

<https://doi.org/10.7236/IIBC.2018.18.6.63>

IIBC 2018-6-7

## 모바일 환경에서 오프로딩 기술 기반 인터넷 방송 영상 처리에 관한 연구

### A Study on the Internet Broadcasting Image Processing based on Offloading Technique on the Mobile Environments

강흥규\*

Hong-gue Kang\*

**요약** 오프로딩(offloading)은 컴퓨팅 자원 및 계산 속도의 한계를 극복하기 위해 로컬 컴퓨터에서 수행하는 어플리케이션의 일부를 컴퓨팅 자원과 처리능력이 우수한 원격지 컴퓨터에 전달하여 처리한 후 결과를 반환받는 방식이다. 최근에는 컴퓨팅 자원과 처리능력에 한계를 갖고 있는 모바일 컴퓨팅 분야에서 처리속도를 높이고 배터리 소모를 줄이기 위해 모바일 게임, 멀티미디어 데이터, 360도 동영상 처리, 인터넷 방송용 이미지 처리 분야에서 응용되고 있다. 본 논문에서는 모바일 장치에서 촬영된 360도 구면 영상을 사용자가 쉽게 영상 내용을 이해할 수 있도록 다양한 평면 영상을 변환하고 내용을 확인할 수 있는 뷰어를 무선 인터넷 환경에서 오프로딩 기술을 적용하여 구현하고 실제 실험 결과를 제시한다. 360도 구면 영상은 인터페이스를 통해 360도 카메라의 영상 획득 위치에 따라 Double Panorama, Quad, Single Rectangle, 360 Overview + 3 Rectangle로 평면영상으로 변환이 성공적으로 수행된다. 실험과정에서 100가지 이상의 360도 구면 영상을 아래 인터페이스를 통해 평면 영상으로 변환을 성공적으로 진행하였다.

**Abstract** Offloading is a method of communicating, processing, and receiving results from some of the applications performed on local computers to overcome the limitations of computing resources and computational speed. Recently, it has been applied in mobile games, multimedia data, 360-degree video processing, and image processing for Internet broadcasting to speed up processing and reduce battery consumption in the mobile computing sector. This paper implements a viewer that enables users to convert various flat-panel images and view contents in a wireless Internet environment and presents actual results of an experiment so that users can easily understand the images. The 360 degree spherical image is successfully converted to a plane image with Double Panorama, Quad, Single Rectangle, 360 Overview + 3 Rectangle depending on the image acquisition position of the 360 degree camera through the interface. During the experiment, more than 100 360 degree spherical images were successfully converted into plane images through the interface below.

**Key Words** : 360 Degree Image Processing, Mobile Internet Environment, Offloading, Internet of Things

\*정희원, 중부대학교 방송영상학과  
접수일자: 2018년 10월 4일, 수정완료: 2018년 11월 6일  
게재확정일자: 2018년 12월 8일

Received: 4 October, 2018 / Revised: 6 November, 2018 /  
Accepted: 8 December, 2018

\*Corresponding Author: hong7@joongbu.ac.kr

Dept. of Broadcasting & visual art, Joongbu University, Korea

## I. 서 론

컴퓨팅 자원과 능력이 제한적인 스마트폰, 태블릿 PC와 같은 모바일 디바이스의 사용이 끊임없이 증가되면서 복잡한 계산을 위해 고성능의 컴퓨팅 자원을 필요로 하는 모바일 어플리케이션에 대한 대처 연구가 중요한 이슈로 부각되고 있다. 특히, 처리속도, 저장능력과 같은 지속적인 성능향상 노력과 더불어 용량이 제한적인 배터리 사용시간을 연장하기 위한 소프트웨어적인 관리기술의 중요성이 강조되고 있다. 따라서 모바일 디바이스에서 실행되는 어플리케이션에 대해 실행시간과 배터리 소비를 최소화하고 이를 관리할 수 있는 연구 시도는 연관된 분야에서 진행되어야 할 중요한 연구 내용이다<sup>[1]</sup>.

최근까지 모바일 디바이스는 CPU 연산 속도, 메모리 용량, 네트워크 대역폭 및 속도 등에서는 비약적인 성장을 거듭하고 있다. 하지만 모바일 디바이스에서 많은 컴퓨팅 자원과 고성능을 요구하는 동영상 코드변환(transcoding), 3D 렌더링, 인식(recognition) 등과 같은 영상처리 어플리케이션은 계산속도 향상, 대용량 메모리, 4G 네트워크, 고해상도 등의 지원에도 불구하고 처리 실행시간과 배터리 사용시간의 제한을 극복하지 못하고 있다.

오프로딩(offloading)<sup>[2]</sup>은 컴퓨팅 자원 및 계산 속도의 한계를 극복하기 위해 로컬 컴퓨터에서 수행하는 어플리케이션의 일부를 컴퓨팅 자원과 처리능력이 우수한 원격지 컴퓨터에 전달하여 처리한 후 결과를 반환받는 방식으로서 “cyber foraging”, “surrogate computing”과도 동일한 의미를 갖는다. 최근에는 컴퓨팅 자원과 처리능력에 한계를 갖고 있는 모바일 컴퓨팅 분야에서 처리속도를 높이고 배터리 소모를 줄이기 위해 모바일 게임, 멀티미디어 데이터, 360도 동영상 처리, 인터넷 방송용 이미지 처리 분야에서 활발하게 응용되고 있다<sup>[3]</sup>.

오프로딩 프레임워크를 개발하고 이를 적용하기 위해서는 그림 1과 같이 오프로딩 단위, 오프로딩 결정시간(정적 vs. 동적), 오프로딩을 적용할 수 있는 어플리케이션, 오프로딩 프레임워크의 환경을 결정해야 한다. 최근 오프로딩 연구는 메소드를 스마트폰에서 실행하면서 오프로딩을 동적으로 결정하며 많은 계산시간이 요구되는 인터넷 방송용 정지 영상 또는 동영상, 얼굴·소리 인식, 게임 등에 적용하여 대표적인 모바일 환경인 스마트폰의 계산속도와 배터리 소모 한계를 극복하는 것이 중요한 연구 이슈이다.

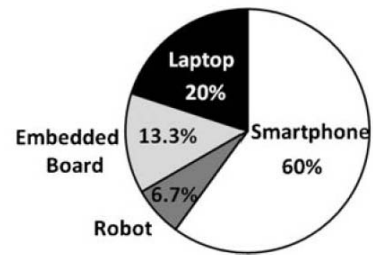


그림 1. 오프로딩을 통한 어플리케이션 개발  
Fig. 1. Application Development through Offloading

본 논문에서는 모바일 장치에서 촬영된 360도 구면 영상을 일반사용자가 쉽게 영상 내용을 이해할 수 있도록 다양한 평면 영상을 변환하고 내용을 확인할 수 있는 뷰어를 무선 인터넷 환경에서 오프로딩 기술을 적용하여 구현하고 실제 실험 결과를 제시한다. 360도 구면 영상은 인터페이스를 통해 360도 카메라의 영상 획득 위치에 따라 Double Panorama, Quad, Single Rectabgle, 360 Overview + 3 Rectangle로 평면영상으로 변환이 성공적으로 수행된다. 실험과정에서 100가지 이상의 360도 구면 영상을 아래 인터페이스를 통해 평면 영상으로 변환을 성공적으로 진행하였다.

## II. 관련 연구

### 1. 오프로딩 기반 관련 연구

모바일 디바이스를 위한 오프로딩 프레임워크 구현은 .NET CLR을 기반으로 어플리케이션의 일부 함수를 RPC 방식으로 원격지 서버에서 실행하는 MAUI<sup>[4]</sup>와 안드로이드 OS의 달빅 가상 기계(Dalvik Virtual Machine; DVM)에서 소스코드의 수정없이 실행환경을 이전하는 Clonecloud<sup>[5]</sup> 연구가 대표적이다. 이러한 연구를 기반으로 메소드 단위의 오프로딩을 통해 어플리케이션의 실행시간을 줄이고 부가적으로 에너지 소비를 줄이려는 연구가 Cuckoo<sup>[6]</sup>, ThinkAir<sup>[7]</sup>, SmartDiet<sup>[8]</sup>에서 진행되었다.

MAUI(Mobile Assistance Using Infrastructure): MAUI는 .NET CLR을 기반으로 동작하는 오프로딩 기술로서 모바일 디바이스(예, 스마트폰)와 원격 서버(예, MAUI 서버)간에 RPC 방식으로 어플리케이션을 실행하여 스마트폰의 계산시간을 줄이고 이를 통해 부가적으로 배터리 소모를 줄이는 것을 목적으로 하고 있다. 어플리케이션 개발자가 원격지 서버에서 실행할 수 있는 함수

를 지정("remotable")하면 로컬 지역의 MAUI 환경에서 함수가 원격지 서버에서 실행되는 데에 필요한 정보를 주고받을 수 있도록 소스코드를 변경한다. 변경된 함수들은 어플리케이션 실행중에 모바일 디바이스 상태와 서버 간의 통신 상태에 따라 로컬 또는 원격지에서 실행될지가 결정되며, 원격지 서버에서 실행될 함수는 모바일 디바이스와 원격지 서버간에 RPC 통신을 이용하여 필요한 정보를 넘기고 서버는 이를 받아 해당 함수를 실행한다.

MAUI 방식의 특징은 원격지 서버에서 실행될 함수를 어플리케이션 실행중에 모바일 디바이스 상태, 서버와의 통신 상태 등에 따라 동적으로 결정함으로써 오프로딩의 효율성을 높일 수 있으며, 최소의 정보만 원격지 서버로 전송함으로써 오프로딩에 필요한 오버헤드를 최소화한다는 것이 장점이다. 하지만 어플리케이션 개발자가 원격지 서버에서 실행가능한 함수를 지정해야 하고, 해당 함수들이 서버에서 실행될 수 있도록 소스코드를 수정해야 하기 때문에 개발자의 부담을 가중시킬 뿐만 아니라 바이너리 형태로 배포되는 어플리케이션에는 적용시킬 수 없다는 단점을 가지고 있다.

CloneCloud: CloneCloud는 안드로이드 OS의 DVM에서 실행되는 어플리케이션의 일부 함수를 오프로딩을 통해 실행속도 향상시키고 이를 통해, 모바일 디바이스의 에너지 소비 부담을 줄이는 것을 목적으로 하고 있다. Clonecloud 시스템은 원격지 클라우드 서버에 로컬과 동일한 실행환경(Clone VM)을 생성한 후 실행에 필요한 정보를 전달한다. 어플리케이션을 오프로딩을 통해 실행하는 과정은 어플리케이션 바이트코드를 분석한 후 오프로딩 후보를 결정하고 각 후보함수의 시작부분(migrate) 및 종료부분(re-integrate)에 명령어를 삽입한다. 후보함수에 스마트폰 및 클라우드 서버의 성능에 대한 프로파일 정보, 스마트폰의 통신 연결 상태 등을 고려하여 오프로딩할 함수를 결정하여 스마트폰 내부에 저장한다. 로컬지역의 스마트폰 DVM에서 어플리케이션의 바이트코드 실행중에 migrate 명령어를 만나게 되면 실행을 멈추고 서버에 어플리케이션의 클론을 만든 후 실행 흐름을 옮긴다. 이 때 서버에 생성된 클론과 스마트폰의 상태를 동일하게 만들어 주기 위해 DVM의 힙 오브젝트, 해당 함수가 실행되던 스레드의 스택 자료구조도 서버로 보낸 후, 서버에 만들어진 클론 어플리케이션을 스마트폰에서 정지되었던 바로 그 위치에서부터 다시 동작시킨다.

CloneCloud 방식의 특징은 안드로이드 DVM을 수정

하는 방식으로 오프로딩을 구현하였기 때문에 어플리케이션을 수정할 필요가 없어 개발자의 작업 편의가 매우 높다는 장점이 있다. 하지만 어플리케이션 시작 전에 통신 상태에 따라 분기할 함수들을 미리 결정한 후, 어플리케이션이 시작될 때의 통신 상태에 따라 미리 정해진 분기 시나리오대로만 동작한다는 점에서 어플리케이션 동작 도중 통신 상태가 바뀌었을 경우 효율이 떨어진다는 것이 단점이다. 또한 분기를 위해 DVM의 힙 오브젝트, 스택 자료구조와 같은 내부 상태를 클라우드 서버로 전송해야 한다는 점과 분기 여부를 결정하는 데 있어서 스마트폰과 클라우드 서버의 성능, 통신 상태만을 고려하기 때문에 스마트폰의 CPU 사용량이나 어플리케이션의 메모리 점유 등을 실시간으로 반영하지 못하는 단점이 있다.

## 2. 파노라마 영상 조합 관련 연구

모바일 장치 등에서 촬영된 360도 영상은 넓은 범위를 한 화면에 표현한 영상이므로 상당히 넓은 범위를 한 번에 볼 수 있다. 특히, 원통형 파노라마(cylindrical panorama)는 가장 널리 사용되는 파노라마 형식으로 좌우 시야각이 360°에 달해 전, 후, 좌, 우를 모두 한 번에 감시할 수 있다. 이는 카메라의 투영면을 가정하는 방식이 다르기 때문에 가능한 촬영 방법이다.<sup>[9]</sup>연구에서는 원통형 파노라마 영상을 만들어내는 사영 변환기법을 제안하였으며,<sup>[10]</sup>연구에서는 PTZ 카메라를 이용해서 파노라마 영상을 조합해내는 방법을 제안하였고,<sup>[11]</sup>에서는 스마트폰 카메라를 이용해서 구면 파노라마 영상을 조합하는 방법을 제안하였다. 하지만 스마트폰 카메라의 자기 센서와 가속도 센서 등을 이용해 구면 파노라마 영상을 구성하게 될 경우 Zero Parallax 문제점을 해결해낼 수 없기 때문에 파노라마 영상에서 찢어짐 현상이 발생한다.

## III. 3장 모바일 환경에서 360도 이미지 처리 시스템

### 1. 전체 시스템

모바일 디바이스에서 360도 카메라는 AR/VR 분야의 급성장, 인터넷 방송의 활성화 등으로 활용 가능성이 매우 높으며, 360 영상 처리에 대한 요구가 급상승 하고 있

다. 특히, 모바일 디바이스에서 360 영상을 촬영하고 사용자의 요구에 따라 평면영상 변환, 움직임 감지, 자동차 또는 동물과 같은 특정 이동 물체를 추적하는 수요가 급상승하고 있지만, 모바일 디바이스가 같은 자원의 한계(처리속도, 메모리, 전력소모 등등) 때문에 무선통신이 발달한 환경에서는 오프로딩 기술이나 모바일 엣지 컴퓨팅 기술과 접목하여 한계를 극복할 수 있다. 본 논문에서는 모바일 디바이스의 360 영상 처리에서 속도 등의 계산 집중적 요소를 초고성능 원격지 서버에 전달하여 시간소비 요소, 전력소비 요소 등을 절감하기 위해 모바일 클라우드 컴퓨팅 분야와 VR/AR, 인터넷 방송 영상 처리 분야 등과 접목하여 360 영상에 대한 다양한 가공 요구를 고성능 원격지 서버에서 실행하여 모바일 디바이스의 계산 및 전력 소비 부담을 줄이고자 한다.

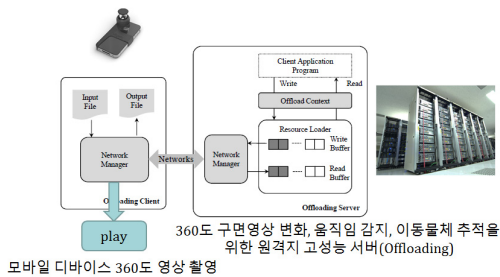


그림 2. 360도 영상 처리 오프로딩 시스템  
Fig. 2. 360 Degree Image Processing Offloading System

360도 구면 영상을 다양한 시점의 평면 뷰(view)로 변환하는 시스템과 결과를 확인할 수 있는 메뉴기반 방식의 뷰어를 설계하고 구현하여 360도 구면 파노라마 영상, 평면 영상, 다양한 시점의 고화질의 영상 결과를 제시한다. 이를 위해, 그림 2와 같이 360도 구면 영상을 평면 영상으로 변환하는 뷰어를 구현하였다.

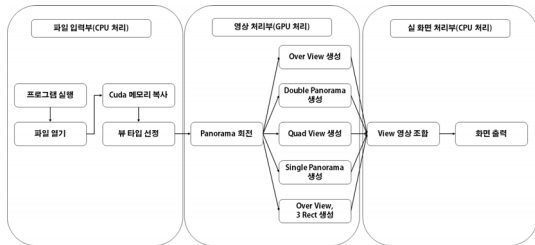


그림 3. 360 영상 뷰어 처리 과정  
Fig. 3. 360 Degree Image Viewer Processing Procedure

## 2. 실험 및 결과

360도 구면 영상에 대한 평면 영상 변환 뷰어는 UI와 기능은 그림 4와 같은 메뉴로 구성되어 있다.

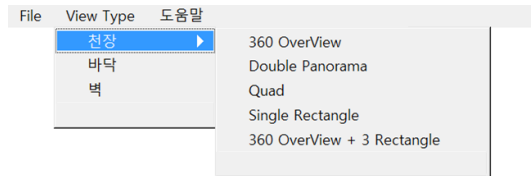


그림 4. 평면 뷰어의 메뉴 및 기능  
Fig. 4. Plain Viewer Menus and Functions

Over View는 구면의 접면을 가정하고 해당 접면에 평행 투영을 수행하여 영상 시점을 획득하였다. 회전에 따라 다양한 시점을 획득할 수 있으며, 이러한 Over View는 Hemispherical 파노라마라고도 불리며 Raw 비디오에서 절반만 취득한 것이라고도 볼 수 있다.



그림 5. 360도 원본 영상  
Fig. 5. 360 Degree Original Image

Double Panorama는 구면 파노라마 영상에서 상단, 하단에 관심사를 맞춘 뷰로서 주요 시점이 파노라마의 상단과 하단에 집중되어 있다. 구면 파노라마 영상의 회전을 이용해 얻어낸 결과와 유사하다. Double Panorama 평면영상 변환에 대한 구체적인 방법은 파노라마 영상의 회전에 대해서는 구면 파노라마 영상의 모든 좌표를 위도, 경도 기준 구면 좌표계를 기반으로 다시 표현하였다. 모든 x, y 영상 좌표계는 구면 각 좌표계로 변환되며 길이값 r은 단일값으로 통일된다. Quad 타입 뷰는 4개의 독립적인 평면 영상이 나타나 있으며 이는 일반적인 카메라로 촬영한 영상과 동일하다. 네 개의 독립적인 평면 영

상은 키보드를 이용하여 원하는 위도, 경도를 중심으로 하는 화면을 얻어낼 수 있다. 즉, 360도 파노라마 영상을 4개로 분할하여 평면 뷰와 같은 변환 영상 결과물을 생성하였다.

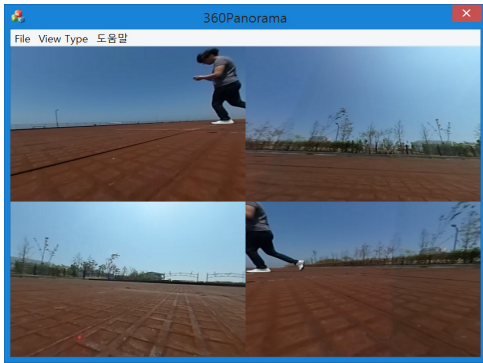


그림 6. Quad 뷰 평면 변환 영상  
 Fig. 6. Quad View Plain Transformation Image

실험에 사용된 영상에서 실험자는 1m 거리에서부터 1m씩 10m까지 총 10개의 위치에서 움직임을 수행하였다. 성능 지표는 프레임별 수행 시간으로서 각 프레임에 대해 처리를 수행하는 데 걸린 시간을 나타낸다. 또한 탐지해낸 중심 위도, 경도의 위치가 잘못된 프레임의 개수와 전체 프레임의 개수를 나타낸다. 본 실험은 건물내부에서 스마트폰으로 360도 영상을 촬영하였으며, 100회 이상의 360도 영상 변환 실험을 진행하였다.

약 2000 프레임대에서 7미터의 거리에서 움직이기 시작하였고, 이 때를 제외하면 DoF 알고리즘의 성능에는 문제가 없어 모두 균일한 0.000000292652초의 수행 시간을 보임을 볼 수 있었다. [그림 7]은 오탐지(error detection) 그래프를 나타낸다.

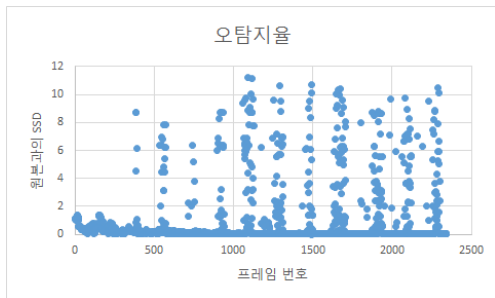


그림 7. 프레임별 오탐지 수준  
 Fig. 7. Error Detection Level per Frames

그래프 상에서 급격히 오탐지가 치솟는 부분이 존재하는데, 이 부분은 측정 거리에서 대기할 때 움직임이 적어서 발생한 오탐지 결과이다. 미세한 움직임도 잡아낼 수 있어야 하지만 거리가 멀어질수록 움직임이 픽셀 단위보다 작을 수 있어 탐지 성능이 떨어짐을 확인할 수 있다.

## IV. 결론

최근까지 모바일 디바이스는 CPU 연산 속도, 메모리 용량, 네트워크 대역폭 및 속도 등에서는 비약적인 성장을 거듭하고 있다. 하지만 모바일 디바이스에서 많은 컴퓨팅 자원과 고성능을 요구하는 동영상 코드변환(transcoding), 3D 렌더링, 인식(recognition) 등과 같은 영상처리 어플리케이션은 계산속도 향상, 대용량 메모리, 4G 네트워크, 고해상도 등의 지원에도 불구하고 처리 실행시간과 배터리 사용시간의 제한을 극복하지 못하고 있다.

본 논문에서는 모바일 장치에서 촬영된 360도 영상을 일반 사용자가 쉽게 영상 내용을 이해할 수 있도록 다양한 평면 영상을 변환하고 내용을 확인할 수 있는 뷰어를 무선 인터넷 환경에서 오프로딩 기술을 적용하여 구현하고 실제 실험 결과를 제시하였다. 이를 통해, 모바일 장치에서 영상 처리에 필요한 시간과 배터리 소모를 줄일 수 있으며, 안정적인 고화질의 평면 영상을 획득하는 효과를 확인하였다.

## Reference

- [1] Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang, "A survey of mobile cloud computing: architecture, applications, and approaches". in Wireless Communications and Mobile Computing, John Wiley & Sons, Ltd., 2011.
- [2] Alexey Rudenko, Peter Reiher, Gerald J. Popek and Geoffery H. Kuenning, "Saving Portable Computer Battery Power through Remote Process Execution", Mobile Computing and Communication Review, Vol. 2, pp.19~26, 1998.
- [3] Karthik Kumar, Jibang Liu, Yung-Hsiang Lu,

- Bharat Bhargava, "A Survey of Computation Offloading for Mobile Systems", Journal of Mobile Networks and Applications, Vol. 18, Issue 1, pp.129~140, 2012.
- [4] Eduardo Cuervo, Aruna Balasubramanian, Dae-ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, Paramvir Bahl, "MAUI: Making Smartphones Last Longer with Code Offload", International Conference on Mobile Systems, Applications, and Services(MobiSys '10), pp.49~62 2010.
- [5] Byung-Gon Chun, Sunghwa Ihm, Petros Maniatis, "CloneCloud: Elastic Execution between Mobile Device and Cloud", International Conference on Mobile Systems, Applications, and Services(MobiSys '11), pp.301~314, 2011.
- [6] R.Kemp, N. Palmer, T. Keilman, and H. pal, "Cuckoo: A Computation Offloading Framework for Smartphones", In Proc. of MobiCASE, 2010.
- [7] Sokol Kosta, Andrius Aucinas, Pan Hui, Richard Mortier, Xinwen Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading", In Proc. of INFOCOM'12, pp. 945~953, 2012.
- [8] Aki Saarinen, Matti Siekkinen, Yu Xiao, Jukka K. Nurminen, Matti Kempainen, Pan Hui, "SmartDiet: offloading popular apps to save energy", In Proc. of ACM SIGCOMM '12, pp. 297~298, 2012.
- [9] G. Tong, J. Gu, "Locating objects in spherical panoramic images," IEEE International Conference on Robotics and Biomimetics, pp. 818-823, 2011.
- [10] Gabriele Fangi, "Photogrammetric Processing of Spherical Panoramas," The Photogrammetric Record, Vol. 28, Issue 143, pp. 293-311, September. 2013.
- [11] J. Cruz-Mota, I. Bogdanova, B. Paquier, M. Bierlaire, J.-P. Thiran, Scale invariant feature

transform on the sphere: Theory and applications, IJCV, Vol. 98(2), pp. 217-241, 2012.

#### 저자 소개

#### 강 홍 규(정회원)



- 중앙 대학교 사진학과 석사
- 아카데미 오 아트 유니버시티 (MFA Motion picture & Video) 석사
- 숭실대학교 미디어 아트 박사과정 수료
- 중부대학교 방송영상학과 교수

이 논문은 2018년도 중부대학교 학술연구비 지원에 의하여 이루어진 것임.