Hierarchical Temporal Memory 를 이용한 Video Codec 연구

권용인, 이종원, 허인구, 이진용, 정재완, 백윤흥
서울대학교 전기컴퓨터공학부
e-mail : yikwon,jwlee,igheo,jylee,jwjung@optimizer.snu.ac.kr, ypaek@snu.ac.kr

Development of Video Codec using Hierarchical Temporal Memory

Yong-In Kwon, Jong-Won Lee, In-Gu Heo, Jin-Yong Lee, Jae-Wan Jung, Yun-Heung Paek
Dept. of Electrical Engineering, SNU

요 약

디지털 비디오 압축기술은 대용량의 영상을 화질 손실을 최소화하면서 압축률을 높이는 데에 그 목적이 있다. 모바일 디바이스에서 인코딩/디코딩시에는 프로세싱 오버헤드를 최소화 시켜야 할 필요성이 있다. 또한 특정 동영상을 압축할 시 불필요한 영상을 줄여 압축률을 높이고 타겟 오브젝트에 집중할 수 있도록 하기 위해서 Hierarchical Temporal Memory 를 사용해 오브젝트를 인식하고 타겟 오브젝트만 선택 압축하는 기술을 제안하고자 한다.

1. 서론

H.264, MPEG4 등의 디지털 비디오 압축 기술은 통신과 멀티미디어 시스템의 영역에서 중요한 역할을 하고 있다.[1] 이와 같은 비디오 코딩 기술은 영상의 화질손실을 최소화 하면서 압축률을 높이는 것이 중요하다.

하지만 이러한 방식의 디지털 비디오 압축 기술은 인코딩과 디코딩에 많은 자원과 시간을 필요로 한다. 그러한 노력에도 불구하고 인코딩 된 영상의 데이터 크기는 상당히 크고 실시간으로 인코딩과 디코딩을 하기에는 높은 프로세싱 오베헤드와 전력소모가 뒤따른다.

한편 용도에 따라 영상의 전부가 아닌 특정 오브젝트만이 압축의 대상이 될 경우가 있다. 예를 들어 길거리를 촬영한 영상의 경우 사람과 자동차, 이륜차만이 관심 있는 대상이고 도로나 배경, 날아다니는 새 등은 압축 대상에서 제외시켜 압축 대상 오브젝트의 수를 줄임으로써 압축된 영상의 데이터 크기를 줄일 수가 있다.

또한 영상에서 압축할 대상인 오브젝트의 구체적인 움직임이나 변화를 세밀하게 표현할 필요가 없는 경우도 있다. 이러한 경우에는 각 오브젝트의 대표 이미지만을 가지고 위치와 크기정보를 추가 하여 인코딩 후 저장함으로써 압축영상의 데이터 크기를 크게 줄일 수 있다.

Robust Motion Segmentation for Content-based Video Coding[2] 에서는 고정되어있는 배경에 움직이는 오브젝트의 Sequence 를 효율적으로 나타냄으로써 영상을 압축하는 기법을 사용하였다. 하지만 이 논문의 경우 영상에 움직이는 오브젝트의 수가 많거나 겹쳐질 경우 표현하기가 어려워지며, 영상 압축을 위한 프로세싱 부하가 커진다.

본 논문에서는 Hierarchical Temporal Memory(HTM)을 이용하여 영상 속 오브젝트를 인식하여 타겟 오브젝트만을 효율적으로 압축하고자 한다.

본 논문의 2 장에서는 오브젝트 인식을 위해 사용될 Hierarchical Temporal Memory 에 대하여 설명하고, 3 장에서는 비디오 압축기술의 Overview 를 설명하도록 하겠다. 그리고 4 장에서는 실험결과를 보여주고 마지막 5 장에서 결론을 맺도록 하겠다.

2. HTM

사람은 쉽게 할 수 있지만 컴퓨터로는 그러하지 못하는 일들이 많이 있다. 시각적인 패턴을 인식하는 일, 언어를 이해하는 일, 촉각을 통하여 사물을 인식하고 다루는 일 등이 사람에게는 쉬운 일이다. 수십 년간의 연구에도 불구하고 이러한 일들을 컴퓨터에서 수행할 수 있는 가시적인 알고리즘이 없다.

인간에게 있어서 이러한 능력은 neocortex 에 의해 수행된다. Hierarchical Temporal Memory (HTM)은 neocortex 의 구조와 알고리즘적인 특성을 본따서 만든 기술이다. 그리하여 HTM 은 여러 종류의 인식작업에 있어서 사람의 능력에 근접하거나 넘어설 수 있도록 해준다.

고전적인 프로그래머블 컴퓨터와는 달리 HTM 은 프로그램 된 계산을 하는 것이 아니고 서로 다른 문제에 대하여 같은 알고리즘을 적용한다. HTM 은 'Learning' 이라고 하는 과정을 통해서 문제를 해결한다. 데이터 센서를 통해 문제들을 노출시켜 'Learning' 의 과정을 거치면 HTM 은 문제를 풀 수 있는 능력을 갖게 된다.

HTM 은 Tree 모양의 계층적 구조의 Node 들로 이루어져있다. 각각의 Node 는 공통적인 'Learning'과 'Memory' 기능을 가지고 있다. HTM 은 이 계층적 구조에 정보를 저장한다. HTM 공간적인 계층적 구조임

과 동시에 시간적으로도 계층적 구조를 지닌다. 따라서 실제 세상을 효과적으로 캡쳐 하고 모델링 할 수 있다.

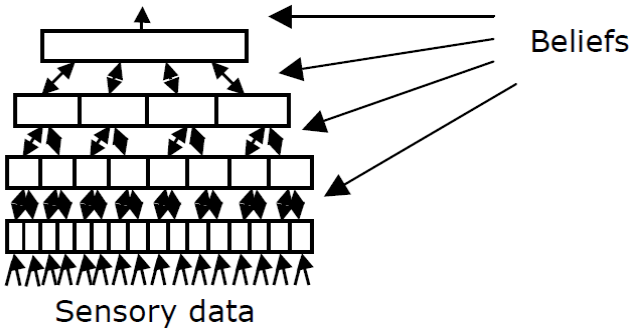

그림 1 HTM Structure

3. System Overview

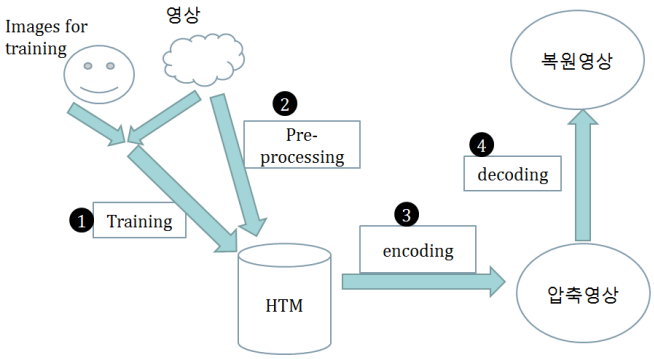

그림 2 System Overview

비디오 인코딩/디코딩 System Overview 는 그림 2 와 같다. 이는 크게 네 단계의 과정을 거친다. 각 과정은 다음과 같다.

1. Learning
2. Pre-processing
3. Encoding
4. Decoding

'Learning' – 이 System 에서 가장 중요한 부분이다. 영상에서 인식할 대상을 선정, 카테고리화 하여 인식이 가능할 수 있도록 HTM 을 형성한다. HTM 은 'Supervised Learning'을 하기 때문에 수작업이 필요하나 한 번 형성된 HTM 은 여러 영상에서 반복적으로 사용할 수 있다. 또한 이 'Learning' 과정을 최적화 알고리즘을 추가하여 수행하면 더 효과적으로 'Learning'을 할 수 있다. 반면 'Learning' 과정은 비교적 많은 프로세싱 자원을 필요로 한다는 단점이 있다.

'Pre-processing' –오브젝트를 인식하기 위해서 영상을 오브젝트 단위로 잘라주는 역할을 한다. 'Pre-processing'을 거치면 영상에서 여러 오브젝트를 추출하게 되고, 추출된 오브젝트는 HTM 의 입력으로 들어가 어떤 카테고리에 속하는지 인식을 하게 된다. 또한 'Pre-processing' 과정에서 각 오브젝트는 위치와 크기 정보가 저장되어 'Encoding' 시 사용된다.

'Encoding' – 'Pre-processing'으로 분할된 오브젝트를 HTM 으로 인식 한 후 나누어지는 카테고리에 따라 영상에 저장할 지를 결정한다. 타겟 오브젝트라면 오브젝트의 대표이미지, 프레임 별 위치와 크기 정보가 저장된다. 대표이미지의 수, Refresh 빈도, fps 등의 값은 옵션에 따라 조정된다.

'Decoding' – 'Decoding'의 경우 HTM 없이 단독 'Decoder' 만으로 'Decoding'이 가능하다. 'Encoding'된 오브젝트 카테고리, 이미지, 위치, 크기 등의 정보를 영상으로 표현한다.

4. 실험

이 장에서는 임의로 제작한 동영상을 'Learning' 과 'Recognition'을 통하여 'Encoding' 하고 'Decoding' 하는 과정에 대한 실험결과를 보여준다. 실험은 200X200 pixel 의 Grayscale Image 의 연속적 Frame 을 이용하였다.

실험에 사용된 오브젝트는 원과 사각형이며 각각 5 개의 'Learning' 용 이미지를 사용해 HTM 을 형성하였다. 그 결과 형성된 HTM 은 2,253KB 의 메모리를 필요로 하였다. 그림 3 은 Training 결과를 보여준다.

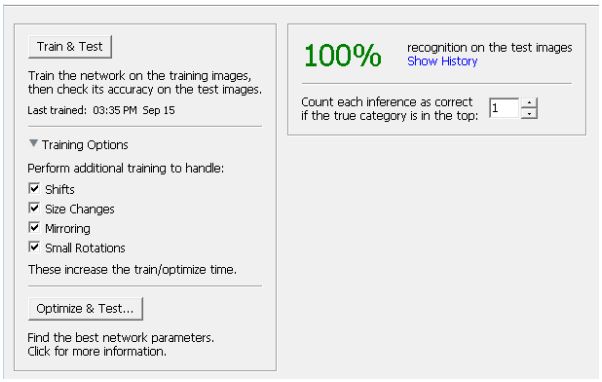

그림 3

이와 같이 형성된 HTM 을 이용하여 1 분 길이의 60fps 동영상을 'Encoding' 한 결과 약 15KB 로 압축이 가능하였다. 결과는 아래 표 1 과 같다.

	HTM	Encoded file	Data of n minutes
Proposed	2,500KB	15KB	(2500+15*n) KB
H.264	-	50MB	(50*n) MB

표 1

HTM 을 이용한 비디오 압축 전후를 그림 4 와

그림 5 에서 비교해 보았다.

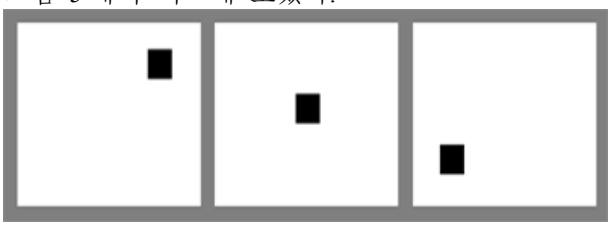

그림 4. Original Frames 0, 30, 60

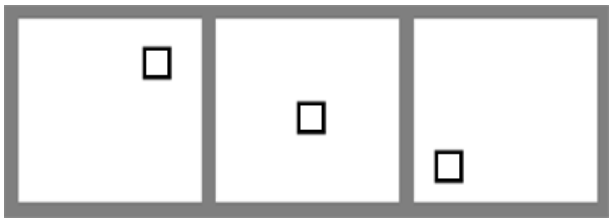

그림 5. Coded-Decoded Frames 0, 30, 60

HTM 을 이용하여 'Learning' 할 시 인식될 '사각형'의 대표 이미지를 '가운데가 빈 선으로 그려진 사각형'으로 설정해 두었기 때문에 Decoding 을 하면 그림 4 와 같은 모습이 된다.

반면, 단순한 이미지인 '원'과 '사각형'이 아닌 좀 더 복잡한 오브젝트들로 'Learning'을 하면 HTM 을 형성하는데 메모리 오버헤드가 어떻게 되는 지를 실험하였다. 인식대상 오브젝트를 세 개로 늘려, '사람', '강아지', '자동차'로 실험을 하였고, 오브젝트 특성상 다양한 이미지를 사용하여 'Learning' 과정을 진행하였다. 이에 따른 메모리 오버헤드는 표 2 와 같다. 전용량은 'Learning' 전 이미지들의 용량 합이고, 후 용량은 'Learning' 후 HTM 을 형성하기 위한 메모리 크기이다.

이미지 수	전용량(MB)	후용량(MB)
3	0.125	2.5
6	0.256	2.5
20	1.12	2.5
60	3.37	2.5
200	10.2	2.6
300	15	2.6
538	27	2.7
665	31.6	2.8

표 2

5. 결론

이 논문에서는 HTM 을 이용한 사물 인식을 통한 비디오 압축 기술에 대하여 설명하였다. 제안된 비디오 압축 기술을 사용하면 더 빠르고 작은 용량으로 인코딩 할 수 있어, 휴대용 단말기에서 사용하기에 용이하다. 위 실험 결과에 따르면 두 가지의 오브젝트를 대상으로 인식, 압축 하였을 시 1 분 길이의 동영상을 압축 할 시 용량이 1/20 배로 줄었다. 한편 추가 1 분당 동영상 압축 용량은 1/3400 배로 더욱 줄어들게 된다. 인식 대상의 오브젝트를 '원'과 '사각형'이 아닌 '사람', '강아지', '자동차' 세 가지로 실험하기 위하여 665 개의 이미지를 이용하여 'Learning' 하더라도 HTM 을 형성하기 위한 메모리는 2.8MB 로 기존의 2.5MB 에 비해 크게 늘지 않아 메모리에 대한 부담은 적다고 할 수 있다.

HTM 을 이용한 사물 인식을 통한 비디오 압축 기술에서 가장 중요하고 많은 프로세싱 자원이 필요한 'Learning' 단계는 수동으로 진행되어야 하고 검증이 필요하다는 점이 이 시스템의 단점이다. 'Learning'을 위해서는 오브젝트의 특성에 따라 많은 수의 이미지가 필요할 수도 있다. 특히 오브젝트의 모양이 다양할수록(ex 앉아있는 사람, 서있는 사람, 사람의 뒷모습 등) 'Learning'에 필요한 이미지의 수가 많아지며 'Learning'이 제대로 되었는지에 대한 검증도 어려워지게 된다.

하지만 한 번 형성된 HTM 은 계속 재사용이 가능하기 때문에, 한 번의 'Learning'만으로 여러 영상에 적용시킬 수 있다. 특히 특정 오브젝트만 인식하여 저장할 필요가 있는 경우나, 단조로운 영상들을 저장하는데 필요한 메모리 오버헤드를 줄이는 데에 도움이 될 것이다.

참고문헌

[1] Mohammed Ghanbari "Standard codecs: image compression to advanced video coding"

[2] F. Odone1 et al., "Robust Motion Segmentation for Content-based Video Coding", RIAO 2000, 6th Conf on Content-Based Multimedia Information Access

[3] George, D., Jaros, B. " The HTM learning algorithms. Technical report, Numenta" , Palto Alto (2006),

ACKNOWLEDGE

본 연구는 교육과학기술부/한국과학재단 우수연구센터 육성사업(과제번호 2010-0001724), 서울시 산학연 협력사업(10560), 2010 년도 정부(교육과학기술부)의 재원으로 한국과학재단의 국가지정연구실사업(No.2010-0018465) 및 IDEC 의 지원을 받아 수행되었습니다.