

종단간 상이한 QoS를 갖는 멀티미디어 연출 재생을 위한 CFG 기반의 변환 경로 생성 알고리즘

전 성 미[†] · 임 영 환^{††}

요 약

멀티미디어 연출을 재생하는 단말기나 통신망이 다양해짐에 따라 멀티미디어 데이터를 저장하고 있는 근원지의 데이터 서비스 품질과 재생하는 목적지의 서비스 품질이 다른 경우가 많이 발생하고 있다. 이러한 멀티미디어 연출을 재생하기 위해서 근원지의 데이터를 목적지에서 요구하는 재생 서비스 품질에 맞추어 변환하는 과정이 반드시 필요하다. 문제는 하나의 멀티미디어 연출 프로그램이라도 그것이 재생되는 단말기나 통신망의 환경에 따라 변환 경로가 달라져야 한다는 점이다. 즉 고정된 변환 경로를 사용할 수 없고 재생되는 단말기나 통신망이 정해지면 그에 맞는 변환 경로를 다시 생성해야 한다는 점이다. 본 논문은 주어진 변환 자원과 통신망을 고려하여 근원지 데이터를 목적지에서 요구하는 재생 서비스 품질에 맞게 변환하는 경로를 찾는 알고리즘과 그 경로가 목적지에서 요구하는 서비스 품질에 맞게 재생 가능한지를 검사하는 방법을 제시한다. 본 논문이 제시한 변환 경로를 생성하는 알고리즘은 변환 자원과 변환 규칙을 문맥 자유 문법(Context Free Grammar)으로 기술할 수 있도록 함으로써 변환 경로를 컴파일러 기법을 이용하여 생성할 수 있도록 하였다. 그리고 재생 가능성을 검사하기 위하여 변환 과정에서 소요되는 시간과 지연 등을 계산하는 방법과 요구하는 시간 내에 이루어 질 수 있는지 검사하는 방법을 제안하였다. 그리고 이러한 방식을 실험실에서 개발한 TransCore라는 멀티미디어 스트림 엔진과 VIP라고 하는 저작도구에 구현하였고 간단한 시나리오에 적용한 결과를 마지막으로 기술하였다.

A Context Free Grammar based Algorithm for Generating Playable Transcoding Paths of the Multimedia Presentation with Different End-to-End QoS

Sungmi Chon[†] · Younghwan Lim^{††}

ABSTRACT

Since various terminals and different networks get involved in playing of a multimedia presentation, the case that the presentation QoS at a destination should be different from the QoS of multimedia data at a source occurs frequently. For playing the multimedia presentation, the multimedia data at a source should be transcoded into the multimedia data satisfying the QoS required at the terminal. The problem of generating the transcoding path is that, for a given multimedia presentation, different transcoding paths depending the terminal and networks should be generated. That is, a fixed transcoding path cannot be applied to a multimedia presentation. Instead, whenever the terminal and network to play the presentation get determined, a proper transcoding path should be regenerated automatically. In this paper, the algorithm for generating the transcoding path and the method for checking the playability of the generated path are proposed. The generating algorithm adopted the technique of Context Free Grammar in describing the set of transcoding resources and a user's transcoding rules in order to utilize the well-known compiler techniques. For the playability check, a method of computing the transcoding time and the delay time between data units are proposed. Finally all the proposed methods were implemented in the stream engine, called TransCore and the presentation-authoring tool, called VIP, we had developed. And the test results with a sample scenario were presented at the last.

키워드 : 멀티미디어 연출(Multimedia Presentation), 멀티미디어 변환(Multimedia Transcoding), 서비스 품질(QoS), 스트림(Stream)

1. 서 론

기존의 분산 멀티미디어 시스템 환경에서 서비스 품질(Quality of Service, QoS)에 관련된 연구는 많이 있다[1-5]. 이러한 연구들의 주요 관심은 통신 서비스 품질이다. 따

라서 대역폭, 지연, 지터, 패킷 손실 확률 등을 서비스 품질의 파라미터로 사용하고 있으며[6, 7], 파일 형식, 색상, 크기, 프레임 율 등과 같은 재생 서비스 품질 파라미터는 근원지와 목적지에서 서로 동일하다고 가정하고 있다[8, 9]. 즉, 근원지와 목적지가 서로 다른 재생 서비스 품질을 갖는 경우에 대한 연구는 드물다.

근원지와 목적지가 서로 다른 재생 서비스 품질을 갖는 경우는 사용자의 다양한 요구에 의해서 또는 목적지의 하

[†] 준 최 원 : 숭실대학교 대학원 컴퓨터학과

^{††} 종신회원 : 숭실대학교 미디어학부 교수

논문접수 : 2002년 7월 29일, 심사완료 : 2002년 9월 25일

드웨어나 네트워크의 제약 때문이다. 이러한 환경에서 멀티미디어 스트림이 최종적으로 목적지가 원하는 재생 서비스 품질을 만족시키기 위해서는 서비스 품질을 바꾸어 주는 일련의 변환 자원들을 통과하여야 한다[10].

그런데 상황에 따라 필요한 변환 자원의 종류가 다르고, 변환 자원이 여러개 필요할 때는 그 변환 자원들을 연결하는 방법도 여러 가지이다. 따라서 근원지와 목적지가 서로 다른 재생 서비스 품질의 멀티미디어 스트림을 재생하기 위해서 상황에 따라 다른 변환 자원들의 연결인 변환 경로를 자동적으로 생성하여야 한다. 또 멀티미디어 스트림이 여러개의 변환 자원을 통과하는 동안 지연되어 목적지가 원하는 대로 재생이 가능하지 않을 수도 있으므로, 여러 개의 변환 자원을 통과한 후에도 목적지가 원하는 대로 재생이 가능한지를 확인하여야 한다.

이를 위해서 본 논문에서는 변환 자원들의 연결인 변환 경로의 자동 생성 방법과 그 변환 경로가 목적지가 원하는 대로 재생이 가능한지 확인하는 재생성 검사 방법을 제안한다. 그런데 이러한 변환 경로의 생성을 시도할 때 아무런 전략을 고려하지 않는다면, 모든 변환 자원의 종류에 대하여 순열의 형태로 연결을 시도할 것이므로 응용 프로그램의 부담이 증가하게 된다. 따라서 응용의 부담을 줄이면서 변환 경로를 생성하는데 CFG(Context Free Grammar)를 이용하는 방법의 틀을 제공하고, 그 결과로 나온 변환 경로가 재생 가능한지를 확인하는 재생 가능한 변환 경로를 생성하는 알고리즘을 제안하였다. 그리고 이러한 방법을 당 실험실에서 개발한 TransCore라는 멀티미디어 스트림 엔진과 멀티미디어 연출 저작 도구인 VIP에 구현하여 적용하였다.

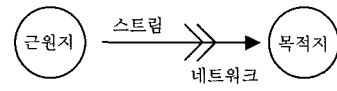
2. 변환 경로 생성의 문제점 및 해결 방향

근원지와 목적지가 서로 다른 재생 서비스 품질을 갖는 경우는 사용자가 원래 가지고 있던 멀티미디어 스트림을 다른 형식으로 재생하기를 원하거나 핸드폰, PDA와 같은 목적지의 하드웨어 또는 PSTN, 이동 통신망과 같은 네트워크의 제약 때문에 멀티미디어 스트림의 재생에 제약을 받을 때 발생할 수 있다.

그런데 서비스 품질과 관련된 기존의 연구들은 서비스 품질 개념이 데이터 전송의 기술적인 특성을 기술하기 위하여 통신 분야에서 처음 나타난 것에서 알 수 있듯이 서비스 품질의 파라미터로 대역폭, 지연, 지터, 패킷 손실 확률등 통신 서비스 품질에 관한 연구가 많다[6, 7]. 또한 서비스 품질을 자원 관리와 연관지어 자원을 할당하는 스케줄링에 활용한 연구도 많이 있다[11-13]. 이러한 연구들은 모두 멀티미디어 연출의 재생을 위해서 근원지와 목적지가 갖고 있거나 요구하는 서비스 품질이 서로 동일함을 기본 조건으로 하고 있다[8, 9].

근원지에 저장된 멀티미디어 스트림을 목적지에서 재생

하려고 할 때 다음과 같은 그림으로 표현하기로 한다.



(그림 2-1) 근원지의 스트림을 목적지에서 재생할 때의 그림 표현법

근원지에 있는 동일한 멀티미디어 스트림에 대하여 상황에 따라 목적지가 원하는 재생 서비스 품질이 다음의 예에서 보는 것처럼 달라질 수 있다

[예 2-1] 환경에 따라 근원지와 목적지가 서로 다른 재생 서비스 품질을 갖는 예

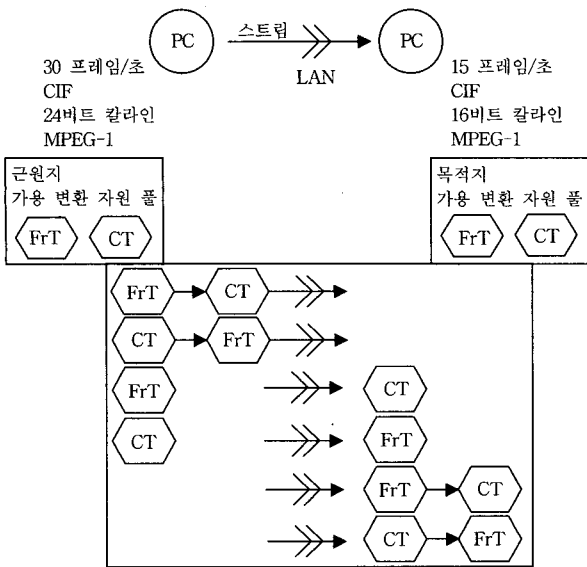
근원지의 재생 서비스 품질	동작 환경			목적지의 재생 서비스 품질	재생 서비스 품질이 달라진 원인
	근원지의 시스템 종류	통신 환경	목적지의 시스템 종류		
30 프레임/초 CIF 24비트 칼라인 MPEG-1 파일	PC	LAN	PC	15 프레임/초 CIF 16비트 칼라인 MPEG-1 파일	사용자의 요구에 의해서
		PSTN	PC	10 프레임/초 CIF 24비트 칼라인 H.263 파일	네트워크의 제약에 의해서
	이동 통신망	휴대폰	5 프레임/초 QCIF 8비트 흑백인 MPEG-4 파일	네트워크와 목적지 하드웨어의 제약에 의해서	

위의 예와 같이 근원지와 목적지가 서로 다른 서비스 품질을 갖을 때 목적지가 원하는 대로 멀티미디어 스트림을 재생하려면 근원지에 있는 데이터를 목적지의 재생 서비스 품질을 만족하도록 변환시켜주는 변환 자원들의 연결이 필요하다.

트랜스코더에 대한 기존의 관련 연구로는 각종 파일 형식을 변환하는 트랜스코더[14-17], 크기를 변환하는 크기 트랜스코더[18, 19], 프레임 율을 변환하는 프레임 율 트랜스코더[20]들이 제안되어 있다.

[예 2-2] 근원지와 목적지의 서로 다른 재생 서비스 품질을 만족시키기 위해서 필요한 변환 자원들이 연결된 예

[예 2-1]의 처음 사례는 PC에 있는 30 프레임/초, CIF, 24비트 칼라인 MPEG-1 파일이 LAN을 통하여 PC인 목적지에 15 프레임/초, CIF, 16비트 칼라인 MPEG-1 파일로 재생되는 예이다. 이 경우에 프레임 율과 색상에 대한 재생 서비스 품질이 근원지와 목적지에서 서로 다르므로 이것을 만족시키기 위해서 프레임 율을 변환해주는 프레임 율 트랜스코더(Frame Rate Transcoder, FrT)와 색상을 변환해주는 색상 트랜스코더(Color Transcoder, CT)가 필요하다는 알 수 있다. 이 두 개의 변환 자원이 모두 시스템에 있을 때 이것들을 연결하는 방법은 여섯 가지가 있음을 (그림 2-2)에서 표현하였다.



(그림 2-2) 필요한 변환 자원들이 연결된 여러 예

이처럼 근원지에 저장되어 있는 멀티미디어 데이터의 서비스 품질과 재생할 목적지의 서비스 품질이 서로 다른 경우에 여러 개의 변환 자원을 통과하여 최종적으로 원하는 서비스 품질로 바뀌게 된다. 이때 발생하는 문제점은 스트림이 연출되는 환경에 따라 변환 자원들의 연결인 변환 과정이 달라지므로 재생될 환경이 목적지와 통신망에 의해서 정해질 때마다 그에 적합한 변환 과정이 필요하다는 점이다.

이에 대한 해결 방향은 다음과 같다.

첫째, 필요한 변환 자원들의 연결인 변환 경로를 만드는 방법으로 먼저 사용자나 프로그래머가 변환 자원의 종류와 순서를 직접 지정하는 방법을 사용할 수 있다. 그러나 멀티미디어 연출 프로그램은 재생될 단말기나 통신망에 따라 필요한 변환 자원이 다르고, 변환 자원에 따라 변환 경로가 달라지므로 일반적인 유무선 인터넷 환경에 이 방법을 적용할 수 없다. 따라서 사용자나 프로그래머가 변환 자원의 종류와 순서를 직접 지정하지 않고, 프로그램에서 변환 경로를 자동으로 생성하는 방법이 필요하다. 그런데 이 경우에 선행 변환 자원의 출력 스트림이 갖는 데이터 형식과 동일한 형식의 입력을 수용하는 모든 가능 자원을 연결하다가 목적지가 원하는 서비스 품질에 도달하면 변환 경로의 생성을 중지하도록 한다면, 변환 자원들간에 연결되는 경우의 수가 너무 많아져서 시스템 부담이 증가할 것이다. 따라서 경험자의 변환 자원 결합에 대한 지침을 시작으로 하는 CFG를 이용하여 변환 경로를 자동 생성하는 방법의 틀을 제안한다.

둘째, 변환 자원이 연결된 변환 경로일지라도 많은 변환 자원을 통과하는 동안 스트림이 지연되어 목적지가 원하는 대로 재생이 가능하지 않을 수도 있으므로, 스트림이 목적지가 원하는 대로 도착하는지를 확인하는 방법이 필요하다. 따라서 변환 경로에 대해서 목적지가 요구하는 대로 재생 가능한지를 확인하는 방법을 제공하여 재생 가능한 변환 경

로를 생성하는 알고리즘을 구현한다.

3. CFG 기반 변환 경로 생성 알고리즘

근원지와 목적지가 서로 다른 재생 서비스 품질을 갖는 환경에서 멀티미디어 스트림이 목적지가 원하는 재생 서비스 품질을 만족시키기 위해서는 서비스 품질을 바꾸어 주는 변환 자원들의 연결인 변환 경로를 생성하여야 한다.

그런데 상황에 따라 필요한 변환 자원의 종류가 다르고, 변환 자원이 여러개 필요할 때는 그 변환 자원들을 연결하는 방법도 여러 가지이므로, 변환 경로를 자동적으로 생성하여야 한다. 변환 경로를 생성할 때 모든 변환 자원의 종류에 대하여 연결을 시도하면, 응용 프로그램의 부담이 증가하므로, CFG를 이용하는 방법의 틀을 제공하여 응용의 부담을 줄이면서 변환 경로를 생성하는 알고리즘을 제안한다.

[정의 3-1] 변환 자원 및 변환 경로의 정의

변환 자원이란 멀티미디어 스트림 처리기에서 근원지와 목적지가 서로 다른 재생 서비스 품질의 멀티미디어 스트림을 재생 하려고 할 때, 목적지가 원하는 재생 서비스 품질로 멀티미디어 스트림을 변환할 때 사용하는 자원이다.

또한 변환 경로란 근원지에 있는 멀티미디어 데이터의 서비스 품질을 목적지인 단말기에서 원하는 서비스 품질에 맞게 변환해 가는 과정을 말한다. 즉, 근원지로부터 목적지 사이에 변환 자원이 연결되어 있는 경로이다.

변환 자원의 종류는 엔코더(Encoder, EC), 디코더(Decoder, DC), 프레임 율 트랜스코더(Frame Rate Transcoder, FrT), 색상 트랜스코더(Color Transcoder, CT), 크기 트랜스코더(Size Transcoder, ST), 파일 형식 트랜스코더(File Transcoder, FT)가 있어서 각각 원하는 형태로 압축을 하거나 압축을 풀며, 프레임 율 · 색상 · 크기 · 파일 형식을 변환시켜주는 기능을 갖는다.

또 변환 자원의 입력 속성값 · 출력 속성값에는 MPEG-1 · H.263과 같이 압축 파일 형식에 대한 표현으로 입출력 속성값을 나타낼 수 있고, RGB · YUV와 같이 비압축 멀티미디어 스트림에 대한 입출력 속성값을 나타낼 수 있다. 그리고, 1에서 30 프레임까지를 변환할 수 있는 프레임 율 트랜스코더와 같이 입출력 속성 외에 기타 속성을 갖고 있는 변환 자원들도 있다.

이러한 변환 자원은 다음과 같이 표현하기로 한다.

[정의 3-2] 변환 자원의 표현 방법

입력 속성 변환 자원의 종류 출력 속성
기타 속성

변환 경로 생성 알고리즘은 사용자가 변환 자원의 종류와 순서를 직접 지정하지 않고 프로그램에서 변환 경로를 자동으로 생성한다. 그런데 이때 모든 변환 자원들의 연결

을 시도하지 않고 경험자의 변환 자원 결합에 대한 지침을 시작으로 하는 CFG를 이용하는 방법의 틀을 제한한다.

CFG 기반 변환 경로 생성에 대한 전반적인 알고리즘은 다음과 같다.

입력 : •연출 제어 프로그램이 가지고 있는 근원지 서비스 품질 정보, 목적지 서비스 품질 정보 •시스템이 가지고 있는 가용 자원 •변환 경로를 생성하기 위한 형식 문법 (S, V _N , V _T , P)
단계 1 : 근원지, 목적지가 가지고 있는 서비스 품질 정보를 멀티미디어 스트림 처리기가 분석하여 근원지와 목적지 사이에 위치시킬 필요가 있는 변환 자원의 종류를 파악한다. 단계 2 : 실제로 시스템이 가지고 있어서 사용할 수 있는 가용 변환 자원을 파악한다. 단계 3 : CFG를 이용하여 시작에서 생성 규칙을 따라 변환 경로를 생성하여 모든 논 터미날이 터미날로 바꾸어 질 때 까지 반복한다. 이때, 터미날로 모두 바뀐 것이 변환 경로이다.
출력 : CFG에 기반하여 생성된 목적지 서비스 품질을 만족하는 모든 변환 경로

(알고리즘 3-1) CFG 기반 변환 경로 생성 알고리즘

(1) CFG 기반 변환 경로 생성 알고리즘의 1 단계 : 필요한 변환 자원 파악 단계

멀티미디어 스트림 처리기의 하나인 트랜스코어(Trans-Core)는 멀티미디어 스트림 연출에 관한 모든 정보를 prs 라는 확장자를 갖는 연출 제어 프로그램에 가지고 있어서 그곳의 서비스 품질 정보를 분석하면, 종단간에 어떤 변환 자원이 필요한지 파악할 수 있다.

[예 3-1] 필요한 변환 자원 파악 단계의 예

근원지에 있는 CIF, 24비트 칼라, 30프레임/초인 MPEG-1 멀티미디어 파일을 목적지에서 QCIF, 8비트 흑백, 5프레임/초인 H.263로 재생하려고 한다.

이 경우, 근원지와 목적지의 재생 서비스 품질이 서로 다르므로, 재생 서비스 품질을 만족시키기 위해서 파일 형식 트랜스코더, 프레임 울 트랜스코더, 크기 트랜스코더, 색상 트랜스코더, H.263 디코더와 같은 변환 자원이 필요하다.

(2) CFG 기반 변환 경로 생성 알고리즘의 2 단계 : 가용 자원 파악 단계

현재 시스템에 실제로 존재하는 가용 변환 자원들을 파악한다. 만약 어떤 변환 자원이 필요해서 그 변환 자원을 포함하여 경로를 생성하였는데, 실제로 그러한 변환 자원이 시스템에 없다면 그 경로는 종단간에 연결이 되지 못할 것이기 때문이다.

[예 3-2] 가용 자원 파악 단계의 예

현재 시스템에는 파일 형식 트랜스코더 2개(입력 속성 MPEG-1, 출력 속성 H.263인 것과 입력 속성 H.263, 출력 속성 MPEG-1인 것), 프레임 울 트랜스코더 4개(입출력 속성이 모두 H.263인 것, 모두 MPEG-1인 것, 모두 YUV인 것과 모두 RGB인 것), 크기 트랜스코더 4개(입출력 속성이

모두 H.263인 것, 모두 MPEG-1인 것, 모두 YUV인 것과 모두 RGB인 것), 색상 트랜스코더 4개(입출력 속성이 모두 H.263인 것, 모두 MPEG-1인 것, 모두 YUV인 것과 모두 RGB인 것), MPEG-1 디코더 1개(입력 속성 MPEG-1, 출력 속성 YUV), H.263 디코더 1개(입력 속성 H.263, 출력 속성 YUV), H.263 엔코더 2개(입력 속성 YUV, 출력 속성 H.263인 것과 입력 속성 RGB, 출력 속성 H.263인 것), MPEG-1 엔코더 2개(입력 속성 YUV, 출력 속성 MPEG-1인 것과 입력 속성 RGB, 출력 속성 MPEG-1인 것)를 가지고 있다고 가정한다.

(3) CFG 기반 변환 경로 생성 알고리즘의 3단계 : CFG를 이용한 변환 경로 생성 단계

근원지와 목적지가 서로 다른 재생 서비스 품질을 갖을 때 목적지가 요구하는 재생 서비스 품질을 만족하기 위해 근원지와 목적지 사이에 필요한 변환 자원들의 변환 경로를 자동으로 생성하여야 한다. 이러한 변환 경로를 아무런 전략없이 생성한다면, 존재하고 있는 트랜스코더들의 순열만큼이 되어 시간 복잡도는 N이 필요한 가용 자원의 개수일 때, $T(N) = O(N!)$ 이 된다. 따라서, 시스템의 부담을 줄이면서 변환 경로를 자동으로 생성하기 위해서 CFG의 형식을 이용한다.

CFG를 사용할 때, 경험자에 의해 트랜스코더간에 순서를 정한 것을 시작인 S로 한다면, 일종의 전문가에 의한 지침이 될 수 있다.

[정의 3-3] CFG 기반 변환 경로 생성에 사용하는 CFG 형식 문법

스트림 처리기에서 근원지와 목적지가 서로 다른 재생 서비스 품질의 멀티미디어 스트림을 재생할 때, 종단간에 위치하는 변환 자원들 간의 연결을 자동적으로 생성하기 위해서 사용하는 형식 문법은 G_{변환경로}이다.

$$G_{\text{변환경로}} = (S, V_N, V_T, P)$$

S는 CFG를 이용한 변환 경로를 생성할 때의 시작 기호이고, V_N는 논 터미날 변환 자원으로 영문자 대문자로 표기한다. 또한 V_T는 입력과 출력 속성을 표현한 가용 자원으로 영문자 소문자로 표기하며, P는 필요한 변환 자원을 선택할 때, 선행 변환 자원의 출력 속성과 동일한 입력 속성을 갖는 변환 자원을 선택하도록 하는 생성 규칙이다.

[예 3-3] 변환 경로 생성에 필요한 CFG인 G_{변환경로}의 예

근원지 PC에는 재생 서비스 품질이 CIF, 24비트 칼라, 30 프레임/초인 MPEG-1 파일이 있고, LAN을 통한 목적지 PC에서 원하는 스트림의 재생 서비스 품질은 모니터에서 QCIF, 8비트 흑백, 5프레임/초인 H.263으로 재생하는 것이다.

이때의 G_{변환경로}은 다음과 같다.

G변환경로 = (S, V_N, V_T, P)

$$V_T = \{ \begin{array}{l} \text{mpeg-1ft}_{h.263}, \quad h.263\text{ft}_{\text{mpeg-1}} \\ \text{mpeg-1firt}_{\text{mpeg-1}}, \quad h.263\text{firt}_{h.263} \\ \text{yuvfirt}_{\text{yuv}}, \quad \text{rgbfirt}_{\text{rgb}} \\ \text{mpeg-1st}_{\text{mpeg-1}}, \quad h.263\text{st}_{h.263} \\ \text{yuvst}_{\text{yuv}}, \quad \text{rgbst}_{\text{rgb}} \\ \text{mpeg-1ct}_{\text{mpeg-1}}, \quad h.263\text{ct}_{h.263} \\ \text{yuvct}_{\text{yuv}}, \quad \text{rgbct}_{\text{rgb}} \\ h.263\text{dc}_{\text{yuv}}, \quad \text{mpeg-1dc}_{\text{yuv}}, \\ \text{yuvECH}_{.263}, \quad \text{rgbECH}_{.263} \\ \text{yuvEC}_{\text{mpeg-1}}, \quad \text{rgbEC}_{\text{mpeg-1}} \end{array} \}$$

$$V_N = \{ \langle \text{TR}_{CT} \rangle, \langle \text{TR}_{DC} \rangle, \langle \text{TR}_{FT} \rangle, \langle \text{TR}_{FRT} \rangle, \langle \text{TR}_{ST} \rangle \}$$

$$P : S ::= \langle \text{TR}_{FRT} \rangle \langle \text{TR}_{ST} \rangle \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle$$

$$\langle \text{TR}_{FT} \rangle ::= \text{mpeg-1ft}_{h.263} \mid h.263\text{ft}_{\text{mpeg-1}} \mid \text{mpeg-1dc}_{\text{yuv}} \text{yuvECH}_{.263} \mid h.263\text{dc}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \mid \epsilon$$

$$\langle \text{TR}_{FRT} \rangle ::= \text{mpeg-1firt}_{\text{mpeg-1}} \mid h.263\text{firt}_{h.263} \mid \text{mpeg-1dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \mid h.263\text{dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvECH}_{.263} \mid \text{mpeg-1dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvECH}_{.263} \mid h.263\text{dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \mid \epsilon$$

$$\langle \text{TR}_{ST} \rangle ::= \text{mpeg-1st}_{\text{mpeg-1}} \mid h.263\text{st}_{h.263} \mid \text{mpeg-1dc}_{\text{yuv}} \text{yuvst}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \mid h.263\text{dc}_{\text{yuv}} \text{yuvst}_{\text{yuv}} \text{yuvECH}_{.263} \mid \text{mpeg-1dc}_{\text{yuv}} \text{yuvst}_{\text{yuv}} \text{yuvECH}_{.263} \mid h.263\text{dc}_{\text{yuv}} \text{yuvst}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \mid \epsilon$$

$$\langle \text{TR}_{CT} \rangle ::= \text{mpeg-1ct}_{\text{mpeg-1}} \mid h.263\text{ct}_{h.263} \mid \text{mpeg-1dc}_{\text{yuv}} \text{yuvct}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \mid h.263\text{dc}_{\text{yuv}} \text{yuvct}_{\text{yuv}} \text{yuvECH}_{.263} \mid \text{mpeg-1dc}_{\text{yuv}} \text{yuvct}_{\text{yuv}} \text{yuvECH}_{.263} \mid h.263\text{dc}_{\text{yuv}} \text{yuvct}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \mid \epsilon$$

$$\langle \text{TR}_{DC} \rangle ::= h.263\text{dc}_{\text{yuv}} \mid h.263\text{dc}_{\text{rgb}} \mid \text{mpeg-1dc}_{\text{yuv}} \mid \text{mpeg-1dc}_{\text{rgb}} \mid \epsilon$$

근원지와 목적지의 재생 서비스 품질이 서로 다른 경우의 멀티미디어를 재생하려고 할 때, 멀티미디어 스트림 처리기는 중단간에 명시되지 않은 변환 자원들을 연결하여 목적지가 원하는 대로 멀티미디어 스트림을 재생하려고 시도할 것이다. 이렇게 여러 변환 자원들로 변환 경로를 생성할 때 CFG 형식 문법을 사용한다.

CFG를 이용할 때, 변환 자원의 결합에 대한 경험자의 지침으로부터 시작해서 생성 규칙을 따라 변환 자원들의 연결을 유도를 해나간다면, 모든 논 터미날이 터미날로 바뀌는 경우가 있는데 이때의 터미날들의 연결이 필요한 가용 자원들로 연결된 변환 경로이다.

[예 3-4] CFG 형식 문법을 사용한 변환 경로 생성의 예위의 예에서, 경험자의 변환 자원 순서에 대한 지침인 $S ::= \langle \text{TR}_{FRT} \rangle \langle \text{TR}_{ST} \rangle \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle$ 로부터 시작한다.

가) 좌단 유도에 의해 다음의 3가지로 유도된다.

$$S ::= \text{mpeg-1firt}_{\text{mpeg-1}} \langle \text{TR}_{ST} \rangle \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle$$

$$S ::= \text{mpeg-1dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \langle \text{TR}_{ST} \rangle \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle$$

$$S ::= \text{mpeg-1dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvECH}_{.263} \langle \text{TR}_{ST} \rangle \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle$$

이때, $h.263\text{firt}_{h.263}$ 으로 유도되지 않은 것은 근원지의 스트림이 H.263이 아닌 MPEG-1이기 때문이다. 또한 여기서 여러 경로로 유도가 가능한데, 이것은 변환 경로가 여러 개 있을 수 있다는 의미이다.

나) 좌단 유도에 의해 다음으로 유도된다.

$$S ::= \text{mpeg-1firt}_{\text{mpeg-1}} \langle \text{TR}_{ST} \rangle \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle$$

$$S ::= \text{mpeg-1firt}_{\text{mpeg-1}} \text{mpeg-1st}_{\text{mpeg-1}} \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle \text{이거나,}$$

$$S ::= \text{mpeg-1firt}_{\text{mpeg-1}} \text{mpeg-1dc}_{\text{yuv}} \text{yuvst}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle \text{이거나,}$$

$$S ::= \text{mpeg-1firt}_{\text{mpeg-1}} \text{mpeg-1dc}_{\text{yuv}} \text{yuvst}_{\text{yuv}} \text{yuvECH}_{.263} \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle \text{로 유도된다.}$$

$$S ::= \text{mpeg-1dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \langle \text{TR}_{ST} \rangle \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle$$

$$S ::= \text{mpeg-1dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \text{mpeg-1st}_{\text{mpeg-1}} \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle \text{이거나,}$$

$$S ::= \text{mpeg-1dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \text{mpeg-1dc}_{\text{yuv}} \text{yuvst}_{\text{yuv}} \text{yuvECH}_{.263} \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle \text{이거나,}$$

$$S ::= \text{mpeg-1dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \text{mpeg-1dc}_{\text{yuv}} \text{yuvst}_{\text{yuv}} \text{yuvECH}_{.263} \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle \text{로 유도된다.}$$

$$S ::= \text{mpeg-1dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvECH}_{.263} \langle \text{TR}_{ST} \rangle \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle$$

$$S ::= \text{mpeg-1dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvECH}_{.263} \text{h.263st}_{h.263} \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle \text{이거나}$$

$$S ::= \text{mpeg-1dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvECH}_{.263} \text{h.263dc}_{\text{yuv}} \text{yuvst}_{\text{yuv}} \text{yuvECH}_{.263} \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle \text{이거나}$$

$$S ::= \text{mpeg-1dc}_{\text{yuv}} \text{yuvfirt}_{\text{yuv}} \text{yuvECH}_{.263} \text{h.263dc}_{\text{yuv}} \text{yuvst}_{\text{yuv}} \text{yuvECH}_{.263} \langle \text{TR}_{CT} \rangle \langle \text{TR}_{FT} \rangle \langle \text{TR}_{DC} \rangle \text{로 유도된다.}$$

이렇게 좌단 유도를 계속해나가면 다음과 같이 논 터미날이 모두 터미날로 바뀌는 경우가 있다.

$$S ::= \text{mpeg-1firt}_{\text{mpeg-1}} \text{mpeg-1dc}_{\text{yuv}} \text{yuvst}_{\text{yuv}} \text{yuvEC}_{\text{mpeg-1}} \text{mpeg-1ct}_{\text{mpeg-1}} \text{mpeg-1ft}_{h.263} \text{h.263dc}_{\text{yuv}}$$

이때의 터미날들의 연결이 필요한 가용 자원들로 연결된 변환 경로이다.

[예 3-4]의 경우, 좌단 유도를 통해 실제로 생성된 변환 경로는 25개이다.

변환 경로가 생성되는 중에 생성 규칙에는 존재하지만

실제로 시스템에 가용 변환 자원이 없거나, 변환 자원의 연결은 가능하지만 그 변환 경로가 목적지의 요구 사항을 만족하지 못하는 형태가 있다면, 중간에 변환 경로를 생성하는 것을 중단한다.

4. 재생 가능한 변환 생성 알고리즘

근원지와 목적지가 서로 다른 재생 서비스 품질의 멀티미디어 스트림을 재생하기 위한 변환 자원들이 CFG를 이용하여 자동으로 연결되었다. 그런데 이렇게 연결된 변환 경로는 여러 개의 변환 자원을 통과하는 동안 지연되어 목적지가 원하는 대로 재생이 가능하지 않을 수도 있으므로, 여러 개의 변환 자원을 통과한 후에도 목적지가 원하는 대로 재생이 가능한지 확인하여야 한다. 즉, 변환 경로가 재생 가능하지를 확인하여 재생 가능한 변환 경로를 생성하도록 한다.

4.1 변환 경로의 재생 가능성 검사

변환 경로를 지나는 스트림은 여러 개의 변환 자원을 통과하는 동안 지연되므로, 최종적으로 목적지가 원하는 대로 재생이 가능한지 확인하기 위해서 재생 가능성 검사를 받아야 한다. 주어진 변환 경로가 재생 가능하다는 것은 목적지가 요구하는 서비스 품질대로 스트림을 변환하는데 드는 소요 시간이 목적지에서 원하는 시점보다 앞서 이루어져야 한다는 의미이다. 이것을 판단하기 위해서 다음 과정을 거쳐 재생성을 검사한다. 즉, 여러 변환 자원을 통과하여 지연된 실제 스트림이 목적지에 도착하는 시간 간격과 목적지가 요구하는 스트림의 도착시간 간격을 비교한다. 이 비교에서 실제로 목적지에 도착하는 스트림의 시간 간격이 요구하는 시간 간격보다 크면, 여러 변환 자원을 통과하는 동안 너무 많이 지연되어서 요구대로 재생할 수 없다는 것을 의미한다.

재생 가능성의 검사 과정은 다음과 같다.

[과정 4-1] 재생 가능성의 검사 과정

입력 : • (알고리즘 3-1)의 출력인 목적지 서비스 품질을 만족하는 모든 변환 경로 • 변환 경로에 포함되어 있는 각 변환 자원의 처리량 • 목적지의 서비스 품질 중 프레임 율
단계 1 : 변환 경로에 대해서 실제 LDU간 도착시간 간격을 모든 변환 자원을 통과한 지연 시간의 총합으로 계산한다. 단계 2 : LDU간 도착시간 간격에 대한 목적지의 요구 사항을 요구 프레임 율로부터 파악한다. 단계 3 : 단계 1, 2의 값을 비교하여 요구 LDU간 도착시간 간격이 실제보다 크면 재생 가능하다고 판정하고, 그렇지 않으면, 재생이 가능하지 않다는 메시지를 출력한다.
출력 : 재생 가능한지 그렇지 않은 지를 판단하는 부울 값

(1) 재생 가능성의 검사 과정 1 단계 : 실제 LDU간 도착 시간 간격 파악 단계

멀티미디어 스트림이 근원지를 출발하여 여러 변환 자원을 통과한 후 목적지에서 재생될 때까지의 시간은 목적지

관점에서 볼 때 실제로 LDU간 도착시간 간격이라고 할 수 있다. 따라서 하나의 변환 경로상에서 모든 변환 자원을 통과하는데 걸리는 시간을 더하면, 실제 LDU간 도착시간 간격을 알 수 있다. 이 시간간격을 알기 위해서 다음과 같은 용어들을 정의한다.

[정의 4-1] 변환 자원 처리량, 지연 및 종단간 지연

변환 자원 처리량이란 각 변환 자원이 단위 시간당 처리할 수 있는 멀티미디어 스트림의 양으로 변환 자원마다 고유한 값을 갖는다. 변환 자원 처리량의 단위는 비트/초이다.

지연이란 하나의 변환 자원을 통과하는데 걸리는 시간이며 변환 자원의 처리량과 변환 자원에 도착한 워크로드에 관련 있다. 지연을 식으로 표현하면 다음과 같다.

$$\text{지연} = \text{워크로드} \times \text{변환자원명} / \text{처리량} \times \text{변환자원명}$$

종단간 지연이란 하나의 변환 경로상에서 모든 변환 자원을 통과하는데 걸리는 시간을 더한 것으로 이것은 실제 LDU간 도착 시간 간격이며, 식으로 표현하면 다음과 같다. 이때 n은 1, 2, ..., 변환 자원의 개수이다.

$$\text{종단간 지연} = \sum_{i=1}^n \text{지연}_i$$

[예 4-1] 하나의 변환 경로에 대한 실제 LDU간 도착 시간 간격 파악의 예

CFG에 의해 생성된 변환 경로 중 하나인 다음과 같은 변환 경로에 대해서

```
S ::= mpeg-1dCyuv yuvftrtyuv yuvECmpeg-1 mpeg-1dCyuv yuvStyuv
      yuvECmpeg-1 mpeg-1dCyuv yuvCtyuv yuvECmpeg-1 mpeg-1ftu.263
      h.263dCyuv
```

프레임 율 트랜스코더, 크기 트랜스코더, 색상 트랜스코더, 파일 형식 트랜스코더, 디코더, 엔코더의 모든 변환 자원을 통과하는데 걸리는 시간인 지연을 더한 종단간 지연을 구하여 실제 LDU간 도착시간 간격을 파악한다.

(2) 재생 가능성의 검사 과정 2 단계 : 요구 LDU간 도착 시간 간격 파악 단계

목적지가 요구하는 LDU간 도착시간 간격 값은 목적지가 요구하는 재생 서비스 품질 중에서 프레임 율의 역수로 파악할 수 있다.

[정의 4-2] 목적지가 요구하는 LDU간 도착시간 간격

목적지가 요구하는 LDU간 도착시간 간격 값은 목적지가 요구하는 재생 서비스 품질 중에서 프레임 율의 역수이다. 이것을 식으로 나타내면 다음과 같다.

$$\text{요구 LDU간 도착시간 간격} = 1/(\text{요구 프레임 율})$$

[예 4-2] 요구 LDU간 도착시간 간격 파악의 예

요구된 재생 서비스 품질의 프레임율이 5 프레임/초이 라면 이것은 1초에 5 프레임이 도착하기를 요구하는 것이며, 이것은 바꾸어 말하면 하나의 프레임이 도착 한 후에 또 하나의 프레임이 도착하려면 0.2초가 요구된다는 것이다. 따라서 요구된 LDU간 도착시간 간격은 프레임 율의 역수인 $1/5 = 0.2$ 초이다.

(3) 재생 가능성의 검사 과정 3 단계 : 실제와 요구 LDU간 도착시간 간격의 비교 단계

재생 가능성을 결정하기 위해 목적지의 LDU 도착 시간 간격과 변환 경로의 실제 LDU간 도착시간 간격을 비교한다.

이 두 값을 비교하여 실제 LDU간 도착시간 간격이 목적지의 요구 LDU 도착시간 간격보다 크면, 목적지가 원하는 대로 멀티미디어 스트림을 재생할 수 없으므로 재생이 불가능하다고 판단하고, 그렇지 않으면 재생 가능하다고 판단한다. 따라서 모든 변환 경로에 대하여 재생 가능한지 그렇지 않은지를 결정한다.

[정의 4-3] 재생 가능성 결정 식

실제 LDU간 도착시간 간격인 종단간 지연과 요구 LDU간 도착시간 간격을 비교하여 재생 가능성 검사를 한다. 이것을 식으로 표현하면 다음과 같다.

$$\text{실제 LDU간 도착시간 간격} \leq \text{요구 LDU간 도착시간 간격} \\ \text{종단간 지연} \leq 1/(\text{요구 프레임 율}) \quad (4-1)$$

식 (4-1)의 결과가 참이면, 요구 LDU간 도착시간 간격이 실제 LDU간 도착시간 간격보다 크거나 같다는 것이므로 멀티미디어 스트림이 여러 개의 변환 자원을 통과하였다 라도 목적지가 원하는 대로 재생 가능하다고 판단한다.

식 (4-1)의 결과가 거짓이면, 요구 LDU간 도착시간 간격이 실제 LDU간 도착시간 간격보다 작으므로 멀티미디어 스트림이 여러 개의 변환 자원을 통과하는 동안에 목적지가 원하는 대로 목적지에 도착할 수 없어서 재생 가능하지 않다고 판단한다.

4.2 재생 가능한 변환 경로 생성 알고리즘

근원지와 목적지가 서로 다른 재생 서비스 품질의 멀티미디어 스트림을 재생하기 위해서 CFG를 이용한 변환 경로를 자동으로 생성하였다. 또 자동으로 생성된 변환 경로 상에서 여러 개의 변환 자원을 통과하여 지연이 된 스트림이 목적지가 원하는 대로 재생 가능한지를 재생 가능성 검사로 확인하여 재생 가능하다고 판단되면, 이 경로는 재생 가능한 변환 경로라고 한다. 즉, 재생 가능한 변환 경로를 생성하는 것은 CFG를 이용하여 변환 경로를 자동 생성하고, 생성된 변환 경로에 대하여 재생 가능성 검사를 하여 재생 가능하다고 판단된 경로를 생성하는 것이다.

재생 가능한 변환 경로를 생성하는 알고리즘은 다음과 같다.

입력 : (알고리즘 3-1)과 [과정 4-1]의 입력
단계 1 : (알고리즘 3-1)에 있는 변환 생성 알고리즘을 수행한다. 단계 2 : 단계 1)의 결과로 생성된 변환 경로에 대해서 [과정 4-1]에 있는 재생 가능성 검사를 수행하여, 결과가 재생 가능하다고 판단되면, 이때의 변환 경로를 재생 가능한 변환 경로로 생성한다.
출력 : 재생 가능한 변환 경로

(알고리즘 4-1) 재생 가능한 변환 경로를 생성하는 알고리즘

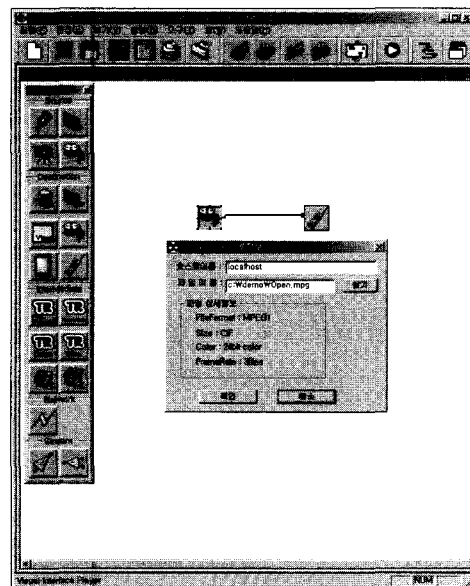
5. 구현 및 실험

5.1 구현 환경 및 도구

멀티미디어 스트림 처리기의 하나인 TransCore는 멀티미디어 데이터 처리 및 입출력, 멀티미디어 관련 동기화 기능을 정의하고 있는 서버에 대한 프레임 워크를 제공한다.

또한 객체지향 개념으로 구현된 TransCore의 API를 이용하면 내부의 상세하고 복잡한 처리 사항들을 모르더라도 쉽고 빠르게 응용 서비스를 개발 할 수 있다. 응용 서비스에 따른 데이터의 흐름은 스트림 계층, 프리젠테이션 계층, 하이퍼 프리젠테이션 계층의 세 가지로 구성된다.

시각적 멀티미디어 연출 편집기의 하나인 VIP(Visual Interface Player)는 TransCore API(Application Programming Interface)를 좀더 쉽고 직관적으로 사용할 수 있도록 하기 위해 비주얼 인터페이스를 제공하는 아이콘 프로그래밍 도구로서 사용자가 직접 코딩하지 않고도 원하는 서비스나 응용 프로그램을 쉽고 빠르게 구현해 볼 수 있는 도구이다. 사용자는 자신이 생각하는 서비스를 VIP가 제공하는 아이콘과 기능을 이용하여 TransCore가 제공하는 모델의 각 메커니즘으로 도식화한 다음 수행하면 인터프리터가 그 명시 프로그램을 해석하여 서버가 제공하는 API를 호출하게 된다.



(그림 5-1) VIP 초기 화면

5.2 실험 결과 및 분석

5.2.1 실험 방법

근원지와 목적지의 서비스 품질 정보는 연출 제어 프로그램을 통해서 알 수 있으며, 시스템이 갖고 있는 가용 자원은 <표 5-1>과 같이 11개와 15개로 실험하였다.

<표 5-1> 가용 자원의 종류 tr₁, tr₂의 기술

가용 자원의 종류 tr ₁	가용 자원의 종류 tr ₂
11개	15개
h.263ftmpeg-1	mpeg-1ft _{h.263} , h.263ftmpeg-1
mpeg-1ftmpeg-1, yuvftrt _{yuv}	mpeg-1ftmpeg-1, h.263ftrt _{h.263} , yuvftrt _{yuv}
mpeg-1stmpeg-1, h.263sth.263	mpeg-1stmpeg-1, h.263sth.263, yuvst _{yuv}
h.263ct _{h.263} , yuvct _{yuv}	mpeg-1ctmpeg-1, h.263ct _{h.263} , yuvct _{yuv}
h.263dc _{yuv} , mpeg-1dc _{yuv}	h.263dc _{yuv} , mpeg-1dc _{yuv}
yuvect _{h.263} , yuvecmpeg-1	yuvect _{h.263} , yuvecmpeg-1

또한 제안한 CFG 기반 변환 경로 생성 알고리즘에서는 경험자의 변환 자원에 대한 순서 결정 지침을 다음과 같이 CFG의 생성 규칙에 반영할 수 있다.

CFG 1은 프레임 율, 크기, 색상, 파일 형식, 디코더의 순서로 변환 자원을 통과하도록 하기 위해서 시작 생성 규칙을 S ::= <TR_{FRT}><TR_{ST}><TR_{CT}><TR_{FRT}><TR_{DC}>로 하였다.

또한 각 변환 자원의 종류에 대해서 가용 자원을 하나씩만 연결하는 지침이다.

즉, <TR_{FRT}>규칙에서 yuvftrt_{yuv}나, h.263ftrt_{h.263}과 같이 하나의 변환 자원은 연결할 수 있지만, mpeg-1dc_{yuv} yuvftrt_{yuv}와 같은 두개 이상의 변환 자원을 통해서 프레임 율을 변환할 수 없다.

CFG 2는 시작 생성 규칙은 CFG 1과 동일하지만, 각 변환 자원의 종류에 대해서 가용 자원을 여러 개 연결하는 것을 허용하는 지침이다.

즉, <TR_{FRT}>규칙에서 yuvftrt_{yuv}나, h.263ftrt_{h.263}과 같이 하나의 변환 자원을 연결할 수도 있고, mpeg-1dc_{yuv} yuvftrt_{yuv}와 같이 두개 이상의 변환 자원을 통해서 프레임 율을 변환할 수도 있는 지침이다.

CFG 3은 파일 형식, 프레임 율, 크기, 색상, 디코더의 순서로 변환 자원을 연결하기 위해서 시작 생성 규칙을 S ::= <TR_{FRT}><TR_{ST}><TR_{CT}><TR_{FRT}><TR_{DC}>에서 S ::= <TR_{FRT}><TR_{ST}><TR_{CT}><TR_{DC}>로 CFG 1, 2)와 다르게 적용하였다. 또, 각 변환 자원의 종류에 대해서 가용 자원을 여러 개 연결하는 것을 허용하는 지침이다.

<표 5-2>에 생성 규칙 CFG 1, CFG 2, CFG 3에 대해

<표 5-2> CFG의 종류 CFG 1, CFG 2, CFG 3의 기술

CFG 1	CFG 2	CFG 3
S ::= <TR _{FRT} ><TR _{ST} ><TR _{CT} ><TR _{FRT} ><TR _{DC} >	S ::= <TR _{FRT} ><TR _{ST} ><TR _{CT} ><TR _{FRT} ><TR _{DC} >	S ::= <TR _{FRT} ><TR _{FRT} ><TR _{ST} ><TR _{CT} ><TR _{DC} >
<TR _{FRT} > ::= yuvftrt _{yuv} mpeg-1ftmpeg-1 h.263ftrt _{h.263} ε	<TR _{FRT} > ::= yuvftrt _{yuv} mpeg-1ftmpeg-1 h.263ftrt _{h.263} mpeg-1dc _{yuv} yuvftrt _{yuv} mpeg-1dc _{yuv} yuvftrt _{yuv} yuvecmpeg-1 mpeg-1dc _{yuv} yuvftrt _{yuv} yuvect _{h.263}	<TR _{FRT} > ::= mpeg-1ft _{h.263} h.263ftmpeg-1 mpeg-1dc _{yuv} yuvect _{h.263} h.263dc _{yuv} yuvecmpeg-1 ε
<TR _{ST} > ::= mpeg-1stmpeg-1 h.263sth.263 yuvst _{yuv} ε	<TR _{ST} > ::= mpeg-1stmpeg-1 h.263sth.263 yuvst _{yuv}	<TR _{ST} > ::= yuvftrt _{yuv} mpeg-1ftmpeg-1 h.263ftrt _{h.263} mpeg-1dc _{yuv} yuvftrt _{yuv} mpeg-1dc _{yuv} yuvftrt _{yuv} yuvecmpeg-1 mpeg-1dc _{yuv} yuvftrt _{yuv} yuvect _{h.263} h.263dc _{yuv} yuvftrt _{yuv} h.263dc _{yuv} yuvftrt _{yuv} yuvecmpeg-1 h.263dc _{yuv} yuvftrt _{yuv} yuvect _{h.263} ε
<TR _{CT} > ::= yuvct _{yuv} mpeg-1ctmpeg-1 h.263ct _{h.263} ε	<TR _{CT} > ::= yuvct _{yuv} mpeg-1ctmpeg-1 h.263ct _{h.263} mpeg-1dc _{yuv} yuvct _{yuv} mpeg-1dc _{yuv} yuvct _{yuv} yuvecmpeg-1 h.263dc _{yuv} yuvct _{yuv} yuvecmpeg-1 h.263dc _{yuv} yuvct _{yuv} yuvect _{h.263} ε	<TR _{ST} > ::= mpeg-1stmpeg-1 h.263sth.263 yuvst _{yuv} mpeg-1dc _{yuv} yuvst _{yuv} mpeg-1dc _{yuv} yuvst _{yuv} yuvecmpeg-1 mpeg-1dc _{yuv} yuvst _{yuv} yuvect _{h.263} h.263dc _{yuv} yuvst _{yuv} h.263dc _{yuv} yuvst _{yuv} yuvecmpeg-1 h.263dc _{yuv} yuvst _{yuv} yuvect _{h.263} ε
<TR _{FRT} > ::= mpeg-1ft _{h.263} h.263ftmpeg-1 ε	<TR _{FRT} > ::= mpeg-1ft _{h.263} h.263ftmpeg-1 mpeg-1dc _{yuv} yuvst _{yuv} h.263dc _{yuv} yuvst _{yuv} yuvecmpeg-1 h.263dc _{yuv} yuvst _{yuv} yuvect _{h.263} ε	<TR _{CT} > ::= yuvct _{yuv} mpeg-1ctmpeg-1 h.263ct _{h.263} mpeg-1dc _{yuv} yuvct _{yuv} mpeg-1dc _{yuv} yuvct _{yuv} yuvecmpeg-1 h.263dc _{yuv} yuvct _{yuv} yuvecmpeg-1 h.263dc _{yuv} yuvct _{yuv} yuvect _{h.263} ε
<TR _{DC} > ::= mpeg-1dc _{yuv} h.263dc _{yuv} ε	<TR _{DC} > ::= mpeg-1dc _{yuv} h.263dc _{yuv} ε	<TR _{DC} > ::= mpeg-1dc _{yuv} h.263dc _{yuv} ε

기술하였다.

CFG 기반 변환 경로 생성 알고리즘을 사용하여 생성한 변환 경로에 대해서 재생 가능성 검사를 할 때, 요구된 LDU간 도착 시간 간격이 0.2초이므로, LDU기 각 변환 경로의 모든 변환 자원을 통과하여 목적지에 실제로 도착하는 시간은 0.2초와 같거나 작아야 재생 가능하다고 판단한다. 이때, 각 변환 자원의 지연은 트랜스코더를 설계한 알고리즘마다 다르므로, 여기서는 6개 이하의 변환 자원을 통과하는 변환 경로를 재생 가능하다고 판단하였다.

5.2.2 실험 결과

<표 5-3>과 같이 근원지 서비스 품질이 CIF, 30프레임/초, 24비트 칼라, MPEG-1 파일이며, 목적지 서비스 품질은 QCIF, 5프레임/초, 8비트 흑백, H.263 파일을 재생하는 경우에 6번의 실험을 하였고, 그 결과를 작성하였다.

<표 5-3> 실험 결과

CFG 종류	가용 자원수	tr ₁ 11개		tr ₂ 15개	
		생성된 변환 경로의 수	재생 가능한 변환 경로의 수	생성된 변환 경로의 수	재생 가능한 변환 경로의 수
CFG 1		(실험 1) 없다.	없다.	(실험 2) 1개	1개
CFG 2		(실험 3) 7개	2개	(실험 4) 42개	5개
CFG 3		(실험 5) 4개	없다	(실험 6) 42개	8개

<표 5-3>의 실험 2, 3, 4)에 의해서 재생 가능한 변환 경로에 대한 예는 다음과 같다.

실험 2의 재생 가능한 변환 경로)

mpeg-1ftmpeg-1 mpeg-1Stmpeg-1 mpeg-1Ctmpeg-1 mpeg-1ft h.263
h.263dc yuv

실험 3의 재생 가능한 변환 경로)

mpeg-1ftmpeg-1 mpeg-1Stmpeg-1 mpeg-1dc yuv yuvCt yuv yuvCC h.263
h.263dc yuv

실험 4의 재생 가능한 변환 경로)

mpeg-1ft h.263 h.263ft h.263 h.263St h.263 h.263Ct h.263 h.263dc yuv

5.2.3 실험 결과의 분석

목적지가 요구하는 재생 서비스 품질을 만족하기 위해서 근원지에서 시작하여 변환 자원들을 연결할 때, 시스템이 가지고 있는 모든 가용 자원들의 연결을 시도하여 변환 경로를 생성할 수 있을 것이다. 이 방법으로 가용 자원의 수가 15개인 경우에 변환 경로를 생성하면, 변환 경로의 수가 100개 이상, 변환 자원의 반복 연결이 발생하는 경우에는 무한개가 생성된다.

그런데 실시간으로 종단간 상이한 재생 서비스 품질을 갖는 멀티미디어 스트림을 재생하려면, 적절한 시간 내에 적절한 변환 경로 하나를 찾는 것이 중요하다.

따라서 경험자의 변환 자원 결합에 대한 지침을 시작으로 하는 CFG를 이용하여 변환 경로를 생성한다면, 가용 자원의 수가 11개의 경우에 생성 규칙 CFG 2, CFG 3에 따라 각각 7개와 4개, 가용 자원의 수가 15개의 경우에는 42개씩의 변환 경로를 생성하였다. 이것은 모든 가용 자원의 연결을 시도했을 때보다 훨씬 적은 수의 변환 경로가 생성된 것을 의미하는 것으로, 경험자가 제시하는 변환 자원 결합 규칙에 의한 변환 경로 생성이 필요함의 의미이다.

그러나, 11개의 가용 자원에 생성 규칙 CFG 1을 적용하였을 때에는 변환 경로를 생성하지 못하였고, 다른 가용 자원의과 생성 규칙에서는 변환 경로가 생성되었다.

이것은 경험자가 제시하는 지침인 생성 규칙에 따라 특정 조합의 가용 자원에 대해서는 변환 자원을 생성하지 못할 수도 있다는 것을 의미한다. 따라서 현재 존재하는 가용 자원에 따른 경험자의 적절한 지침이 필요하다.

또한 최종적으로 목적지에서 스트림을 재생하는 것이 목적이므로 찾은 변환 경로가 목적지의 요구대로 스트림이 도착하여 재생하는지 검사하는 것도 중요하다.

6. 결 론

근원지와 목적지가 서로 다른 재생 서비스 품질은 사용자의 다양한 요구나 목적지의 하드웨어나 네트워크의 제약 때문에 생길 수 있다. 이러한 환경에서 멀티미디어 스트림은 최종적으로 목적지가 원하는 재생 서비스 품질을 만족시키기 위해서 일련의 변환 자원들을 통과하여야 한다. 그런데 환경에 따라 변환 자원의 종류가 다르므로 근원지와 목적지가 서로 다른 재생 서비스 품질의 멀티미디어 스트림을 재생하기 위해서는 상황에 따라 모두 다른 변환 자원들의 연결인 변환 경로를 자동적으로 생성하여야 한다.

이를 위해서 본 논문에서는 변환 자원들에 대한 연결인 변환 경로를 응용의 부담을 줄이면서 자동 생성하기 위해서 경험자의 변환 자원에 대한 순서 결정 지침을 시작으로 하는 CFG를 이용하였다. 또한 그 결과로 나온 변환 경로가 여러 변환 자원을 통과하는 동안 생긴 지연에도 불구하고 목적지가 요구하는 데로 재생이 가능한지를 확인하는 방법을 제안하고 구현하였다.

참 고 문 헌

[1] 강의선, "압축상태에서 MPEG2 P프레임을 H.263 P프레임으로 변환하기 위한 Guided Search 방법 연구", 숭실대학교 대학원 컴퓨터학과 석사학위논문, pp.1-15, 2001.
 [2] 박대원, "휴대폰에 추가 소프트웨어 없이 영상 스트림 전송 방법 연구", 숭실대학교 대학원 컴퓨터 학과 석사학위논문, pp.17-19, 2001.
 [3] 오미경, "이동형 운영체제를 위한 적응적 스트림 처리기 설계"

- 및 구현”, 숭실대학교 대학원 컴퓨터학과 석사학위논문, pp. 3-7, 2001.
- [4] 이조원, “압축된 영역에서의 영상 크기 축소 트랜스코더의 성능 향상 방법의 연구”, 숭실대학교 대학원 컴퓨터학과 석사학위논문, pp.5-13, 2001.
- [5] Akbar, Mohammad M., Manning, Eric G., Shoja, Gholamali C., “Admission control and quality adaptation in the distributed multimedia server system (DMSS),” Proceedings of SPIE Internet Multimedia Management Systems II, pp.246-257, 2001.
- [6] Anderson, D. P., “Metascheduling for Continuous Media,” ACM Transactions on Computer Systems, Vol.11, No.3, pp.226-252, 1998.
- [7] Chen MJ. Chu MC. Pan CW. k., “Efficient Motion-Estimation Algorithm for Reduced Frame-Rate Video Transcoder,” IEEE Transactions on Circuits & Systems for Video Technology, Vol.12, pp.269-275, 2002.
- [8] Dan, A., Sitaram, D., “Session Scheduling and Resource Sharing in Multimedia Systems,” Multimedia Information Storage and Management, Kluwer Academic Publishers, pp.257-269, 1996.
- [9] Guobin Shen, Bing Zeng, Ya-Qin Zhang, arid Ming L. Liou, “Transcoder with Arbitrarily Resizing Capability,” Proceedings of the IEEE International Symposium on Circuits and Systems, Vol.5, pp.25-28, 2001.
- [10] Hafid A, Vonbochmann G, Dssouli R, “A Quality of Service Negotiation Approach with Future Reservations(NAFUR)-A Detailed Study,” Computer Networks & Isdn Systems, Vol.30, No.8, 1998.
- [11] Huang V. Zhuang WH., “QoS-Oriented Access Control for 4G Mobile Multimedia CDMA Communications,” IEEE Communications Magazine, Vol.40, No.3, pp.118-125, 2002.
- [12] Jin Soo Jang, Woo Hyun Ahn, Daeyeon Park, “A Server Framework Scheduling Multimedia plications In. peg Syste Environment,” Proceedings of the 8th International Conference on Parallel and Distributed Systems, pp.562-569, 2001.
- [13] Kim, Y. H., “Dynamic QoS Control Technique in Multimedia Multicast Communication,” Master’s Dissertation, pp. 13-21, 1996.
- [14] Kwang-deok Seo, Seong-cheol Heo, Jae-kyoon Kim, “A Rate Control Algorithm Based on Adaptive R-Q Model for MPEG-1 to MPEG-4 Transcoding in DCT Domain,” Proceedings of the 2002 IEEE International Conference on Communications, Vol.1, pp.109-113, 2002.
- [15] Lin, C., Chang, S., “Modeling and Transformation of Multimedia Data,” Multimedia Information Storage and Management. Kluwer Academic Publishers, pp.303-307, 1996.
- [16] Lombaedo, A., Schembra, G., Morabito, G., “Traffic Specification for the Transm ission of Stored MPEG video on the Internet,” Multimedia, IEEE Transaction, Vol.3, Issue 1, pp.5-10, March, 2001.
- [17] Nahrstedt, K., Smith, J. M., “The QoS Broker,” IEEE, Spring, 1995.
- [18] Niklas Bjork and Charilaos Christopoulos, “Transcoder Architectures for Video Coding,” IEEE Transactions on Consumer Electronics, Vol.44, No.1, February, 1998.
- [19] Ozgur Gurbuz, Henry Owen, “Dynamic Resource Scheduling for Variable QoS Traffic in W-CDMA,” Proceedings of the 1999 IEEE International Conference on Communications, Vol.2, pp.703-707, 1999.
- [20] Raghavan, S. V., Tripathi, S. K., “Networked Multimedia Systems,” Prentice Hall. New Jersey, pp.34-37, pp.72-73, pp.193-194.



전 성 미

e-mail : smchon@hananet.net

1980년 숭실대학교 정보과학대학 전자계산학과 졸업(학사)

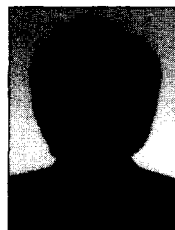
2000년 이화여자대학교 교육대학원 교육공학과 졸업(석사)

2000년~현재 숭실대학교 대학원 컴퓨터학과 박사 재학중

1995년~1999년 LG CNS 기술대학원 전문과장

2000년~현재 가천 길대학 전산 정보학과 겸임교수

관심분야 : 멀티미디어 스트리밍, 멀티미디어 통신, 멀티미디어 콘텐츠



임 영 환

e-mail : yhlm@computing.ssu.ac.kr

1977년 경북대학교 수학과 졸업(학사)

1979년 한국과학원 전산학과 졸업(석사)

1985년 Northwestern University 전산학과(박사)

1979년~1996년 한국전자통신연구소 책임연구원

1996년~현재 숭실대학교 미디어학부 부교수

관심분야 : 멀티미디어