



## 저작자표시 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#) 

공학석사학위논문

QoS를 고려한 스마트폰 GPU의  
에너지 최적화 기법

2015년 2월

서울대학교 대학원

전기·컴퓨터공학부

김도한

# QoS 를 고려한 스마트폰 GPU 의 에너지 최적화 기법

지도교수 김 지 흥

이 논문을 공학석사 학위논문으로 제출함

2015 년 2 월

서울대학교 대학원

전기·컴퓨터공학부

김 도 한

김도한의 공학석사 학위논문을 인준함

2015 년 2 월

위 원 장 :	신 현 식	(인)
부위원장 :	김 지 흥	(인)
위 원 :	민 상 렬	(인)
위 원 :	하 순 회	(인)

# 요약

모바일 GPU의 소비 전력이 전체 시스템 전력 소모량에서 큰 비중을 차지하게 됨에 따라 스마트폰에서 중요한 전력 관리 대상으로 부상하였다. 그러나 모바일 GPU로써 스마트폰에 널리 탑재되고 있는 ARM Mali-400MP의 장치 드라이버를 분석한 결과, 기본으로 제공되는 전력 관리 기법인 단순 장치 사용률 기반의 동적 주파수 및 전압 스케일링(DVFS) 알고리즘은 부하가 요구하는 성능보다 높은 성능으로 GPU를 동작하도록 하여 비효율적인 전력 에너지 소모가 발생하였다.

따라서 본 논문은 장치 사용률뿐만 아니라 스마트폰 응용의 서비스 품질 요소로써 초 당 프레임 수(FPS)와 배터리 시간을 함께 고려하여 기존 DVFS 알고리즘에서 발생한 비효율적인 주파수 상승을 방지할 수 있는 전력 에너지 최적화 기법을 제안한다.

본 연구에서 제안하는 기법은 응용 별로 GPU가 제공하는 동작 주파수 별로 FPS를 관측하여 성능 하락이 어느 정도 되는지 관측하고 이를 바탕으로 배터리 잔량을 함께 고려하여 목표 FPS를 유지하며 전력 소모를 최소화할 수 있는 최적의 주파수로 GPU를 동작하도록 한다.

제안한 에너지 최적화 기법을 적용하면 기존 장치 사용률 기반의 DVFS 기법 대비 평균 9.9%, 최대 23%의 에너지 효율 향상이 가능하였으며 FPS는 평균 0.4 정도의 차이를 보이며 성능 하락이 거의 없었다.

추가로 배터리 잔량이 부족하여 사용자가 응용을 사용하는 평균 시간을 만족할 수 있도록 어느 정도의 성능 하락을 허용한 경우 평균 8.5분의 배터리 시간 연장이 가능하였다.

주요어 : 모바일 GPU, 서비스 품질, 초 당 프레임 수, 배터리 시간

학번 : 2013-20752

# 목차

요약	i
목차	iii
그림목차	v
표목차	vi
제 1 장 서론	1
1.1 연구 배경 . . . . .	1
1.2 연구 동기 . . . . .	5
1.3 연구 기여 . . . . .	7
1.3 논문 구성 . . . . .	9
제 2 장 장치 사용률 기반 DVFS의 평가	10
2.1 장치 사용률 기반 DVFS . . . . .	10
제 3 장 QoS를 고려한 스마트폰 GPU 에너지 최적화 기법	15
3.1 QoS를 고려한 GPU 에너지 최적화 기법의 알고리즘. . . . .	15
3.2 동작 주파수 간 장치 사용률 예측 . . . . .	18
3.3 배터리 잔량을 고려한 FPS 최대 하락 허용 값의 동적 조절 기 법 . . . . .	22

제 4 장 실험 결과	26
4.1 실험 환경 . . . . .	26
4.2 실험 결과 . . . . .	28
제 5 장 결론	34
5.1 결론 . . . . .	34
5.2 향후 연구 . . . . .	36
참고문헌	37
Abstract	38

# 그림목차

그림 1 게임 응용 실행 시 각 컴포넌트 별 전력 소모량 . . . . .	5
그림 2 ARM Mali-400MP의 DVFS 알고리즘 . . . . .	10
그림 3 Pixelgun3D 실행 시 주파수 천이 . . . . .	11
그림 4 Minecraft 실행 시 주파수 천이 . . . . .	12
그림 5 QoS를 고려한 에너지 최적화 기법 플로우 차트 . . . . .	15
그림 6 장치 사용률 예측이 없을 때 제안 기법의 FPS 하락 . . .	18
그림 7 FPS 최대 하락 허용 값 동적 조절 기법 . . . . .	22
그림 8 FPS 최대 하락 허용 값에 따른 에너지 소모량 . . . . .	23
그림 9 FPS 최대 하락 허용 값에 따른 평균 FPS 차이. . . . .	24
그림 10 제안 기법의 소모 에너지 . . . . .	28
그림 11 제안 기법의 평균 FPS . . . . .	29
그림 12 Bowling 3D 주파수 천이의 변화 . . . . .	30
그림 13 Counter Strike 주파수 천이의 변화 . . . . .	31
그림 14 FPS 최대 하락 허용 값 동적 조절에 따른 에너지 소모량 .	32
그림 15 FPS 최대 하락 허용 값 동적 조절에 따른 배터리 시간 . .	32
그림 16 실제 배터리 시간과 응용의 평균 사용 시간 차이 . . . .	33



# 표목차

표 1 동작 주파수 고정 시 게임 응용의 FPS 분포 . . . . .	13
표 2 인접 동작 주파수 간 GPU 처리량의 상관관계. . . . .	20
표 3 다음 주파수에서 성능하락이 예상되는 GPU 사용률 . . . . .	20
표 4 FPS 최대 하락 허용 값 동적 조절을 위한 학습 데이터 . . . . .	27
표 5 제안 기법의 에너지 이득 . . . . .	28
표 6 제안 기법의 평균 FPS. . . . .	29

# 제 1 장 서 론

## 1.1 연구 배경

스마트폰이 최근 널리 보급되어 생활환경의 핵심 장치로 부상함에 따라 고사양, 고화질의 시각 콘텐츠가 폭발적으로 증가하였다. 이러한 시각 콘텐츠인 고사양의 3D 게임, 유려한 UI 등을 문제없이 구동하기 위해서는 스마트폰에서도 데스크톱 PC 와 같이 고성능의 GPU 가 필요하게 되었다. 이에 스마트폰 제조사들은 모바일 GPU 의 성능 개선을 위하여 CPU 와 같이 코어의 수를 늘린 멀티코어 형태로 개발하거나 복잡한 셰이더 연산을 지원하도록 내부 구조의 복잡도를 증가시키게 되었다. 이러한 복잡도 증가로 모바일 GPU 의 전력 소모가 시스템 전체 전력 소모에서 차지하는 비중이 계속해서 증가될 것으로 예상되었으며 연구 결과 [1]에 따르면 최근 그 비율이 평균 20%에 이를 정도로 제한된 용량의 배터리를 전력 원으로 사용하는 스마트폰에 있어 매우 중요한 전력 관리 대상으로 대두되었다.

한편, 스마트폰에 탑재된 모바일 GPU 의 전력 소비와 관련된 연구들은 주로 계측 장비 없이 응용 및 시스템 개발자에게 소모 전력을 측정할 수 있도록 전력 예측 모델을 만들거나 그 필요성을 주장하는 연구들[1] - [4]이 주를 이루었으나, 실제 에너지 최적화 기법과 관련된 연구는 부족한 실정이다. 그 이유로 대부분의 제조사들이 모바일 GPU 의 내부 구현을 모두 공개하지 않아 스마트폰 이전의 모바일 GPU 를 대상으로 한 연구[5]와 같이 렌더링 파이프라인을 고려하는 등 GPU 에 특화된 다양한 에너지 최적화 기법 연구가 제한적이기 때문이다. 따라서 제조사들이 GPU 장치 드라이버를 통하여 주로 공개하고 있는 대표적인 에너지 최적화 알고리즘으로 부하 량에 따라 성능을 동적으로 조절하는 DVFS(Dynamic Voltage Frequency Scaling)나 DPM(Dynamic Power Management)등의 효율을 향상할 수 있는 방향으로 에너지 최적화 연구가 필요하다.

본 논문에서는 최근 스마트폰에 많이 탑재되고 있는 모바일 GPU IP(Intellectual Property)중 하나인 ARM 사의 Mali-400MP 를 선정하여 먼저 장치 드라이버에서 기본으로 제공하는 에너지 최적화 기법인 GPU 장치 사용률(GPU Utilization) 기반 DVFS 알고리즘의

효율을 평가한다. 평가 결과, 실제 사용자가 체감 가능한 GPU 성능 지표인 초 당 프레임 수, FPS(Frames per Second)의 저하가 없음에도 불구하고 비효율적인 동작 주파수 상승이 빈번하게 발생하는 문제점을 밝힌다.

따라서 본 논문에서는 장치 사용률뿐만 아니라 GPU 의 성능 지표인 FPS 를 함께 고려하여 기존 DVFS 알고리즘에서 발생한 비효율적인 전력 소모량을 줄이고 동시에 스마트폰 배터리 시간을 고려한 전력 에너지 최적화 기법을 개발한다. 즉, 실제 사용자가 체감할 수 있는 FPS 의 저하가 없는 범위에서 동작 주파수 상승을 방지하여 전력 에너지 소모량을 줄이고, 배터리 잔량이 일정량 이하 일 때 사용자가 배터리 시간의 연장을 원한다면 모바일 GPU 의 성능을 어느 정도 하락시켜 그 목표를 달성할 수 있도록 한다. 이로써 에너지 효율 향상과 함께 실제 GPU 의 성능과 스마트폰의 사용시간에 대한 일정한 서비스 품질, 즉 QoS(Quality of Service)를 보장할 수 있는 전력 에너지 최적화 기법을 제안한다. 제안한 기법을 평가하기 위하여 기존 장치 사용률 기반의 DVFS 알고리즘과 제안한 기법을 각각 적용하여 전력 에너지 소모량과 성능 차이를 확인하였다. 실험 결과 제안 기법의 전력 소모량은 기존 기법 대비 평균 9.9%의

에너지 이득을 얻을 수 있었으며, 이 때 평균 FPS 는 0.4 정도의 차이로 성능 하락이 거의 없었다. 또한 배터리 잔량이 10%이하일 경우 사용자가 응용을 사용하는 평균 시간을 고려하여 FPS 하락을 응용 요구치의 최대 10%까지 허용했을 경우 증가된 에너지 이득으로 평균 8.5 분의 배터리 시간 연장이 가능하였다.

## 1.2 연구 동기

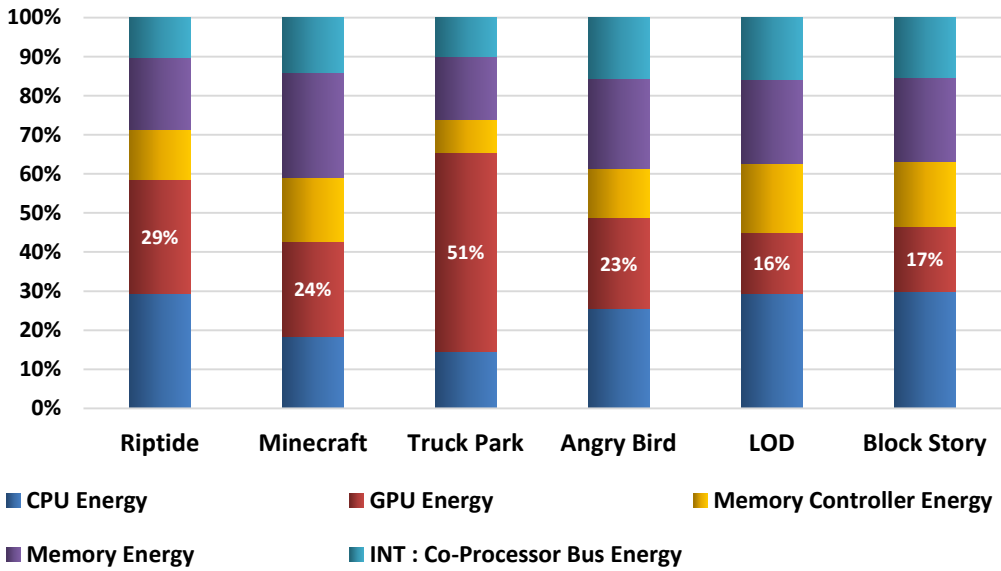


그림 1 게임 응용 실행 시 각 컴포넌트 별 전력 소모량

스마트폰에서 모바일 GPU가 차지하는 전력 에너지의 비중을 알아보기 위하여 갤럭시 S3 개발보드에서 6개의 게임 응용을 1분 동안 실행하며 GPU를 포함한 5개 주요 컴포넌트(CPU, GPU, Memory Controller, Memory, Co-processor Bus)의 에너지 소모량을 실측하였으며 그

결과는 그림 1과 같다. 이 때 모바일 GPU는 주요 컴포넌트들의 전체 에너지 소모량 중 평균 27%를 차지하였고 Truck Park 응용에서 최대 51%를 보이며 상당한 비중을 차지하였다. 또한 모바일 GPU의 장치 드라이버에서 제공하는 함수로 GPU 처리를 요청한 프로세스 번호를 분석한 결과 현재 실행중인 전경 응용(Foreground Application)이 GPU를 독점하여 사용하는 것을 알 수 있었다. 이러한 실험 결과로부터 모바일 GPU가 스마트폰에서 소비하는 전력 에너지의 양이 상당하고 전경 응용의 성능 하락이 없는 선에서 전력 소비를 최소화하여 에너지 효율 향상의 가능성을 알 수 있다.

## 1.3 연구 기여

연구의 핵심 기여는 두 가지로 정리할 수 있다.

첫 번째로, GPU의 장치 사용률(Device Utilization) 뿐만 아니라 스마트폰 환경에서 응용의 서비스 품질 평가 지표로써 FPS와 배터리 시간을 함께 고려한 모바일 GPU 에너지 최적화 기법을 제안한 점이다. 기존 모바일 GPU 에너지 최적화 기법들은 장치 사용률(Utilization)을 중심으로 단순한 DVFS를 구현하였으나, 본 연구의 실험을 통하여 장치 사용률이 실제 사용자가 느끼는 성능을 직접적으로 반영하지 못하는 문제점을 밝힌다. 이러한 문제점을 해결하기 위하여 장치 사용률(Utilization)뿐만 아니라 사용자가 실제로 체감하는 성능 지표로써 FPS를 함께 고려한 전력 에너지 최적화 기법을 제안한다. 추가로, 스마트폰의 사용 시각 측면에서 사용자 경험을 해치지 않도록 배터리 잔량이 일정량 이하가 되면 어느 정도의 프레임 수 하락을 허용하여 응용의 사용 시간을 연장할 수 있는 기법을 제안한다.

두 번째로 실제 스마트폰에 탑재된 모바일 GPU를 대상으로 성능 하락 없이 실측 GPU 전력을 줄인 점이다. 기존의 모바일 GPU와는 다르게 스마트폰에 탑재된 대부분의 모바일 GPU는 제조사의 IP(Intellectual Property)에 의하여 내부 구현이 공개되어 있지 않은 상황이다. 따라서



기존의 많은 모바일 GPU를 대상으로 한 전력 에너지 최적화 기법 [5]을 적용하기 힘들다. 따라서 본 연구는 스마트폰에 탑재된 모바일 GPU를 대상으로 한 에너지 최적화 기법 개발이 제한적인 상황에서 제조사들이 장치 드라이버를 통하여 일부 공개한 단순한 전력 관리 기법을 중심으로 이에 대한 성능을 평가하고 모바일 GPU 특성에 알맞는 에너지 최적화 기법을 제안한다.

## 1. 4 논문 구성

본 논문의 구성은 다음과 같다. 먼저 2장에서는 대상 모바일 GPU인 ARM Mali-400MP의 장치 드라이버에서 제공하는 전력 관리 기법인 장치 사용률(Utilization) 기반의 DVFS를 먼저 평가하여 문제점을 밝히고 개선의 필요성을 설명한다. 3장에서는 서비스 품질의 요소로써 초당 프레임 수와 배터리 시간을 함께 고려한 에너지 최적화 기법에 대해 설명한다. 4장에서는 기존 장치 드라이버에서 제공한 DVFS 알고리즘과 본 논문이 제안한 에너지 최적화 기법의 전력 소모량과 성능을 평가하고 마지막으로 5장에서 결론을 맺는다.

## 제 2 장 장치 사용률 기반 DVFS의 평가

### 2.1 장치 사용률 기반 DVFS

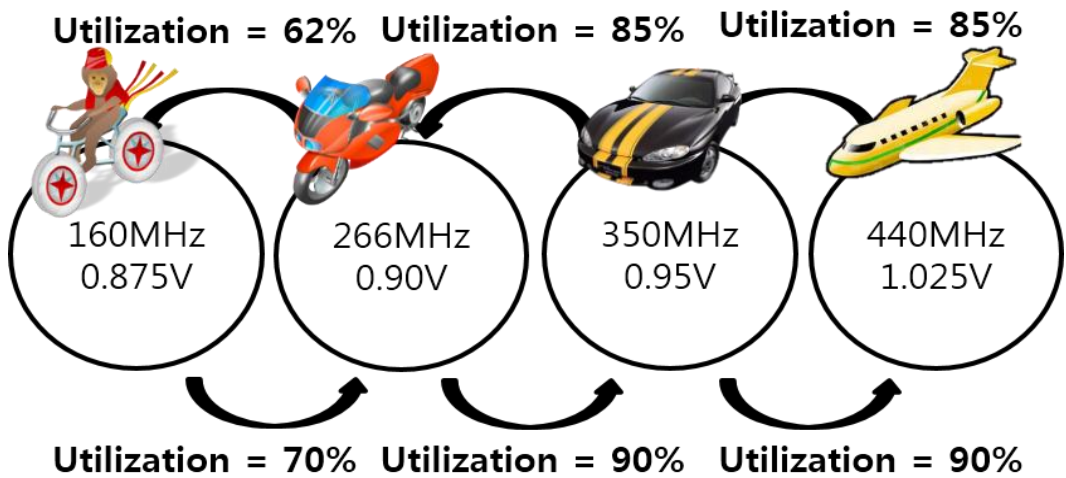


그림 2 ARM Mali-400MP에서 기본으로 제공하는  
장치 사용률(Utilization) 기반의 DVFS 알고리즘

본 논문의 연구 대상 모바일 GPU로 최근 스마트폰 및 태블릿 PC에서 널리 사용되는 ARM Mali-400MP를 선정하였으며 장치 드라이버를 분석하였다. 분석 결과, 장치 드라이버에서 기본적으로 장치 사용률 기반의 DVFS를 제공하며, 그 알고리즘은 그림 2와 같다. 기본으로 총

4개의 동작 주파수 (160MHz, 266MHz, 350MHz, 440MHz) 및 전압 상태(0.875V, 0.90V, 0.95V, 1.025V)가 제공되고, 단위 시간마다 GPU 내부에서 1개 이상의 코어가 동작한 시간으로 장치 사용률을 산출하며, 이를 이용하여 미리 정의된 장치 사용률 임계값(Utilization Threshold)과 비교하여 동작 주파수 및 전압을 결정한다. 예를 들어 현재 동작 주파수가 160MHz일 때, 단위 시간 동안 산출한 장치 사용률 값이 70%를 넘는다면 GPU 부하가 현재 주파수로 처리하기에는 무리라고 판단하여 266MHz로 조절한다. 이 때 몇 가지 게임응용을 2분씩 실행하며 시간에 따른 DVFS에 의한 주파수 천이를 관찰하였다.

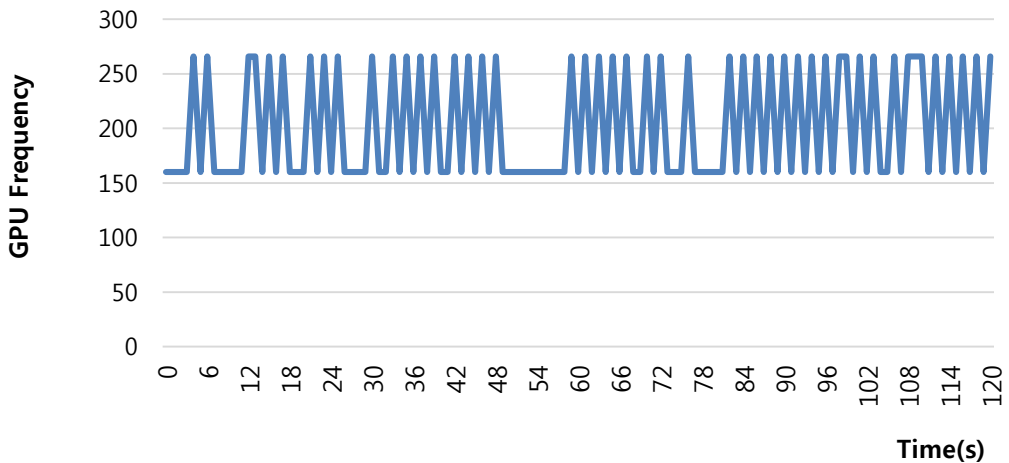


그림 3 기존 DVFS 기법에서 Pixelgun3D 실행 시 시간에 따른 주파수 천이

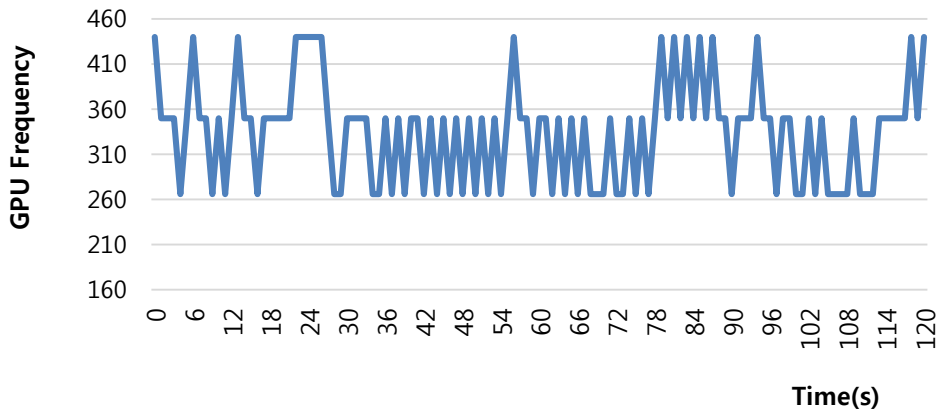


그림 4 기존 DVFS 기법에서 Minecraft 실행 시 시간에 따른 주파수 천이

그림 3과 4는 게임 응용 Minecraft 및 Pixelgun3D를 2분간 실행하며 주파수 천이를 나타낸 것이다. 결과에서 알 수 있는 특징으로 상위 주파수로 천이하자마자 다시 하위 주파수로 바뀌는 출렁임이 상당히 많은 것을 알 수 있다. 이는 그림 1의 알고리즘에서 GPU 부하의 양이 현재 주파수 영역에서 처리할 수 있음에도 불구하고 공격적인 DVFS를 위하여 동작 주파수 레벨을 상승시켜 나타난 결과로 볼 수 있다. 이로 미루어 볼 때 그림 3과 4의 출렁임이 존재하는 부분을 성능 하락이 없이 처리 가능하다면 낮은 주파수 영역에서 실행하게 함으로써 전력 이득을 얻을 것으로 기대 된다. 이를 확인하기 위하여 7개의 응용을 대상으로 4개의 주파수 대역에서 각각의 주파수를 고정시켜 성능 하락에 대해서

관찰하였으며 그 결과는 표 1과 같다. 이 때 성능 지표로 GPU의 연산과 가장 관련이 깊으며 사용자가 느낄 수 있는 초당 프레임(FPS)를 이용하였다.

표 1 동작 주파수 고정 시 게임 응용의 FPS 분포

	Bus Simulator	Counter Strike	Fireturck park	Minecraft	Pixelgun	Riptide	Truck Park
FPS@160MHz	49.52	<b>59.8</b>	52.6	38.2	<b>58.38</b>	39.9	33.07
FPS@266MHz	<b>57.08</b>	59.84	<b>58.3</b>	<b>57</b>	58.47	49.76	46.81
FPS@350MHz	58.29	59.86	59.04	58.58	58.45	<b>52.51</b>	<b>58.39</b>
FPS@440MHz	59.53	59.93	59.44	59.49	58.85	51.41	58.5
FPS@DVFS	58.75	59.76	58.7	58.6	59.1	52.1	57.9
DVFS 시 평균 Frequency	<b>333.37</b>	<b>205.55</b>	<b>274.48</b>	<b>353.32</b>	<b>210.8</b>	<b>396.39</b>	<b>354.66</b>

표 1을 통하여 알 수 있는 점은 장치 사용률 기반의 DVFS 알고리즘 적용 시 평균 주파수보다 작은 주파수 영역에서도 FPS의 하락 없이 성능이 유지된다는 점으로 음영 표시된 부분으로부터 알 수 있다. 이를 미루어보아 기존 장치 사용률 기반의 DVFS에서 공격적으로 설정된

사용률 임계 값으로 인하여 현재 부하가 요구하는 것 이상으로 동작 주파수를 설정하여 비효율적인 전력 소모량이 발생하는 것을 알 수 있으며, 장치 사용률을 이용한 DVFS로부터 사용자가 실제 느끼는 성능이 직접적으로 연관 짓기 힘들다는 것을 알 수 있다.

## 제 3 장 QoS를 고려한 스마트폰

### GPU 에너지 최적화 기법

#### 3. 1 QoS를 GPU 에너지 최적화 기법의 알고리즘

앞서 제기한 기존 장치 사용률 기반의 DVFS의 문제점을 해결하기 위하여 본 절에서는 QoS를 고려한 스마트폰 GPU 에너지 최적화 기법을 제안한다.

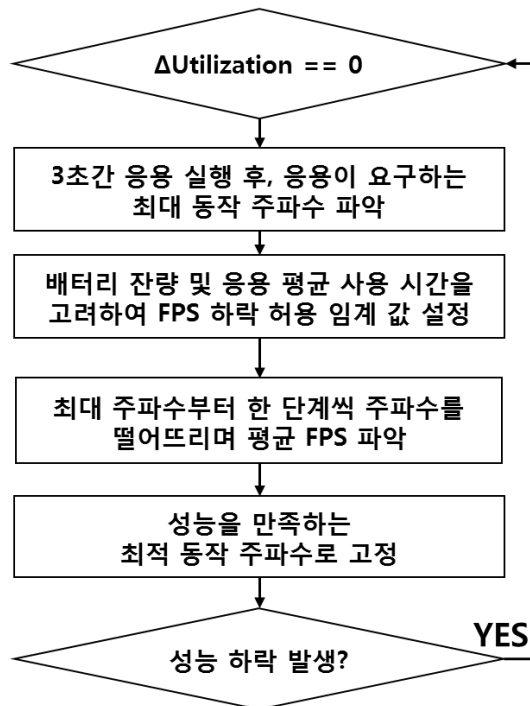


그림 5 QoS를 고려한 에너지 최적화 기법 플로우 차트



그림 5는 제안 기법 알고리즘의 플로우 차트이다. 알고리즘은 GPU가 유휴상태(IDLE)로부터 부하의 처리가 요청될 때 시작되며 이는 장치 사용률이 0으로부터 변화가 있음을 통하여 감지한다. 먼저 3초간 기존 장치 사용률 기반의 DVFS를 실행하며 현재 응용의 현재 장면에서 요구하는 최대 동작 주파수와 FPS를 파악한다. 그 후, 배터리의 잔량과 응용의 평균 사용 시간을 고려하여 FPS 최대 하락 허용 값을 설정한다. 배터리의 잔량이 충분할 때에는 평균 FPS의 5%(60프레임을 요구하는 응용의 경우 최대 2프레임 이하까지 허용)까지 하락을 허용하고 배터리의 잔량이 10% 이하로 부족할 때에는 5%보다 큰 범위까지 FPS 하락을 허용한다. 그 다음, 파악한 최대 동작 주파수로부터 한 단계씩 주파수를 떨어뜨려 3초간 고정시킨 후 평균 FPS를 파악한다. 이 때 고정된 주파수에서 FPS가 허용한 범위보다 큰 하락이 있으면 한 단계 위의 주파수를 성능 하락 없이 전력 소모가 최소인 최적 동작 주파수로 판단한다. 또한 현재 주파수에서 FPS의 하락이 없더라도 동작 주파수 간 장치 사용률 예측을 통하여 다음 한 단계 하위 동작 주파수에서 성능 하락이 있을 것으로 예측되면 현재 주파수를 최적 동작 주파수로 판단한다. 마지막으로 파악한 최적 동작 주파수로 GPU를 동작시킨다. 그 후 단위 시간마다 FPS의 하락이 있는지, 장치 사용률이 기존 DVFS의 하락 임계 값보다 낮은지 체크한다. FPS의 하락이 발생하거나 장치

사용률의 값이 임계 값보다 낮으면 부하의 특성이 바뀐 것으로 간주하여 성능을 만족시키는 최적 주파수를 파악하기 위하여 알고리즘을 처음부터 다시 시작한다. 위와 같은 알고리즘을 통하여 FPS와 배터리 시간의 QoS를 일정 수준으로 보장하고 모바일 GPU의 소모 에너지가 최적의 효율을 제공할 수 있도록 한다.

### 3.2 동작 주파수 간 장치 사용률 예측

앞서 제안한 기법에서 주파수 별 FPS를 파악할 때, 지속적으로 주파수를 한 단계씩 낮추다 보면 성능하락이 발생하는 주파수까지 도달할 수 있다. 따라서 본 절에서는 FPS 파악 중 장치 사용률을 이용하여 성능하락이 발생하는 않는 주파수까지 FPS를 파악하도록 하는 동작 주파수 간 장치 사용률 예측에 대하여 기술한다.

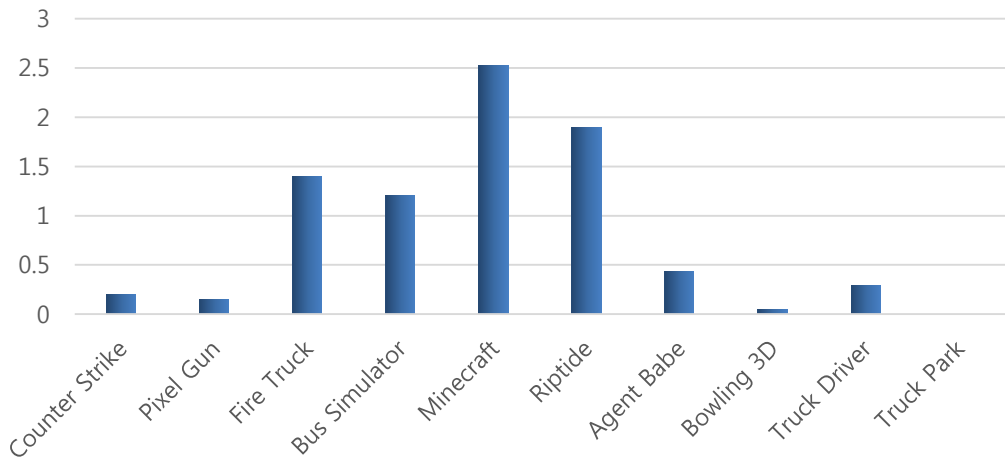


그림 6 장치 사용률 예측이 적용되지 않은 제안 기법과 기존 DVFS 기법의 FPS 차이

그림 6은 주파수 대역별 FPS 하락에 대한 예측 없이 응용이 요구하는 최대 동작 주파수로부터 가장 낮은 주파수(160MHz)까지 FPS를 파악

할 때 기존 장치 사용률 기반 DVFS 대비 FPS 하락을 보인 그래프로 11개의 게임 응용을 2분 동안 실행 하였을 때 FPS를 평균 낸 값을 서로 비교한 것이다. 이 때 성능 하락은 평균 0.8 최대 2.53 프레임만큼의 하락을 보였다.

프레임 하락의 주요 원인으로서는 제안 기법의 알고리즘을 통하여 결정된 최적 동작 주파수가 응용이 요구하는 부하의 특성이 바뀌어 더 이상 최적의 주파수가 아닌 경우와 FPS의 하락이 나타나는 지점까지 FPS 학습을 하는 경우에 발생하였다. 따라서 성능 하락이 없는 지점까지만 FPS를 파악하여 성능을 개선할 필요가 있다.

이를 위하여 한 단계 아래 주파수에서 성능 하락을 예측하는 방법으로 GPU 장치 사용률을 이용한다. 즉, 현재 GPU의 장치 사용률을 이용하여 다음 주파수 영역에서 장치 사용률이 100%를 초과할 것으로 예상된다면 성능 하락이 발생하는 것으로 판단하고 학습을 중단한다. 따라서 주파수 별 GPU의 상대적인 성능 차이를 파악할 필요가 있다. 표 2는 7개의 게임 응용을 4개의 주파수 영역에서 각각 고정시키고 2분 동안 실행하는 동안 주파수 간 평균 GPU의 처리량의 비를 구한 것이다. 이 때 평균 GPU의 처리량은 평균 GPU 장치 사용률의 역수이며 인접한 주파수 영역 사이에서의 GPU 처리량의 비를 구한 것이다. 예를 들어 표2의 Pixelgun의 266MHz/160MHz 평균 GPU 처리량의 비는 해당 응용이

요구하는 GPU 연산 부하를 266MHz 주파수 영역에서 평균적으로 1.39배 더 처리한다는 의미이다.

표 2 인접 동작 주파수 간 GPU 처리량의 상관관계

	266MHz/160MHz 평균 GPU 처리량의 비	350MHz/266MHz 평균 GPU 처리량의 비	440MHz/350MHz 평균 GPU 처리량의 비
Pixelgun	1.396892	1.174364	1.006554
Truck Parking	1.086265	1.047223	1.130351
Riptide	1.026568	1.070847	1.082357
Minecraft	1.045459	1.092878	1.14873
Firetruck Parking	1.223403	1.158902	1.112028
Bus Simulator	1.109841	1.073794	1.117688
Counter Strike	1.410116	1.196575	1.168586
처리량의 평균비	1.18	1.11	1.11

표 3 다음 주파수 영역에서 성능 하락이 예상되는 현재 주파수 영역에서의 GPU 장치 사용률

266MHz	350MHz	440MHz
85%	90%	90%

$$Utilization_{curfreq-1} = \frac{Utilization_{curfreq}}{Average\ GPU\ Throughput\ Ratio} \quad (1)$$

표 3은 앞서 표 2에서 구한 인접 동작 주파수 간 GPU 처리량의 평균 비를 식 (1)과 같이 계산하여 한 단계 아래 인접 동작 주파수에서 GPU 장치 사용률이 100%으로 예측될 때 현재 주파수 영역에서의 GPU 장치 사용률을 구한 것이다. 이를 이용하여 현재 동작 주파수에서 FPS 파악을 하고 평균 GPU 장치 사용률을 구하였을 때, 표 3의 값보다 크거나 같으면 다음 주파수 영역에서 성능 하락이 있는 것으로 간주하고 FPS 파악을 중단하여 성능 하락을 방지한다.

### 3. 3 배터리 잔량을 고려한 FPS 최대 하락 허용 값의 동적 조절 기법

배터리의 잔량이 부족한 경우 특정 응용에 대하여 사용자가 사용하는 평균 시간을 보장해주기 위하여 어느 정도 성능 저하를 감수함으로써 그 목표를 이룰 수 있다. 따라서 본 연구에서 제안하는 전력 에너지 최적화 기법에서 일정 수의 프레임 하락을 허용함으로써 배터리 시간을 연장할 수 있다. 이를 위해 QoS를 고려한 GPU 에너지 최적화 알고리즘 그림 5에서 FPS 최대 하락 허용 값을 기존 95%보다 낮게 설정하여 최적 동작 주파수를 선정하도록 한다.

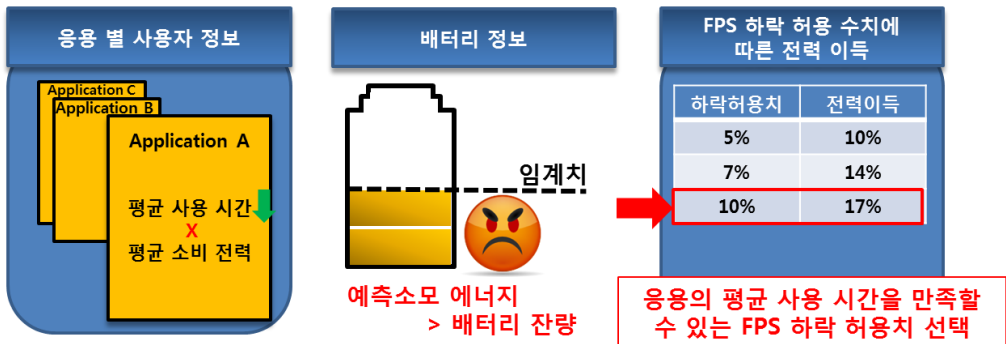


그림 7 배터리 잔량을 고려한 FPS 최대 하락 허용 값의 동적 조절 기법 구조도

그림 7은 FPS 최대 하락 허용 값을 동적으로 조절하는 기법의 구조도이다. 구체적으로 FPS 최대 하락 허용 값은 현재 배터리 잔량이 응용이 사용하게 될 에너지 예측 값보다 작으면 동적으로 조절하도록 한다. 이 때 응용의 예측 소모 에너지는 응용의 평균 사용 시간과 평균 소비 전력을 이용하여 산출하고, 이를 위하여 GPU 장치 드라이버 내에서 응용의 평균 사용 시간 및 평균 소비 전력을 수집하도록 구현하였다.

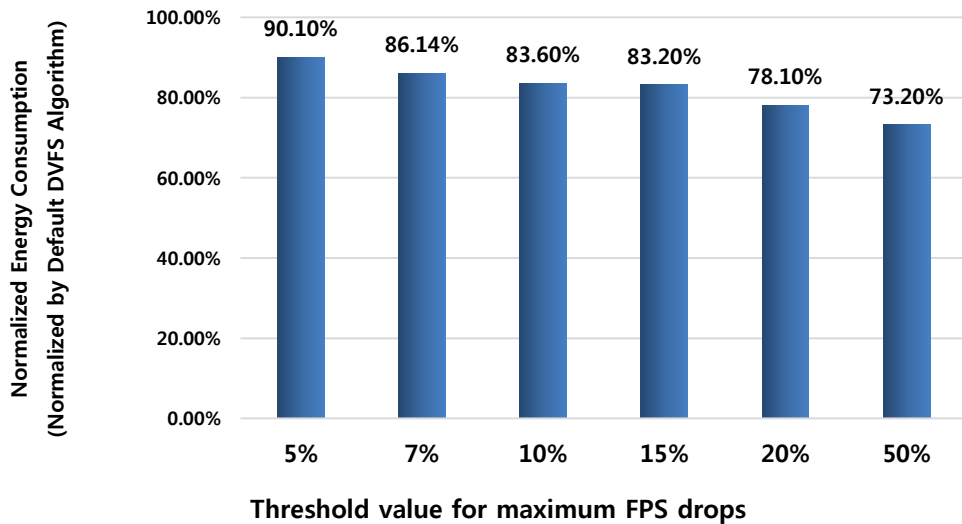


그림 8 FPS 최대 하락 허용 값에 따른 기존 장치 사용률 기반의 DVFS 대비 제안 기법의 에너지 소모량



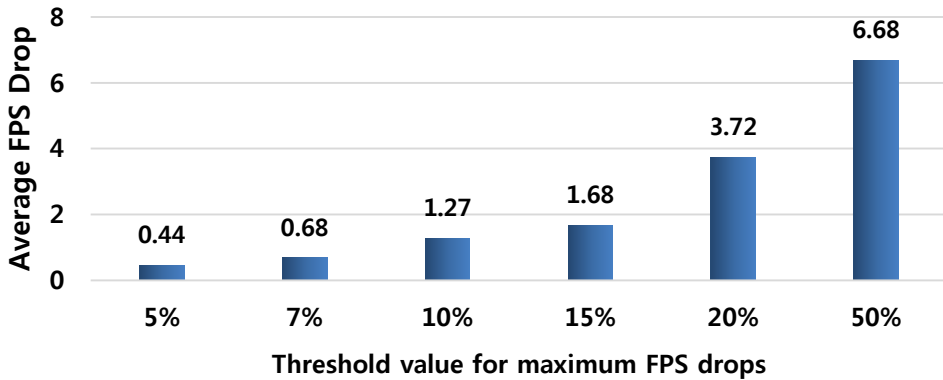


그림 9 FPS 최대 하락 허용 값에 따른 기존 장치  
사용률 기반의 DVFS 대비 제안 기법의 평균 FPS 차이

FPS 최대 하락 허용 값의 결정은 허용 값에 따른 에너지 이득을 이용하여 해당 범위까지 FPS 하락을 허용한 경우 예측되는 응용의 에너지 소모량이 배터리 잔량보다 작은 값으로 결정한다. 이 때 FPS 최대 하락 허용 값에 따른 에너지 이득은 실험을 통한 오프라인 분석으로 결정하였으며 그 결과로 그림 8과 9는 11개의 응용을 대상으로 배터리의 잔량이 10% 이하일 때, FPS 최대 하락 허용 값을 다양하게 변화시키며 기존 DVFS 기법 대비 소모 에너지와 평균 FPS 하락을 측정하였다. 실험 결과 허용 값이 증가함에 따라 에너지 이득과 평균 FPS 하락 값은 증가하였으며 사용자 경험을 크게 해치지 않는 10% 정도 까지가 현실적인 허용수치가 될 것으로 예상된다. 10%만큼 하락을 허용할 경우 기본 5%

만큼 허용할 때보다 증가된 평균 16.4%의 에너지 이득으로 배터리 시간 연장이 가능할 것으로 예상되어 FPS 최대 하락 허용 값은 5%~10%내에서 동적 조절하도록 하였다. 이러한 기법으로 배터리 잔량을 고려하여 응용의 평균 사용 시간에 대한 서비스 품질을 유지할 수 있도록 하였다.

## 제 4 장 실험 결과

### 4.1 실험 환경

실험은 ARM Mali-400MP가 탑재된 갤럭시 S3 개발 보드에서 진행되었다. 기존 GPU 장치 사용률 기반의 DVFS와 제안한 FPS 학습 기반의 전력 관리 기법을 비교하기 위하여 Google Play Store로부터 11개의 게임 응용을 선정하고 2분 동안 플레이할 때 소비된 전력 에너지와 성능 지표인 평균 FPS를 산출하였다. 이 때 게임은 최대한 동일한 시나리오로 수행하였다.

실험을 진행한 개발 보드에서는 5개 컴포넌트(CPU, GPU, Memory Controller, Memory, Co-processor Bus) 각각의 실측 전력 소모량을 측정할 수 있도록 단자(Shunt)가 노출되어있다. 전력 실측은 NI사의 DAQ USB-6289를 이용하였으며 초당 1000번의 전압 전류 샘플을 평균 내어 소비 전력을 산출하였으며 이를 이용하여 2분 동안 소모 에너지를 구하였다.

성능 지표인 FPS는 Android 프레임 워크의 SurfaceFlinger에서 화면을 그리는 함수의 초당 요청 횟수를 이용하여 산출하였다.

마지막으로 배터리 잔량을 고려하여 FPS 최대 하락 허용 값 동적 조절

실험을 위하여 배터리 잔량 모델로 갤럭시 S3의 배터리 용량의 27%만큼인 7756560mJ만큼을 GPU 에너지의 총량(Energy Budget)으로 설정하였다. 이 때 FPS 최대 하락 허용 값은 GPU 에너지 총량의 10% 이하일 때 5%, 7%, 10% 중 소모 에너지를 예측계산하고 GPU 에너지 총량과 비교하여 동적 조절하도록 하였다. 마지막으로 응용 별 평균 사용 시간과 평균 소모 전력의 학습 데이터는 다음 표 4와 같다.

표 4 FPS 최대 하락 허용 값 동적 조절을 위한 응용 별 평균 사용 시간 및 소모 전력 학습 데이터

	평균 사용 시간	평균 소모 전력
Counter Strike	102	144
Pixel Gun	88	172
Fire Truck	75	178
Bus Simulator	53	269
Riptide	35	420
Agent Babe	67	217
Bowling 3D	70	210
Truck Driver	54	274
Truck Park	23	608
Beach Bug	30	518
Minecraft	65	210

## 4.2 실험 결과

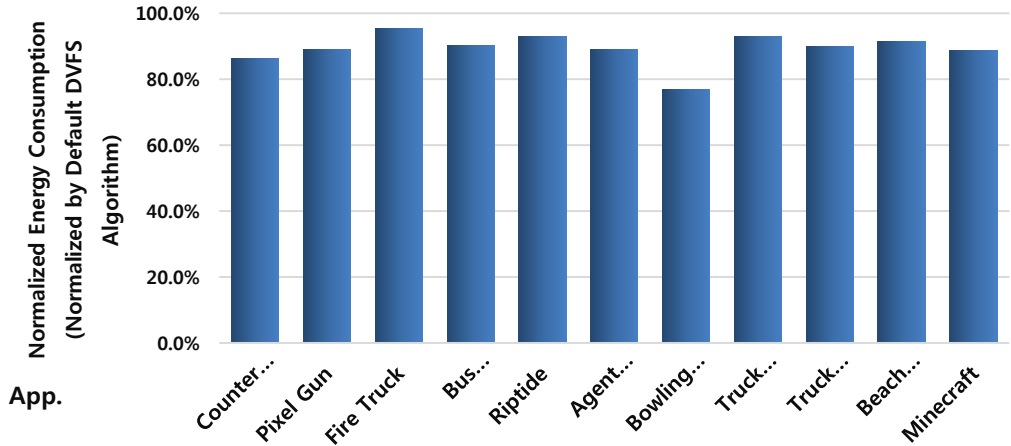


그림 10 기존 GPU 장치 사용률 기반의 DVFS 대비  
제안 기법의 소모 에너지

표 5 기존 GPU 장치 사용률 기반의 DVFS 대비 FPS 학습 기반 전력  
관리 기법의 에너지 이득

에너지 이득 (%)	
Counter Strike	4.3%
Pixel Gun	10.9%
Fire Truck	7.8%
Bus Simulator	9.0%
Riptide	9.6%
Agent Babe	19.8%
Bowling 3D	23.0%
Truck Driver	7.5%
Truck Park	13.6%
Beach Bug	4.7%
Minecraft	14.0%

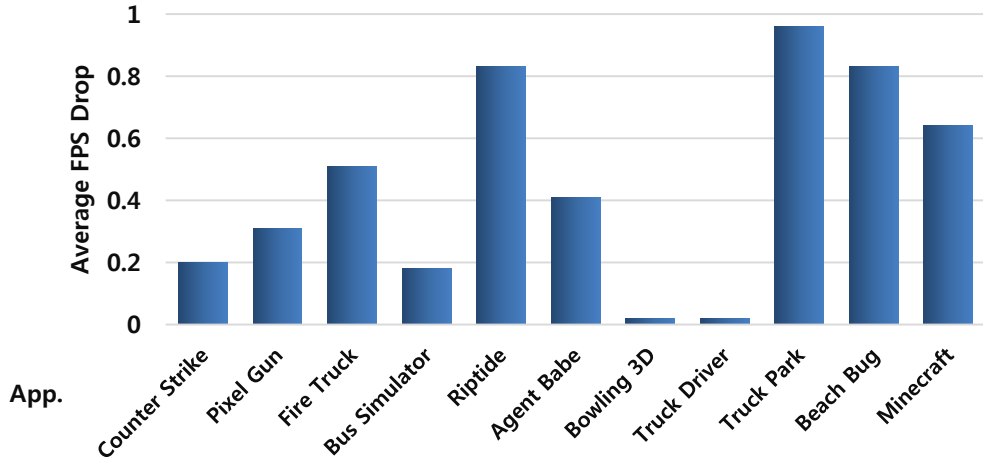


그림 11 기존 GPU 장치 사용률 기반의 DVFS 대비 FPS 학습 기반 전력 관리 기법의 평균 FPS 차이

표 6 기존 GPU 장치 사용률 기반의 DVFS 대비 FPS 학습 기반 전력 관리 기법의 평균 FPS 하락

FPS 하락	
Counter Strike	0.2
Pixel Gun	0.31
Fire Truck	0.51
Bus Simulator	0.18
Riptide	0.83
Agent Babe	0.41
Bowling 3D	0.02
Truck Driver	0.02
Truck Park	0.96
Beach Bug	0.83
Minecraft	0.64

그림 10과 표 5는 기존 GPU 장치 드라이버에서 제공하는 장치 사용률 기반 DVFS 기법과 본 논문에서 제안한 FPS 학습 기반의 전력 기법의 전력 에너지를 비교한 것이다. 그 결과 평균 9.9% 최대 23.0%의 에너지 이득을 얻을 수 있었다.

이와 함께 성능 차이를 비교하기 위하여 평균 FPS를 비교하였으며 그 결과는 그림 11과 표 6과 같다. 그 결과 평균 0.42만큼의 1 프레임 미만의 저하가 나타났으며 이는 성능 하락이 없는 미미한 수치로 사용자 경험을 저해하지 않은 것을 알 수 있다.

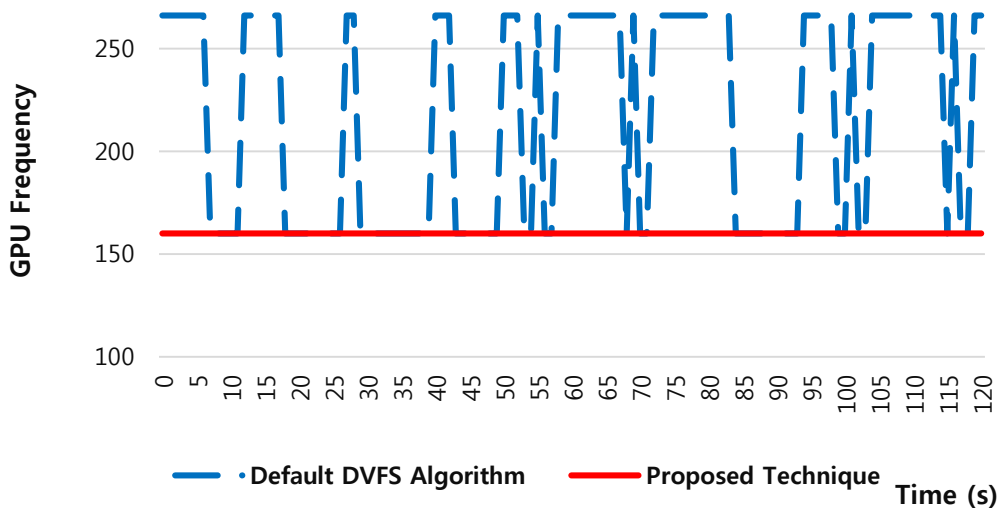


그림 12 Bowling 3D를 플레이 할 때 기존 GPU 장치 사용률 기반 DVFS와 제안 기법의 주파수 천이 비교

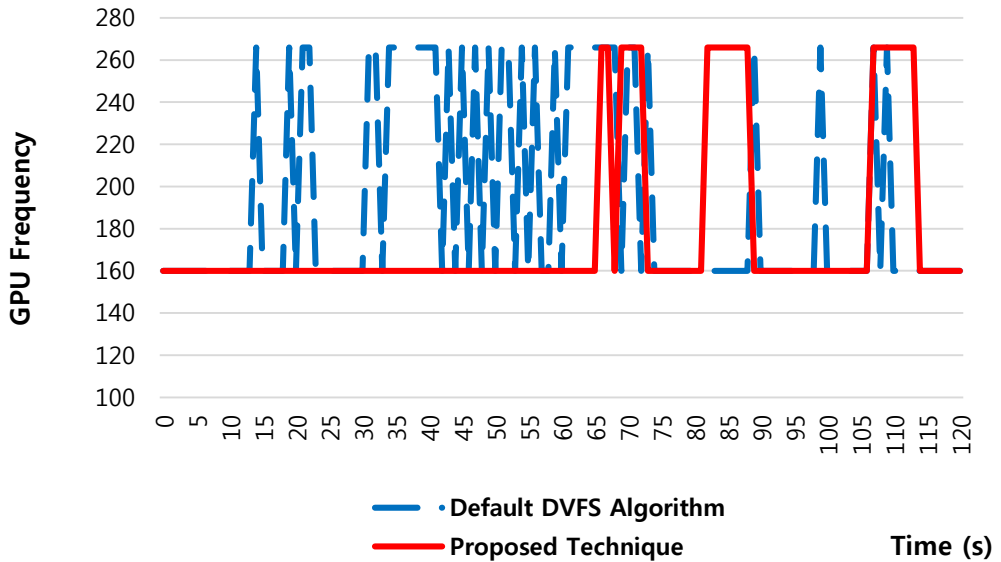


그림 13 Counter Strike를 플레이 할 때 기존 GPU 장치 사용률 기반 DVFS와 제안 기법의 주파수 천이 비교

그림 12, 13은 게임 응용 Bowling 3D과 Counter Strike를 기존 장치 사용률 기반 DVFS와 제안 기법을 각각 적용하여 2분 동안 수행하며 얻은 주파수 천이의 결과이다. 결과에서 볼 수 있듯이 제안 기법에서 실행한 두 응용에서 GPU의 동작 주파수 160MHz에서 266MHz로의 상승 수가 줄어든 것을 알 수 있다. 이로써 기존 DVFS에서 발생한 비효율적인 주파수 상승을 방지하여 성능 하락 없이 에너지 이득을 얻을 수 있었다.



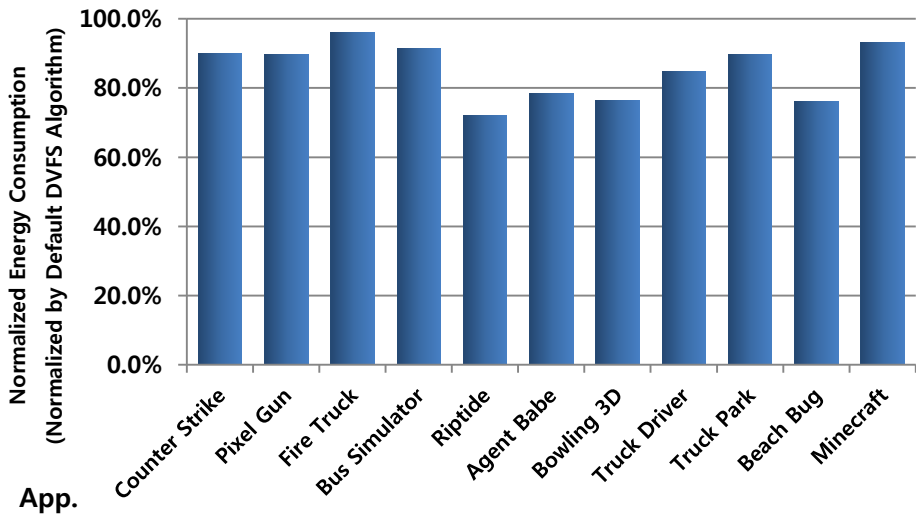


그림 14 FPS 최대 하락 허용 값 동적 조절에 따른 기존 DVFS 대비 에너지 소모량

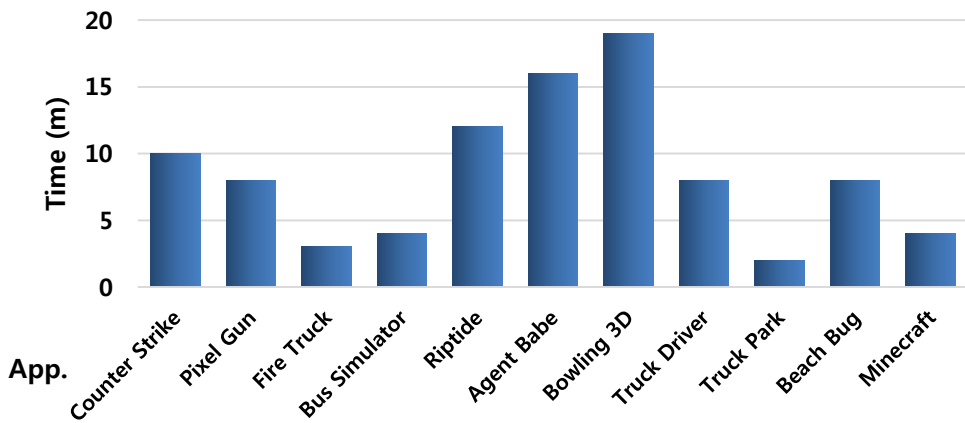


그림 15 FPS 최대 하락 허용 값 동적 조절에 배터리 연장 시간

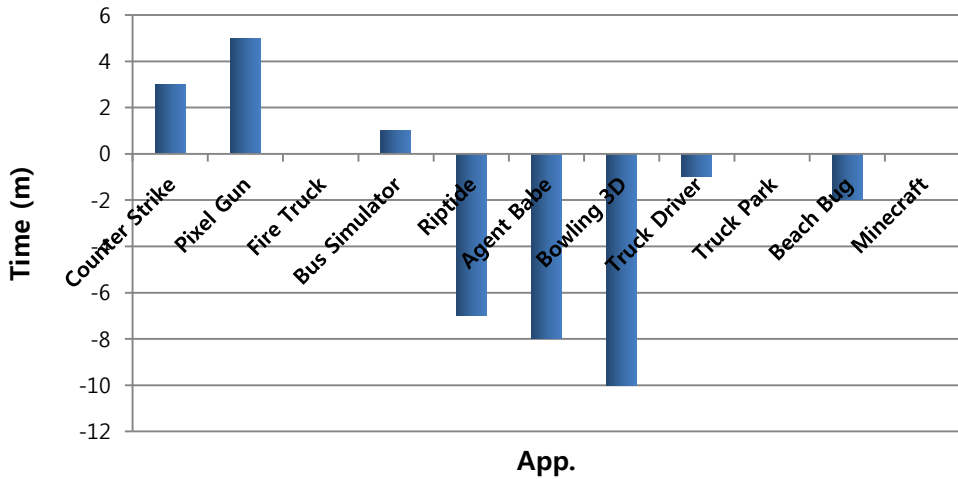


그림 16 FPS 최대 하락 허용 값 동적 조절에 따른 연장 시간과 사용자 평균 사용 시간과의 차이

그림 14-16 은 배터리의 잔량이 10%이하일 때 FPS 최대 하락 허용 값을 동적 조절한 실험 결과이다. 그림 14, 15 에서와 같이 평균 12.6%의 에너지 이득을 보였으며 배터리 시간은 Bowling 3D 응용에서 최대 19분 연장이 가능하였다. 이 때 실제 연장된 배터리 시간과 표 4의 입력한 평균 사용 시간과의 차를 그림 16 과 같이 구하였다. 이 때 5개의 응용(Riptide, Agent Babe, Bowling 3D, Truck Driver, Beach Bug)에서 예측한 에너지 이득 값보다 실제 이득이 큰 값을 가지며 사용자의 평균 사용 시간보다 긴 시간을 보장할 수 있었다.

## 제 5 장 결 론

### 5. 1 결론

최근 모바일 GPU가 고성능화됨에 따라 소모하는 전력이 스마트폰 전체 전력 소모에서 차지하는 비중이 증가하여 중요한 전력 관리 대상으로 대두되었다. 따라서 본 논문은 ARM Mali-400MP를 전력 관리 연구 대상 모바일 GPU로 선정하여 장치 드라이버를 분석하였다. 그 결과 기본으로 제공되는 전력 에너지 최적화 기법인 장치 사용률 기반의 DVFS에서 사용자가 실질적으로 인지할 수 있는 전경 응용의 FPS 하락이 없음에도 불구하고 동작 주파수를 상승시켜 비효율적인 에너지 소모가 있었음을 밝혔다.

따라서 본 논문은 GPU의 장치 사용률뿐만 아니라 FPS와 배터리 잔량을 함께 고려하여 응용의 성능을 유지하며 전력 소모가 최소인 주파수로 GPU를 동작하도록 하는 기법을 제안함으로써 QoS를 보장하고 높은 에너지 효율을 제공할 수 있도록 하였다. 제안한 방법을 평가한 결과, 기존 DVFS 기법 대비 에너지 이득은 평균 9.9% 최대 23%를 얻을 수 있었으며 FPS는 평균 0.44만큼의 하락을 보여 사용자 경험을

크게 저해하지 않았다. 추가로 배터리의 잔량과 응용의 평균 사용 시간을 함께 고려하여 FPS 최대 하락 허용 값을 동적으로 조절할 경우, 평균 8.5분의 배터리 시간 연장이 가능하였다.

## 5. 2 향후 연구

본 논문에서 제안한 전력 최적화 기법은 GPU의 장치 사용률과 사용자 중심의 FPS를 함께 고려하여 성능 저하 없이 에너지 이득을 얻을 수 있는 기법이다. 추가적으로 가능한 개선 연구로써 주파수 간 장치 사용률 예측을 통하여 다른 주파수로 천이 시 성능 하락에 대한 예측을 3.2절에서와 같이 여러 응용을 주파수 대역 별로 수행하여 얻은 데이터를 기반으로 처리량의 단순 평균을 이용하였는데, 이보다 정확한 모델링을 통하여 그 정확도를 높일 수 있을 것으로 기대된다. 예를 들어 GPU의 부하 특성을 세분화하여 Geometry 중심의 연산인지 Pixel 연산인지에 따라 주파수 별 처리량이 바뀔 수 있기 때문에 이러한 특성을 고려하여 보다 정확한 성능 예측을 할 수 있을 것으로 기대된다.

또한 FPS 최대 하락 허용 값의 동적 조절 기법에서 허용 값에 따른 에너지 이득을 오프라인 분석을 하였는데, 이를 응용 별로 온라인 분석을 함으로써 보다 정확하게 배터리 시간 연장을 할 수 있을 것으로 기대된다.

마지막으로 현재 연구는 ARM Mali-400MP를 기반으로 하여 커널 및 프레임워크 수정이 불가피하나 기법을 보다 범용성 있게 적용할 수 있도록 모듈로 개발하여 다양한 모바일 GPU에서 적용될 수 있도록 연구를 할 예정이다.

## 참고문헌

- [1] Kim, M. and Chung, S., “Accurate GPU Power Estimation for Mobile Device Power Profiling” , *Proc. of 2013 IEEE International Conference on Consumer Electronics*, 2013.
- [2] Yoon, C., Kim, D., Jung, W., and Kang, C., “Appscope: Application Energy Metering Framework for Android Smartphone using Kernel Activity Monitoring” , *Proc. of USENIX Annual Technical Conference*, 2012.
- [3] W. Jung, C. Kang, C. Yoon, D. Kim, and H. Cha, “Devscope: A Non-intrusive and online power analysis for smartphone hardware components” , *Proc. of the 8th IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, 2012.
- [4] Wang, C., F. Yan, Y. Guo, and X Chen, “Power Estimation for Mobile Applications with Profile-Driven Battery Traces,” *Proc. of Low Power Electronics and Design*, 2013.
- [5] C. Park, H. Kim, J. Kim, “A Low-Power Implementation of 3D Graphics System for Embedded Mobile Systems” *Proc. of 2006 IEEE Workshop on Embedded Systems for Real Time Multimedia*, 2006.

# Abstract

## A Qos-Aware Energy Optimization Technique for Smartphone GPUs

Dohan Kim

Dept. of Computer Science and Engineering

College of Engineering

The Graduate School

Seoul National University

The power consumption of the mobile GPU in the smart phone has been increased to improve performance of the GPU. Therefore it is important to optimize the power consumption of the mobile GPU. However, when we analyze the device driver of ARM Mali-400MP, the existing power management technique shows the inefficiency increment of the operating frequency, despite the current operating frequency can handle the GPU workload.

Therefore, we propose a novel energy optimization technique for smartphone GPUs, which lowers the GPU frequency more aggressively for a higher energy efficiency with a negligible negative impact on the GPU performance. In order to achieve the Quality of Service (QoS) specified by the smartphone application, the proposed optimization technique employs the minimal acceptable GPU frequency based on average Frames per Second (FPS) for each GPU frequency level. Additionally,

when the battery capacity is low, the technique drops some frames to extend the smartphone battery life.

Our experimental results on a smartphone development board show that the proposed technique can reduce the GPU energy consumption by up to 23% over the default DVFS algorithm with only a 0.45 frame drop. When the battery level is low and the technique allows some FPS drop, we can extend battery time by up to 19 minutes.

**keywords : Mobile GPU, Quality of Service, Frames Per Second, Battery Time**  
**Student number : 2013-20752**