

ELBP 분류기를 이용한 초해상도 기법의 CUDA 최적화

최지훈, *송병철
인하대학교

jiji12641@hanmail.net, *bcsong@inha.ac.kr

CUDA Optimization of Super-Resolution Algorithm using ELBP Classifier

Ji Hoon Choi, *Byung Cheol Song
Inha University

요 약

저해상도 영상을 고해상도 영상으로 복원하기 위한 다양한 방법의 초해상도 기법이 존재한다. 다양한 기법들 중에서도 ELBP 분류기를 이용한 초해상도 기법[1]은 단일 영상 기반의 초해상도 기법으로 사전에 학습된 필터를 이용하여 고해상도 영상을 획득하는 기법이다. 그러나 해당 알고리즘을 일반적인 CPU 환경에서 수행할 경우 실시간으로 영상을 획득하는데 어려움이 존재한다. 본 논문에서는 지역메모리를 이용한 GPU 환경에서의 최적화를 수행하여 ELBP 분류기를 이용한 초해상도 기법의 가속성을 보인다. 먼저, 알고리즘에 대하여 간단히 설명하고 CUDA 가속화 기법[2]을 차례로 적용했을 때 얻을 수 있는 가속 성능을 확인한다. 최종적으로 본 논문은 CPU 환경과 비교했을 때 5 배의 가속 효과를 얻을 수 있다.

1. 서론

최근 디스플레이 기술이 향상됨에 따라 저해상도의 영상을 선명한 화질의 고해상도 영상으로 복원하는 초해상도 기법에 대한 연구가 활발히 이루어지고 있다. 초해상도 기법은 크게 다수의 영상을 이용하는 기법과 단일 영상을 이용하는 기법으로 나눌 수 있다.

다수의 영상을 이용하는 기법의 경우 비교적 선명한 영상을 획득할 수 있는 반면 입력 영상 간 정합과정이나 최적화 과정 등에서 매우 많은 연산량을 필요로 하기 때문에 실제로 적용하기에 어려움이 있다. 단일 영상을 이용하는 기법은 복원 기반의 기법, 학습 기반 및 자기 학습 기반 기법, 학습된 필터를 이용한 기법 등으로 나눌 수 있는데 단일 영상을 이용하는 경우 다수의 영상을 사용하는 기법과 비교해 성능은 떨어지나 적은 연산량을 가지고 고해상도 영상을 획득할 수 있는 장점이 있다.

ELBP 분류기를 이용한 초해상도 기법은 학습된 필터를 이용한 단일 영상 기반의 초해상도 기법 중의 하나이다. 해당 기법은 입력 영상의 각 패치에 대하여 제안된 ELBP 분류기를 이용해 고주파 정보 합성을 위한 필터의 주소 값을 선택한다. 선택된 위치에 필터가 존재하는 경우 해당 필터를 이용하여 최종적으로 고주파를 합성하는 과정을 수행한다.

그러나 ELBP 필터를 이용해 고주파를 합성하는 과정에서 사용되는 연산량이 많아 일반적인 C 언어를 이용하여 수행할 경우 실시간으로 고해상도의 영상을 획득하기에 어려움이 있다. 따라서 본 논문에서는 CUDA 를 이용한 최적화를 수행하여 GPU 환경에서 ELBP 를 이용한 초해상도 기법의 가속성을 확인한다.

2. ELBP 분류기를 이용한 초해상도 기법

LBP 는 입력 패치의 중심 밝기 값을 기준으로 이웃한 픽셀 밝기 값의 대소를 비교해 이를 이진수의 값으로 추출하여 해당 패치를 대표하는 특성으로 이용한다. 제안 기법은 중심 픽셀로부터의 거리가 1 과 2 인 각각 8 개의 픽셀을 이용하여 해당 픽셀들의 평균 밝기 값을 기준으로 대소를 비교한 LBP 패턴을 추가로 이용한다. 따라서 총 24 비트로 이루어진 ELBP 패턴을 분류기로 사용한다.

$$ELBP(X) = \sum_{p=0}^{P-1} S(X_{1,p} - \mu_1) \cdot 2^p + S(X_{2,p} - \mu_2) \cdot 2^{p+P} + S(X_{1,p} - X_c) \cdot 2^{p+2P} \quad (1)$$

사전에 충분히 많은 수의 저해상도-고해상도 패치 쌍을 이용하여 제안된 ELBP 를 기준으로 동일한 패턴을 갖는 저해상도 패치들로 클래스를 분류한다. 이어서 식 (2)를 이용해 각 클래스에 최적화된 필터를 생성하는 학습 과정을 수행한다.

$$HR(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \omega_{i,j}(m, n) LR(m, n) \quad (2)$$

합성 과정에서는 저해상도 영상이 입력되면 각 패치의 ELBP 값을 추출하고 이를 필터의 주소 값으로 사용하여 사전에 학습한 필터를 선택한다. 최종적으로 선택된 필터 정보와 식(2)를 이용해 필터링 과정을 수행함으로써 고해상도 영상을 획득한다. 단, ELBP 에 해당하는 필터가 존재하지 않을 경우 해당 패치는 일반적인 선형 보간법을 이용하여 복원한다.

3. CUDA 가속화 기법

먼저, CUDA 가속화 기법으로 공유 메모리를 이용한다. 각 과정에 이용되는 영상 정보는 블록 별로 일정 범위의 인근 영역으로 제한되어 있고, 일부는 서로 다른 쓰레드에서 규칙성 있게 중복되어 사용된다. 따라서 각 블록의 연산에 필요한 영상 정보를 사전에 지역 메모리인 공유 메모리로 할당하여 빠른 속도로 필요한 정보를 이용할 수 있도록 한다.

두 번째로, 상수 메모리를 이용하여 가속화를 수행한다. 저해상도 영상 패치의 ELBP 에 해당하는 필터가 존재하지 않을 경우, 일반적인 선형 보간법을 이용해 고해상도 영상을 획득한다. 본 논문에서는 Lanczos 필터를 이용하여 영상을 확대하는데 Lanczos 필터는 6 개의 필터 탭을 사용한다. 이와 같이 고정적으로 사용되는 필터 계수 등을 읽기 전용 메모리인 상수 메모리로 선언하면 가속 성능을 확인할 수 있다. 그러나 상수 메모리는 사용할 수 있는 용량이 제한되어 있어 매우 큰 크기를 차지하는 ELBP 필터는 상수 메모리로 이용할 수 없다.

따라서 ELBP 필터는 텍스처 메모리를 이용해 가속화한다. 입력 패치의 필터 주소 값이 정해지면 각각의 필터가 가지고 있는 625 개의 필터 계수에 동일한 패턴으로 접근하기 때문에 공간적으로 캐시되어 사용되는 텍스처 메모리를 활용하기에 적합하다.

가속화 기법을 적용한 이후, 각 커널에서 사용하고자 하는 쓰레드의 차원 및 크기를 조절해 입력 영상의 크기에 최적화된 값을 탐색한다. 최적의 쓰레드 차원 및 크기는 일반적인 경향이 존재하나 실험적인 결과로 최적의 값을 판단할 수 있다.

4. 실험 결과

GPU 환경에서의 가속 성능을 비교하기 위하여 3.2GHz 로 동작하는 i5-4570 CPU 와 GeForce GT 730 을 이용하였고, 실험 영상은 768x512 해상도의 Kodak 표준 영상* 중 18 개 영상을 사용하였다.

CPU 환경에서 알고리즘을 수행하는데 걸리는 시간을 측정하면 평균적으로 65.52 ms 가 소요되는 것을 확인할 수 있다. 각 부분별로 소요 시간을 확인해보면 2D FIR 필터링 부분에서 45.33 ms 로 약 69.2%의 매우 많은 연산 시간을 차지하고 ELBP 추출 및 주소 할당이나 고주파 합성 부분에서 각각 8.00 ms, 6.72 ms 로 적지 않은 시간이 소요되는 것을 확인할 수 있다.

표 1. CPU/GPU 환경에서의 각 과정 별 평균 수행 시간 비교

	CPU(msec)	GPU(msec)	가속 성능
Lanczos	3.53	2.31	×1.53
패치 variance 계산	1.94	0.61	×3.18
ELBP 추출 및 주소 값 할당	8.00	1.18	×6.78
2D FIR 필터링	45.33	23.26	×1.95
고주파 합성	6.72	1.64	×4.10
Total	65.52	29.00	×2.26

* Kodak 표준 영상(<http://r0k.us/graphics/kodak/>)



그림 1. Kodak 표준 영상(7 번 영상(좌) 13 번 영상(우))

표 2. 입력 영상에 따른 수행 시간 비교

	7 번 영상		13 번 영상	
	CPU(msec)	GPU(msec)	CPU(msec)	GPU(msec)
Lanczos	3.00	2.25	3.00	2.31
패치 variance 계산	2.00	0.62	2.00	0.62
ELBP 추출 및 주소 값 할당	6.00	1.13	14.00	1.74
2D FIR 필터링	45.00	19.34	54.00	34.62
고주파 합성	6.00	1.62	6.00	1.65
Total	62.00	24.96	79.00	40.64
가속 성능	×2.48		×1.94	

각 과정에 대하여 GPU 환경에서의 최적화를 개별적으로 수행한 결과, 총 29.00 ms 로 평균적으로 2.26 배의 가속 성능을 보인다. 연산 과정이 전체적으로 동일하게 이루어지는 패치의 variance 계산이나 고주파 합성 부분에서는 약 3 배 이상의 가속 성능을 확인할 수 있으며 특히 ELBP 추출 및 주소 할당 부분은 약 6.78 배로 향상된다.

2D FIR 필터링의 경우 최적화 이전과 비교 했을 때 약 1.95 배의 가속성을 보이는데 이는 입력되는 패치의 특성에 따라 서로 다른 필터를 불러온 다음 필터링이 수행되기 때문에 완전한 병렬 연산 처리가 이루어지기 어렵다. 따라서 영상의 특성에 따라 연산 시간도 비교적 큰 차이를 보인다.

그림 1 은 실험에 사용한 영상 중 2 개 영상의 예로 7 번은 일정 영역이 평탄한 특성을 보이고 13 번 영상의 경우 영상의 전 영역에 걸쳐 다양한 패치 특성을 가진다. 다른 과정에서는 두 영상이 유사한 가속 성능을 보이지만 영상의 특성에 따라 2D FIR 필터링에 대한 가속 성능이 차이가 나는 것을 알 수 있다. 이는 전체 알고리즘의 수행 시간에 직접적인 영향을 미친다.

표 3. 커널 통합에 따른 전체 수행 시간 비교

	CPU(msec)	GPU(msec)	가속 성능
Total	65.12	27.67	×2.35

추가로 패치 variance 계산, ELBP 추출 및 주소 값 할당, 2D FIR 필터링 커널은 각 블록 연산 과정에서 동일한 영역을 이용하여 연산이 이루어지기 때문에 해당 커널은 1 개로 통합하여 1 회의 공유 메모리 할당만으로 연산 수행이 가능하다. 3 개 커널을 통합하여 알고리즘을 적용할 경우 약 27.67ms 가 소요되어 최종적으로 약 2.35 배의 가속 성능을 보인다.

5. 결론

본 논문은 ELBP 분류기를 이용한 초해상도 기법의 실시간성을 위하여 CUDA 를 이용해 GPU 환경에서의 메모리 최적화 및 커널 통합 등을 실시하였다. 결과적으로 768x512 해상도의 영상을 2 배 확대할 때 35Hz 이상의 실시간성을 확인하였다.

참고문헌

- [1] D. Y. Choi and B. C. Song, "Fast Super-Resolution Algorithm using ELBP Classifier," Proc. IEEE VCIP, Singapore, Dec. 2015.
- [2] J. Sanders and E. Kandrot, "CUDA by Example: An Introduction to General-Purpose GPU programming," Addison-Wesley, 2010.