEEE*, 1997.

[8] Ren Egawa, A. Aydin Alatan and Ali N. Akansu, "Compressed Domain MPEG-2 Video Editing with VBV Requirement," *IEEE*, 2000.

[9] Susie J. Wee and Bhaskaran Vasudev, "Splicing MPEG Video Streams in the Compressed Domain", *IEEE*, Jun, 1997.

### 저 자 소개



#### 김 동 준

- 2000년 8월 : 연세대학교 공과대학 전자공학과 졸업
- 2002년 8월 : 연세대학교 대학원 전기전자공학과 졸업
- 2002년 8월~현재 : (주)프리컴시스템 부설연구소 연구원



#### 최 윤 식

- 1975년 3월~1979년 2월 : 연세대학교 전기공학과 공학사
- 1978년 9월~1979년 9월 : (주)대한전선 중전기 사업부 연구직 사원
- 1979년 9월~1981년 8월 : 연세대학교 전기공학과 공학석사
- 1981년 8월~1984년 5월 : 미국 Case Western University system 공학과 MSEE
- 1984년 5월~1984년 8월 : 미국 University of Illinois, Urbana-champaign, 전기공학과 핵융합연구소실 Technical Assitant
- 1984년 8월~1987년 5월 : 미국 Penn.State University 전기공학과 MS
- 1984년 8월~1987년 7월 : 미국 Penn.State University, Dept. of E.E. Graduate Instructor
- 1987년 8월~1990년 12월 : 미국 Purdue University School of Electrical Eng. Ph.D
- 1988년 1월~1990년 10월 : Purdue University, School of E.E. Research Assitant
- 1990년 11월~1993년 2월 : (주)현대전자 산업전자연구소 책임연구원
- 1993년 3월~현재 : 연세대학교 전기전자공학과 조교수, 부교수
- 2000년 3월~현재 : 연세대학교 신호처리연구소 소장