Journal of Korea Multimedia Society Vol. 14, No. 7, July 2011(pp. 849-857)

주문형 게임 서비스를 위한 장면 기술자 기반 고속 게임 부호화기

전찬웅[†], 조현호[††], 심동규[†††]

요 약

주문형 게임 서비스는 서버에서 실행하는 게임을 동영상 부호화하여 클라이언트에 전송하고, 클라이언트에서 비디오 복호화를 통해 게임을 즐길 수 있게 해 준다. 다수의 사용자가 네트워크상에서 실시간 게임 서비스를 즐기기 위해서는 초고속 게임 인코더가 필요하다. 본 논문에서 제안한 방법은 장면 기술자를 정의하고, 이를 게임 영상을 부호화하는 부호화기에 부가적인 정보로 입력함으로써 움직임 예측, 율·왜곡 최적화와 같은 복잡도가 높은 부호화 과정을 생략하여 부호화기를 고속화한다. 장면 기술자를 움직임 벡터로 사용하고, 장면 기술자를 이용하여 매크로블록 모드를 결정해 부호화기를 고속화한다. 제안하는 방법의 성능 평가를 위해 H.264/AVC의 오픈 소프트웨어인 x264와 비교한 결과, x264에 어셈블리 코드가 포함되지 않은 경우에 대해서 약 192%의 부호화 속도 향상을 확인하였고, x264에서 일부 모듈에 대해서 어셈블리 최적화를 반영한 결과에 대해서는 86%의 부호화 속도가 향상되는 것을 확인할 수 있었다. 부호화기의 고속화 결과 60 FPS의 부호화 속도를 넘어 주문형 게임을 실시간으로 수행할 수 있게 되었다.

Fast Game Encoder Based on Scene Descriptor for Gaming-on-Demand Service

Chan Woong Jeon[†], Hyun Ho Jo[††], Dong Gyu Sim[†††]

ABSTRACT

Gaming on demand(GOD) makes people enjoy games by encoding and transmitting game screen at a server side, and decoding the video at a client side. In this paper, we propose a fast game video encoder for multiple users over network with low-powered devices. In the proposed system, the computational complexity of game encoders is reduced by using scene descriptors, which consists of an object motion vector, global motion, and scene change. With additional information from game engines, the proposed encoder does not need to perform various complexity processes such as motion estimation and rate-distortion optimization. The motion estimation and rate-distortion optimization skipped by scene descriptors. We found that the proposed method improved 192 % in terms of FPS, compared with x264 software. With partial assembly code, we also improved coding speed by 86 % in terms of FPS. We found that the proposed fast encoder could encode over 60 FPS for real-time GOD applications.

Key words: Gaming-On-Demand(주문형 게임), Fast Encoder Algorithm(부호화기 고속화 방법), Streaming (스트리밍), Motion Estimation(움직임 예측)

※ 교신저자(Corresponding Author) : 심동규, 주소 : 서울시 노원구 광운로 20 (139-701), 전화 : 02)941-6470, FAX : 02)914-6471, E-mail : dgsim@kw.ac.kr
접수일 : 2011년 2월 15일, 수정일 : 2011년 5월 5일
완료일 : 2011년 6월 10일
[†] 준회원, 광운대학교 임베디드소프트웨어공학과
(E-mail : jcw@kw.ac.kr)
[††] 준회원, 광운대학교 컴퓨터공학과
(E-mail : idjhh@kw.ac.kr)
[†††] 종신회원, 광운대학교 컴퓨터공학과 부교수
(E-mail : dgsim@kw.ac.kr)
※ 본 논문은 중소기업청에서 지원하는 2010년도 산학연공동기술개발사업(No.00042548)의 연구수행으로 인한 결과물임을 밝힙니다.

1. 서 론

전자 통신 기술이 비약적으로 발전함에 따라, PMP, DMB 플레이어, 모바일 태블릿, 넷북 등의 모바일 멀티미디어 기기의 사용이 보편화 되었다. 따라서 사용자들은 인터넷, 비디오 스트리밍, 모바일 게임 등의 멀티미디어 서비스를 언제, 어디서나 즐길 수 있게 되었고, 제공되는 서비스의 종류도 증가하고 있는 추세이다. 특히, 모바일 단말의 성능이 향상되면서 모바일 단말에서 동작하는 게임에 대한 사용자의 요구가 간단하게 즐길 수 있는 아케이드 게임이나 슈팅 게임에서 복잡한 기능이 요구되는 전략 시뮬레이션 게임이나 롤플레잉 게임으로 확대되고 있다. 그리고 스마트 폰, 모바일 태블릿 등 고성능의 프로세서를 가진 모바일 단말을 통해 기존에 PC에서나 즐길 수 있었던 고사양의 게임들을 시간과 공간에 대한 제약 없이 즐기고자 하는 요구가 증가하고 있다. 이러한 사용자의 요구에 따라 등장한 기술이 주문형 게임 서비스이다.

주문형 게임 서비스는 저사양의 장치에서도 고사양의 게임을 즐길 수 있도록, 고성능의 서버에서 동작하고 있는 게임의 영상 및 오디오를 부호화하여 저사양의 클라이언트에 비트스트림을 전송한다. 모바일 단말기와 같은 클라이언트에서는 수신한 비트스트림을 복호화한 후 영상 및 오디오를 재생시켜 줌으로써 사용자가 모바일 단말에서 직접 게임을 플레이하는 것과 같은 서비스를 제공한다. 주문형 게임 시스템에서 서버의 부호화 속도와 단말의 복호화 속도는 매우 중요하며 주문형 게임 서비스의 핵심적인 기술 요소가 된다. 일반적인 게임의 화면 출력 속도는 30~60 frame per second (FPS)로 주문형 게임 시스템에서 부호화기와 복호화기는 이 속도를 만족해야 한다. 복호화기는 부호화기에서 수행하는 움직임 예측, 모드 결정 과정을 거치지 않기 때문에 부호화기에 의해 생성된 비트스트림을 복호 하는데 복잡도가 낮아 저사양의 장치에서도 실시간 복호가 가능하다. 그러나 부호화기는 동영상의 부호화 시 다양한 후보 중에서 가장 유사한 블록을 찾는 움직임 예측/보상, 다양한 부호화 모드들 중 객관적인 화질을 만족시키며 비트스트림으로 표현했을 때 가장 적은 비트량으로 표현할 수 있는 모드를 선택하는 율・왜곡 최적화를 이용한 모드 결정 등의 복잡한 과정을 거치기 때문에 복호화기에 비해 복잡도가 매우 높아 실시간으로 부호화하기 어렵다. 따라서 주문형 게임 서비스를 실행하기 위한 부호화기의 고속화가 필요하다.

본 논문은 게임 실행 시, 게임 엔진에서 생성되는 물체들의 위치나 움직임 정보, 화면의 변화를 장면 기술자라 정의하고, 주문형 게임 시스템 서버에서의 게임 화면 부호화 시 장면 기술자를 이용한 부호화기 고속화 방법을 제안한다. 논문의 구성은 다음과 같다. 2장에서는 제안하는 부호화기 고속화 방법에 대하여 설명한다. 3장에서는 실험 결과를 통하여 제안하는 방법의 성능을 평가하고 결과를 분석한다. 4장에서 결론을 맺는다.

2. 제안하는 장면 기술자를 이용한 부호화기 고속화 방법

본 장에서는 주문형 게임 시스템을 서비스하기 위해 본 논문에서 제안하는 방법에 대하여 설명한다. 본 논문에서는 게임 플레이 시 생성되는 게임 정보 중 비디오 부호화의 고속화에 이용될 수 있는 정보를 장면 기술자라 정의하고, 장면 기술자를 이용하여 H.264/AVC[1] 부호화기를 고속화하는 방법에 대하여 제안하였다. H.264/AVC 부호화기를 고속화하는 방법으로는 부호화기에서 높은 복잡도를 차지하는 움직임 예측 및 모드 결정 단계를 고속화하는 방법이 많은데, 일반적으로 임계값을 이용한 부호화기 고속화 연구[2-4], 경계를 기반으로 한 화면내 예측 고속화 연구[5]등의 연구가 진행되고 있다. 하지만 임계값은 영상 혹은 부호화 환경에 의존도가 높은 값으로 임계값을 잘못 정하였을 경우 부호화 효율 저하를 가져올 수 있다. 또한, 경계 기반의 방법은 영상처리를 통해 구한 경계가 실제 물체의 경계가 아닐 수도 있고, 경계를 구하는 계산 복잡도가 높아 문제가 된다. 이러한 문제점을 극복하기 위해 본 논문에서는 게임 화면의 특성을 이용하였다. 게임 엔진은 게이머의 조작에 의하여 게임을 실행하고, 실행한 결과를 출력한다. 즉, 게이머가 움직이고자 하는 물체를 선택하고, 선택된 물체가 어디로 움직일 것인지, 어떻게 행동할 것인지 등을 결정하면, 게임 엔진은 어떤 배경을 출력할 것인지, 어떤 물체들을 출력할 것인지 등을 판단한다. 따라서 게임 화면의 부호화 시에 게임 엔진에서 생성되는 부가적인 정보를 이용하게 돼

면 게임 화면의 부호화를 더욱 빠르고 정확하게 수행할 수 있다. 이와 유사한 연구로 3D 공간상에서 배경 맵을 구현하는 방법인 Skybox와 Skydome의 정보를 이용하여 매크로블록 분할, 움직임 예측을 고속화하여 게임 영상 부호화를 고속화한 연구[6,7]가 있다. 하지만 3D 게임의 렌더링시에 생성되는 정보인 Skybox, Skydome을 이용하기 때문에, 2D 게임을 포함하는 모든 게임에 적용하기 어려우며, 영상의 부호화 속도가 QVGA 영상에서 약 16%, VGA 영상에서 약 14%가 향상되어 속도 향상이 크지 않았다. 따라서 본 논문에서는 모든 게임에서 적용될 수 있는 게임 엔진에서 생성되는 부가정보를 이용하여 장면 기술자를 생성하고, 이를 통해 게임 화면의 부호화를 고속화하는 방법들을 제안하였다.

2.1 장면 기술자를 이용한 주문형 게임 시스템

그림 1은 제안하는 장면 기술자를 이용한 주문형 게임 시스템의 블록도이다. 제안하는 방법의 게임 엔진은 부호화기를 고속화하기 위해 장면 기술자를 동영상 부호화기에 입력한다. 그림 2는 그림 1의 동영상 부호화기를 자세하게 그린 블록도이다. H.264/AVC 부호화기는 이전에 복원된 영상을 이용하여 현재 부호화할 블록을 예측하고, 예측한 영상과 원본 영상의 차분을 정수 변환, 양자화, 엔트로피 부호화하여 영상을 부호화한다. 장면 기술자를 이용하는 H.264/AVC 부호화기는 장면 기술자를 입력받는데, 장면 기술자는 게임 내의 물체의 움직임 또는 배경의 변화 등의 정보를 가지며, H.264/AVC 부호화기의 움직임 예측과 모드 결정 과정을 생략할 수 있도록 한다.

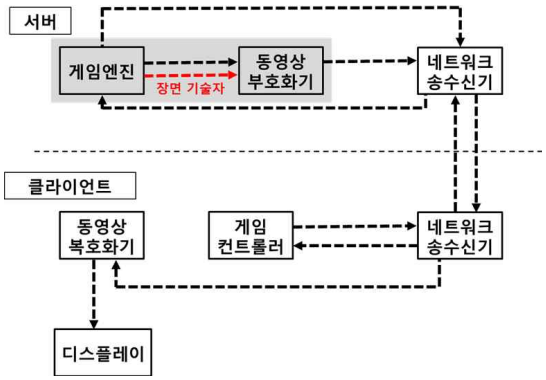

그림 1. 장면 기술자를 이용한 주문형 게임 시스템 블록도

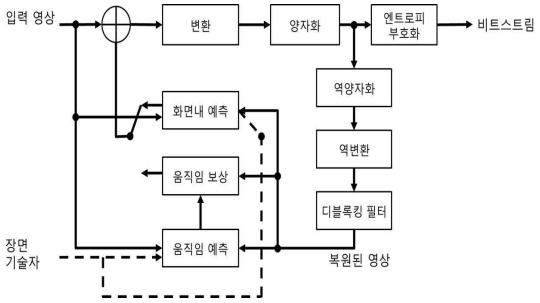

그림 2. 장면 기술자를 이용한 H.264/AVC 부호화기 블록도

2.2 장면 기술자 정의

본 논문에서는 부호화기를 고속화하기 위하여 표 1과 같이 매크로블록 단위로 장면 기술자를 정의하였다. 장면 기술자는 크게 영상에 변화가 있는 영역과 없는 영역으로 구분을 지었으며, 현재 부호화할 매크로블록이 이전 프레임의 동일한 위치의 매크로블록과 비교하여 변화가 없는 영역을 UR(Unchanged Region) 기술자, 변화가 있는 영역을 CR(Changed Region) 기술자로 정의한다.

변화가 있는 영역은 움직이는 물체인지, 배경인지를 판단하여 CR 기술자의 하위 기술자를 정의하는데, 물체는 OBJ 기술자, 배경은 BG 기술자로 정의한다. 그리고 움직이는 영역의 물체와 배경은 모두 움직임을 가지기 때문에 OBJ, BG 기술자의 하위 기술자로 움직임 정보를 MI 기술자라 정의하였다.

표 1. 장면 기술자의 정의

기 술 자			설 명
UR			현재 부호화할 매크로블록이 이전 프레임 동일한 위치의 매크로블록과 비교해 변화가 없음
CR			현재 부호화할 매크로블록이 이전 프레임 동일한 위치의 매크로블록과 비교해 변화가 있음
	OBJ		변화 있는 영역의 물체
	BG		변화 있는 영역의 배경
		MI	물체, 배경의 움직임 정보

2.3 모드 결정 고속화

H.264/AVC 코덱은 화면내 예측, 화면간 예측 모두를 실행해보고 율·왜곡 최적화를 통해 가장 부호화 효율이 높은 모드를 선택한다. 그림 3의 좌측 그림

은 이전에 복원된 프레임이고, 우측 그림은 현재 부호화할 프레임이다. 게이머는 두 개의 물체에 이동하라는 명령을 게임 엔진에 전하였고, 명령에 따라 물체를 이동하는 화면을 출력한다. 그림 3에서 이전 프레임과 비교해 현재 부호화할 프레임에서 변화가 있는 부분은 우측 그림의 색이 칠해진 매크로블록이다. 변화가 있는 영역인 색이 칠해진 매크로블록들은 CR 기술자를, 변화가 없는 영역인 색이 칠해지지 않은 매크로블록들은 UR 기술자를 가진다.

H.264/AVC 부호화기의 모드 결정에서 현재 부호화할 매크로블록이 이전에 복원된 프레임의 연관된 위치의 매크로블록과 유사하다면 스킵 모드로 선택될 확률이 높다. 때문에 게임 엔진에서 생성된 장면 기술자 UR을 이용해 스킵 모드의 선택을 대신 할 수 있다. 따라서 UR인 매크로블록의 경우 모드 결정 과정을 생략하고 스킵 모드로 선택하여 부호화한다. 그리고 CR 기술자를 가진 매크로블록은 배경인지, 물체인지를 판단하여 하위 기술자 BG와 OBJ 중 한 가지를 가지게 되고, 각각의 CR-BG, CR-OBJ인 매크로블록들은 움직임 정보를 나타내는 하위 기술자 MI를 가진다. 장면 기술자 MI는 물체 또는 배경의 움직임 정보를 나타내는 기술자로 매크로블록의 움직임 벡터와 동일하다. 부호화기에서 움직임 예측을 수행하여 움직임 벡터가 탐색 영역을 벗어나면, 움직임 벡터의 크기가 매우 커져 부호화 효율이 떨어져 화면내 예측 모드로 부호화될 확률이 높으므로, 장면 기술자 MI와 탐색 영역을 이용하여 현재 부호화할 매크로블록을 화면내 예측 모드로 부호화할 것인지, 화면간 예측 모드로 부호화할 것인지를 결정할 수 있다. 그림 4와 같이 장면 기술자 MI가 부호화기의 탐색 영역보다 크다면 매크로블록이 화면내 예측 모드로 부호화될 확률이 높으므로 매크로블록을 화면내 예측 모드로 결정한다. 반대로 장면 기술자 MI가 부호화기의 탐색 영역보다 작다면 화면간 예측 모드로 결정되고, 움직임 보상 후 잔차 신호는 정수 변환, 양자화, 엔트로피 부호화 과정을 거쳐 부호화된다.

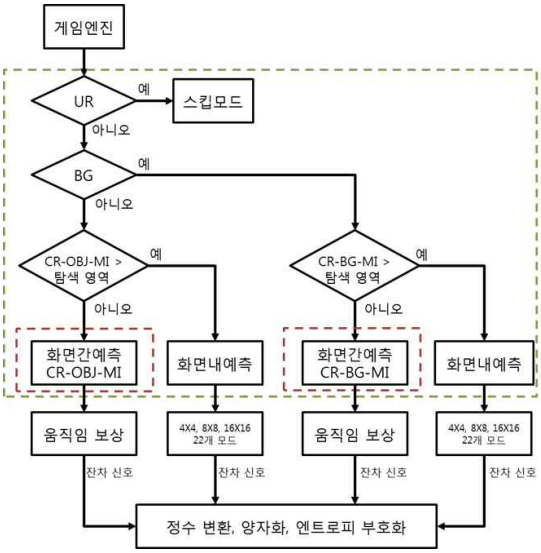

그림 4. 장면 기술자를 이용한 부호화 흐름도

2.4 움직임 예측 고속화

비디오 부호화의 움직임 예측은 이전에 복원된 프레임에서 현재 부호화할 블록과 가장 유사한 블록을 찾는 과정이다. 그림 5의 좌측 화면은 복원된 이전 프레임, 우측 화면은 부호화하려는 현재 프레임이다.

게이머는 좌측 그림의 사각형에 그려진 두 개의 물체를 우측의 그림과 같이 이동시키려 한다. 게임 엔진은 게이머가 물체들의 이동할 지점을 지정하면, 우측 그림의 화살표와 같이 물체의 이동정보를 알 수 있다. 그림에서 물체의 이동에 1초가 걸리고 게임 엔진이 60 FPS로 게임 화면을 출력한다면, 물체는 프레임과 프레임 사이 이동거리의 1/60만큼씩 이동한다. 1/60초 동안 물체가 움직인 좌표를 계산하고, 현재 프레임에 존재하는 물체의 좌표에서 이전 프레임에 존재하는 물체의 좌표를 뺀다면 물체의 움직임 벡터를 알 수 있고, 이 움직임 벡터는 MI 기술자로

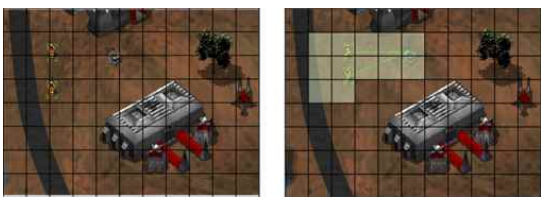

그림 3. 영상에서 변화한 영역과 변하지 않은 영역의 예

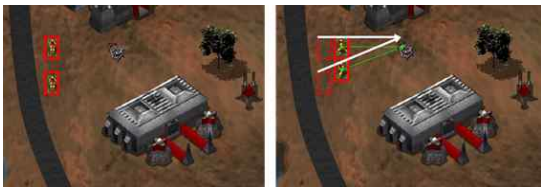

그림 5. 게이머에 의해 움직이는 물체의 예

정의된다.

그림 6은 장면 기술자 MI를 이용한 움직임 예측/보상의 블록도이다. 장면 기술자 MI를 통하여 움직임 예측 없이 움직임 벡터를 얻고, 움직임 보상을 수행하여 잔차 신호를 생성한다. 게임 엔진에서 생성한 물체의 움직임 벡터는 움직임 예측하여 생성되는 움직임 벡터 대신 사용된다. 움직임 예측은 움직임 벡터를 찾기 위해 수행하는 과정이므로, 게임 엔진에서 생성한 MI 기술자를 부호화하여 움직임 예측 과정을 생략한다. 장면 기술자 MI를 이용한 움직임 예측 과정의 생략으로 인하여, 부호화기의 복잡도를 크게 낮출 수 있으며, 부호화 시간 또한 많이 감소하여 부호화 속도를 높일 수 있다.

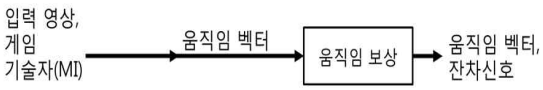

그림 6. 장면 기술자를 이용한 움직임 예측/보상 블록도

3. 실험 결과

제안하는 부호화기 고속화 방법을 시뮬레이션하기 위하여 주문형 게임 시스템을 다음과 같이 구성하였다. 게임 엔진은 그림 7과 같은 640×640 크기의 영상을 출력하는 오목 게임을 이용하였고, 서버에서는 H.264/ AVC 부호화기 중 부호화 속도가 가장 빠른 x264 소프트웨어[8]를, 클라이언트의 H.264/AVC 복호화기는 일반적으로 가장 많이 사용하는 FFMPEG 복호화기[9]를 사용하였다. 또한, UDP 프로토콜을 이용해 서버와 클라이언트 사이의 비트스트림 송수신을 수행하였다[10]. 제안하는 주문형 게임 시스템의 부호화기 시뮬레이션 및 성능 평가를 위하여 표 2의 실험 환경을 구성하고, 표 2의 실험 조건으로 실험하였다.

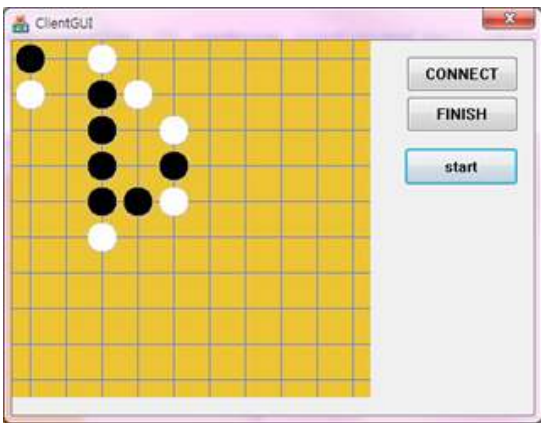

그림 7. 주문형 게임 시스템을 위한 오목 게임 실행 예

본 논문에서 제안한 주문형 게임 시스템의 부호화기 성능을 표 3~8에 나타내었다. 표 2의 실험 조건에 대해 600프레임 5회씩 실험하였고, 각 QP에 대한 5회 실험 평균 비트율(Kbits/s), PSNR(dB), FPS (frame/s)를 측정하였다. 제안한 방법의 성능은 BDbitrate(%), BDPSNR(dB)과 FPS의 향상 정도(%)로 나타내었다[11]. 본 논문을 위한 주문형 게임 시스템의 부호화기로 사용한 x264 소프트웨어는 부호화기를 고속화하기 위한 방법으로 어셈블리 코드를 사용하고 있다. x264 소프트웨어에서 사용하는 MMX 어셈블리 코드는 특정한 플랫폼에서만 지원되어 모든 플랫폼에서 사용하기에는 무리가 있다. 따라서 제안하는 장면 기술자를 이용한 주문형 게임 시스템 부호화기의 고속화 성능을 측정하기 위하여 어셈블리 코드를 포함한 경우와 어셈블리 코드를 제외한 두 가지 경우의 실험을 모두 진행하였다.

주문형 게임 시스템의 어셈블리 코드를 포함한 x264 소프트웨어가 640×640 오목 영상 600프레임을 부호화한 결과는 표 3~5와 같다. 제안하는 방법 1은 2.4절의 움직임 예측을 고속화하는 방법이며, 표 3에 성능을 나타내었다. 제안하는 방법 1은 어셈블리 코드를 포함한 x264 소프트웨어와 비교하여 부호화 속도가 약 53% 향상되고, BDbitrate와 BDPSNR은 약 1.233%, 0.227dB의 성능 향상이 있었다. 제안하는 방법 2는 2.3절의 모드 결정 고속화 방법이며, 표 4에 성능을 나타내었다. 제안하는 방법 2는 부호화 속도가 약 86% 향상되었다. BDbitrate와 BDPSNR은 약 2.109%, 0.057dB의 성능 저하가 발생하였다. 제안하는 방법 1, 2를 모두 적용한 결과를 표 5에 나타내었

표 2. 실험 환경 및 실험 조건

실험 환경		실험 조건	
CPU	Intel Core2 Quad 2.33GHz	GOP 구조	IPPP
RAM	4GB	I 프레임 주기	30
부호화기	x264 부호화기	QP	22, 27, 32, 37

표 3. 어셈블리 코드를 포함한 x264 소프트웨어에 제안한 방법 1을 적용한 실험 결과

QP	x264			방법 1			성능		
	비트율	PSNR	FPS	비트율	PSNR	FPS	BDbitrate	BDPSNR	FPS 향상
22	259.39	53.885	118	256.09	53.901	177	-1.233	0.227	49.6%
27	205.90	47.258	116	203.52	47.287	173			49.6%
32	159.30	42.570	116	157.79	42.594	178			53.5%
37	117.68	39.172	111	116.76	39.199	177			59.0%

표 4. 어셈블리 코드를 포함한 x264 소프트웨어에 제안한 방법 2를 적용한 실험 결과

QP	x264			방법 2			성능		
	비트율	PSNR	FPS	비트율	PSNR	FPS	BDbitrate	BDPSNR	FPS 향상
22	259.39	53.885	118	253.12	50.113	215	2.109	-0.507	81.4%
27	205.90	47.258	116	204.23	46.65	214			84.0%
32	159.30	42.570	116	158.53	42.416	215			84.8%
37	117.68	39.172	111	117.68	39.172	214			92.2%

표 5. 어셈블리 코드를 포함한 x264 소프트웨어에 제안한 방법 1, 2를 적용한 실험 결과

QP	x264			방법 1, 2			성능		
	비트율	PSNR	FPS	비트율	PSNR	FPS	BDbitrate	BDPSNR	FPS 향상
22	259.39	53.885	118	250.52	50.105	215	1.025	-0.323	81.5%
27	205.90	47.258	116	202.17	46.682	213			83.9%
32	159.30	42.570	116	157.28	42.442	215			85.3%
37	117.68	39.172	111	116.87	39.199	216			93.5%

다. 제안하는 방법 1, 2를 모두 적용한 결과 방법 2와 유사한 부호화 속도 향상률을 보였지만, BDbitrate와 BDPSNR은 약 1.025%, 0.323dB 의 성능 저하만이 발생하여 방법 2만을 적용하는 경우에 비하여 성능이 향상되었다.

주문형 게임 시스템의 어셈블리 코드를 제외한 x264 소프트웨어의 실험 결과는 표 6~8과 같다. 어셈블리 코드를 포함한 결과와 제외한 결과의 비트율과 PSNR은 동일하다. 표 6의 제안하는 방법 1의 실험 결과, 부호화 속도가 약 127% 향상되었다. 제안하는 방법 2의 실험 결과, 표 7의 결과와 같이 부호화 속도가 약 192% 향상되었다. 제안하는 방법 1, 2를 모두 적용한 결과를 표 8에 나타내었다. 제안하는 방법 1, 2를 모두 적용한 결과 방법 2와 유사한 부호화 속도를 보였다. 실험 결과 어셈블리 코드를 포함한 x264 소프트웨어는 약 111 FPS의 부호화 속도를 보였고, 이는 평균 60 FPS를 초과하여 주문형 게임 시스템의 부호화기로 사용하기에 적합한 부호화 속도

표 6. 어셈블리 코드를 제외한 x264 소프트웨어에 제안한 방법 1을 적용한 실험 결과

QP	x264			방법 1			성능		
	비트율	PSNR	FPS	비트율	PSNR	FPS	BDbitrate	BDPSNR	FPS 향상
22	259.39	53.885	21	256.09	53.901	48	-1.233	0.227	122.5%
27	205.90	47.258	21	203.52	47.287	46			120.0%
32	159.30	42.570	21	157.79	42.594	47			127.3%
37	117.68	39.172	19	116.76	39.199	47			137.7%

표 7. 어셈블리 코드를 제외한 x264 소프트웨어에 제안한 방법 2를 적용한 실험 결과

QP	x264			방법 2			성능		
	비트율	PSNR	FPS	비트율	PSNR	FPS	BDbitrate	BDPSNR	FPS 향상
22	259.39	53.885	21	253.12	50.113	61	2.109	-0.507	183.1%
27	205.90	47.258	21	204.23	46.650	60			187.5%
32	159.30	42.570	21	158.53	42.416	60			188.2%
37	117.68	39.172	19	117.68	39.172	61			207.3%

표 8. 어셈블리 코드를 제외한 x264 소프트웨어에 제안한 방법 1, 2를 적용한 실험 결과

QP	x264			방법 1, 2			성능		
	비트율	PSNR	FPS	비트율	PSNR	FPS	BDbitrate	BDPSNR	FPS 향상
22	259.39	53.885	21	250.52	50.105	61	1.025	-0.323	183.5%
27	205.90	47.258	21	202.17	46.682	60			188.2%
32	159.30	42.570	21	157.28	42.442	60			188.9%
37	117.68	39.172	19	116.87	39.199	61			208.0%

를 보였다. 하지만 x264 소프트웨어의 MMX[12] 어셈블리 코드는 특정한 플랫폼에서만 사용할 수 있는 코드이므로, 다양한 플랫폼의 단말기들에서 사용할 수 없다. 어셈블리 코드를 제외한 실험 결과 약 20 FPS의 부호화 속도를 보였고, 평균 60 FPS를 만족하지 않아 주문형 게임 시스템의 부호화기로 사용하기 위해서는 x264 소프트웨어의 고속화가 필요하다. 본 논문에서 제안한 고속화 방법을 적용한 세 가지 실험 결과 모두 60 FPS에 근접한 부호화 속도를 보였다.

x264 소프트웨어에 제안하는 방법 1을 고속화 한 결과 BDbitrate와 BDPSNR은 약 1.233%와 0.277dB의 성능 향상을 보였다. x264의 움직임 예측은 그 과정을 간소화하기 위해 탐색 영역 전체를 탐색하지 않고, 3단계의 hexagonal 탐색을 사용한다. x264의 hexagonal 탐색 시 첫 번째 단계로 정수 화소 단위의 (-2, 0), (-1,2), (1,2), (2,0), (1,-2), (-1,-2)를 찾는다. 이는 움직임 벡터가 정수 화소 2 이하의 작은 움직임이 많은 영상의 부호화에 유리하다. 하지만 부호화하는 오목 영상은 움직임 벡터의 크기가 크기 때문에 대부분이 화면내 예측 모드로 선택 되어 잔차 신호가 커지고 이 때문에 비트율이 높아지고, PSNR이 낮아진다. 제안하는 방법 1은 움직임 기술자를 이용하여 오목 영상을 부호화할 때, 주변의 바둑돌 모양의 위치를 움직임 벡터로 사용하기 때문에, 움직임 벡터의 값은 커지지만, 잔차신호가 작아지고 양자화 에러가 작아져, 비트율이 낮아지고, PSNR이 높아진다. 제안하는 방법 2는 율・왜곡 최적화를 거치지 않고 매크로블록을 스킵 모드로 결정하기 때문에 비트율은 작아지지만, PSNR이 크게 떨어지게 된다. 제안하는 방법 2를 적용한 결과와 방법 1, 2를 모두 적용한 결과의 부호화 속도가 거의 유사한 이유는 오목 게임은 움직임이 거의 없기 때문이다. 오목 게임은 바둑돌을 놓는 순간, 화면에 변화가 생기고, 바둑돌 주변의 소수의 매크로블록만이 움직임을 갖기 때문에 대부분의 매크로블록은 UR 기술자를 가져 스킵 모드로 결정되고, 소수의 매크로블록만이 CR-OBJ-MI 기술자를 가져 화면내 예측 또는 화면간 예측 모드로 결정된다. 전체 1600개 매크로블록 중 바둑돌이 그려지는 4개의 매크로블록만이 CR-OBJ-MI 기술자를 가져 1%도 되지 않았고, 99% 이상의 매크로블록은 UR 기술자를 가졌다. 방법 1, 2를 모두 적용한 결과에서 화면내 예측 또는 화면간 예측으로 결정되는 매크로블록이 소수이기 때문에 방법 2와 비교하여 부호화 속도에 큰 차이가 나지 않았다.

4. 결 론

본 논문에서는 주문형 게임 시스템에서 장면 기술자를 이용한 H.264/AVC 부호화기의 고속화를 제안하였다. 주문형 게임 시스템에서 사용되는 H.264/AVC 부호화기는 부호화 복잡도가 매우 높아서 게임을 즐기기 위한 부호화 속도를 만족시키기 힘들다.

H.264/AVC 부호화기의 부호화 속도를 만족시키기 위하여 게임 영상의 부호화에 게임 정보를 이용하였다. 게임 영상의 부호화에 이용한 게임 정보를 장면 기술자라 정의하였고, H.264/AVC 부호화기인 x264 소프트웨어의 모드 결정 과정, 움직임 예측 과정을 고속화하였다. 장면 기술자를 이용해 고속화한 x264 소프트웨어의 부호화 속도와 고속화하지 않은 x264 소프트웨어의 부호화 속도를 비교하여, 장면 기술자를 이용한 x264 소프트웨어는 기존 x264 소프트웨어에 비하여 어셈블리 코드를 포함했을 때 약 86%, 어셈블리 코드를 포함하지 않았을 때 약 192% 부호화 속도가 향상되었다. 어셈블리 코드를 포함한 경우 60 FPS를 넘어 부호화기의 고속화가 의미가 없었다. 하지만 어셈블리 코드를 포함하지 않은 경우 부호화 속도가 크게 향상되었고 평균 61 FPS의 성능을 보여 60 FPS의 프레임 레이트를 가지는 주문형 게임 서비스가 가능함을 보였다.

추후 연구 진행 과제로는 정의한 장면 기술자를 일반적으로 적용하기 위하여 게임의 종류별로 장면 기술자를 정의하여 부호화기를 고속화하는 연구가 필요하다. 게임은 특성이 다르므로 게임 영상의 부호화에 항상 같은 장면 기술자를 사용할 수 없으며, 게임마다 다른 장면 기술자를 정의해야 한다. 게임을 스포츠, 액션, 아케이드 등의 종류별로 특성을 파악하여 장면 기술자를 게임의 종류별로 정의하여 같은 종류의 게임에서는 같은 장면 기술자를 사용하여 부호화기를 고속화할 수 있는 연구가 필요하다.

참 고 문 헌

[1] Final Draft International Standard of Joint Video Specification, ITU-T Rec. H.264, ISO/IEC 14496-10, ISO/IEC JCT1/SC29/WG11, 2003.

[2] H. S. Kim, S. H. Kim, and Y. S. Ho, “Fast Mode Decision Algorithm Using Mode Classification for H.264,” International Conference on Information and communication Technologies (ITC2004), pp. 51-58, Thailand, 2004.

[3] M. R. Mohammadnia, H. Taheri, and S. A. Motamedi, “Fast H.264/AVC Intra Mode Decision implementation on DM648 DSP,” 2009 International Conference on Signal Acquisition and Processing (ICSAP2009), pp. 53-56, Kuala Lumpur, 2009.

[4] Y. H. Huang, T. S. Ou, and H. H. Chen, “Fast H.264 Selective Intra Mode Decision for Inter-frame coding,” Picture Coding Symposium 2009 (PCS2009), pp. 1-4, Chicago, 2009.

[5] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu, and S. Wu, “Fast Mode Decision Algorithm for Intra Prediction in H.264/AVC Video Coding,” *Transactions on Circuits and System for Video Technology*, Vol.15, No.7, pp. 813-822, 2005.

[6] A. Laikari, P. Fechteler, B. Prestele, P. Eisert, and J. Laulajainen, “Accelerated Video Streaming for Gaming Architecture,” 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), pp. 1-4, Tampere, Finland, 2010.

[7] P. Fechteler and P. Eisert, “Depth Map Enhanced Macroblock Partitioning for H.264 Video Coding of Computer Graphics Content,” IEEE Int. Conf. on Image Processing (ICIP2009), pp.3441-3444, Cairo, Egypt, 2009.

[8] x264 project website, http://www.videolan.org/developers/x264.html.

[9] FFMPEG project website, http://www.ffmpeg.org.

[10] 김병용, 이동진, 심동규, 권재철, “초저지연 비디오 통신을 위한 RTP 기반 립 싱크 제어 기술에 관한 연구,” 한국멀티미디어학회, Vol.10, No.8, pp. 1039-1051, 2007.

[11] G. Bjøntegaard, “Calculation of average PSNR differences between RD-Curves,” *ITU-T SG16/Q.6*, VCEG-M33, Austin, TX, 2001.

[12] A. Peleg and U. Weiser, “MMX Technology Extension to the Intel Architecture,” *IEEE Micro*, Vol.16, No.4, pp. 42-50, 1996.

전 찬 웅

2009년 한국기술교육대학교 전자공학과 학사
2011년 광운대학교 임베디드소프트웨어공학과 석사
관심분야 : 영상처리, 영상압축

조 현 호

2008년 광운대학교 컴퓨터공학과 학사
2010년 광운대학교 컴퓨터공학과 석사
2010년~현재 광운대학교 컴퓨터공학 박사과정
관심분야 : 영상처리, 영상압축

심 동 규

1999년 서강대학교 전자공학과 공학박사
1999년~2000년 (주) 현대전자
2000년~2002년 (주) 바로비젼
2002년~2005년 Univ. of Washington

2005년~현재 광운대학교 컴퓨터공학과 부교수
관심분야 : 영상신호처리, 영상압축, 컴퓨터비전