

DPCM 영상 부호화기에서 예측 오차를 줄이기 위한 변환된

영상에서의 PRA 적용

문주희, 고종석, 김재균

한국과학기술원 전기및 전자공학과

Application of PRA to The Difference Image for Prediction Error Reduction in DPCM Image Coding

Joo Hi MOON, Jong-Seog KOH, Jae-kyoon KIM

Dept. of Electrical Engineering, KAIST

ABSTRACT

This paper proposes a conversion method to reduce prediction error produced when PRA(Pel Recursive Algorithm) motion estimation method is used in real image. The method is to get a spatial difference image from a given raw image and then to apply any PRA method to the difference image. The algorithm proposed in this paper is compared with several algorithms including the ubiquitous Netravali and Robbins' based on the performance and the hardware complexity. Computer simulation shows that the difference image conversion method is about 4.5dB better than the other algorithms with regard to prediction error power.

1. 서 론

영상 신호의 데이터 감축 방법은 크게 예측 부호화 방식(predictive coding)과 변환 부호화 방식(transform coding)으로 구분된다. 낮은 전송율에서 변환 부호화 방식은 예측 부호화 방식에 비하여 감축 능력이 우수한 반면 하드웨어가 훨씬 복잡하다[1]. 예측 부호화 방식 중에서 물체의 운동에 의한 변화를 보상하여 예측하는 방식인 interframe 운동 보상(Motion Compensation) 방식이 우수한 성능을 갖는 것으로 알려져 있다[2,3]. 이러한 운동 보상 방식에는 이동 정보를 구하는 방법에 따라 크게 나누어 BMA(Block Matching Algorithm)[2]과 PRA(Pel Recursive Algorithm)[3]이 있다. BMA는 하드웨어 구현이 용이한 반면 운동 보상 효과가 PRA에 비하여 떨어진다. 본 논문에서는 PRA방법에 의한 운동 보상형 예측 부호화 방식에 대해 알아 보고 이 방식에 의한 예측 오차를 더욱 줄이기 위하여, 먼저 실제 영상을 공간차 영상(Spatial Difference Image)으로 변환한 다음에 PRA방법을 적용하여 예측하는 방식을 제안하였다.

2. PRA 움직임 추정 방식과 문제점

PRA방식은 주사선을 따른 이전 화소에서의 이동 벡터를 보정하여 현재 화소의 움직임을 찾아가는 방식이다[3]. 즉 현재 화소 \vec{x}_t 의 이동 벡터를 \vec{D}_t 이전

화소 \vec{x}_{t-1} 의 이동 벡터를 \vec{D}_{t-1} 라고 할 때 인접한 화소들의 이동 벡터간에 correlation이 크므로 $|\vec{D}_t - \vec{D}_{t-1}|$ 이 작은 값을 갖을 것이라는 가정하에서 \vec{D}_{t-1} 로 부터 \vec{D}_t 를 추정하는 방법이다. \vec{D}_t 의 공간 반복(spatial recursion)식은 다음과 같다.

$$\begin{aligned} \vec{D}_t &= \vec{D}_{t-1} - (\epsilon/2) \nabla_{\vec{x}_t} [DFD(\vec{x}_t, \vec{D}_{t-1})]^2 \\ &= \vec{D}_{t-1} - \epsilon \cdot DFD(\vec{x}_t, \vec{D}_{t-1}) \cdot \nabla I(\vec{x}_t - \vec{D}_{t-1}, t - \tau) \end{aligned} \quad (1)$$

여기서

$$DFD(\vec{x}_t, \vec{D}_{t-1}) \equiv I(\vec{x}_t, t) - I(\vec{x}_t - \vec{D}_{t-1}, t - \tau) \quad (2)$$

이고 $I(\vec{x}_t, t)$ 와 $I(\vec{x}_t - \vec{D}_{t-1}, t - \tau)$ 는 각각 현재 frame t의 \vec{x}_t 위치에서의 밝기(intensity)와 이전 frame t- τ 의 $\vec{x}_t - \vec{D}_{t-1}$ 위치에서의 밝기를 나타낸다. ϵ 은 수렴 속도와 수렴 허용 오차를 결정하는 양의 상수로서 보통 0.01~0.0001의 값을 갖는다. 또한 $\nabla I(\vec{x}_t - \vec{D}_{t-1}, t - \tau)$ 는 이전 frame의 $\vec{x}_t - \vec{D}_{t-1}$ 위치에서의 기울기를 나타내며 그림 1.과 같이 계산된다. 그런데 (1)식에 의하여 추정된 이동 벡터는 실제 이동 벡터 \vec{D}_t 로 수렴해가는 값으로 정확한 이동 벡터라고 볼 수 없다. 더우기 (1)식은 영상내의 물체가 평행 이동을 하였다든 가정하에 움직인 영역에서만 적용되는 식이므로 실제 영상의 많은 부분에서 매우 부정확하게 된다. 일반적으로 실제 영상에서는 상당히 많은 부분에서 밝기의 변화가 급격하므로 수렴과정에 있거나 또는 평행 이동이 아닌 복잡한 운동을 하는 경우 예측 오차의 값이 상당히 커지게 된다. 실제 영상에서 이와 같은 문제는 상당히 심각한 고려 대상이 된다.

3. 공간차 영상에 이용한 PRA 방식의 적용

앞에서 언급한 실제 영상에서의 큰 예측 오차 $DFD(\vec{x}_t, \vec{D}_{t-1})$ 는 다음과 같이 줄일 수 있다. 새로 제안된 방법에 의한 예측 오차를 DFD_0 라고 하면

$$\begin{aligned} DFD_0(\vec{x}_t, \vec{D}_{t-1}) &\equiv DFD(\vec{x}_t, \vec{D}_{t-1}) - DFD_0 \\ &= (I(\vec{x}_t, t) - I(\vec{x}_t - \vec{D}_{t-1}, t - \tau)) - (\hat{I}(\vec{x}_t, t) - \hat{I}(\vec{x}_t - \vec{D}_{t-1}, t - \tau)) \\ &= (I(\vec{x}_t, t) - \hat{I}(\vec{x}_t, t)) - (I(\vec{x}_t - \vec{D}_{t-1}, t - \tau) - \hat{I}(\vec{x}_t - \vec{D}_{t-1}, t - \tau)) \end{aligned} \quad (3)$$

여기서 DFD_0 가 $DFD(\vec{x}_t, \vec{D}_{t-1})$ 과 같은 부호를 갖고 크기가 비슷한 값을 갖게 할 수 있다면 $|DFD_0| < |DFD|$ 로 될 것이다. 이런 조건을 만족하는 DFD_0 는 현재 예측되는 \vec{x}_t 위치 주위의 화소들에 대하여 DFD를 구함으로써 얻을 수 있다.

예를 들어 \hat{I} 의 값으로서 I값의 적당한 예측치를 이용할 수 있다. 즉 가장 간단한 예측 방법인 이전 화소 예측 방식(previous pel prediction)을

이용하면 $\vec{a} = (1, 0)$ 라고 할 때, $DFD_0 = I(\vec{x}_1 - \vec{a}, t) - I(\vec{x}_1 - \vec{a}, t - \tau)$ 로 된다. 이 값은 \vec{D}_{-1} 을 이용하여 같은 line 의 바로 이전 화소에서 구한 DFD가 되며 추정된 이동 벡터가 실제 이동 벡터에 수렴하고 있는 경우 대체로 $DFD(\vec{x}_1, \vec{D}_{-1})$ 과 같은 부호를 갖고 크기가 비슷하게 된다.

이처럼 DFD₀ 를 구하여 DFD를 줄이는 방법은 결국 공간차(spatial difference) $I(\vec{x}_1, t) - I(\vec{x}_1 - \vec{a}, t)$ 에 대한 예측 오차를 구하는 것과 동일하다. 즉 실제 영상을 공간차 영상으로 변환하여 움직임 보상하는 것과 같게 된다. 또한 공간차 영상으로 변환하였을 때 frame간의 물체의 운동에 관한 정보는 그대로 존재하게 되므로 실제 영상에서의 마찬가지로 공간차 영상에서도 이동 벡터를 추정할 수 있다. PRA 방식에서는 예측 오차가 이동 벡터 추정시에 쓰여 지므로 공간차 영상에서 움직임을 추정하는 것이 계산량을 줄이는 방법이므로 유리하다. 이상과 같은 방법은 실제 하드웨어의 큰 증가 없이 예측기의 성능을 향상시키게 된다.

(3) 식에서 DFD₀ 를 구하는 방법으로 다음과 같은 공간차 영상을 사용하였다.

- 2개의 화소에 근거한 공간차 영상 (2 pels based Spatial Difference Image)

실제 영상의 밝기를 I, 변환된 영상의 밝기를 I*라 하면

$$I(\vec{x}, t) \xrightarrow{\text{conversion}} I^*(\vec{x}, t) = I(\vec{x}, t) - I(\vec{x} - (1, 0), t)$$

$$I^*(\vec{x}, t) \xrightarrow{\text{inverse conversion}} I(\vec{x}, t) = I^*(\vec{x}, t) + I(\vec{x} - (1, 0), t) \quad (4)$$

여기서 $I(\vec{x} - (1, 0), t)$ 는 이전 화소의 밝기이므로 변환은 causal 하다.

4. 컴퓨터 시뮬레이션 결과 및 고찰

본 논문에서 제안한 변환 영상에서의 PRA 적용 방식의 성능을 조사하기 위하여 각 화소들이 8bit로 양자화되고 16개의 연속된 frame으로 구성된 "Cronkite" 영상을 이용하여 시뮬레이션을 하였다. 비교 대상으로서 운동 보상 방식의 성능을 알아보기 위하여 그림 2.(a)의 CRC(Conditional Replenishment Coding)[4] 방식을 사용하였다. 실제 영상에서 PRA의 성능 측정을 위하여 그림 2.(b)의 기존의 방식[3]을 사용하였으며 하드웨어의 관점에서 새로이 제안된 방식과 비슷한 그림 2.(c)의 방법[5]을 추가 비교 대상으로 삼았다. 그림 2.(d)는 (4)식을 이용하여 제안된 방식이다. 위의 모든 방식에서 움직임 영역을 구분하기 위하여 Netravali가 사용한 영상 분할(Image Segmentation)방법[4]을 이용하였다. 또한 이동 벡터의 공간 반복(Spatial recursion)은 그림 3. 과 같이 Sabri가 제안한 방법[6]을 약간 수정하여 사용하였다. 성능 비교의 기준으로서 다음과 같이 정의되는 SNR을 사용하였다.

$$SNR = 10 \log_{10}(255^2 / \text{prediction error power}) \text{ [dB]}$$

여기서 예측 오차 전력은 영상 분할 후 움직임 영역으로 판정된 영역에서 계산한 값이다. 그 이유는 기존의 방식을 비롯한 모든 방식이 배경 영역에서는 거의 일정한 성능을 갖기 때문이다.

시뮬레이션 결과를 그림 4. 에 나타내었다. 그림 4. 에서 운동 보상 방식이 CRC 보다 아주 우수한 성능을 갖음을 알 수 있다. 또한 공간차 영상으로 변환하는 방식이 실제 영상에서의 PRA 방식 보다 4.5dB 정도 높음을 알 수 있다.

5. 결론

본 논문에서는 PRA 운동 보상형 예측 방식을 공간차 영상으로 변환하여 적용하는 방식을 제안하였다. 이 방식은 결국 인접 화소간의 중복성을 먼저 제거하고 그 후에 PRA 방식을 이용하여 인접 frame간의 중복성을 제거하는 방식이 된다. Cronkite 영상을 사용하여 시뮬레이션을 한 결과 기존의 다른 방식보다 새로이 제안된 방식에서 움직임 영역의 예측 오차 전력에 있어서 4.5dB 정도의 개선이 이루어 졌다. 또한 영상의 중복성을 제거하기 위하여 PRA 방식을 사용하는 경우에 인접 화소간의 중복성을 먼저 제거하는 것이 효율적임을 알 수 있었다. 인접 화소 간의 중복성이 제거된 공간차 영상에서 움직임을 추정하는 데 필요한 정보는 그대로 존재한다. 따라서 공간차 영상에서 움직임을 추정함으로써 하드웨어의 증가를 방지할 수 있다.

[References]

- [1]. A.K. JAIN, "Image data compression: a review," IEEE Proc., vol.69, No.3, Mar.1981.
- [2]. T. KOGA, et al, "Motion compensated interframe coding for video conferencing," NTC, Nov.29-Dec.3 1981, New Orleans, LA, U.S.A.
- [3]. A.N. NETRAVALI, J.O. ROBBINS, "Motion compensated television coding; Part I," BSTJ, vol.58, Mar. 1979.
- [4]. F.W. MOUNTS, "A video encoding system with conditional picture-element replenishment," BSTJ, Sep. 1969.
- [5]. N. MUKAWA, et al, "An interframe coding system for video teleconferencing signal transmission at a 1.5 Mbit/s rate," IEEE Trans., COM-32, Mar. 1984.
- [6]. S. SABRI, "Movement compensated interframe prediction for NTSC color TV signals," IEEE Trans., COM-32, No.8, Aug. 1984.

$$\begin{matrix} \times & & \times \\ I_1 & & I_2 \\ & \circ & \\ & & \\ \times & & \times \\ I_3 & & I_4 \end{matrix} \quad \begin{matrix} EDIF = (I_2 - I_1 + I_4 - I_3) / 2 \\ LDIF = (I_3 - I_1 + I_4 - I_2) / 2 \\ \\ VI = [EDIF \quad LDIF] ; \text{gradient} \\ \circ ; \text{displaced location} \\ \text{in previous frame} \\ I_1, I_2, I_3, I_4 ; \text{intensity} \end{matrix}$$

그림 1. 기울기 계산을 위한 방법

Fig. 1. Intensity Gradient Calculation

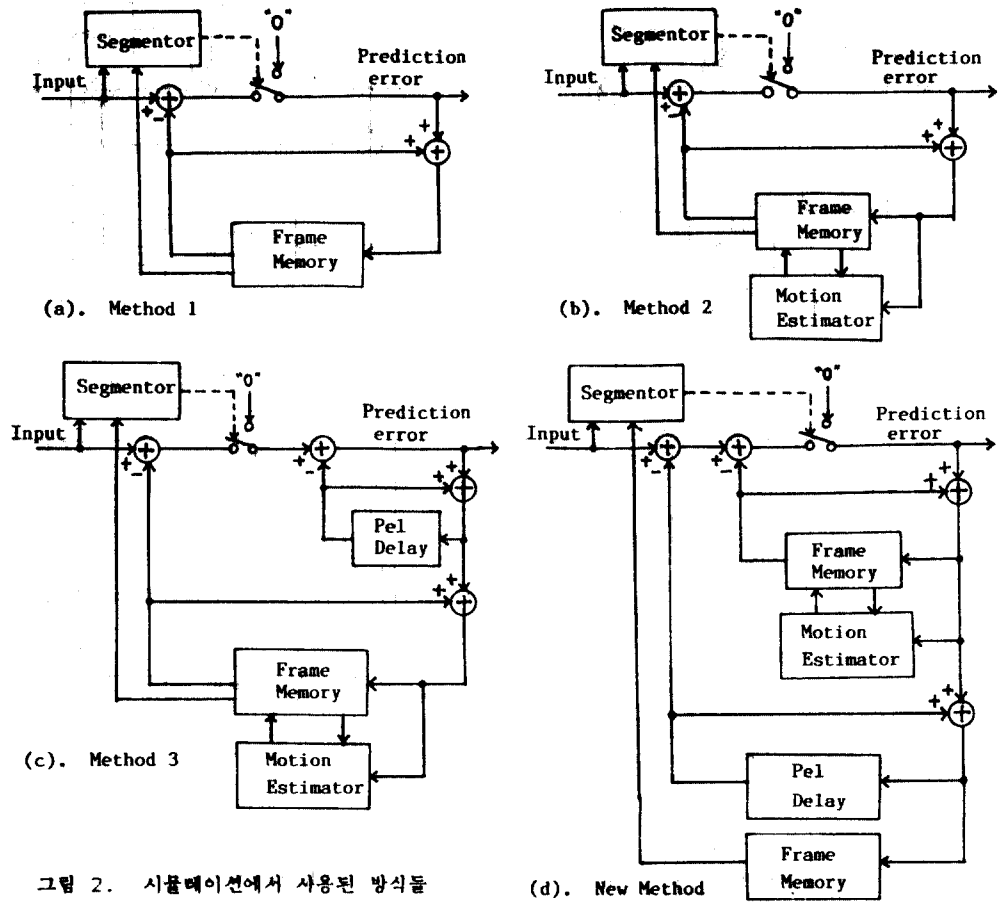


그림 2. 시뮬레이션에서 사용된 방식들
 Fig. 2. Schemes used in the Simulation

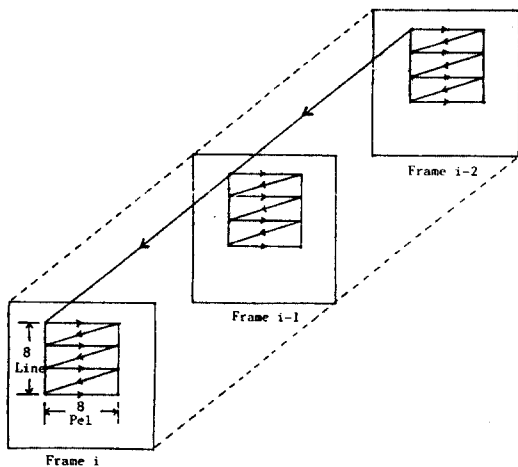


그림 3. 이동 벡터의 공간 반복 방법
 Fig. 3. Recursion Method of Motion Vector

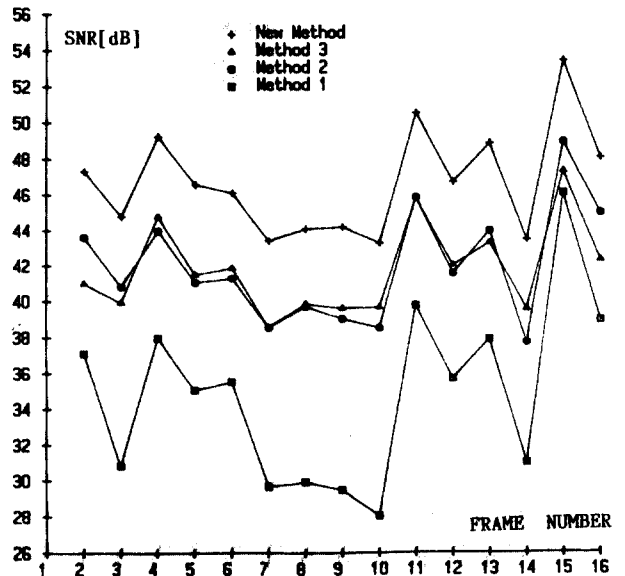


그림 4. 시뮬레이션 결과
 Fig. 4. SIMULATION RESULTS