

# SIMD 구조의 다중 프로세서를 이용한 NTGST의 병렬고속화

김복만, 서경석, 김종화, 최홍문  
경북대학교 대학원 전자공학과  
전화 : 053-940-8613 / 핸드폰 : 011-534-1395

## Parallel Speedup of NTGST on SIMD type Multiprocessor

Bok-Mann Kim, Kyung-Seok Seo, Jong-Hwa Kim, and Heung-Moon Choi  
Department of Electronics, Graduate School, Kyungpook National University  
E-mail : bkmann@pdp.knu.ac.kr

### 요약

본 논문에서는 SIMD (Single Instruction stream and Multiple Data stream)형 병렬 구조의 다중 프로세서를 이용하여 NTGST (noise-tolerant generalized symmetry transform)를 병렬 고속화하였다. 먼저 NTGST의 화소 및 영상 영역간의 계산 독립성을 이용하여 영상을 분할하여  $P$ 개의 프로세서에 할당하고, 이들 각각을  $N$ 개의 데이터를 한번에 처리하는 SIMD 구조로 병렬화하여  $NP$ 에 비례하는 속도 향상을 얻었다. 실험에서 MMX 기술의 펜티엄 III 프로세서를 2개 사용하여 제안한 알고리즘이 기존의 NTGST 보다 8 배 가까이 고속으로 처리됨을 확인하였다.

### I. 서론

최근, 산업 각 분야에서는 컴퓨터 비전을 이용한 물체의 검출이 많이 사용되고 있고, 단순 배경 속의 특정 단일 물체의 검출뿐만 아니라 사전 정보 없이 복잡한 배경에서 임의의 다중 물체의 효과적인 검출이 요구되고 있다.[5]. 주목연산자 (attention operator)로서 NTGST[1]는 이러한 요구에 부합되며, 다양한 응용분야에서 이미 그 실용성과 성능을 인정받고 있다.

NTGST는 두 화소의 명도변화의 크기와 대칭성뿐만 아니라 방사 (radial) 방향 명도변화 방향의 수렴 및 발산을 대칭 기여도에 반영시킴으로써 명도변화 방향의 수렴과 발산이 일관되지 않은 불규칙 잡음 영역과 복잡한 배경의 대칭 기여도가 상쇄되고 물체영역에서는 대칭 기여도가 누적된다. 따라서, NTGST는 불규칙한 잡음과 복잡 배경 하에서 크기와 형태 및 명도가 다양한 물체의 동시 검출이 필요한 응용분야에 적합하다. 또한 NTGST는 LCD (liquid crystal display) 생산을 위한 BLU (back light unit)의 얼룩 및 이물질 검출 또는 복잡 배경에서의 임의 다중 물체 분할 등에 사용되고 있다.

하지만 NTGST는 계산 집약적이므로 고해상도의 BLU 검사와 같은 응용분야에서는 고속화가 필요하다. 특히 단일 프로세서를 사용하는 순차 알고리즘의 고속화에는 한계가 있으므로, 다중 프로세서를 이용한 병렬 알고리즘의 개발이 시급한 현안이다.

따라서 본 논문에서는 SIMD 구조의 다중 프로세서를 이용한 NTGST의 병렬화 방법을 제안하였다. 먼저 NTGST 알고리즘을  $N$ 개의 데이터를 한번에 처리하는 SIMD 병렬 구조에 적합하도록 변환하고, 개인 컴퓨터에서 일반적으로 사용되는 Pentium 프로세서의 MMX 기술을 이용하여 병렬화 하였다. 이 과정에서 불필요한 반복문을 제거하고 조건문을 순차적으로 풀어서 최

적의 병렬성을 얻도록 하였다. 또한, NTGST 연산에서 화소나 영역간에 서로 독립인 점을 이용하여 다중 프로세서에서 나누어 처리하도록 함으로써 더욱 고속화하였다. 즉, 처리할 하나의 영상이 입력되면 이를  $P$ 개의 프로세서에 각각 분할하고  $N$ 개의 데이터를 SIMD 병렬 처리함으로써  $NP$ 에 비례한 속도 향상을 이룰 수 있도록 하였다.

실험에서는 MMX 기술을 채용한 Pentium III 프로세서 2개를 사용하여 평균적으로 8배 정도의 속도 향상을 얻을 수 있었다.

## II. NTGST 알고리즘

Park 등[1]은 명도 변화 방향이 수렴하는 경우에는 음극성 (-), 발산하는 경우에는 양극성 (+)을 갖도록 명도 변화의 방향성까지 고려하여 대칭성을 대칭 기여도에 반영함으로써 잡음에 강건한 NTGST를 주목연산자로 제안하였다.

NTGST에서는 두 화소의 명도변화 방향의 대칭성을 부각시키며 수렴과 발산을 구분할 수 있도록 위상 가중 함수 (phase weight function)  $P(i, j)$ 를

$$P(i, j) = \sin\left(\frac{\theta_j + \theta_i}{2} - \alpha_{ij}\right) \times \sin\left(\frac{\theta_j - \theta_i}{2}\right) \quad (1)$$

와 같이 정의하였다. 여기서  $\alpha_{ij}$ 는 두 화소들을 연결하는 직선이 수평선과 이루는 각도이고  $\theta_i$ 와  $\theta_j$ 는 각 화소의 명도변화 방향이다. 두 화소의 명도변화 방향이 수렴하면  $P(i, j)$ 는 음극성(-)을 가지며, 발산하면 양극성(+)을 갖는다. 두 화소의 명도변화 방향이 완전히 마주보는 방향일 때 음의 최소값을 가지며, 완전히 반대 방향으로 향할 때 양의 최고 값을 가진다. 그리고,  $\theta_i = \theta_j$  혹은  $\theta_i + \theta_j = 2\pi$ 와 같이 명도변화 방향의 대칭성이 완전히 없을 때는 0의 값을 가진다. 따라서, 이 위상 가중 함수를 이용하면 두 화소의 명도 변화 방향의 수렴과 발산을 구분하면서도 대칭성을 잘 부각시킬 수 있다.

한편, 두 화소간의 거리가 가까울수록 큰 값을 갖도록 거리 가중 함수 (distance weight function)를

$$D_\sigma(i, j) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{\|p_i - p_j\|}{2\sigma}\right) \quad (2)$$

와 같이 정의하였다. 여기서  $\sigma$ 는 대칭 변환할 영역의

크기를 나타낸다. 따라서 두 화소의 거리가 가까울수록, 명도변화 방향이 대칭일수록, 그리고 명도변화 크기가 클수록 큰 대칭 기여도가 나타나게 된다. 화소의 명도변화 크기  $r_k$ 와 위상 가중 함수 및 거리 가중 함수를 이용하여 두 화소의 대칭 정도를 나타내는 대칭 기여도  $C(i, j)$ 를

$$C(i, j) = D_\sigma(i, j)P(i, j) r_i r_j \quad (3)$$

와 같이 구한다.

이와 같이 구성된 NTGST를 복잡한 배경에 물체가 존재하는 영상에 적용하는 경우 복잡한 배경에 대한 대칭기여도의 극성을 다르게 하여 상쇄시키고 물체에 의한 대칭기여도 만을 누적하여 복잡한 배경에서도 물체를 검출 할 수 있다.

그러나, 한 점에 대해 대칭 기여도를 누적할 때 마스크내의 모든 대칭점에 대해서 부동소수점 연산을 수행해야 하므로 영상의 크기가 커지거나 마스크의 영역이 넓어질 경우 처리시간이 많이 소요된다.

## III. NTGST의 병렬화

본 논문은 NTGST를 SIMD 구조의 다중 프로세서를 사용하여 병렬 고속화 하였다. 그림 1에는 기존의 NTGST 알고리즘을 수정하여 SIMD 구조의 다중 프로세서에 사용 가능하도록 변환한 병렬 알고리즘을 나타내었다.

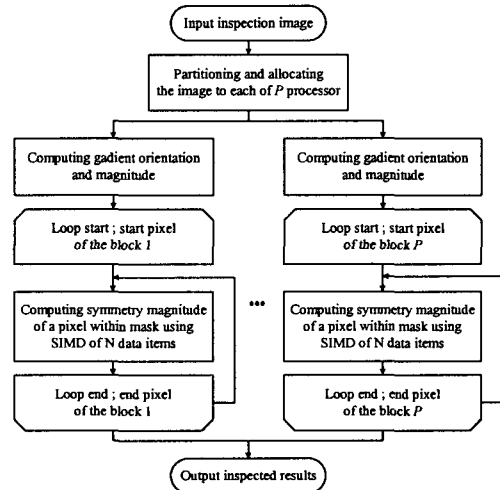


그림 1 NTGST 병렬 알고리즘  
Fig. 1. Parallel NTGST algorithm.

## SIMD 구조의 다중 프로세서를 이용한 NTGST의 병렬고속화

### 3.1 SIMD 구조를 이용한 병렬화

하나의 명령으로 여러 개의 데이터를 처리할 수 있는 SIMD 구조는 CPU의 처리능력을 쉽게 향상시킬 수 있다. SIMD 구조는 반복적이고 병렬성이 강한 루틴에 적합하며, 멀티미디어의 처리나 그래픽, 음성통신 등 많은 부분이 이러한 성향을 가진다.

현재 많이 사용하고 있는 펜티엄 프로세서도 SIMD 구조의 64bit MMX 레지스터를 가지며 펜티엄 III는 더욱 발달한 SIMD 구조의 128bit XMM 레지스터를 독립적으로 가지고 있다. 그림 2는 펜티엄 III의 SIMD 용 레지스터를 나타내었다.

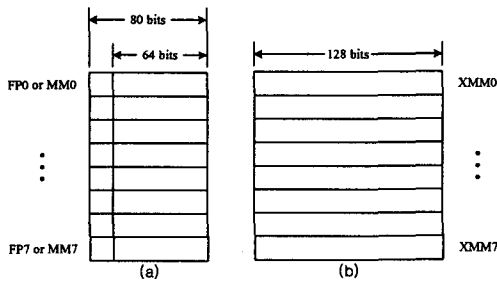


그림 2. 펜티엄 III의 SIMD 레지스터 (a) 8개의 64비트 MMX technology/IA X87-FP 레지스터, (b) 8개의 새로운 128비트 레지스터

Fig. 2. SIMD registers in the Pentium III (a) eight 64-bit MMX technology/Intel Architecture X87-FP registers and (b) new eight 128-bit registers.

펜티엄 III 이상에서 지원되는 아래와 같은 SIMD-FP (floating point)를 사용하면 4개의 float형 데이터를 한꺼번에 병렬 처리할 수 있다.

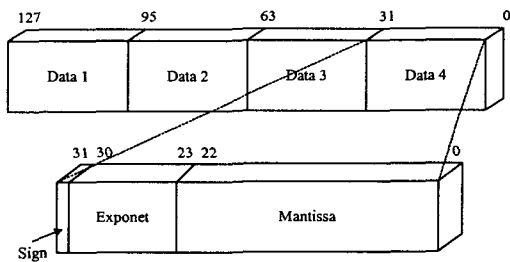


그림 3. SIMD-FP의 데이터형  
Fig. 3. Data type of SIMD-FP.

NTGST의 대칭도 맵을 구하는데 사용되는 식(3)에서 식(1)의 위상 가중 함수와 식(2)의 거리 가중 함수,

그리고 각 대칭점의 명도변화 크기를 SIMD-FP에서 동시에 4개의 화소에 대해 수행 할 수 있으므로 4배의 속도 향상을 기대할 수 있다.

### 3.2 다중 프로세서를 이용한 병렬화

하드웨어의 가격이 저렴해 지고 대부분의 운영체제가 다중 프로세서를 지원하므로 다중 프로세서를 사용한 병렬화가 가능하게 되었다.

단일 프로세서 환경에서 멀티 쓰레드는 단지 시분할로 처리되는 것뿐이어서 시간을 많이 소요하는 처리에 있어서는 한계가 있지만 다중 프로세서 환경에서는 프로세서 수만큼 동시에 수행될 수 있다. NTGST 알고리즘은 영상 영역간의 의존성이 없으므로 영상을 프로세서의 수에 맞게 분할하여 멀티 쓰레드로 처리함으로써 프로세서의 수에 비례한 속도 향상을 얻을 수 있다.

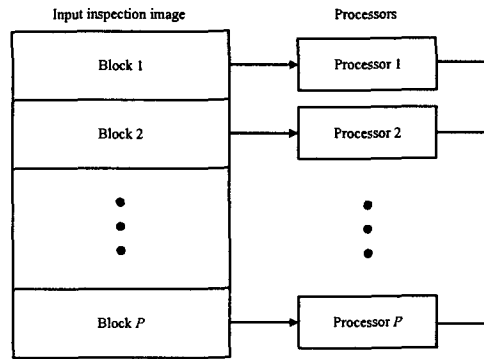


그림 4. 다중 프로세서에서의 영상 분할 및 할당  
Fig. 4. Image partition and allocation onto multiprocessor.

## IV. 실험 결과

제안한 알고리즘의 성능을 평가하기 위해서 개인용 컴퓨터에서 널리 쓰이는 MMX SIMD 병렬화 기술의 Intel Pentium III 프로세서 2개를 사용하여 다양한 크기의 영상에 대해서 기존의 NTGST와 처리 시간을 비교하였다. 표 1에서 볼 수 있는 바와 같이 평균적으로 8배의 속도 향상을 이루었다. 이를 위해 비교적 시간이 많이 걸리는 각도 계산과 정현 함수의 계산을 검색표 (look-up table)를 이용하여 SIMD 구조에 적합하게 알고리즘을 수정하고, 4개의 데이터를 한번에 처리하는 SIMD-FP를 사용하는 프로세서 2개를 이용하여 영상을 두 부분으로 나누어 각각을 프로세서에 할당하

여 동시에 처리 하도록 하였다.

표 1. 기존 NTGST와 제안한 병렬 NTGST 처리시간 비교

Table 1. Processing time of conventional NTGST and the proposed parallel NTGST.

[sec]

Image size (BMP type)	4M bytes	12M bytes	50M bytes
Conventional NTGST	12	33	130
The proposed parallel NTGST	1.4	3.8	16

## V. 결론

계산 집약적인 NTGST 알고리즘을 SIMD 구조의 다중 프로세서를 사용하여 병렬화 하였다.  $N$ 개의 데이터를 처리할 수 있는 SIMD 구조의 프로세서를  $P$ 개 사용할 경우  $NP$ 에 비례한 속도 향상을 얻을 수 있다.

따라서, NTGST 알고리즘의 적용에 제한요소로 작용하는 속도 문제를 이러한 병렬화의 구현으로 해결하여 PC레벨에서도 충분히 활용이 가능하도록 하였다.

## 참고문헌

- [1] 박창준, "물체 위치 검출을 위한 잡음에 강건한 주목 연산자," *경북대학교 박사학위 논문*, 2000
- [2] 박창준, 조상현, 최홍문, "잡음에 강건한 주목 연산자의 구현과 효과적인 다중 물체 검출," *전자공학회 논문지*, vol. 38SP, no. 1, pp. 89-96, Jan. 2001.
- [3] C. J. Park, W. G. Oh, S. H. Cho, and H. M. Choi, "An efficient context-free attention operator for BLU inspection of LCD," *Proc. of the IASTED SIP*, pp. 251-256, Las Vegas, Nevada, USA, Nov. 2000.
- [4] K. S. Seo, C. J. Park, S. H. Cho, and H. M. Choi, "Context-free marker-controlled watershed transform for efficient multi-object detection and segmentation," *IEICE Transaction on Fundamentals*, Vol. E84-A, No. 6, June 2001.
- [5] D. Reissfeld, H. Wolfson, and Y. Yeshurun, "Context-free attentional operators: The generalized symmetry transform," *IJCV*, vol. 14, pp. 119-130, Jan. 1995.

- [6] A. Peleg and U. Weiser, "MMX Technology Extension to the Intel Architecture," *IEEE Micro*, vol. 16, no. 4, pp. 42-50, Aug. 1996.
- [7] A. Peleg, S. Wilkie, and U. Weiser, "Intel MMX for Multimedia PCs," *Communications of the ACM*, vol. 40, no. 1, pp. 25-38, Jan. 1997.
- [8] Intel Architecture Software Developer's Manual Volume 1: Basic Architecture, Order Number 243190, Intel, 1999.
- [9] B. Forta, P. Fonte, and G. Brewer, *Windows 2000 Developer's Guide*, M&T Books