

<https://doi.org/10.7236/JIIBC.2024.24.2.35>
JIIBC 2024-2-6

다초점 촬영과 초점후처리 기능을 가진 스마트폰 카메라 앱의 성능평가

Performance Evaluation of Smartphone Camera App with Multi-Focus Shooting and Focus Post-processing Functions

박채원*, 김경미*, 유송연*, 김유진*, 황기태**, 정인환***, 이재문***

Chae-Won Park*, Kyung-Mi Kim*, Song-Yeon Yoo*, Yu-Jin Kim*,
Kitae Hwang**, In-Hwang Jung***, Jae-Moon Lee***

요약 본 논문은 이전 연구에서 구현된 OnePIC 앱의 실행 성능과 저장 성능을 분석하여 OnePIC 앱의 실용성을 검증한다. OnePIC 앱은 다양한 초점을 가진 사진들을 촬영한 후, 원하는 초점의 사진을 얻을 수 있는 카메라 앱이다. 본 논문은 성능평가를 위해 거리별 다초점 촬영 시간과 객체별 다초점 촬영 시간을 세부적으로 분석하였다. 성능평가는 실제 스마트폰에서 실측하는 방식으로 진행되었다. 거리별 다초점 촬영 시간은 5장에 0.84초, 객체 감지 시간은 객체 개수와 관계없이 0.19초, 객체별 다초점 촬영 시간은 5장에 4.84초 정도로 측정되었다. 다초점 사진을 한 장에 저장한 All-in-JPEG 파일 크기와 각각 JPEG 파일로 저장한 경우를 비교한 결과, All-in-JPEG 파일 크기가 미묘하게 줄어 저장 공간의 큰 이득은 없었다. 그렇지만 All-in-JPEG은 다초점 촬영된 사진들의 관리 면에서 매우 용이하다. 결론적으로 본 논문의 성능평가 결과, OnePIC 앱은 촬영 시간과 사진의 저장 크기 및 관리 면에서 실용적인 것으로 판단된다.

Abstract In this paper, we validate the practicality of the OnePIC app implemented in the previous study by analyzing the execution and storage performance. The OnePIC app is a camera app that allows you to get a photo with a desired focus after taking photos focused on various places. To evaluate performance, we analyzed distance focus shooting time and object focus shooting time in detail. The performance evaluation was measured on actual smartphone. Distance focus shooting time for 5 photos was around 0.84 seconds, the object detection time was around 0.19 seconds regardless of the number of objects and object focus shooting time for 5 photos was around 4.84 seconds. When we compared the size of a single All-in-JPEG file that stores multi-focus photos to the size of the JPEG files stored individually, there was no significant benefit in storage space because the All-in-JPEG file size was subtly reduced. However, All-in-JPEG has the great advantage of managing multi-focus photos. Finally, we conclude that the OnePIC app is practical in terms of shooting time, photo storage size, and management.

Key Words : JPEG, Multi-Focus, Focus Movement, Smartphone Camera

*비회원, 한성대학교 컴퓨터공학부

**중신회원, 한성대학교 컴퓨터공학부(교신저자)

***정회원, 한성대학교 컴퓨터공학부

접수일자 2024년 2월 1일, 수정완료 2024년 3월 1일

계재확정일자 2024년 4월 5일

Received: 1 February, 2024 / Revised: 1 March, 2024 /

Accepted: 5 April, 2024

*Corresponding Author: calafk@hansung.ac.kr

Dept. of Computer Engineering, Hansung University, Korea

I. 서 론

스마트폰 카메라를 이용하여 좋은 사진을 얻는 방법은 다양하게 연구되어왔다[1,2,3]. OnePIC 앱은 다양한 객체나 거리에 초점을 맞춘 사진들을 동시에 촬영하여 한 장의 사진에 저장해두고 촬영 후에도 사용자가 원하는 곳으로 초점을 변경시켜 원하는 사진을 얻도록 한다 [4]. OnePIC 앱은 다초점 촬영 기술과 여러 장의 사진을 한 장의 사진에 저장하도록 하는 저장 기술로 구성된다. 객체마다 초점이 맞는 사진들을 촬영하는 객체별 다초점 촬영과 거리마다 초점이 맞는 사진들을 촬영하는 거리별 다초점 촬영을 통해 여러 곳에 초점이 맞는 사진들을 촬영한다. 촬영된 사진들은 한 장의 사진 안에 저장될 수 있도록 구현한 All-in-JPEG 파일[5]로 저장하여 사용자가 원하는 곳에 초점이 맞춰진 사진을 얻을 수 있게 한다.

본 논문은 OnePIC 앱의 실행 시간을 세부적으로 분석하고 실용성을 평가하기 위해 다음과 같이 성능을 평가하였다. 거리별 다초점 촬영에서 렌즈 초점 거리 이동 시간을 포함한 촬영 시간과 객체별 다초점 촬영 시, 객체를 감지하는 시간, 객체별로 렌즈 초점 거리를 조절하면서 촬영하는 시간을 측정하고 평가하였다. 또한, 한 장의 사진 안에 여러 장의 사진을 저장한 All-in-JPEG 파일 크기를 측정하고 기존 JPEG[6] 파일 크기와 비교하였다.

본 논문은 2장에서 지난 연구에서 구현된 OnePIC 앱의 구조와 촬영 시간을 분석하고, 3장에서 성능평가에 대해 기술하며, 4장에서 결론을 맺는다.

II. OnePIC 앱의 구조와 촬영 시간 분석

1. OnePIC 앱 시스템 구성

OnePIC 앱은 그림 1과 같이 카메라를 제어하여 촬영하는 카메라 모듈과 촬영된 여러 사진을 한 장의 All-in-JPEG 파일로 저장하는 All-in-JPEG 입출력 모듈로 구성된다.

OnePIC 앱은 카메라 사용 시, 안드로이드에서 제공하는 저수준의 Camera2 API[7]를 사용하여 동작한다. 특별히, 객체별 다초점 촬영은 Tensorflow Lite의 Efficient Det Object detection model[8]을 사용하여 객체들의 위치를 판단한다. 촬영된 사진들은 All-in-JPEG 입출력 모듈을 통해 한 장의 All-in-JPEG 파일로 저장한다.

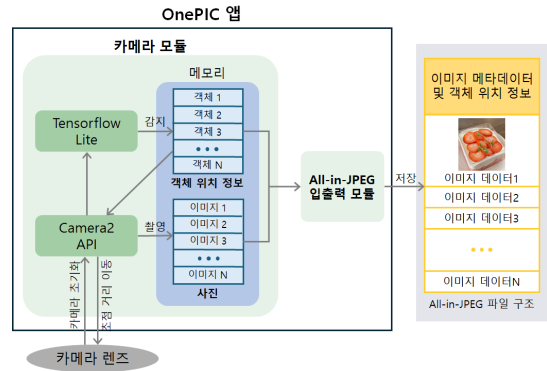


그림 1. OnePIC 앱의 시스템 구성
Fig. 1. System view of OnePIC

2. OnePIC 앱의 촬영 시간 분석

이 절에서는 OnePIC 앱의 사용자 실용성에 영향을 미치는 다초점 촬영 시간을 평가하기 위해, 그 시간을 세부적으로 분석한다.

가. 거리별 다초점 촬영 시간 분석

거리별 다초점 촬영은 그림 2와 같이 카메라를 초기화하고 카메라가 초점을 맞출 수 있는 렌즈 초점 거리의 최댓값 N 을 알아낸다. 그리고 거리별로 초점이 맞춰진 m 장의 사진을 촬영하기 위해 렌즈 초점 거리를 N/m 만큼 움직이면서 사진을 촬영하여 메모리에 저장한다. 메모리에 저장된 사진들은 한 장의 All-in-JPEG 파일로 저장된다.

Algorithm 1. Distance Focus Shooting

```

Initialize camera
 $N \leftarrow \text{get\_max\_focal\_length}()$ 
imageList  $\leftarrow$  empty
focalLength  $\leftarrow 0$ 
for imageNum from 0 to  $m$  do
    adjust_focus(focalLength)
    image  $\leftarrow$  capture_image()
    imageList.append(image)
    focalLength  $\leftarrow$  focalLength +  $\frac{N}{m}$ 
end for
save_All-in-JPEG(imageList)
    
```

그림 2. 거리별 다초점 촬영 알고리즘
Fig. 2. Algorithm of distance focus shooting

OnePIC 앱에서 거리별 다초점 촬영의 시간 흐름은 그림 3과 같이 Init, Preview, Shooting의 세 과정으로 구성된다.

Init 과정에서는 OnePIC 앱을 처음 실행할 때 카메라를 초기화하고 이 시간을 T_{caminit} 이라고 정의한다.

Preview는 카메라 렌즈에 비친 화면을 보여주는 과정으로 이 시간을 $T_{preview}$ 라고 정의한다. Shooting 과정은 T_{dist} 와 T_{dsave} 로 구성되며 다음과 같이 정의한다.

- $T_{dshooting}$: 거리별 다초점 촬영과 All-in-JPEG 파일로 저장하는 시간 ($T_{dist} + T_{dsave}$)
- T_{dist} : 거리별로 초점이 맞춰진 m 장의 사진을 촬영하는 시간 ($\sum_{i=1}^m (T_{df_i} + T_{ds_i})$)
- T_{df_i} : i 번째 촬영을 위한 초점 거리 이동 시간
- T_{ds_i} : i 번째 사진 촬영 + 메모리 저장 시간
- T_{dsave} : 메모리에 저장된 사진들을 모두 All-in-JPEG 파일로 저장하는 시간

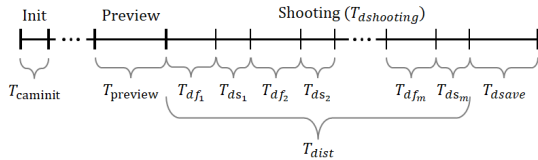


그림 3. 거리별 다초점 촬영 시간 흐름
 Fig. 3. Time flow of distance focus shooting

나. 객체별 다초점 촬영 시간 분석

객체별 다초점 촬영은 그림 4와 같이 진행되며 그림 5와 같이 Init, Preview, Shooting의 세 과정으로 구성된다.

```

Algorithm 2. Object Focus Shooting
Initialize camera
ObjDetectionList ← detect_objects( )
imageList ← empty
for each object in objDetectionList do
    focalLength ← calculate_focal_length(object)
    adjust_focus(focalLength)
    image ← capture_image( )
    imageList.append(image)
end for
save_All-in-JPEG(imageList)
    
```

그림 4. 객체별 다초점 촬영 알고리즘
 Fig. 4. Algorithm of object focus shooting

Init 과정은 거리별 다초점 촬영의 Init 과정과 동일하다. Preview 과정에서는 Tensorflow Lite를 통해 렌즈에 비치는 객체를 감지하며 이 시간을 T_{detect} 라고 정의한다. Shooting 과정은 T_{obj} 와 T_{osave} 로 구성되며 다

음과 같이 정의한다.

- $T_{oshooting}$: 객체별 다초점 촬영과 All-in-JPEG 파일로 저장하는 시간 ($T_{obj} + T_{osave}$)
- T_{obj} : 객체별로 초점이 맞춰진 m 장의 사진을 촬영하는 시간 ($\sum_{i=1}^m (T_{of_i} + T_{os_i})$)
- T_{of_i} : i 번째 촬영을 위한 초점 거리 이동 시간
- T_{os_i} : i 번째 사진 촬영 + 메모리 저장 시간
- T_{osave} : 메모리에 저장된 사진들과 객체 위치 정보 모두 All-in-JPEG 파일로 저장하는 시간

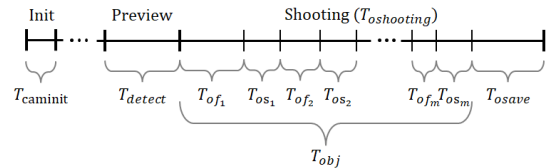


그림 5. 객체별 다초점 촬영 시간 흐름
 Fig. 5. Time flow of object focus shooting

III. 성능평가

이 절에서는 OnePIC 앱의 실용성을 평가하기 위해, T_{dist} , T_{detect} , T_{obj} 그리고 All-in-JPEG 파일 크기를 측정하고 평가한 내용을 기술한다. 본 논문에서의 성능 평가는 Galaxy Note 9를 사용하여 실시하였다.

1. 거리별 다초점 촬영 시간 평가

거리별 다초점 촬영 시간은 그림 3과 같이 m 번의 연속된 촬영 시간의 합인 T_{dist} 에 의해 결정되므로 T_{dist} 를 측정한 결과, 그림 6과 같은 성능을 보였다. 본 실험은 촬영장 수를 3장, 5장, 7장, 10장으로 바뀌며 측정하였다.

성능평가 결과, 거리별 다초점 촬영 시간은 촬영 장수에 선형적으로 비례한다. 1장의 사진을 촬영하는데 평균 0.16초가 소요되고 10장을 촬영한다고 했을 때 약 1.15초 정도 소요되어 사용자가 크게 불편함을 느끼지 않는다고 판단된다. 초점 거리를 더 세분화하고자 10장의 사진을 촬영할 수도 있지만 5장으로도 충분하며 촬영 시간이 1초도 걸리지 않기 때문에 사용자는 불편함을 느끼지 않는다고 판단된다.

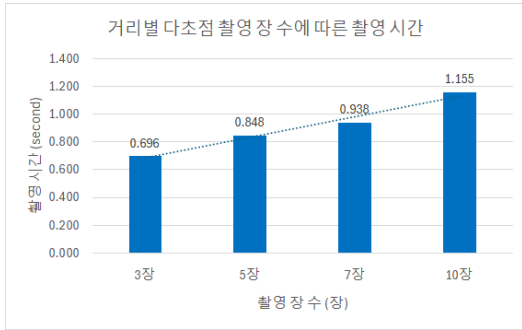


그림 6. 거리별 다초점 촬영 장 수에 따른 촬영 시간
Fig. 6. Distance focus shooting time according to the number of images

2. 객체별 다초점 촬영 시간 평가

객체별 다초점 촬영 시간은 그림 5와 같이 객체를 감지하는 T_{detect} 와 m번의 연속 촬영 시간 T_{obj} 로 구성되며, 본 실험에서는 이들을 각각 측정하였다.

T_{detect} 를 측정한 결과는 그림 7과 같이 감지 객체 개수를 3개, 5개, 7개, 10개, 20개로 측정하였다. 실험 결과, T_{detect} 는 감지되는 객체 개수와 상관없이 약 0.19초로 일정한 수치를 보인다. 미미한 수준의 오차는 실험 상황에 의해 발생한 것으로, 무시해도 되는 수치라고 판단된다.

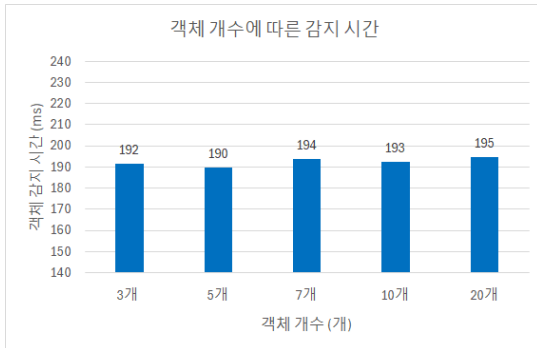


그림 7. 객체 개수에 따른 객체 감지 시간
Fig. 7. Object detection time according to the number of objects

T_{obj} 는 객체별로 초점 거리를 이동시키면서 촬영한 시간의 합으로 그림 8과 같으며 촬영 장 수를 3장, 5장, 7장, 10장으로 측정하였다.

성능 측정 결과, 예상할 수 있듯이 감지되는 객체가 많을수록 T_{obj} 는 선형적으로 증가하였다. 이러한 결과

는 촬영 장 수(감지 객체 개수)에 따라 렌즈 초점 거리 이동 시간이 길어지기 때문에 나오는 당연한 결과이다. 10장의 촬영 시간은 10초 정도로 짧지 않지만 3개의 객체에만 초점을 맞추고자 한다면 촬영 시간이 2.8초 정도로 길지 않다. 또한, 사용자는 촬영 시간을 고려하여 스스로 객체 개수를 선택할 수 있다. 예를 들어 많은 사람이 모여 촬영하는 경우, 사람들마다 초점 거리를 모두 맞출 필요가 없기 때문에 촬영자가 3명 정도만 감지하도록 설정해도 원하는 결과물을 얻는데 문제가 없다.

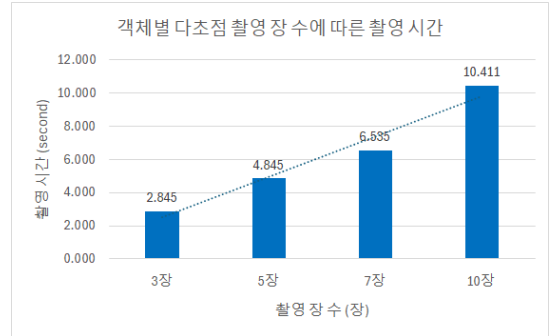


그림 8. 객체별 다초점 촬영 장 수에 따른 촬영 시간
Fig. 8. Object focus shooting time according to the number of images

본 실험의 결과, 객체별 다초점 촬영 시간이 긴 이유는 가까이 있는 객체와 먼 객체를 정렬하지 않고 감지된 순서대로 촬영하여 렌즈 초점 거리가 임의로 움직여 이동 시간이 길어지기 때문이라고 판단된다. 만약 렌즈 초점 거리를 줄이는 방향으로 객체들을 정렬한 후 촬영을 시작한다면, 초점 거리 이동 시간은 매우 짧아질 것이다. 이 방법은 추후 연구 과제로 남겨두고자 한다.

3. 저장 성능

가. 일반 JPEG 파일과 All-in-JPEG 파일 크기 비교
All-in-JPEG은 일반 JPEG과 달리 m 장의 JPEG으로 구성된 저장 포맷이고 All-in-JPEG 헤더 부분에 공통적인 메타데이터를 저장한다.

본 실험에서는 m 장의 일반 JPEG 파일 크기와 m 장의 사진을 담은 All-in-JPEG 파일 크기를 측정하고 비교하였다. 실험 결과, 그림 9는 일반 JPEG 파일에 대한 All-in-JPEG 파일의 상대적 크기 비율을 보여준다.

실험 결과, All-in-JPEG 파일이 일반 JPEG 파일보다 크기가 작은 것으로 판단된다. 예를 들어 일반 JPEG 파일을 5개 합친 크기는 18.59MB이고 5장의 사진을 담은

All-in-JPEG 파일은 18.29MB로 0.3MB의 미미한 차이를 보인다. 이는 일반 JPEG 파일의 헤더 부분의 크기가 이미지에 비해 매우 작기 때문에 공통된 헤더 부분의 크기를 줄인다 해도 전체 크기에 있어서 큰 변화는 없는 것으로 판단된다.

All-in-JPEG의 근본적인 목적은 파일 크기를 줄이고자 함에 있지 않다. All-in-JPEG은 다초점 사진들을 한 장의 파일 속에 저장함으로써 사용자가 원하는 곳에 초점이 맞춰진 사진을 추출하거나 사진을 전송하는 등 관리의 편리함을 가진다.

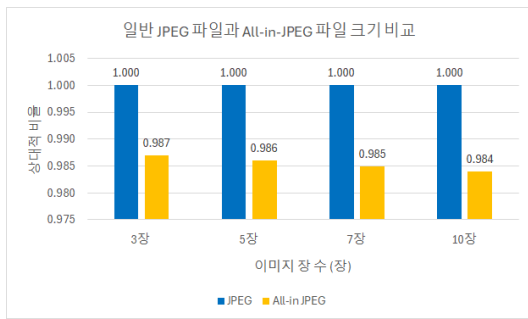


그림 9. 일반 JPEG 파일과 All-in-JPEG 파일 크기 비교
 Fig. 9. Size comparison between JPEG files and All-in-JPEG files

IV. 결 론

본 논문에서는 이전 연구에서 구현된 OnePIC 앱의 실용성을 검증하기 위해 다양한 성능평가를 실시하였다. 성능평가 결과, 초점 거리를 5구간으로 나눠 사진을 촬영할 때 전체 0.84초 정도로 측정되었고 객체별 다초점 촬영 시, 객체 감지 시간은 객체 개수와 관계없이 전체 0.19초 정도로 측정되었으며, 5개의 객체에 초점을 맞춘 사진들을 모두 촬영하는 시간은 4.84초 정도로 측정되었다. 또한, 일반 JPEG 파일과 All-in-JPEG 파일의 크기를 비교한 결과, All-in-JPEG 파일 크기가 미미한 정도로 줄었지만, 여러 장의 사진을 관리할 수 있는 편리함을 가진다고 판단된다. 본 논문에서 촬영 시간과 저장 성능을 평가한 결과, OnePIC 앱은 실생활에서 높은 실용성을 가지는 것으로 판단된다.

References

- [1] S. Yousefi, M. Rahman, N. Kehtarnavaz, "A new auto-focus sharpness function for digital and smart-phone cameras", IEEE Transactions on Consumer Electronics, vol. 57, no. 3, pp. 1003-1009, August 2011, DOI: <https://doi.org/10.1109/ICCE.2011.5722691>
- [2] <https://helpx.adobe.com/kr/photoshop/using/reduce-camera-shake-induced-blurring.html>
- [3] <https://helpx.adobe.com/kr/photoshop/using/adjusting-image-sharpness-blur.html>
- [4] Chae-Won Park, Kyung-Mi Kim, Song-Yeon Yoo, Yu-Jin Kim, Kitae Hwang, In-Hwan Jung, Jae-Moon Lee, "Camera App of Smartphone with Multi-Focus Shooting and Focus Post-processing Functions", The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 24, No. 1, pp.189-196, Feb.29, 2024. DOI: <https://doi.org/10.7236/JIIBC.2024.24.1.189>
- [5] Yu-Jin Kim, Kyung-Mi Kim, Song-Yeon Yoo, Chae-Won Park, Kitae Hwang, In-Hwan Jung, Jae-Moon Lee, "Preliminary Study on All-in-JPEG with Multi-Content Storage Format extending JPEG", The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 23, No. 5, pp.183-189, Oct.31, 2023. DOI: <https://doi.org/10.7236/JIIBC.2023.23.5.183>
- [6] <https://ko.wikipedia.org/wiki/JPEG>
- [7] <https://developer.android.com/reference/android/hardware/camera2/package-summary>
- [8] <https://www.kaggle.com/models/tensorflow/efficientdet/frameworks/tfLite/variations/lite0-detection-meta-data/versions/1?tfhub-redirect=true>

저 자 소 개

박 채 원(비회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 웹 공학, UI/UX

김 경 미(비회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 웹 공학, 분산 시스템

유 승 연(비회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 웹 공학, 분산 시스템

김 유 진(비회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 웹 공학, 분산 시스템

황 기 태(중신회원)



- 서울대학교 컴퓨터공학과 학사
- 서울대학교 컴퓨터공학과 석사
- 서울대학교 컴퓨터공학과 박사
- 현재 한성대학교 컴퓨터공학과 교수
- 경력
University of Florida 방문 교수
- 관심분야 : 모바일 시스템, IoT, 인공지능

정 인 환(정회원)



- KAIST 정보및통신공학과 박사
- 삼성전자 수석연구원
- 현재 한성대학교 컴퓨터공학부 교수
- 관심분야 : IoT, 분산시스템, 모바일 시스템

이 재 문(정회원)



- 한양대학교 전자공학과(학사)
- 한국과학기술원 전기및전자공학과 (석사)
- 한국과학기술원 전기및전자공학과 (박사)
- 현재 한성대학교 컴퓨터공학부 교수
- 경력
한국통신 연구개발단
- 관심분야 : 기계학습, 게임프로그래밍, 감성컴퓨팅

※ 본 연구는 한성대학교 교내 학술 연구비를 지원받았음