

論文98-35C-1-3

Template Matching을 위한 새로운 알고리즘 및 ASIC 칩 구현 (A New Template Matching Algorithm and its ASIC Chip Implementation)

徐承完*, 鮮于明勳*

(Seung Wan Seo and Myung Hoon Sunwoo)

요 약

본 논문은 실시간 영상처리를 위한 새로운 template matching 알고리즘과 이를 이용한 병렬처리 방식의 고속 ASIC 칩을 구현한다. Template matching 처리시 CC 및 SAD 방식은 많은 게이트 카운트와 연산량의 부담을 가진다. 이러한 하드웨어 부담과 연산량을 줄이기 위하여 양자화 처리 및 사전처리 방식 등을 이용하였는데, 이 방식들 중 MPPM 방식은 양자화 오류 및 데이터 정확도의 한계를 가진다. 따라서 본 논문은 MPPM 알고리즘을 이용한 template matching 처리에 비하여 5% 에러의 부가적 백색 가우시안 잡음(AWGN) 환경에서 22% 이상의 정확도를 개선한 EMPPM 알고리즘을 제안한다. 제안된 병렬처리 하드웨어 구조는 8 비트 SAD 연산을 사용하는 것에 비하여 60% 이상의 게이트를 감소시켰으며, MPPM 방식을 이용한 Ranganathan & Venugopal 처리구조 보다 사전 덧셈에 의해 28%의 게이트를 감소시켰다. 제안된 병렬처리 방식 ASIC 칩은 VHDL 모델링 후 SYNOPSIS™ tool로 논리합성 구현하였다. 논리합성은 삼성 0.6 μ m SOG 셀 라이브러리를 사용하였으며 CADANCE™ tool을 이용하여 타이밍 시뮬레이션을 수행하였다. 구현된 프로세서 칩은 16 \times 16 기준영상을 처리하며 총 게이트 수는 35,827개이다. 칩의 최대동작 주파수는 100 MHz(worst case 53 MHz)이며 초당 200 Mpixels/sec로 동작한다.

Abstract

This paper proposes a new template matching algorithm and its chip design. The CC and SAD algorithms require the massive amount of computation. Hence, several algorithms using quantization schemes have been proposed to reduce the amount of computation and its hardware cost. The proposed algorithm called the EMPPM improves at least 22% of the noise margin compared with the MPPM algorithm. In addition, the proposed architecture can reduce the gate count by more than 60% of that used in the SAD algorithm without using quantization schemes and 28% of the MPPM algorithm. The VHDL models have been simulated by using the CADANCE™ and logic synthesis has been performed by using the SYNOPSIS™ with 0.6 μ m SOG(sea-of-gate) cell library. The implemented chip consists of 35,827 gates, operates at 100 MHz(worst case 53 MHz) and performs the template matching with the speed of 200 Mpixels/sec.

I. 서 론

Template matching은 탐색영상(search image)에

서 찾고자 하는 기준영상을 전후좌우로 이동시키면서 상관성을 계산하여 최대 상관값을 갖는 위치를 찾는 것이다. 이러한 template matching은 영상 통신에 사용되는 동화상 압축, 지문 감지, 문자 감지 등 각종 화상의 유사성을 인지하는 패턴 인식, 공장 자동화에 사용되는 물체 감지, 그밖에 영상등록(image regist-

* 正會員, 亞州大學校 電子工學府

(School of Electronic Eng., Ajou Univ.)

接受日字:1997年10月27日, 수정완료일:1998年1月5日

ration) 등에 사용되고 있다^[1-6]. Template matching은 연산량이 방대하여 실시간 처리를 위해서는 하드웨어 구현이 필수적이다. 이러한 template matching 알고리즘에는 크게 CC(cross correlation) 방식과 SAD(sum of absolute difference) 방식이 있다. CC 방식은 곱셈 연산을 사용하여 정확한 정합을 구할 수 있으나 많은 곱셈기를 사용하므로 하드웨어 크기가 매우 크며 복잡한 산술연산으로 인해 연산량이 많다. 이에 반해 SAD 방식은 두 값의 차이를 구하여 곱셈기 대신 뺄셈기와 절대값을 사용하기 때문에 CC 방식보다 하드웨어 크기가 작아지나 여러 비트에 대한 산술연산 때문에 여전히 많은 하드웨어 및 연산량을 요구한다^[4-8].

하드웨어의 크기 및 연산량을 줄이기 위해 low-level feature map^[9, 10], binary map^[11] 방식의 영상 사전처리(image preprocessing)를 하거나 MPPM(Moment Preserving Pattern Matching)^[12] 등과 같이 적은 비트 수로 양자화하여 template matching을 수행하는 알고리즘들이 연구되어 왔다. 그러나, 이러한 알고리즘들은 연산 오차에 의해 잡음에 대한 오류 및 데이터 정확도(accuracy)의 한계를 갖는다.

본 논문에서는 MPPM 알고리즘을 응용하여 정확도 및 잡음에 대한 영향을 개선한 새로운 알고리즘인 EMPPM(Enhanced Moment Preserving Pattern Matching)을 제안하며 이를 적용한 template matching의 아키텍처(architecture)를 제안한다. EMPPM은 MPPM이 2 비트를 3 레벨(level)로 연산하는데 비해 2 비트를 4 레벨로 연산함으로써 5% 에러의 가우시안 랜덤 잡음(gaussian random noise)에 대해 22% 이상의 정확도 개선을 이루었으며 연산블럭의 구조를 개선시킴으로서 direct-form^[4, 5] 아키텍처에 비하여 60%, 행을 먼저 더하는 덧셈기 구조를 이용함으로써 PE 내의 레지스터를 가진 열을 더하는 Ranganathan & Venugopal의 MPPM 아키텍처^[3]에 비해 28%의 게이트를 감소시켰다.

제안하는 template matching 아키텍처는 32×32 기준영상에 대하여 C 프로그램으로 알고리즘의 타당성과 잡음에 대한 영향을 확인하였다. 또한 설계된 16×16 template matching 아키텍처는 VHDL (VHSIC Hardware Description Language)로 모델링하고 SYNOPSISTM tool을 이용하여 논리 합성 수행 후 CADANCETM tool을 이용하여 타이밍 시뮬레이션을

수행하였다. 사용된 셀은 삼성 0.6 μ m SOG(Sea-of-Gate) 셀 라이브러리를 이용하였으며 최대 동작 주파수는 100 MHz이고 16×16 기준영상에 대한 256×256 탐색영상의 처리 시간은 0.332 ms이다. 총 게이트 수는 35,827개이며 처리 속도는 최대 100 MHz이다.

본 논문은 다음과 같이 구성된다. II장에서는 기존의 알고리즘을 분석하고, 제안하는 EMPPM 알고리즘을 가우시안 랜덤 잡음에 대한 영향 분석을 통하여 기존의 MPPM 알고리즘과 비교한다. III장에서는 EMPPM 알고리즘을 바탕으로 구현된 template matching을 위한 병렬 처리 프로세서의 아키텍처를 서술한다. IV장에서는 기존의 아키텍처와의 비교분석을 통한 성능 평가를 기술하며, V장에서는 연구 성과를 기술한다.

II. 새로운 알고리즘의 수식적인 모델

1. 기존의 template matching 알고리즘

Template matching 알고리즘은 식 (1)과 같이 일반화한 식으로 나타낸다.

$$SS(x, y) = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} f(S(x+i, y+j), T(i, j)) \quad (1)$$

$S=(S(x, y): 0 \leq x < N, 0 \leq y < N)$ 는 탐색영상의 범위이고, $T=(T(i, j): 0 \leq i < K, 0 \leq j < K)$ 는 기준영상의 범위이며 f 는 상관성을 찾는 함수이다. 기준영상이 탐색영상 내를 이동하면서 각 위치마다 하나의 출력 $SS(x, y)$ 를 구하므로 기준영상은 탐색영상 내에서 $(N-K+1)^2$ 번 이동한다. 하나의 출력 $SS(x, y)$ 를 구하기 위해 함수 f 는 $S(x, y)$ 부터 $S(x+K-1, y+K-1)$ 까지 기준영상을 찾기 위해 총 K^2 번의 연산을 수행한다. 또한 전체 영상에 대한 연산량은 $K^2(N-K+1)^2$ 이 된다. 따라서 탐색영상이 기준영상에 비해 큰 경우 $K \ll N$ 이고 필요한 연산량의 근사식은 K^2N^2 이 된다^[2, 3].

Template matching 중에 상관성을 찾기 위한 방식으로는 식 (2)의 CC^[1]와 식 (3)의 SAD^[6]가 있다.

$$SS_{cc}(x, y) = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} S(x+i, y+j) \cdot T(i, j) \quad (2)$$

$$SS_{sad}(x, y) = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} |S(x+i, y+j) - T(i, j)| \quad (3)$$

식 (1)에서의 함수 f 가 곱셈인 CC는 일반적인 방식으로 식 (2)처럼 계산되어 정확도가 높은 반면 VLSI 구현시 많은 게이트가 요구되며, SAD 방식은 CC에 비하여 정확도는 낮아지나 함수 f 가 뺄셈과 절댓값이므로 적은 게이트로 구현될 수 있다. 그러나, 두가지 방식 모두 연산량이 방대하여 비트 수가 큰 데이터에 적용한다면 많은 연산처리 시간과 게이트를 필요로 하며, 성능 대 비율을 떨어뜨린다. 따라서 Template matching 알고리즘은 데이터 비트 수를 줄여서 양자화하거나 사전처리한 후 template matching를 수행하는 것이 일반적이는데 이러한 template matching 알고리즘에는 영상 사전처리 방식으로 통계적 방법을 이용한 binary map^[11]과 영상의 기울기(gradient), 경계선(edge) 등을 이용한 low-level feature map^[9, 10], 2 비트로 양자화한 MPPM^[12] 등이 있다. Binary map이나 low-level feature map의 경우 계산의 복잡성이나 하드웨어 비용은 줄일 수 있으나 탐색영상이 커질수록 잡음에 약한 단점과 특히 low-level feature map의 경우 사전처리를 위한 계산적 부담을 가지게 된다^[9-12].

이러한 단점들을 개선하고 각각의 장점을 취한 알고리즘이 MPPM이다^[12]. MPPM은 한 비트로 양자화된 모멘트 보존 양자화(moment preserving quantization) 함수 2개로 이루어진 쌍함수(pairing function)를 구한 후 CC를 수행 정합점을 찾는다. 양자화는 기준영상의 상위의 범위와 하위의 범위를 정하고 상위의 범위 보다 높은 값과 하위의 범위보다 낮은 값이 서로 같은 '00'을 갖도록 한다. 또한 평균을 기준으로 상·하위 범위의 사이에 존재하는 값은 '01', '10'으로 양자화하며 평균값은 '11'로 한다. 이때 각 비트는 독립적이기 때문에 '00'은 0, '10'과 '01'은 1, '11'은 2를 의미한다. 따라서 2비트의 양자화된 데이터로 CC를 수행할 때 3레벨 연산을 수행함으로 정확한 데이터 값을 추출하는 것에 장애 요인이 되며, 연산 코드(code) 활용율이 낮은 단점이 있다.

따라서 본 논문에서는 MPPM 알고리즘을 개선한 새로운 알고리즘 EMPPM을 제안하고 EMPPM을 바탕으로 고속의 병렬처리구조를 이용한 아키텍처를 제안한다. EMPPM은 VLSI 구현시 적은 량의 게이트를 사용하여 데이터 정확도를 높일 수 있으며 잡음 마진과 동작 속도를 증대시킬 수 있는 장점을 가지고 있다.

2. 제안하는 EMPPM 알고리즘

기존의 MPPM 방식은 2비트 데이터를 3레벨로 연산하므로써 연산 코드 활용율이 낮아 정확도가 떨어지는 단점이 있다. 이를 개선한 EMPPM 알고리즘은 4레벨로 연산하였으며 양자화 방식은 기존의 MPPM의 양자화 방식과 같이 확률분포에 비례한 모멘트 보존 양자화 방식을 사용하였다.

확률분포에 비례한 양자화를 위해 기준영상의 평균과 표준편차를 구해야 한다. $K \times K$ 기준영상에서 256 그레이 레벨(gray level)을 가진 8비트 데이터를 2비트, 4그레이 레벨로 양자화하기 위해 기준영상의 평균과 표준편차를 구하면 식 (4), 식 (5)와 같다. 식 (4)에서 $T(i,j)$ 는 기준영상 i, j 화소의 그레이 레벨이며 따라서 η 는 기준영상의 평균 그레이 레벨 값이다. σ 는 기준영상의 표준편차로서 식 (5)로 표현된다.

$$\eta = \frac{1}{K^2} \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} T(i,j) \tag{4}$$

$$\sigma = \left[\frac{1}{K^2} \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} T(i,j)^2 - \eta^2 \right]^{1/2} \tag{5}$$

양자화의 상위 레벨의 경계결정과 하위 레벨의 경계결정을 위한 파라미터 u, l 을 정의하였다. u 는 상위 레벨의 경계결정에 사용되는 값으로 식 (6)과 같이 정의되며 l 은 하위 레벨의 경계결정에 사용되는 값으로 식 (7)과 같이 정의된다.

$$u = \sigma \cdot \left(\frac{K^2 - q}{q} \right)^{1/2} \text{ for } T(i,j) \geq \eta \tag{6}$$

$$l = \sigma \cdot \left(\frac{q}{K^2 - q} \right)^{1/2} \text{ for } T(i,j) \leq \eta \tag{7}$$

식 (6)과 식 (7)에서의 q 는 기준영상에서 평균 그레이 레벨 즉 η 보다 큰 그레이 레벨 값을 갖는 화소의 개수이다.

u 와 l 을 이용한 8비트 256레벨을 2비트 4레벨로 양자화 및 부호화 하는 과정은 식 (8)과 같다.

$$T_q(i,j) = \begin{cases} a_3, & \text{if } \eta + u \times r \leq T(i,j) \\ a_2, & \text{if } \eta \leq T(i,j) < \eta + u \times r \\ a_1, & \text{if } \eta - l \times r \leq T(i,j) < \eta \\ a_0, & \text{if } 0 \leq T(i,j) < \eta - l \times r \end{cases} \tag{8}$$

u 와 l 은 (6), (7)에서 알 수 있듯이 평균 그레이 레벨

η 보다 큰 화소의 개수가 많을 수록 u 값이 작아지고 l 값은 커진다. 따라서 상위 양자화 레벨 $\eta+u \times r$ 가 평균에 가까워지며 하위 양자화 레벨 $\eta-l \times r$ 는 평균과의 차이가 많이 나게 되어 4 레벨로 양자화할 때 각 레벨에 분포하는 화소수를 균일하게 맞추어 준다. 식 (8)에서 r 값은 범위 조정값(scale factor)으로서 일반적으로 2이며 표준편차가 커 상위 또는 하위 양자화 레벨이 8 비트 그레이 레벨 표현 범위를 넘어설 경우 1로 조정함으로써 보다 정확한 값으로 양자화 되게 한다. EMPPM 알고리즘은 두 비트로 양자화된 데이터는 4개의 레벨로 연산처리 함으로서 같은 2 비트를 사용하여 3개 레벨로 연산처리하는 MPPM 방식에 비하여 부호화 효율과 잡음 마진율을 증대시켜 준다. a_3, a_2, a_1, a_0 는 각각 2 비트의 양자화된 데이터이며 $T_q(i,j)$ 의 값을 영역별로 나타낸다.

Template matching은 기준영상과 탐색영상의 상관성을 계산하므로 탐색영상의 양자화도 기준영상의 양자화 레벨을 따르게 된다. 따라서 탐색영상의 양자화는 식 (9)와 같다.

$$S_q(i,j) = \begin{cases} a_3, & \text{if } \eta+u \times r \leq S(i,j) \\ a_2, & \text{if } \eta \leq S(i,j) < \eta+u \times r \\ a_1, & \text{if } \eta-l \times r \leq S(i,j) < \eta \\ a_0, & \text{if } 0 \leq S(i,j) < \eta-l \times r \end{cases} \quad (9)$$

식 (9)에서의 S_q 데이터는 기준영상으로 부터 구한 양자화 레벨을 사용하므로 사전처리 연산량이 줄어든다. 양자화된 새로운 기준영상 T_q 와 탐색영상 S_q 에 대해서 SAD를 적용하면 식 (10)과 같다.

$$QSAD(x,y) = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} |T_q(i,j) - S_q(x+i,y+j)| \quad (10)$$

하나의 $QSAD(x,y)$ 값을 연산하기 위해 K^2 번의 SAD 연산과 이의 누적 연산이 필요하며, $N \times N$ 영상에 대하여 $(N-K+1)^2$ 번의 QSAD 연산을 반복 수행한다. 연산된 결과값 중 가장 작은 값을 갖는 부분의 영상이 가장 높은 상관성을 가지게 되며, 이때 EMPPM 알고리즘의 QSAD 연산은 양자화된 두 비트의 데이터를 4개의 레벨을 갖는 2진수로 인식하여 연산함으로써 2 비트의 데이터 중 1의 개수로 최대 2의 값을 가지는 3개의 레벨로 인식하는 MPPM의 CC 연산에 비하여 정확도가 개선된다. 또한 SAD 연산을 2 비트로 수행하여 고속 연산처리를 할 수 있고 하드

웨어의 크기도 줄일 수 있다.

3. EMPPM과 MPPM 알고리즘의 잡음에 대한 성능 비교

본 절에서는 기존의 MPPM 알고리즘과 제안한 EMPPM 알고리즘의 시뮬레이션 결과를 제시하고 잡음에 대한 영향을 비교 분석함으로써 제안한 EMPPM 알고리즘의 성능 개선을 확인한다. 두 알고리즘의 성능 비교 분석을 위하여 실험에서는 그림 1(a)의 256×256 크기와 256 그레이 레벨을 갖는 도시 그림을 탐색영상으로 한다. 탐색영상은 그림 1(b)의 히스토그램을 가지며 평균 142, 표준편차 49를 갖는다. 잡음 영향을 조사하기 위하여 탐색영상 그림 1에서 (181,181), (171,105) 위치의 그림 2(a)와 (b)에 나타난 32×32 영상 T1, T2를 기준영상으로 추출하였다. 그림 2(c)와 (d)는 그림 2(a)와 (b)의 T1, T2 히스토그램 분포를 나타낸 것으로 신빙성 있는 데이터 값을 구하고자 다양한 분포를 가지도록 표본화한 것이다. T1의 경우 평균이 103, 분산이 51이며, T2의 경우 평균이 120, 분산이 60이다.

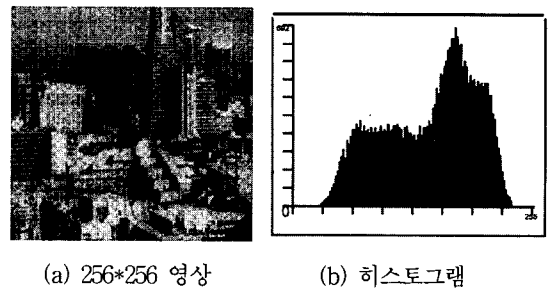


그림 1. 256×256 크기의 탐색영상

Fig. 1. The search image of 256×256 size.

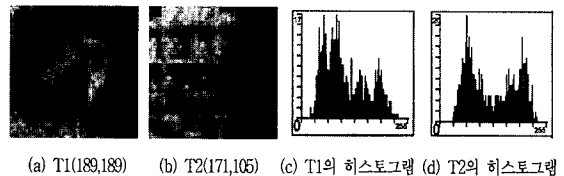


그림 2. 탐색영상에서 추출한 2개의 기준영상(32×32)과 히스토그램

Fig. 2. Two template images(32×32) sampled from searching image and their histograms.

그림 3, 4, 5, 6은 가로축 값의 AWGN 분산을 지닌 잡음을 탐색영상에 EMPPM, MPPM, 8 비트 SAD 알고리즘을 적용한 template matching 연산을

100번씩 반복 수행한 후 에러가 발생할 확률 및 원위치와의 거리차를 비교한 그래프이다. 이때 그래프에서 에러가 발생할 확률은 잡음의 영향에 의하여 탐색영상에서 기준영상의 원위치를 찾지 못하는 에러의 개수를 나타내며, 이는 적을 수록 유리하다. 원위치와의 거리차는 잡음의 영향에 의하여 탐색영상에서 기준영상의 원위치를 찾지 못하고 원위치에서 벗어난 거리의 평균을 나타낸다. 이때 사용된 프로그램 언어는 C 언어이다.

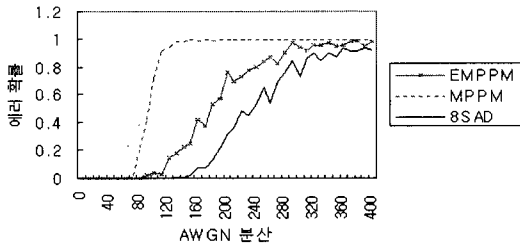


그림 3. T1 위치 기준영상의 에러확률 분석
Fig. 3. The error probability analysis of T1 template image.

그림 3은 T1의 기준영상을 가진 데이터에 대한 잡음 분산과 에러확률에 대한 도표이다. 5% 에러 확률을 갖는 AWGN 분산값은 EMPPM의 경우 120, MPPM의 경우 80, 8 비트 SAD의 경우 160이다. 이와 같이 EMPPM은 MPPM에 비해 AWGN 비율로 33% 이상 정확도가 개선된다. 또한 8 비트 SAD와 비교하면 정확도는 떨어지나 하드웨어 구성시 EMPPM의 경우 2 비트의 양자화된 데이터를 사용하므로 8 비트 SAD를 사용하는 것에 비하여 연산속도 및 하드웨어 비용을 대폭 줄일 수 있다.

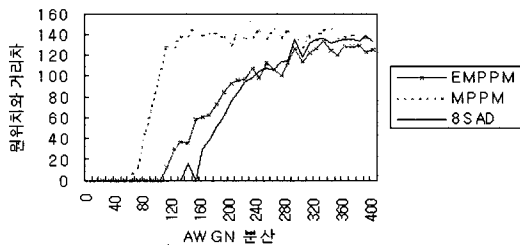


그림 4. T1 위치 기준영상의 원위치와 거리분석
Fig. 4. The true position analysis of T1 template image.

그림 4는 T1의 기준영상을 가진 데이터에 대하여 잡음 분산과 원위치와의 거리 분석 도표이다. 정확한

위치로 부터 거리가 벗어나기 시작한 지역은 EMPPM의 경우 90, MPPM의 경우 70, 8 비트 SAD의 경우 130의 잡음 분산을 지낸다. 그림 5는 T2의 기준영상을 가진 데이터에 대하여 잡음 분산과 에러확률에 대한 도표이다. 5% 에러 확률을 갖는 AWGN의 분산 값은 EMPPM의 경우 90, MPPM의 경우 70, 8 비트 SAD의 경우 150이다. 그림 6은 T2의 기준영상을 가진 데이터에 대하여 잡음 분산과 원위치와의 거리 분석 도표이다. 정확한 위치로 부터 거리가 벗어나기 시작한 지역은 EMPPM 70, MPPM 60, 8 비트 SAD 110의 잡음 분산을 지낸다.

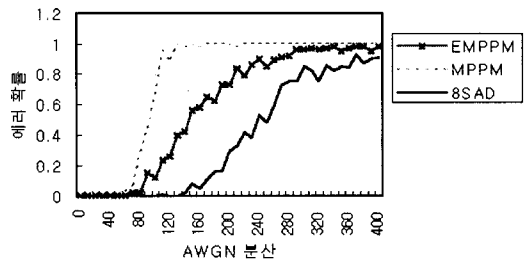


그림 5. T2 위치 기준영상의 에러확률 분석
Fig. 5. The error probability analysis of T2 template image.

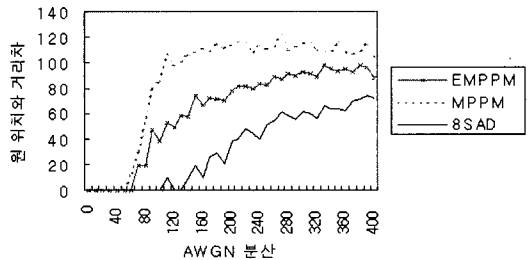


그림 6. T2 위치 기준영상의 원위치와 거리분석
Fig. 6. The true position analysis of T2 template image.

그림 3, 4, 5, 6을 살펴보면 EMPPM은 MPPM보다 8 비트 SAD에 더 근접하여 나타난다. 또한 기준영상의 분산이 적을 수록 8 비트 SAD에 근접함을 알 수 있다. 위의 실험을 결과로 EMPPM 알고리즘이 MPPM 알고리즘에 비하여 부가적 잡음의 허용도가 최소 22%에서 최대 33% 개선됨을 확인하였다. 이는 MPPM의 2 비트를 3 레벨로 연산처리하는 CC방법 대신 EMPPM은 2 비트를 4 레벨로 연산처리하는 SAD를 이용함으로써 데이터 정확도가 향상되기 때문이다. 비록 EMPPM이 8 비트 SAD를 사용한 것 보

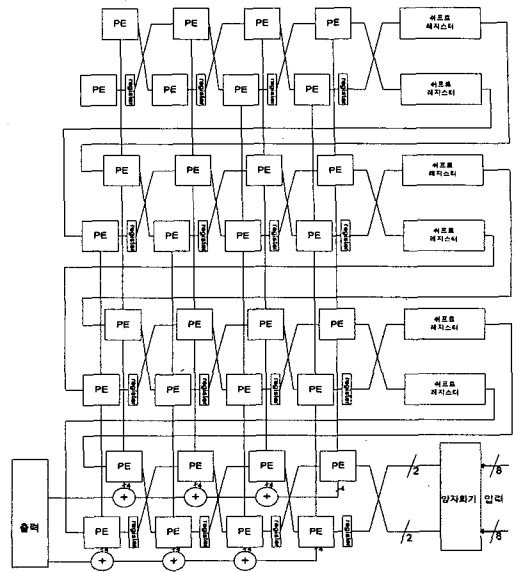
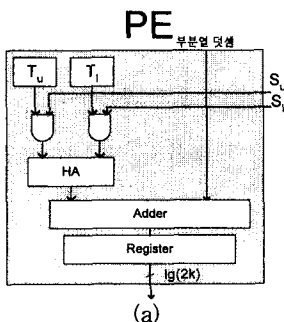
다 부가적 잡음의 허용도가 떨어지나 8 비트를 2 비트로 사용함으로 하드웨어의 크기를 줄일 수 있으며 연산속도를 개선시킬 수 있다.

III. EMPPM 알고리즘을 이용한 Template Matching 아키텍처

본 장에서는 template matching 아키텍처를 4×4 기준영상에 적용하여 설명한다. 일반적으로 중복되지 않는 두개의 SAD 값을 구하기 위해서는 탐색영역을 두 부분으로 나누거나 탐색영역의 프레임(frame)들을 두 그룹으로 나누어야 하며 그에 따라 사전 처리과정과 저장 매체가 요구된다. 제안된 아키텍처는 Ranganathan & Venugopal의 구조를 이용하여 기준영상에 대한 사전처리가 요구되나 탐색영상에 대한 저장 매체나 사전처리 과정 없이 실시간으로 입력되는 탐색영상의 홀수 화소와 짝수 화소를 동시에 처리하는 방식으로 중복되지 않는 2개의 QSAD 값을 연산하는 장점을 가지고 있다.

1. 기존의 MPPM을 이용한 하드웨어 구조

제안하는 EMPPM 구조와 기존의 MPPM 구조의 비교를 위해 먼저 MPPM 알고리즘을 이용한 하드웨어 구조에 대해 살펴본다. 그림 7은 MPPM 알고리즘을 이용한 Ranganathan & Venugopal^[3]이 제안한 하드웨어 구조이다. 이 구조는 두 개의 입력으로 동시에 두개의 template matching 연산을 한다. 그림 7(a)는 PE 구조를 나타낸다. PE는 기준영상에 대한 데이터 값을 가진 T_u , T_l 레지스터, 입력되는 탐색영상 데이터와 레지스터에 저장된 기준영상 데이터에 대한 2 비트 MPPM 연산을 수행하는 HA(Half Adder)와 AND 게이트로 구성된다. 또한 열 방향으로 연결된 PE들의 데이터를 더하는 덧셈기(Adder)와 레지스터가 추가된다.



(b)

그림 7. Venugopal이 제안한 하드웨어

(a) PE 구조, (b) 연산 블록 구조

Fig. 7. The Venugopal's system architecture.

(a) PE organization, (b) Processing block architectures

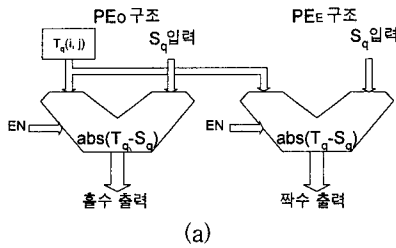
그림 7(b)는 연산블럭 구조를 나타낸다. 입력은 탐색영상의 연속된 2개의 화소가 양자화기를 거쳐 2개의 2 비트 데이터로 양자화된다. 입력 데이터는 쉬프트 레지스터를 통하여 template matching 연산을 위한 열과 행을 맞춘다. 열로 연결된 PE들은 고속 연산을 위하여 파이프라인 구조로 구성한다. 파이프라인 구조를 위해 대략적으로 3 비트 덧셈기 32개와 3비트 레지스터 32개가 필요하다. PE들이 파이프라인 구조로 연결되어 있으므로 임계경로는 그림 7 (b)의 연산블럭 구조 하단에 있는 3개의 4 비트 덧셈기들이다. PE 연결은 임계경로가 아니기 때문에 열 방향으로 PE마다 레지스터로 파이프라인 시킬 필요가 없으며 파이프라인을 위해 사용된 덧셈기도 PE 내에 한개씩 가질 필요가 없다. 또한 MPPM의 2 비트를 3 레벨로 연산처리하는 CC방법을 사용하여 정확도가 떨어진다.

2. 제안한 EMPPM을 이용한 하드웨어 구조

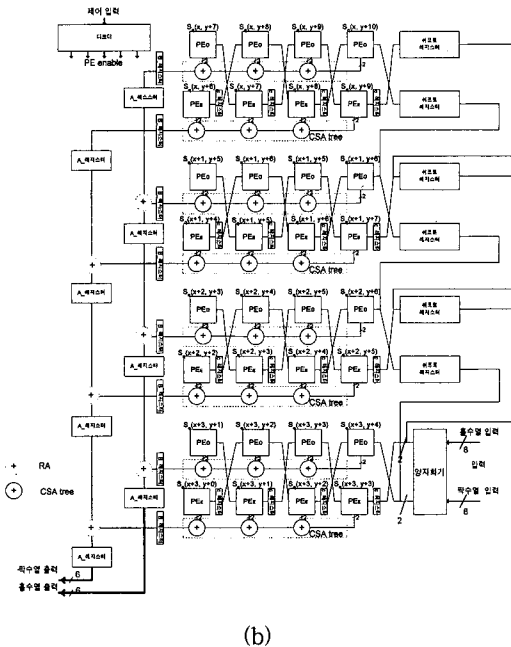
MPPM 방식의 단점을 줄이기 위해 본 논문에서는 EMPPM 알고리즘을 이용한 새로운 아키텍처를 제안한다. 그림 8은 template matching 연산의 정확도를 높이면서 하드웨어 부담을 줄이기 위해 제안된 아키텍

처이다. 그림 8 (a)는 PE 구조들로 짝수열 출력을 위한 PE_E와 홀수열 출력을 위한 PE_O는 2비트 SAD 연산을 한다. $T_q(i,j)$ 레지스터에 저장된 기준영상에 대한 데이터와 계속 입력되는 탐색영상 데이터 S_q 는 $abs(T_q - S_q)$ 를 통하여 $|T_q(i,j) - S_q(x+i,y+j)|$ 연산을 수행한다. 기준영상의 크기를 조절하는 EN 신호가 PE의 enable을 결정한다. 그림 8 (b)는 제안하는 연산 블럭 구조이다. 제안하는 연산 블럭 구조는 양자화기, 32개의 PE(16개 PE_O와 16개 PE_E), 8개의 CSA(Carry Save Adder) tree, 쉬프트 레지스터들로 구성된다.

$\times r$ 값을 저장하고 입력되는 기준영상이나 탐색영상 8 비트 데이터를 4 레벨의 2 비트로 양자화한다. 기준영상의 8 비트 데이터 2개가 짝수열 화소 $T(i,2n)$ 과 홀수열 화소 $T(i,2n+1)$ 이 한 쌍이 되어 동시에 양자화기로 입력된다. 입력된 데이터는 양자화기를 통해 식 (8)에 나타난 $T_q(i,j)$ 의 2 비트 데이터가 된다. 기준영상의 짝수열 화소와 홀수열 화소가 동시에 입력되므로 $T_q(i,2n), T_q(i,2n+1)$ 이 동시에 계산된다. 계산된 $T_q(i,2n), T_q(i,2n+1)$ 데이터는 PE 어레이를 통해 각 PE_O의 $T_q(i,j)$ 레지스터에 저장된다. 기준영상이 모두 입력되면 탐색영상이 입력된다.



탐색영상에 대한 8 비트 데이터 2개가 짝수열 화소 $S(x,2n)$ 과 홀수열 화소 $S(x,2n+1)$ 이 한 쌍이 되어 동시에 양자화기에 입력된다. 양자화기는 기준영상에 대하여 구해진 $\eta + u \times r, \eta - u \times l, \eta$ 값을 그대로 탐색영상에 이용한다. 입력된 데이터는 양자화기를 통해 식 (9)에 나타난 $S_q(i,j)$ 의 2 비트 데이터가 된다. 짝수열 화소와 홀수열 화소가 동시에 입력되므로 $S_q(x,2n), S_q(x,2n+1)$ 이 동시에 계산된다. 양자화된 데이터는 지그재그(Zig-Zag) 방식으로 연결된 PE_E와 PE_O 및 쉬프트 레지스터로 입력된다. 입력되는 데이터는 쉬프트 레지스터를 통해 열과 행이 맞추어져 모든 PE에 입력된다. 이때 쉬프트 레지스터는 파이프라인 초기화($\log_2 K^2$)를 줄이기 위해 한 열을 $(N/2 - 1)$ 개로 구성한다. 16개의 PE_E는 $|T_q(i,j) - S_q(x+i,y+j)|$ 값을 구하며 16개의 PE_O는 $|T_q(i,j) - S_q(x+i,y+1+j)|$ 값을 구하게 된다.



각 PE에서 구해진 2 비트 SAD 값을 합산하여 식 (10)과 같은 QSAD(x,y) 값을 구하기 위해 CSA tree와 RA(Ripple Adder)를 이용한다. Ranganathan & Venugopal 구조에서는 열을 더하는 파이프라인을 이용하여 모든 PE내에 덧셈기와 불필요한 레지스터의 사용으로 하드웨어 부담을 많이 가지게 한다. 이를 본 구조에서는 임계경로를 고려하여 행을 먼저 더하는 방식을 사용한다. 하나의 행에 고속의 데이터 처리와 하드웨어 크기를 줄이기 위하여 CSA tree를 사용한다. 이때 CSA tree는 홀수행과 짝수행의 덧셈을 위해 그림 8 (b)와 같이 PE_O와 PE_E 출력에 각각 위치한다. 따라서 Ranganathan & Venugopal 구조에서 사용하는 덧셈기 32개와 레지스터 32개를 제안된 구조에서는 행을 더하는 CSA tree를 사용하므로써 3 비트 덧셈기(CSA tree) 8개, RA 4 비트 덧셈

그림 8. 제안된 아키텍처

(a) PE 구조들, (b) 제안하는 연산 블럭 구조

Fig. 8. The proposed architecture.

(a) PE organizations, (b) Proposed processing block architectures

본 구조의 데이터 흐름은 먼저 양자화기를 통하여 호스트에서 처리한 세개의 경계값 $\eta, \eta + u \times r, \eta - l$

기 6개, 4 비트 레지스터 16개로 줄일 수 있다. 또한 본 구조는 정확도를 개선을 위해 MPPM의 2 비트를 3 레벨로 연산처리하는 CC 방법을 사용하는 대신 EMPPM의 2 비트를 4 레벨로 연산처리하는 SAD 방법을 사용하였다. 제안된 구조는 동시에 2개의 입력을 처리하므로 짝수열 출력 $QSAD(x, 2n)$ 과 홀수열 출력 $QSAD(x, 2n+1)$ 을 동시에 출력한다.

외부 제어 입력은 디코더를 통해 기준영상의 크기를 가변시킬 수 있다. 따라서 4×4 는 기본으로 지원하며 3×3 이하의 기준영상을 처리할 수 있다. 또한 여러개의 칩을 사용하여 5×5 이상의 기준영상도 처리할 수 있다. 만일 기준영상의 크기가 줄어들면 사용하지 않는 PE 내의 $T_q(i, j)$ 레지스터에 0이 입력되고 EN 신호가 0이 되어 2 비트 SAD 출력값은 0이 출력된다. 따라서 최종출력 값에는 0이 더해지므로 영향을 주지 않는다. 탐색영상 전체에 대해 위의 연산을 $(N-K+1) \times (N-K+1)/2$ 번 반복한 후 최소값을 가지는 위치 정보를 찾는다. Template matching은 연산 블록구조에서 $N^2/2$ 의 연산 시간이 소요된다.

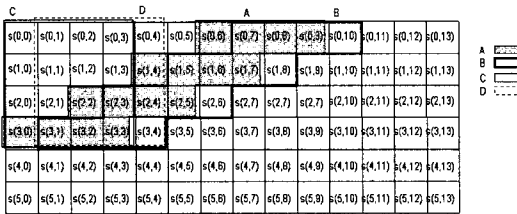


그림 9. 전체 탐색영상에서 형성되는 영상 윈도우 A, B와 덧셈기 배열 블록에서 연산되는 영상 윈도우 C, D

Fig. 9. The image window A, B formed in search image, The image window C, D processed in adder array block.

예를 들어 그림 8 (b)의 PE 위의 x와 y에 0을 대입한다. 하나의 쉬프트 레지스터 열이 $(N/2-1)$ 개로 구성되어 있으므로 16개의 PE_E은 그림 9의 A와 같은 영상 윈도우를 갖는다. 또한 16개의 PE_O는 그림 9의 B와 같은 영상 윈도우를 갖는다. 그림 9의 영상 윈도우 A는 한 클럭이 지나면 한 행에서 2 칸씩 이동하여 S(0,8), S(0,9), S(0,10), S(0,11), S(1,6), ..., S(3,2), S(3,3), S(3,4), S(3,5)의 영상 윈도우를 가진다. 또한 한 클럭에 2개의 탐색영상 윈도우를 동시에 수행하므로 B의 영상 윈도우도 한 행에서 2 칸씩 오른쪽으로 이동한다. 그림 9의 영상 윈도우는 아래에서

위로 각각의 행이 2 칸씩 늦어진다. 이는 쉬프트 레지스터가 파이프라인 초기화 시간을 줄이기 위해 한행에 $(N/2-1)$ 개로 구성되어 있기 때문이다. 따라서 원래의 template matching 연산을 위한 그림 9의 영상 윈도우 C, D를 얻기 위해서 A 레지스터와 RA를 파이프라인으로 연결한다. 이때 RA와 CSA tree를 그대로 연결하면 임계경로가 되므로 B 레지스터를 사용하여 임계경로를 줄인다. 탐색영상이 기준영상과 template matching 연산을 수행하여 식 (10)의 데이터 $QSAD(0,0)$ 과 $QSAD(0,1)$ 을 동시에 출력함을 나타낸다.

본 하드웨어 구조는 EMPPM 방식을 사용하므로 정확도가 개선되고 direct-form^[4, 5]에 비하여 연산 시간이 적게 걸리며, Ranganathan & Venugopal^[3] 구조의 열을 더하는 덧셈기 대신 본 구조에서는 행을 더하는 덧셈기를 사용하여 레지스터와 덧셈기를 줄일 수 있어 게이트 수를 줄였다. 구현된 칩은 16×16 기준영상을 처리한다. 16×16 기준영상에 대하여 데이터 처리속도는 200 Mpixels/sec로서 256×256 탐색영상의 처리 시간은 0.332 ms로 초당 3,012개의 영상을 처리한다.

IV. 칩 구현 및 성능 평가

그림 10은 구현된 16×16 template matching 칩을 나타낸 것이다. 제안하는 아키텍처는 VHDL로 모델링하고 SYNOPSISTM tool을 이용하여 논리 합성 수행 후 CADANCETM tool을 이용하여 타이밍 시뮬레이션을 수행하였다. 사용된 셀은 삼성 SOG $0.6 \mu\text{m}$ 라이브러리이다. 최대동작 주파수는 100 MHz이고 데이터 처리속도는 200 Mpixels/sec로서 256×256 탐색영상의 처리 시간은 0.332 ms로 초당 3,012개의 16×16 기준영상을 처리한다.

표 1은 8 비트 SAD 연산을 이용한 direct-form^[4, 5]과 Ranganathan & Venugopal^[3]이 제안한 아키텍처, 본 논문에서 제안하는 아키텍처의 성능을 비교 분석한 표이다. Direct-form과 Ranganathan & Venugopal이 제안한 아키텍처의 게이트 수는 4×4 기준영상에 대하여 제안한 아키텍처의 구현방식과 같은 방식으로 COMPASSTM tool을 이용하여 구현한 결과이다. 처리속도의 경우 128×128 의 확장된 기준영상 적용시 direct-form 아키텍처의 경우 21 비트

덧셈기가 임계경로가 되며 데이터 처리속도는 한개의 입력에 한개의 template matching을 처리하므로 80 Mpixels/sec이다. Ranganathan & Venugopal 아키텍처는 15 비트 덧셈기가 임계경로가 되며 2개의 입력에 2개의 template matching을 처리하므로 200 Mpixels/sec이다. 여러개의 칩을 이용하여 확장된 EMPPM 방식은 15 비트 덧셈기가 임계경로가 되며 2개의 입력에 2개의 template matching을 처리하므로 200 Mpixels/sec이다.

Ranganathan & Venugopal 보다 28%의 게이트를 감소시켰다.

V. 결 론

본 논문에서는 고속의 영상 이미지 병렬 처리 프로세서를 제안한다. Template matching은 기준영상과 탐색영상을 비교하여 위치 및 상관성을 찾아내는 것으로 많은 연산량 때문에 고속처리 및 VLSI 구현이 힘들다. 따라서 본 논문에서는 고속의 데이터 처리를 위한 새로운 양자화 template matching 알고리즘을 제안하고 그 구조를 설계하여 고속의 이미지 프로세서 칩을 개발하였다.

제안하는 EMPPM 알고리즘은 기존의 MPPM 알고리즘의 2 비트 3 레벨 연산 대신 2 비트 4 레벨 연산을 사용하므로써 5% 에러의 AWGN 분포에 대하여 22% 이상 개선하였다. 이를 바탕으로 제안된 병렬 처리 아키텍처는 동시에 2개의 입력으로 2개의 기준영상을 처리한다. 이는 MPPM 알고리즘을 이용한 Ranganathan & Venugopal의 아키텍처를 EMPPM 알고리즘으로 적용하여 개선하였다. 이때 제안된 아키텍처는 PE 출력 덧셈기를 CSA tree로 사용하여 행을 더함으로써 Ranganathan & Venugopal이 제안한 아키텍처에 비하여 약 28%의 하드웨어를 감소하였다.

설계된 16×16 template matching 칩의 최대 동작 주파수는 100 MHz이며 데이터 처리속도는 200 Mpixels/sec로 8비트 SAD 아키텍처에 비하여 2배 이상의 처리속도를 가진다. 또한, 설계시 확장성을 고려하여 기준영상 및 탐색영상의 변화에 성능이 무관한 특성을 지닌다. 제안된 병렬이미지 프로세서는 top-down IC 설계 방식에 따라 설계되었으며 SYNO-PSYS™ tool을 이용하여 논리 합성 수행후 CADANCE™ tool을 이용하여 타이밍 시뮬레이션을 수행하였다. 구현된 칩은 게이트 수가 35,827개이며 최대동작 주파수는 100 MHz이다.

참 고 문 헌

[1] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing. Addison-Wesley Publishing Company, 1992.

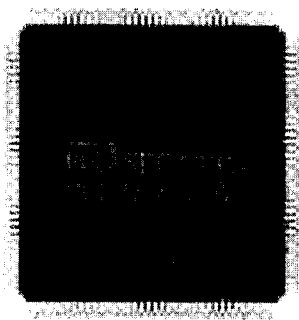


그림 10. 구현된 template matching 칩
Fig. 10. The implemented template matching chip.

표 1. 성능 비교
Table 1. Performance comparison.

항목 \ 아키텍처	Direct-form [4,5]	Ranganathan & Venugopal[3]	제안한 아키텍처	
양자화 방식	없음	MPPM	EMPPM	
연산방식	8비트 SAD	2비트 CC	2비트 SAD	
데이터 처리 속도	80 Mpixel/sec	200 Mpixel/sec	200 Mpixel/sec	
4×4 게이트 수	3900	2110	1650	
N×N search image	초당처리 영상수	$80M / N^2$ (frame/sec)	$200M / N^2$ (frame/sec)	$200M / N^2$ (frame/sec)
	요구되는 행 버퍼	K×N 바이트	$2*(K-1)*(N-K)$ 비트	$2*(K-1)*N$ 비트

따라서, 본 논문에서 제안하는 아키텍처는 데이터 처리속도면에서 direct-form 보다 40% 빠르며 Ranganathan & Venugopal과 같다. 또한 direct-form의 마지막 덧셈 연산을 위한 $\log_2(K^2)$ 의 파이프라인 초기화가 필요치 않다. 게이트 수를 비교할 때 4×4 구현시 1,650개로 Ranganathan & Venugopal 보다 28% 적고 direct-form 보다 60% 줄어든다. 이는 한 클럭(clock)에 2개의 연산을 수행하도록 하여 direct-form 보다 2배 이상의 연산능력을 보유하며, 연산 블록과 덧셈기의 배열을 보다 효과적으로 하여

- [2] S. Venugopal, "A VLSI Architecture for Image Template Matching," M. Thesis, Dept. of Computer Science and Engineering, University of South Florida, 1994.
- [3] N. Ranganathan and S. Venugopal. "An Efficient VLSI Architecture for Template Matching Based on Moment Preserving Pattern Matching," in *proc. ICPR-D*, 1994, pp. 388-390.
- [4] T. Fukushima, "Image Signal Processor Computes Fast Enough for Grey-scale Video," *Electronic Design*, pp. 209-215, October, 1984.
- [5] J. Jaggernauth, A. C.P. Loui and A. N. Venetsanopoulos, "Real-time Image Processing by Distributed Arithmetic Implementation of Two-dimensional Digital Filters," *IEEE Trans. on Acoust. Speech Signal Pro., ASSP-33*, pp. 1546-1555, 1984.
- [6] C. H. Chou and Y. C. Chen, "A VLSI Architecture for Real-Time and Flexible Image Template Matching," *IEEE Trans. on Circuits and Systems*, vol. 36, no. 10, pp. 1336-1342, 1989.
- [7] Z. Fang and X. Li and L. M. NI, "Parallel Algorithms for Image Template Matching on Hypercube SIMD Computers," *IEEE Trans. on Pattern Analysis and Machine Intelligence*. vol. 9, no. 6, pp. 835-841, 1987.
- [8] S. Ranka and S. Sahni, "Image Template Matching on SIMD Hypercube Multicomputers," Dept. of Computer Science, University of Minnesota, 1989.
- [9] R. Y. Wong, "Sequential Scene Matching Using Edge Features," *IEEE Trans. on Aerosp. Elec. Syst. Aes-14*, pp. 128-140, 1978.
- [10] M. Svedlow, C. D. McGlem and P. E. Anuta, "Image Registration: Similarity Measure and Preprocessing Math Comparisons," *IEEE Trans. on Aerosp. Elec. Syst. AES-14*, pp. 141-149, 1978.
- [11] A. Venot, J. F. Lebruchec and J. C. Roucayrol, "A New Class of Similarity Measure for Robust Image Registration," *Comput. Vision Graph. Image process.* 28, 1984, pp. 176-184.
- [12] C. H. Chou and Y. C. Chen, "Moment Preserving Pattern Matching," *Pattern Recognition*, vol. 23, no. 5, pp. 461-474, 1990.
- [13] X. Qu and X. Li, "Parallel Template Matching Algorithms," *Proc. of the International Conference on Parallel Processing*, 1988, pp. 223-225.

 저 자 소 개



徐承完(正會員)

1997년 2월 아주대학교 전자공학 학사. 1997년 2월 ~ 현재 아주대학교 전자공학 석사과정. 주관심분야는 영상, 통신 및 신호처리용 ASIC 설계

鮮于明勳(正會員) 第34卷 C編 第8號 參照