레이더 포인트 클라우드 클러스터링을 위한 고정 GMM distance-score 추론 커널의 FPGA 구현 및 정량 비교

여리은, *이성주
세종대학교 반도체시스템공학과 및 메타버스융합전공, *세종대학교 AI 융합전자공학과 및 지능형드론융합전공

rieun@itsoc.sejong.ac.kr, *seongjoo@sejong.ac.kr

FPGA Implementation and Quantitative Evaluation of a Fixed GMM Distance-Score Inference Kernel for Radar Point Cloud Clustering

Rieun Yeo, *Seongjoo Lee
Dept. of Semiconductor Systems Engineering and Convergence Engineering for Metaverse Sejong Univ., *Dept. of AI Convergence Electronic Engineering and Convergence Engineering for Intelligent Drone, Sejong Univ.

요 약

본 연구는 레이더 포인트 클라우드와 같은 3 차원 좌표 기반 다객체 환경에서, 오프라인에서 학습된 고정(global) GMM 의 distance-score 추론(E-step)만을 FPGA 로 가속하고 소프트웨어와 동등 조건에서 비교한다. 공분산은 대각(diagonal)으로 가정하고, 각 포인트에 대해 Mahalanobis 거리 기반 score 를 계산한 뒤 argmax 로 best_k 를 출력하는 커널을 Q16 고정소수점으로 구현하였다. 실험에서 Q16 은 float 대비 클러스터 선택 불일치율이 0.4%로 낮았으며, 경량 커널에서는 AXI4-Lite 기반 순차 전송 오버헤드가 성능에 크게 영향을 주는 것을 확인하였다.

Ⅰ. 서 론

3D/4D 레이더는 프레임당 수천 개 포인트를 생성하므로 객체 단위 클러스터링이 필수이며, GMM 은 분산/형상을 반영하는 확률적 모델링으로 레이더 데이터에 적합하다[1]. 그러나 EM/VB-GMM 의 반복 학습을 임베디드 SoC 에서 실시간으로 수행하는 것은 부담이 크고, 소프트웨어적 기법(VB 기반 모델링 등)만으로는 프레임레이트 · 자원 제약을 동시에 만족시키기 어렵다[2]. FPGA 에서 EM 엔진을 직접 구현한 연구도 있으나 업데이트/제어가 복잡해 SoC 관점에서 설계 · 자원 부담이 증가할 수 있다[3]. 따라서 학습은 오프라인에서 수행하고, 온라인 단계에서는 고정 파라미터를 갖는 추론 커널만 가속하는 구조가 현실적이며, 본 연구는 이 설정에서 고정소수점 양자화 및 PS-PL 인터페이스 오버헤드가 성능/정확도에 미치는 영향을 동일 조건으로 비교한다.

Ⅱ. 고정 GMM distance-score 커널 및 Q16 구현

본 논문에서 사용하는 GMM 은 D차원 특징 벡터 $x \in \mathbb{R}^D$에 대해 K개의 가우시안 컴포넌트로 구성되며, 하드웨어 구현 복잡도를 줄이기 위해 공분산을 대각(diagonal)으로 가정한다. 즉, $\Sigma_k = \mathrm{diag}(\sigma_{k,1}^2, \ldots, \sigma_{k,D}^2)$로 두고 축별 분산만을 저장한다. 고정(global) GMM 파라미터 $(\mu_k, \sigma_k^2, \pi_k)$가 주어졌을 때, 입력 포인트 x에 대한 컴포넌트 k의 Mahalanobis 거리 제곱과 score 는 다음과 같이 정의된다.

$$d_k^2(x) = \sum_{d=1}^{D} \frac{(x_d - \mu_{k,d})^2}{\sigma_{k,d}^2}$$

$$\mathrm{score}_k(x) = \pi_k \exp\left(-0.5\, d_k^2(x)\right)$$

엄밀한 posterior 계산에는 score 의 정규화가 필요하지만, 본 논문의 목적은 "가장 가능성 높은 컴포넌트 선택"이므로 $\mathrm{best}_k(x) = \arg\max_k \mathrm{score}_k(x)$를 최종 클러스터로 사용한다. 결과적으로 제안 커널은 포인트당 K개의 d_k^2와 score 를 평가하고 argmax 로 best_k 를 출력하는 distance-exp-score-argmax 구조로 정리된다.

하드웨어 구현을 위해 입력 포인트와 파라미터 (μ, σ^2, π)를 16-bit signed Q16 으로 통일하고, 본 구현에서는 FRAC_BITS=9(스케일 $2^9 = 512$)로 float↔Q16 변환을 수행한다(포맷/범위는 표 1). 커널 내부에서는 $(x-\mu)$가 signed 이므로 부호 연산을 유지한 상태에서 제곱 · 누산 및 π곱으로 인해 비트폭이 증가하는 구간은 확장 비트폭으로 처리하고, 필요한 지점에서 FRAC_BITS 시프트로 Q16 스케일을 정규화한다. 나눗셈은 결과도 Q16 스케일을 유지하도록 분자를 FRAC_BITS 만큼 스케일 업한 정수 나눗셈(restoring/shift-subtract)으로 구성하며, 최종 출력은 포화(saturation) 정책으로 16-bit 범위에 안전하게 매핑한다. 또한 $\exp(\cdot)$는 하드웨어 비용을 줄이기 위해 LUT 기반 근사(필요 시 shift-add 결합)로 구현하여 score 계산을 효율화하였다.

Ⅲ. HW/SW co-design 및 제어 방식(압축)

그림 1 은 PC(MATLAB)에서의 오프라인 학습/양자화

* 교신저자

와 ZCU104 보드에서의 런타임 추론을 분리한 HW/SW co-design 구조를 보여준다. PC 에서는 데이터 전처리 및 global GMM 학습을 수행한 뒤, 학습된 (μ, σ^2, π)를 Q16 으로 양자화하여 보드에서 사용 가능한 형태로 전달한다. 보드에서는 PS(ARM, Vitis)가 DDR 에 적재된 Q16 포인트/파라미터를 바탕으로 PL 의 GMM IP 를 구동하고, 각 포인트의 클러스터 할당 결과(best_k, best_score)를 회수해 저장한다.

PL 의 GMM IP 는 AXI4-Lite 인터페이스 계층과 Q16 추론 커널 계층으로 구성된다. 인터페이스 계층은 PS 가 레지스터 접근만으로 포인트 좌표와 클러스터 파라미터, 제어 신호를 전달하고 결과를 읽을 수 있도록 하며, 커널은 distance–exp–score–argmax 파이프라인으로 동작한다. 포인트 좌표 (x, y, z)는 포인트당 1 회 입력되고, 클러스터 k 의 $(\mu_k, \sigma_k^2, \pi_k)$ 는 $k = 0 \sim K-1$ 순서로 입력된다. 각 k 입력마다 d_k^2 와 score_k 를 계산해 best_k/best_score 를 갱신하고, 마지막 클러스터 입력에서 결과를 확정(out_valid)하여 PS 가 동기적으로 결과를 읽도록 한다. 본 구조는 MATLAB 과 동일 입력/파라미터 조건에서 하드웨어 구동을 가능하게 해 비교를 단순화하는 반면, 레지스터 기반 순차 전송이 포함되어 경량 커널에서는 PS–PL 트랜잭션 오버헤드가 전체 지연에 영향을 줄 수 있으므로, 본 논문은 이를 end-to-end 관점에서 함께 평가한다.

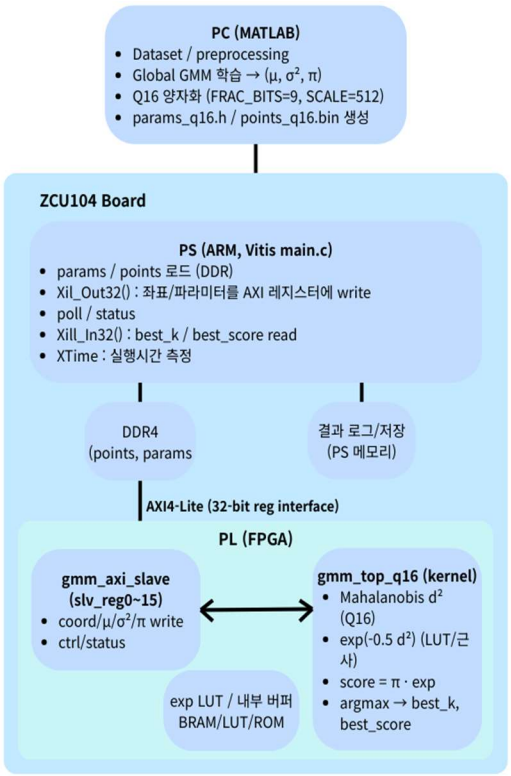

그림 1: HW/SW 블록 다이어그램(PC 학습/양자화 ↔ ZCU104 추론)

Ⅳ. 실험 결과 및 결론

Method	Total time(s)	per-sample (μs)
MATLAB–float	8.3605	8.3605
MATLAB–Q16	8.4074	8.4074
MATLAB–float+copy	13.7186	13.7186
FPGA–Q16	17.6	17.6

표 1. 실행 시간 비교 (distance-exp-score 커널, K=4, D=3)

실험은 5000 프레임, 프레임당 객체 4 로 구성된 총 100 만 포인트 데이터셋을 대상으로 수행했으며, 입력 특징은 $[x, y, z]$만 사용하였다. 공정한 비교를 위해 MATLAB 에서 학습한 global GMM 파라미터를 고정하여 동일한 distance–exp–score–argmax 커널을 소프트웨어와 하드웨어에서 각각 실행하였다. 표 1 에 요약된 per-sample 평균 시간은 MATLAB–float/Q16 ≈ 8.4 μs, MATLAB–float+copy ≈ 13.7 μs, FPGA–Q16 ≈ 18.6 μs 로 측정되었다. 여기서 MATLAB–float+copy 는 매 샘플마다 (μ, σ^2, π)를 재설정하는 비용을 포함한 경우로, 순수 커널 연산보다 데이터 이동/설정 비용이 실행 시간을 지배할 수 있음을 보여준다. 현 FPGA 프로토타입 또한 PS 가 AXI4-Lite 레지스터를 통해 포인트와 파라미터를 순차 전송하며 구동되기 때문에, $K = 4, D = 3$처럼 커널이 경량인 설정에서는 연산 이득보다 PS–PL 트랜잭션 오버헤드가 전체 지연에 크게 반영됨을 확인하였다.

정확도 측면에서는 Q16(FRAC_BITS=9) 양자화 적용 시 float 대비 best_k mismatch 가 0.4%로 관측되어, 16-bit Q16 정밀도가 클러스터 선택 결과를 대부분 유지함을 보였다. 종합하면 본 연구는 고정 GMM 추론 커널을 Q16 으로 하드웨어 구현했을 때도 소프트웨어와 유사한 클러스터 할당 결과를 얻을 수 있음을 검증했으며, 동시에 경량 커널에서 인터페이스 오버헤드가 성능을 제한하는 주요 요인임을 정량적으로 제시하였다. 향후에는 (μ, σ^2, π)를 PL 내부 BRAM 에 상주시켜 런타임 전송을 제거하고, 포인트만 스트리밍 입력하는 구조로 전환함으로써 PS–PL 트랜잭션 오버헤드를 줄이고 처리량을 개선할 계획이다.

ACKNOWLEDGMENT

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No. 2023R1A2C1006340)과 정부(교육부)의 재원으로 한국연구재단의 이공분야 대학중점연구소지원사업의 지원(No. 2020R1A6A1A03038540)을 받아 수행된 연구이며, 검증을 위한 EDA 관련 툴은 IDEC 의 지원을 받았음.

참 고 문 헌

[1] F. Jin, A. Sengupta, S. Cao, and Y.-J. Wu, "MmWave radar point cloud segmentation using GMM in multimodal traffic monitoring," in Proc. IEEE RadarConf, 2020.

[2] F. Valente and C. Wellekens, "Variational Bayesian speaker clustering," in Proc. EUSIPCO, 2004.

[3] C. Guo and W. Luk, "A fully-pipelined expectation–maximisation engine for Gaussian mixture models," in Proc. IEEE Int. Conf. Field-Programmable Technology (FPT), 2012.