

스플라인 곡선을 이용한 블록화 현상 감소 회로의 설계

박성모[†] · 김희정^{**} · 최진호^{***} · 김지홍^{****}

요 약

블록화 현상이란 입력 영상에 대해 매우 낮은 비트율로 블록 기반 부호화 방식을 수행할 때 복원 영상에서 나타나는 블록 형태의 왜곡을 의미한다. 본 논문에서는 스플라인 곡선 생성 방식을 적용한 블록화 현상 감소 알고리즘에 대해 회로를 설계하고 구현하여 동작을 확인한다. 설계된 회로는 영상 데이터의 저장을 위한 메모리, 알고리즘 실행을 위한 산술/논리 연산회로, 제어 신호를 발생하는 제어 블록으로 구성된다. 산술/논리 연산회로는 처리될 화소와 블록 경계간의 거리에 따른 가중치 계산 회로와 블록 경계에서 유리 B 스플라인을 적용한 화소값 계산 회로로 이루어진다. 모의실험을 통해 본 논문에서 설계된 회로는 매우 우수한 블록화 현상 감소 기능을 갖는 것을 알 수 있다.

Circuit Design of a Blocking Effect Reduction Algorithm using B-Spline Curve

Sung Mo Park[†], Hee Jung Kim^{**}, Jin-Ho Choi^{***} and Ji Hong Kim^{****}

ABSTRACT

The blocking effect results from independent coding of each image block and becomes highly visible, especially coded at very low bit rates. In this paper, a blocking effect reduction circuit is designed which is composed of a memory, arithmetic and logic unit, and control block. The circuit is based on a rational open uniform B-spline curve that uses to produce a smooth curve through a set of control points. The weight values and the modified pixel values in a rational open uniform B-spline curve are calculated using arithmetic and logic circuits. The simulation results show that the circuit has excellent performance for all pattern of the blocking effects.

Key words: blocking effect, spline curve, VHDL

1. 서 론

영상 정보의 압축을 위한 주요 방법 중의 하나인 블록 기반 부호화 기법은 입력 영상을 일정한 크기의 블록으로 분할한 후 각 블록을 독립적으로 처리하며, 알고리즘의 단순성에 비해 매우 높은 압축 성능을 갖는다. 그러나 입력 영상에서의 지역적 상관성이 고

접수일 : 2003년 4월 4일, 완료일 : 2003년 5월 27일

[†] 부산외국어대학교 컴퓨터전자공학과 석사과정

^{**} 정회원, 부산외국어대학교 컴퓨터전자공학과 박사과정

^{***} 정회원, 부산외국어대학교 컴퓨터공학부 부교수

^{****} 준회원, 동의대학교 컴퓨터영상공학부 조교수

려되지 않기 때문에 인접 블록들 간의 경계 영역에서 연속성이 깨어져 복원 영상에서 블록 형태의 왜곡이 일어날 수 있는데, 이러한 현상을 블록화 현상(blocking effect)이라고 한다[1]. 블록화 현상의 감소를 위한 기법들은 전체 블록 기반 부호화 과정에서 적용되는 위치에 따라 전처리 기법과 후처리 기법으로 분류된다. 전처리 기법은 영상 부호화 과정에서 적용되는 블록화 현상 감소 기법을 의미하며, 그 종류에는 블록들 간의 상관성을 유지하기 위하여 블록 형성 시 인접한 블록이 부분적으로 서로 중복되도록 설정하는 방식인 블록 중복 기법, LOT(Lapped Orthogonal

Transform)를 사용하여 블록 중복 방법에 비해 비트율의 증가 없이 블록화 현상을 감소시키는 LOT 기법, 전체 영상 영역을 상관성의 높고 낮음에 따라 두 개의 영역으로 분할한 후 이들을 서로 달리 처리하는 CTC(Combined Transform Coding) 기법 등을 들 수 있다[2,3]. 후처리 기법은 복원 영상에 적용되는 기법들을 통칭하며, 그 종류에는 입력 영상의 고주파 성분은 제거하고 저주파 성분만을 보존하는 가우시안 저역통과 필터(Gaussian lowpass filter), 영상의 특성에 적응적으로 부합하는 필터 계수를 사용하는 적응적 필터 방법(adaptive filter method), 블록의 특성에 따라 설정된 임계치를 기준으로 1차원의 서로 다른 필터 계수를 블록 내부와 경계 영역에 각각 적용시켜 블록의 내부보다 경계 부분을 좀 더 평탄화시키는 평균적 임계 필터(average threshold filter) 등이 있다[4].

근래에 들어 JPEG, MPEG 등 영상 압축에 관한 국제 표준들의 제정으로 인하여 블록화 현상 감소를 위한 전처리 기법보다는 후처리 기법의 활용도가 증가되고 있다. 또한 영상 신호의 실시간 처리를 필요로 하는 다양한 분야에서 블록화 현상 감소를 위한 성능과 함께 처리 속도가 중요한 요소로 부각되고 있다. 이러한 점을 감안할 때 블록화 현상 감소를 위한 집적회로로 구현하여 이를 영상처리시스템의 한 처리 요소로서 활용하는 방안은 처리 속도 향상을 위한 하나의 효율적 방법으로 사용될 수 있다.

본 논문에서는 최근 제안된 유리 B 스플라인 곡선을 이용한 블록화 현상 감소 기법을 회로로 설계하고 구현한다. 유리 B 스플라인 곡선을 이용한 블록화 현상 감소 기법은 후처리 기법에 속하며, 기존 기법들에 비해 단순하고 소요 계산량이 매우 적다. 그리고 블록화 현상 감소를 위해 처리될 화소와 블록 경계 간의 거리에 따른 가중치를 알고리즘에 도입하여 매우 우수한 블록화 현상 감소 성능을 갖는다[5].

설계된 회로는 영상 데이터의 저장을 위한 메모리, 알고리즘 실행을 위한 산술/논리 연산회로, 제어 신호를 발생하는 제어 블록으로 구성된다. 산술/논리 연산회로는 가중치를 계산하기 위한 회로와 블록 경계의 화소들에 대해 유리 B 스플라인 생성 기법을 적용하여 블록화 현상이 감소된 새로운 화소 값을 계산하는 회로로 이루어진다. 설계된 회로의 기본적인 동작 방식은 다음과 같다. 먼저 256×256 크기의 영상 데이터를 입력하여 메모리에 저장하고, 영상 데

이터에서 블록 경계에 위치한 화소들에 대해 유리 B 스플라인 곡선 알고리즘을 적용한다. 이 때 계산된 새로운 화소값은 메모리의 해당 번지에 다시 저장되며, 계산이 완료된 후 메모리에 저장된 영상 데이터를 파일로 출력한다.

본 논문의 2장에서는 유리 B 스플라인 곡선을 이용한 블록화 현상 감소 기법에 대해 간략히 살펴보고, 3장에서는 설계된 회로의 구조 및 동작 방식에 대하여 기술한다. 4장의 모의 실험에서는 설계된 회로의 동작과 실험 결과를 제시하며, 5장의 결론으로 논문을 맺는다.

2. 유리 B 스플라인 곡선을 이용한 블록화 현상 감소 기법

본 논문에서 구현하고자 하는 블록화 현상 감소 기법은 컴퓨터 그래픽스 분야에서 부드러운 곡선을 생성할 때 사용하는 유리 B 스플라인 곡선을 이용한 다[5,6]. 유리 B 스플라인은 두 B 스플라인의 비(ratio)를 의미하며, 식 (1)과 같이 표현된다.

$$P(u) = \frac{\sum_{k=0}^n w_k P_k B_{k,d}(u)}{\sum_{k=0}^n w_k B_{k,d}(u)} \quad (1)$$

식 (1)에서 P_k 와 $B_{k,d}(u)$ 는 각각 곡선을 그릴 때 사용될 $n+1$ 개 제어점의 좌표 벡터와 $d-1$ 차수의 블렌딩 함수를 의미하며, w_k 는 각 제어점에 대한 가중치이다. 유리 B 스플라인 곡선을 이용한 블록화 현상 감소 기법에서는 영상의 화소값을 제어점의 좌표로 사용하며, 인접한 블록들의 화소들 중 블록 경계를 기준으로 3개씩의 화소값을 이용한다[5]. 따라서 식 (1)에서 제어점의 수를 6으로 설정하며, 곡선 함수의 계산량을 감소시키기 위해 스플라인 곡선의 차수를 3으로 설정한다. 그리고 제어점에 대한 가중치 w_k 는 처리될 화소와 블록 경계 간의 거리를 고려하여 차등적으로 설정되도록 한다. 즉 블록 경계에서 동일 위치에 있는 화소들의 값에 대한 절대차(absolute difference)와 거리의 비율을 각 제어점에서의 가중치로 설정한다.

그림 1은 입력 영상에서 블록 경계에 위치한 6개의 화소값 $P_0 \sim P_5$ 에 대해 유리 B 스플라인 곡선을 적용하여 블록화 현상이 감소된 새로운 화소값 $P'_0 \sim P'_5$ 을 생성하는 개념을 나타낸다. 그림 1에서 블

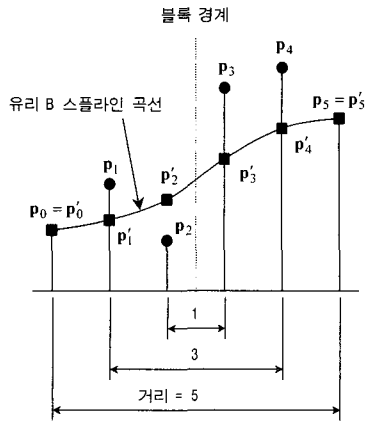


그림 1. 유리 B 스플라인 곡선을 이용한 블록화 현상 감소 기법의 개념

수 있는 것처럼 P_0 과 P_5 , P_1 과 P_4 , 그리고 P_2 와 P_3 이 블록 경계를 기준으로 서로 동일한 위치에 존재하며 각각 5, 3, 1의 거리를 가지므로, 각 제어점에 대한 가중치는 식 (2)와 같이 설정된다[5].

$$\begin{aligned} w_0 = w_5 &= |p_0 - p_5|/5 \\ w_1 = w_4 &= |p_1 - p_4|/3 \\ w_2 = w_3 &= |p_2 - p_3|/1 \end{aligned} \quad (2)$$

그리고 유리 B 스플라인 곡선에서 사용되는 노트 벡터를 $\{0, 0, 0, 0, 1, 2, 3, 3, 3, 3\}$ 으로 설정함으로써, 블록화 현상의 감소를 위해 수정된 새로운 화소값 $P'_0 \sim P'_5$ 은 식 (3)과 같이 결정된다.

$$\begin{aligned} p'_0 &= P(0) \\ p'_1 &= P(3/5) \\ p'_2 &= P(6/5) \\ p'_3 &= P(9/5) \\ p'_4 &= P(12/5) \\ p'_5 &= P(3) \end{aligned} \quad (3)$$

표 1. 곡선 함수에서의 블렌딩 함수 값

$B_{k,d}(u)$	0	3/5	6/5	9/5	12/5	3
$B_{0,4}(u)$	1111101000B	0001000000B	0000000000B	0000000000B	0000000000B	0000000000B
$B_{1,4}(u)$	0000000000B	1000101110B	0010011111B	0000000010B	0000000000B	0000000000B
$B_{2,4}(u)$	0000000000B	0101010110B	1001001100B	0100011010B	0000100100B	0000000000B
$B_{3,4}(u)$	0000000000B	0000100100B	0100011010B	1001001100B	0101010110B	0000000000B
$B_{4,4}(u)$	0000000000B	0000000000B	0000000010B	0010000000B	1000101110B	0000000000B
$B_{5,4}(u)$	0000000000B	0000000000B	0000000000B	0000000000B	0001000000B	1111101000B

수정된 화소값으로 사용되는 곡선 함수의 각 점에서의 블렌딩 함수 값은 표 1과 같다. 표 1은 설계된 회로에서 식 (3)의 유리 B 스플라인 곡선값을 계산할 때 사용되며, 이진수로 표기되어 있다.

3. 회로 구현

본 논문에서 설계된 회로의 개략도를 그림 2에 나타내었으며, 회로 동작 시 영상 데이터의 흐름은 다음과 같다. 먼저 입력 파일로부터 원 영상 데이터가 메모리로 입력되며, 가중치 계산을 위한 산술/논리 연산부(ALU_w)에서는 메모리의 영상 데이터 중 블록 경계의 화소값 $P_0 \sim P_5$ 를 입력하여 식 (2)에 정의된 가중치 $w_0 \sim w_5$ 를 계산한다. 그리고 블록 경계의 화소값과 ALU_w에서 계산된 가중치는 유리 B 스플라인 곡선 적용을 위한 산술/논리 연산부(ALU_p)로 입력된다. ALU_p에서는 식 (1)과 (3)을 수행하여 블록화 현상이 감소된 새로운 화소값 $P'_1 \sim P'_4$ 를 계산한 후, 이들을 메모리에서 $P_1 \sim P_4$ 가 저장되었던 주소에 저장한다. 이러한 연산은 블록의 경계 영역에 위치하는 수평 및 수직 방향의 모든 화소들에 대하여 적용된다. 이를 위해 설계된 회로에서는 먼저 수평 방향의 블록화 현상을 감소하기 위하여 수평 방향의

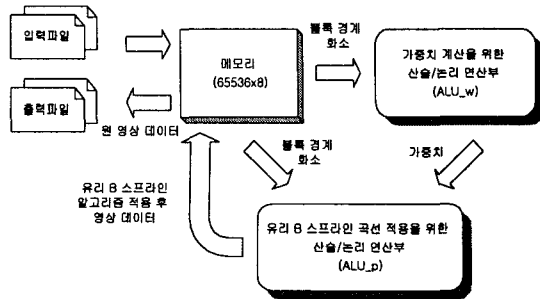


그림 2. 설계된 회로의 개략도

블록 경계에 위치한 화소값들을 읽어 연산을 수행한 후 메모리에 저장한다. 그리고 이어서 수직방향의 블록 경계에 있는 화소값들을 읽은 다음, 연산을 수행하고 메모리에 다시 저장한다.

그림 3은 설계된 회로의 내부 구조를 나타낸 것이다. 회로는 VHDL을 이용하여 설계하였으며, 설계 툴은 synopsys를 사용하였다[7,8]. 그림 3의 제어 블록으로부터 산술/논리 연산부 ALU_w, ALU_p로 입력되는 제어신호 input_en은 메모리에 저장된 원 영상 데이터를 산술/논리 회로에서 읽기 위한 신호이다. 그리고 sub는 뺄셈 연산을 수행하기 위한 제어신호이며, div와 shift는 나눗셈 연산시 뺄셈과 시프트 연산을 수행하기 위한 제어신호이다. 그리고 alout은 각 산술/논리 연산부에서 연산의 결과를 출력하는데 필요한 제어신호이다.

가중치 계산을 위한 산술/논리 연산부 ALU_w는 3개의 연산회로 ALU_w05, ALU_w14, ALU_w23으로 구성된다. 이들의 회로 구성은 동일하지만, 서로 다른 화소값들을 입력하여 연산한다. 즉 ALU_w05는 식 (2)를 이용하여 가중치 w_0 와 w_5 를 계산하는 회로이고, ALU_w14와 ALU_w23은 각각 가중치 w_1 와 w_4 , 그리고 w_2 와 w_3 을 계산하는 회로이다.

그리고 유리 B 스플라인 곡선 적용을 위한 산술/논리 연산부 ALU_p는 서로 동일하게 설계된 4개의 연산회로 ALU_p1, ALU_p2, ALU_p3, ALU_p4로 구성되어 있다. 이들은 식 (1)과 (3)에 따라 각각 P'_1, P'_2, P'_3, P'_4 를 계산한다. 이와 같이 산술/논리 연산부 ALU_w와 ALU_p를 각각 하나씩의 연산회로로 구성한 후 순차적으로 연산을 수행할 수도 있지만,

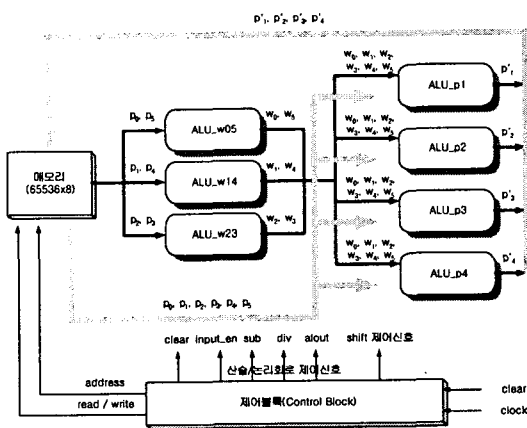


그림 3. 전체 회로의 블록 다이어그램

방대한 분량의 영상 데이터에 대한 처리 속도를 증가시키기 위하여 병렬 계산을 할 수 있도록 다수의 연산회로로 구성하였다. 각 회로 블록에서 일어나는 동작을 타이밍 순서에 따라 표현한 흐름도를 그림 4에 나타내었으며, 회로의 동작 순서는 다음과 같다. 먼저 256×256 크기의 영상 데이터를 읽어 메모리에 저장한 다음, 블록 경계의 화소 $P_0 \sim P_5$ 를 순차적으로 6 사이클 동안 읽는다. ALU_w05, ALU_w14, ALU_w23에서는 각각 P_0 와 P_5, P_1 와 P_4, P_2 와 P_3 를 이용하여 식 (2)에 따라 가중치 $w_0 \sim w_5$ 를 연산한다. 데이터의 입출력 및 가중치 연산은 22 사이클 동안 수행된다. 계산된 가중치와 화소값 $P_0 \sim P_5$ 는 ALU_p1, ALU_p2, ALU_p3, ALU_p4에 입력되며, 여기에서는 식 (1)과 (3)에 정의된 유리 B 스플라인 곡선을 적용하여 블록화 현상이 감소된 새로운 화소값 $P'_1 \sim P'_4$ 를 57 사이클 동안 계산하며, 계산된 데이터는 메모리의 해당 번지에 4 사이클 동안 저장된다.

그림 5 (a)는 가중치 계산을 위한 산술/논리 연산부 ALU_w에서 메모리 주소, 블록 경계 화소값 $P_0 \sim P_5$, 연산에 필요한 제어신호, 그리고 출력 신호의 파형이다. 그림 5 (a)에서 alout1_data는 가중치 w_2 와 w_3 , alout3_data는 가중치 w_1 와 w_4 , alout5_data는 가중치 w_0 와 w_5 의 계산값이다. 그리고 그림 5 (b)는 블록 경계에서 유리 B 스플라인 곡선을 적용하여 화소값을 계산하는 산술/논리 연산회로의 입출력 파형이다.

그림 5 (b)에서 pout1_high, pout2_high, pout3_high, pout4_high는 각각 블록 경계 영역에서 계산된

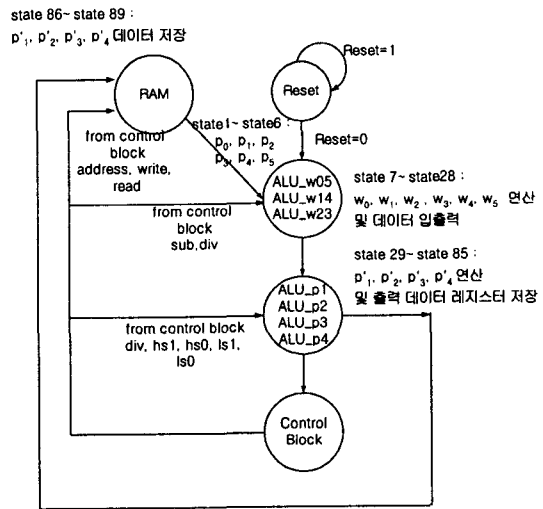
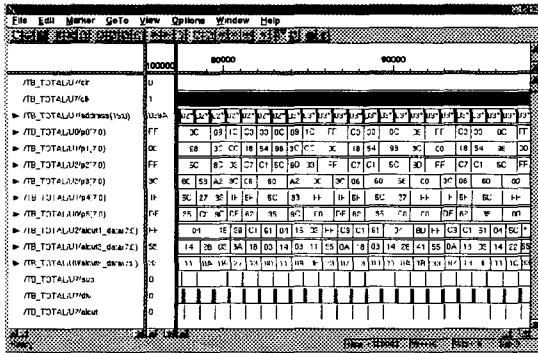
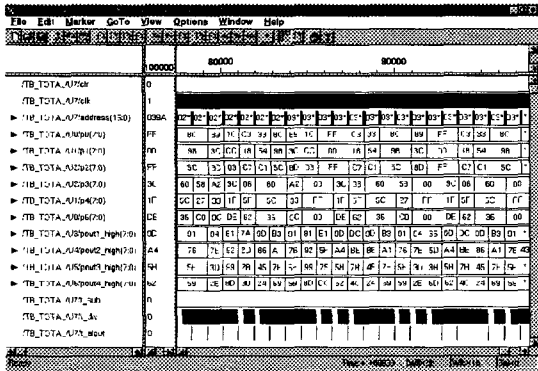


그림 4. 연산을 위한 상태도



(a)



(b)

그림 5. 설계된 회로의 출력 파형 (a) ALU_w05, ALU_w14, ALU_w23의 입출력 파형 (b) ALU_p1, ALU_p2, ALU_p3, ALU_p4의 입출력 파형

P'_1, P'_2, P'_3, P'_4 의 영상 데이터이다.

설계된 회로의 합성 결과를 그림 6에 나타내었다. 전체 회로는 가중치 계산을 위한 연산회로 ALU_w05, ALU_w14, ALU_w23과 유리 B 스플라인 곡선을 적용하여 블록 경계에서 영상 데이터를 계산하는 ALU_p1, ALU_p2, ALU_p3, ALU_p4, 제어 신호를 생성하는 제어블록, 그리고 메모리로 구성되어 있다.

4. 모의 실험

본 논문에서 구현한 회로를 시험 영상에 대해 모

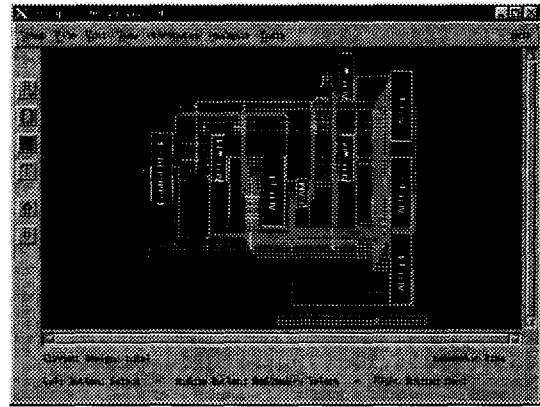


그림 6. 합성된 회로 블록

의 실험하였다. 시험 영상은 JPEG baseline 시스템을 이용하여 다양한 비트율로 압축한 256×256 크기의 Lena, plane, baboon, boat, pepper 영상 등이다. 표 2는 영상의 특징에 따라 다소 차이는 있으나 블록 현상이 나타나기 시작하는 0.25bpp 이하의 주요 비트율에서의 비트율 대비 PSNR 값을 나타낸다. 블록화 현상 감소 등을 포함하는 영상 개선 과정에서의 화질 향상 정도를 나타내는 척도로서 PSNR과 같은 정량적 방법이 갖는 한계가 있다. 그러나 표 2를 볼 때, 본 논문에서 구현된 회로 적용 시 비트율 0.25bpp에서 약 0.1dB 만큼 화질이 향상되었고 0.15bpp에서 약 0.7dB 만큼 PSNR 값이 증가되어, 비트율 감소에 따라 블록화 현상이 심해질수록 감소 효과가 급격히 향상됨을 볼 수 있다.

비트율 0.25bpp에서의 시험 영상과 본 논문에서 구현한 회로를 적용했을 때의 출력 영상을 그림 7에 나타내었다. 그림 7에서 왼쪽의 영상들(a, c, e, g, i)은 시험영상을 나타내며, 오른쪽의 영상들(b, d, f, h, j)은 출력 영상을 각각 나타낸다. 이 그림에서 볼 때 시험 영상에서는 블록화 현상이 나타남을 볼 수 있으며, 출력 영상에서는 PSNR 값이 소폭 증가하였음에도 불구하고 시험 영상에서 존재하던 블록화 현상이

표 2. 시험 영상에 대한 비트율 대비 PSNR (단위: dB)

시험 영상 비트율	Lena		plane		baboon		boat		pepper	
	처리 전	처리 후	처리 전	처리 후	처리 전	처리 후	처리 전	처리 후	처리 전	처리 후
0.25	30.28	30.38	29.37	29.35	21.53	21.66	26.34	26.57	30.51	30.64
0.20	28.83	29.23	27.07	27.48	20.77	21.03	24.76	25.28	28.77	29.25
0.15	25.54	26.31	24.06	24.72	19.86	20.21	22.04	22.83	25.48	26.34

매우 감소하였음을 볼 수 있다. 시험 영상과 출력 영상의 상세한 비교를 위해 그림 7 (a)와 (b)의 Lena 영상에서 주요 부분에 대한 확대 영상을 그림 8~10에 나타내었다. 눈 부분에 대한 확대 영상인 그림 8에서 볼 때, 눈동자 부분에 형성되어 있던 블록화 현상

이 매우 감소된 것을 볼 수 있으며, 또한 눈동자와 눈썹 주위에 큰 형태로 존재하던 블록화 현상도 제거된 것을 알 수 있다. 그림 9의 입 부분 확대 영상에서 볼 때 입과 코 부근의 블록화 현상이 매우 감소된 것을 볼 수 있으며, 그림 10의 모자 부분 확대 영상에

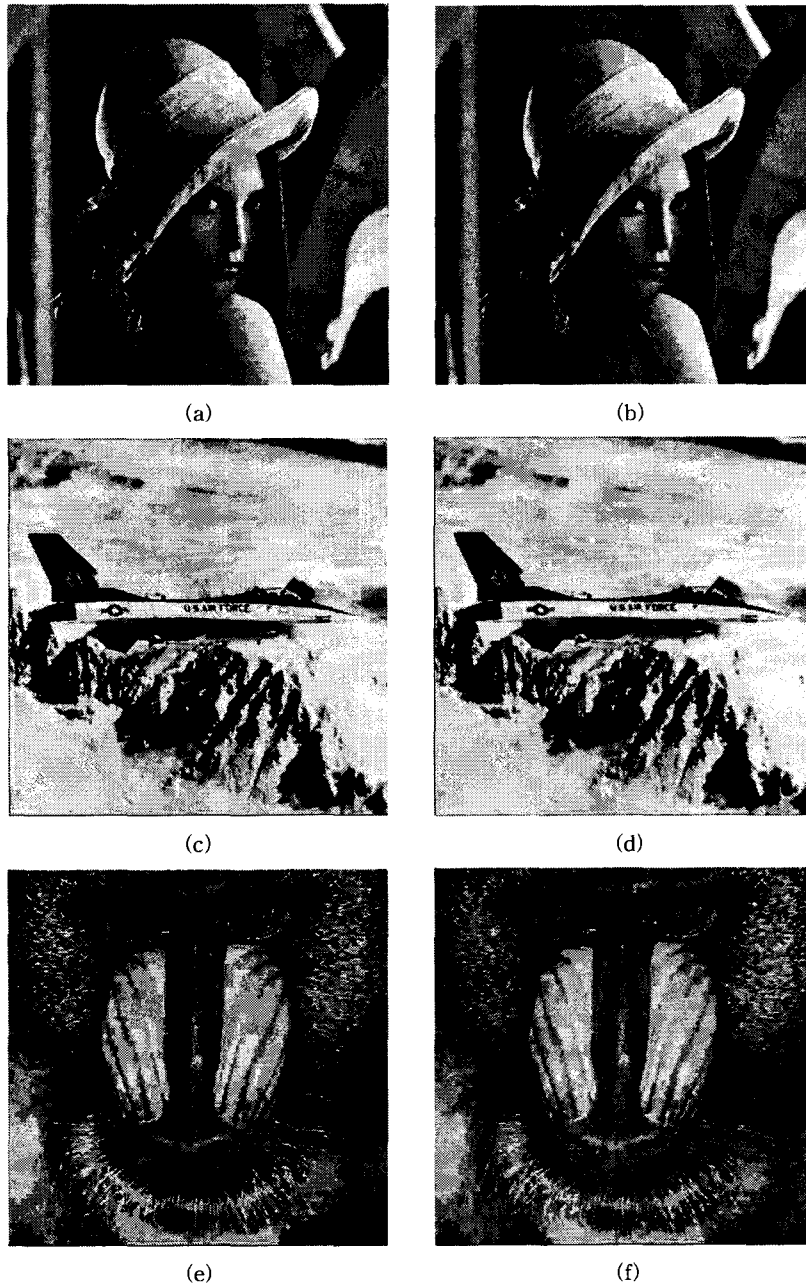


그림 7. 비트율 0.25bpp에서의 시험 영상(a: Lena, c: plane, e: baboon, g: boat, i: pepper)과 출력 영상(b, d, f, h, j)

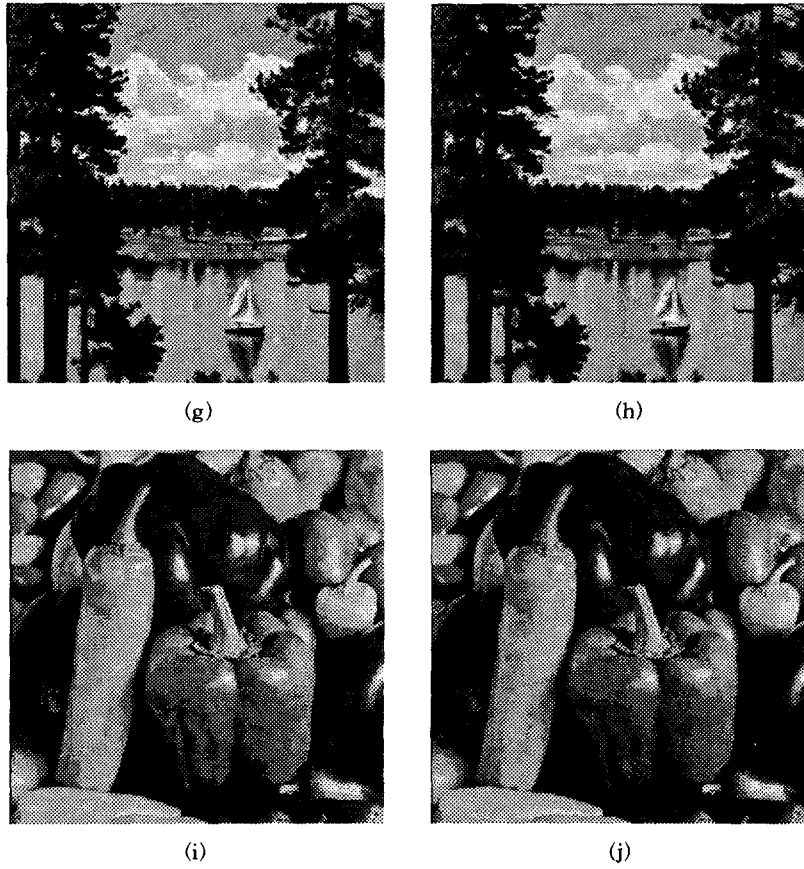


그림 7. 계속

서도 블록화 현상이 급격히 감소되어 모자의 무늬가 시각적으로 쉽게 인식될 수 있는 것을 볼 수 있다. 이상의 결과로부터 본 논문에서 설계된 회로가 정상적으로 동작함을 알 수 있다.

5. 결 론

본 논문에서는 최근 제안된 B 스플라인 곡선을 이용한 블록화 현상 감소 알고리즘을 회로로 설계하고

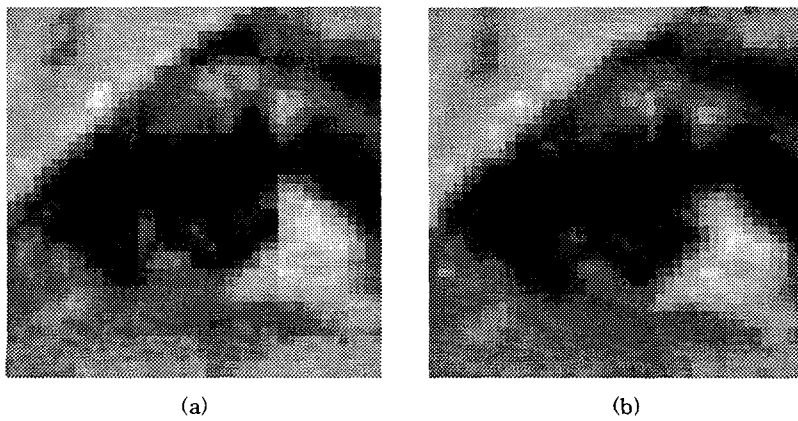
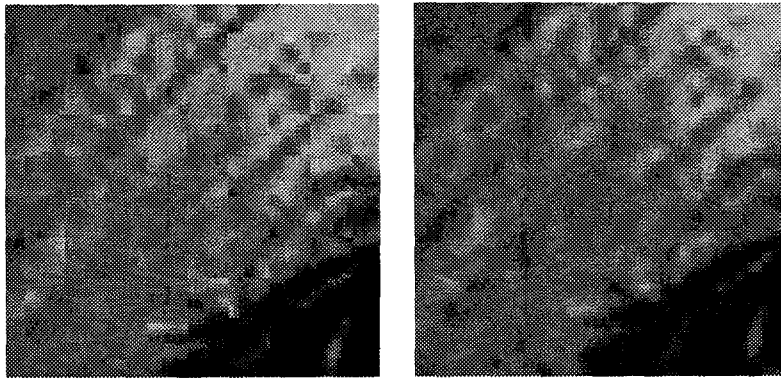


그림 8. Lena 영상에서 눈 부분의 확대 영상 (a) 시험 영상 (b) 출력 영상



(a) (b)

그림 9. Lena 영상에서 입 부분의 확대 영상 (a) 시험 영상 (b) 출력 영상



(a) (b)

그림 10. Lena 영상에서 모자 부분의 확대 영상 (a) 시험 영상 (b) 출력 영상

구현하였다. 설계 시 하드웨어 기술 언어인 VHDL을 이용하였으며, 설계 툴은 synopsys를 사용하였다. 설계된 회로에서는 메모리에 저장된 원 영상 데이터에 대해 먼저 수직 방향의 블록화 현상 감소 알고리즘을 적용한 후 그 결과를 메모리에 저장한다. 이어서 메모리의 중간 결과 영상에 대해 수평 방향의 블록화 현상 감소 알고리즘을 적용하여 생성된 최종 결과 영상을 메모리에 저장한 후, 이를 파일 형식으로 출력한다. 수직 및 수평 처리에 있어서 동일한 회로가 사용되며, 가중치와 알고리즘 적용을 위한 다수의 산술/논리 연산회로들을 구성함으로써 병렬 계산이 가능하여 처리 시간이 단축되도록 하였다. 모의실험을 통해 본 논문에서 설계된 회로가 정상적으로 동작하여 블록화 현상을 효과적으로 제거할 수 있음을 알 수 있었다.

참 고 문 헌

- [1] Jae S. Lim, *Two-Dimensional Signal And Image Processing*, Prentice-Hall, Inc., 1990.
- [2] Y. Zhang, R. L. Pickholtz, and N. H. Loew, "A New Approach to Reduce the Blocking Effect of Transform Coding," *IEEE Trans. Comm.*, Vol. 41, No. 2, pp.299-302, Feb. 1993.
- [3] H. S. Malvar and D. H. Staelin, "The LOT: Transform Coding Without Blocking Effects," *IEEE Trans. Acoust., Speech, Sig. Proc.*, Vol.37, No.4, pp.553-559, Apr. 1994.
- [4] T. Jarske, P. Haavisto, and I. Defee, "Postfiltering Methods for Reducing Blocking Effects," *IEEE Trans. Acoust., Speech, Sig. Proc.*, Vol.40, No.3, pp.521-526, Aug. 1994.

[5] 김희정, 김지홍, "유리 개방형 균일 B 스플라인 곡선을 이용한 블록 효과 감소, 한국멀티미디어 학회 논문지," 제 5권 4호, pp.386-392, 2002. 8.

[6] D. Hearn and M. P. Baker, *Computer Graphics*, Prentice-Hall, Inc., 1997.

[7] 김영철, 정연모, 조중휘, 홍윤식, 디지털 시스템 설계를 위한 VHDL, 홍릉과학출판사, 1998.

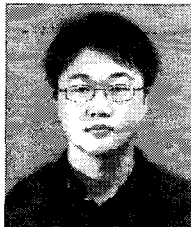
[8] 조경순, 조중휘, VHDL 이론과 실습, 반도체설계교육센터, 1999.

[9] A. S. Al-Fohoum and A. M. Reza, "Combined Edge Crispiness and Statistical Differencing for Deblocking JPEG Compressed Images," *IEEE Trans. Image Proc.*, Vol. 10, pp.1288-1298, Sep. 2001.

[10] T. Meier, K. N. Ngan, and G. Crebbin, "Reduction of Blocking Artifacts in Image and Video Coding," *IEEE Trans. Cir. Sys. for Vid. Tech.*, Vol. 9, pp.490-500, Apr. 1999.

[11] H. Choi and T. Kim, "Blocking-artifact Reduction in Block-coded Images Using Wavelet-based Subband Decomposition," *IEEE Trans. Cir. Sys. for Vid. Tech.*, Vol. 10, pp. 801-805, Aug. 2000.

[12] G. C. Stockman and L. G. Shapiro, *Computer Vision*, Prentice-Hall, Inc., 2001.



박 성 모

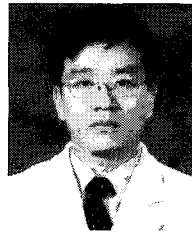
1998년 2월 부산외국어대학교 컴퓨터공학과(공학사)
 1998년 9월~현재 부산외국어대학교 컴퓨터전자공학과 박사과정 재학중

관심분야: VLSI 설계, 컴퓨터그래픽스



김 희 정

1996년 2월 부산외국어대학교 컴퓨터공학과(공학사)
 1999년 8월 부산외국어대학교 교육대학원 전산교육전공(교육학석사)
 2001년 3월~현재 부산외국어대학교 컴퓨터전자공학과 박사과정 재학중
 관심분야: 컴퓨터그래픽스, 컴퓨터비전, 컴퓨터에니메이션



최 진 호

1985년 2월 부산대학교 전자공학과(공학사)
 1987년 2월 한국과학기술원 전기 및 전자공학과(공학석사)
 1992년 2월 한국과학기술원 전기 및 전자공학과(공학박사)

1992년 2월~1996년 2월 현대전자 메모리연구소
 1996년 3월~현재 부산외국어대학교 컴퓨터공학부 부교수
 관심분야: VLSI 설계



김 지 홍

1986년 2월 경북대학교 전자공학과(공학사)
 1988년 2월 경북대학교 대학원 전자공학과(공학석사)
 1996년 8월 포항공과대학교 대학원 전자전기공학과(공학박사)

1988년 2월~1997년 2월 한국전자통신연구원 선임연구원
 1997년 3월~2002년 2월 부산외국어대학교 컴퓨터전자공학부 조교수
 2002년 3월~현재 동의대학교 컴퓨터영상공학부 조교수
 관심분야: 영상통신, 컴퓨터그래픽스, 컴퓨터비전

교신저자

최 진 호 608-738 부산시 남구 우암동 55-1번지 부산외국어대학교 컴퓨터공학부