

멀티 스레드를 이용한 영상 처리 속도 분석

허태성^o, 장형규^{*}

^o인하공업전문대학 컴퓨터정보공학과,

^{*}인하공업전문대학 컴퓨터정보공학과

e-mail: tshur@inhac.ac.kr^o, gudrb7834@naver.com^{*}

Analysis of Image Processing Speed Using Multi-Threads

Hur Tai-sung^o, Jang Hyeong-kyu^{*}

^oDept. of Computer Science Engineering, Inha Technical College,

^{*}Dept. of Computer Science Engineering, Inha Technical College

● 요약 ●

본 논문에서는 멀티 스레드를 이용하여 UI에 영상을 표현할 때, 걸리는 시간 및 적정 사용 방법을 연구하였다. 영상처리는 오픈 API인 OpenCV를 사용하여 처리하였으며, 1~4개의 스레드로 영상을 다운로드하는 시간부터 GUI에 보여주는 시간까지를 계산하는 기능을 구현하여 영상을 다운로드하고, GUI에 넣어주어야 할 때 스레드의 개수가 퍼포먼스에 관여하는 처리속도를 알아볼 수 있다.

키워드: 스레드(Thread), 영상 처리(Image Download), GUI(Graphic User Interface)

I. Introduction

사용자는 흔히 영상을 볼 수 있는 소프트웨어 프로그램을 사용할 수 있다. 사용자에게 영상이 보여지는 프로그램에서 사용자가 중요시 여기는 기능에는 영상 퀄리티, 속도 등의 기능이 있을 것이며, 그 중 스레드 개수가 영상 디스플레이 속도에 미치는 영향에 대한 연구를 진행해보려 한다.

II. Preliminaries

1. Related works

1.1 영상 처리[1]

영상 처리(Image processing) 또는 화상 처리는 넓게는 입출력이 영상인 모든 형태의 정보처리를 가리키며, 주로 사진이나 동영상 처리하는 것이 대표적인 예이다. 대부분의 영상 처리 기법은 화상을 2차원 신호로 보고 여기에 표준적인 신호 처리 기법을 적용하는 방법을 사용한다.

1.2 스레드[2]

스레드는 프로그램(프로세스) 실행의 단위이며, 하나의 프로세스는 여러개의 스레드로 구성이 가능하다. 또한 하나의 프로세스를 구성하는 스레드들은 프로세스에 할당된 메모리, 자원등을 공유한다.

III. The Proposed Scheme

본 프로그램은 4개의 JPEG 파일을 한 개부터 네 개까지의 스레드를 이용하여 다운로드 받고 GUI[3]에 보여주기가까지의 시간을 보여주는 기능을 포함한다. 영상을 디스플레이 하기 위해 영상 처리는 OpenAPI 인 OpenCV[4]를 사용했으며, 예시에 사용된 JPEG 파일의 용량은 423KB이다.

측정에 사용된 하드웨어 스펙은 하기와 같다.

Table 1. 하드웨어 스펙

하드웨어 스펙	
메모리	32768MB RAM
CPU	11th Gen Intel(R) Core i7-11700F

진행방식으로는

1. 전처리 없이 이미지를 디스플레이하기 위해, 이미 로컬 폴더에 있는 JPEG 영상을 GUI에 디스플레이하는 시간을 측정하였다. 4개의 JPEG 영상을 디스플레이 하기 위해, 스레드는 4의 약수(1,2,4)의 개수로 사용하였으며, 1개의 스레드의 경우 메인 스레드만 사용하였다. 디스플레이 시간은 하기와 같다.

Table 2. 전처리가 없는 이미지 디스플레이 시간

GUI 디스플레이 시간			
Thread Count	1	2	4
Time(ms)	99	97	99

2. 전처리 이후 이미지 디스플레이를 하기 위해, 파일을 로컬 폴더가 아닌 서버에 업로드하고 해당 파일을 로컬 폴더에 다운로드 후 GUI에 디스플레이 시간 측정하였다. 이 경우에도 4개의 JPEG 영상을 디스플레이 하기 위해서, 4의 약수(1,2,4) 개의 스레드를 사용하였으며, 1개의 스레드는 메인 스레드만 사용하였다. 측정된 디스플레이 시간은 하기와 같다.

Table 3. 전처리 이후 이미지 디스플레이 시간

GUI 디스플레이 시간			
Thread Count	1	2	4
Time(ms)	194	134	112

해당 결과에서 알 수 있듯이, 여러 전처리 프로세싱[5]을 거치지 않고 바로 GUI에 보여주면 싱글 혹은 멀티 스레드의 차이가 거의 없는 것을 확인할 수 있다. 하지만 전처리 프로세스를 처리하고 디스플레이 할 경우, 스레드의 개수에 따라 점점 차이가 나는 것을 확인할 수 있으며, 걸린 시간의 비율은 하기와 같다.

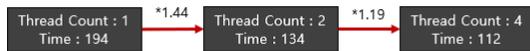


Fig. 1. 스레드 개수에 따른 걸린 시간의 비율

IV. Conclusions

영상을 열기만 하는 환경에서는 싱글 스레드와 멀티 스레드를 사용하는 것이 처리 속도에 큰 영향을 주지 않는다는 것을 확인할 수 있었다. 하지만 영상을 디스플레이하기 전에 다운로드 등과 같은 전처리 프로세싱을 처리하고 디스플레이 시에는 처리하는 스레드 수에 따라 유의미한 차이가 보여진다. 위 결과로 영상을 디스플레이만 하기 위한 이벤트에서는 퍼포먼스를 위해 멀티 스레드를 사용해도 처리 속도가 크게 변하지 않을 것이다. 하지만 다운로드 혹은 영상처리 등의 프로세스를 처리하고 디스플레이 하는 이벤트의 경우 멀티 스레드를 구현하는 것이 퍼포먼스에 도움이 될 수 있다. 본 연구에서 진행한 멀티 스레드에 따른 영상 처리 속도는 JPEG파일을 다운로드하는 환경에서만 연구를 진행하였다. 추후 추가적인 데이터 처리 및 영상 처리 등의 다양한 상황에 따른 퍼포먼스의 시간 측정은 더 많은 샘플 데이터를 만들 수 있을 것이며 해당 샘플 데이터를 이용하면 더욱 정확한 통계를 만들 수 있을 것이다. 이 연구에서는 추후 이러한 기능들을 추가하고 더 정확한 통계를 얻어내는 것을 목표로 한다.

REFERENCES

- [1] Image Processing?, <https://terms.naver.com/entry.naver?docId=393820&cid=42441&categoryID=42441>
- [2] Thread?, <https://terms.naver.com/entry.naver?docId=2424059&cid=51399&categoryID=51399>
- [3] GUI?, <https://namu.wiki/w/GUI>
- [4] What is OpenCV?, <https://studium-anywhere.tistory.com/22>
- [5] Processing?, <https://www.opentutorials.org/module/3431/20322>