

Mobile Ad-Hoc Cloud 기반 파노라마 이미지 생성[☆]

Panoramic Image Generation in Mobile Ad-Hoc Cloud

박 용 석¹ 김 현 식² 정 중 문^{1*}
Yong-Suk Park Hyun-Sik Kim Jong-Moon Chung

요 약

본 논문은 모바일 스마트 단말에서 파노라마 이미지 생성 프로세스를 단축하기 위해 모바일 애드혹 클라우드를 활용하는 방법을 제시한다. 파노라마 이미지 생성과 관련된 작업을 효율적으로 모바일 애드혹 클라우드에 분배하기 위해 이미지 획득 및 정렬 방법을 제안하고 작업분배에 대한 오프로딩 결정 알고리즘을 제안한다. 제안된 방법을 안드로이드 운영체제 기반의 모바일 스마트 단말에 직접 적용하여 파노라마 이미지 생성 프로세스에 미치는 영향을 분석한다.

☞ 주제어 : 파노라마, 이미지 스티칭, 모바일 애드혹 클라우드

ABSTRACT

This paper proposes the use of mobile ad-hoc cloud for reducing the process time of panoramic image generation in mobile smart devices. In order to effectively assign tasks relevant to panoramic image generation to the mobile ad-hoc cloud, a method for image acquisition and sorting and an algorithm for task distribution and offloading decision making are proposed. The proposed methods are applied to Android OS based smart devices, and their effects on panoramic image generation are analyzed.

☞ keyword : Panorama, Image Stitching, Mobile Ad-Hoc Cloud

1. 서 론

파노라마 이미지는 관람자에게 실제 경관을 보는 것과 같은 느낌을 가지게 하는 실감 미디어로 이미지 스티칭(image stitching)이 기반 기술로 사용된다. 이미지 스티칭은 중첩되는 영역을 가지는 여러 이미지들을 조합하여 하나의 큰 이미지로 만드는 기법이다[1]. 다수의 복잡한 연산 단계로 구성된 이미지 스티칭은 스마트폰과 같은 모바일 스마트 단말에서 구동될 경우 처리시간이 길고 많은 전력을 소비하게 된다. 현재 다수의 모바일 스마트 단말에는 이미지 스티칭 기반의 파노라마 이미지 생성 기능이 내장되어 있으나, 대부분 수평 단방향으로 생성된

이미지만을 처리하도록 구현되어있다. 이는 연속된 이미지를 취득함으로써 단말의 연산처리를 최소화하려는 전략으로, 여러 이미지를 비교할 필요가 없어지며 이미지를 순차적으로 스티칭하여 빠르게 파노라마를 생성할 수 있다. 랜덤한 이미지를 입력으로 제공할 경우 연산 모든 이미지를 비교하게 되어 시간이 $O(n \log(n))$ 으로 증가하게 되지만 순차적으로 입력 이미지가 제공될 경우, 선형(linear) 연산시간으로 줄어든다. 하지만 이렇게 수평 한 방향으로 파노라마를 생성함으로써 상하 경관이 제한되어 실감 미디어의 효과가 줄어들게 된다.

그림 1에서와 같이 파노라마 생성을 위한 이미지를 자유로운 시점에서 취득하고 처리하기 위해서는 스마트 단말의 자원이 제한적이다. 스마트 단말이 필요한 연산 자원을 확보하기 위해 클라우드 또는 원격 서버로의 연산 오프로딩(computation offloading) 방법이 제안되었다[2]. 스마트 단말은 결과만 표출하는 썬 클라이언트(thin client)의 역할을 하고 입력 이미지를 스티칭하고 파노라마를 생성하는 과정은 원격서버에서 처리하게 된다.

원격 서버나 클라우드를 활용하는 infrastructure로의 연산 오프로딩 방식은 네트워크 환경이 원활할 경우 문제가 없지만 네트워크가 제한적일 경우 컴퓨팅 자원을 활용하지 못하게 된다.

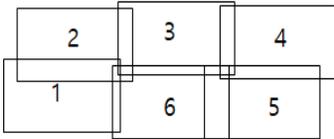
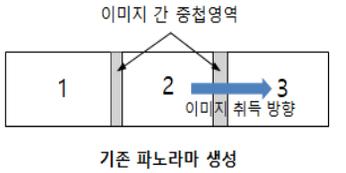
¹ School of Electrical and Electronic Engineering, Yonsei University, Seoul, 03722, Korea.

² Contents Convergence Research Center, Korea Electronics Technology Institute, Seoul, 03924, Korea.

* Corresponding author (jmc@yonsei.ac.kr)

[Received 14 September 2017, Reviewed 21 September 2017, Accepted 25 September 2017]

[☆] 본 연구는 중소벤처기업부 및 한국산업기술진흥원의 중소중견기업지원사업 월드클래스300 R&D의 일환으로 수행하였음. [S245334, 비전기반 지능형 학습 알고리즘 및 영상처리 테스트 응용 소프트웨어 개발]



이미지 취득 방향이 자유로운 파노라마 생성
(그림 1) 파노라마 이미지 취득

(Figure 1) Image Acquisition for Panorama

최근 필요한 연산 자원을 획득하기 위해 peer-to-peer 방식으로 인근에 위치한 다른 스마트 기기들과 애드혹(ad-hoc) 네트워크를 형성하여 분산 방식으로 연산 오프로딩을 수행할 수 있는 방법이 제안되었다[3, 4]. 많은 사람들이 스마트폰을 보유하고 있으며, 한 사람이 다수의 스마트 기기를 보유하는 경우도 많다. 이와 같이 주변 근 거리에 위치한 스마트 단말들의 잉여 리소스를 활용하여 오프로딩 하는 방법을 모바일 애드혹 클라우드(MAC, Mobile Ad-hoc Cloud)라고 한다.

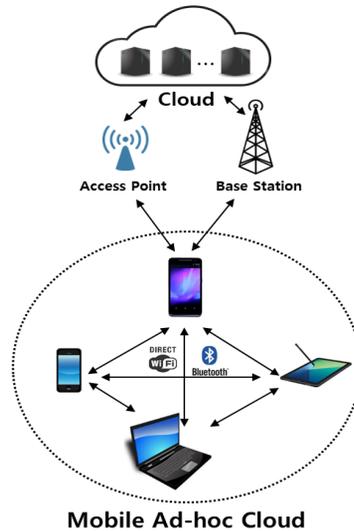
본 논문에서는 파노라마 이미지 생성을 MAC을 활용하는 방법을 제안한다. MAC에 대한 개요와 파노라마 이미지 생성과정을 설명하고, 파노라마 이미지 생성과 관련된 작업을 MAC으로 분배하기 위해 이미지 획득 및 정렬 방법을 제안하고 작업분배에 대한 오프로딩 결정 알고리즘을 제안한다.

2. Mobile Ad-hoc Cloud

모바일 컴퓨팅과 무선통신 기술의 발전은 스마트폰과 태블릿과 같은 모바일 스마트 단말의 기능과 성능을 비약적으로 향상시켰다. 이로 인해 사용자들은 일반 데스크톱 환경에서 사용하던 애플리케이션들을 모바일 환경에서 사용하기를 바라고 있으며, 많은 전산 자원이 요구되는 모바일 애플리케이션들이 등장하고 있다. 하지만 데스크톱 컴퓨팅 환경과 비교할 경우, 모바일 스마트 단말들은 전산처리 능력, 메모리, 저장 공간, 전력(배터리) 등 여러 부분에서 제약이 발생해 많은 전산 자원이 요구되는

애플리케이션을 홀로 구동하기에는 어려움이 있다. 이를 해결하기 위해 모바일 스마트 단말은 클라우드 컴퓨팅(Cloud Computing)의 도움을 받아 필요 시 수행해야 하는 복잡한 연산 태스크(task)를 클라우드 서버에서 처리하도록 오프로딩(offloading)하여 사용자의 요구사항을 만족시키고 있다.

하지만 복잡한 연산을 처리하기 위해 클라우드 컴퓨팅에만 의존할 수는 없다. 네트워크 연결 상태가 좋지 않거나 인터넷이 없는 환경에서는 클라우드 서버로의 접속이 제한된다. 또는 클라우드 서비스와 관련된 비용이 부담스러울 수도 있다. 이와 같은 상황에서는 그림 2과 같이 MAC이 해결책을 제시할 수 있다[5]. MAC은 근거리에서 위치한 다수의 모바일 스마트 단말들이 peer-to-peer 통신을 통해 애드혹 네트워크를 형성하고 서로의 자원을 공유해 주어진 태스크를 분산하여 함께 처리할 수 있다. 모바일 스마트 단말의 성능이 향상되면서, 다수의 모바일 스마트 단말 자원들을 통합하여 사용할 경우, 복잡하고 높은 연산량이 요구되는 태스크들도 분산처리를 통해 실행할 수 있게 된다.



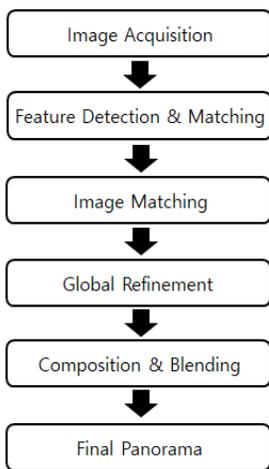
(그림 2) 모바일 애드혹 클라우드
(Figure 2) Mobile Ad-hoc Cloud

3. 파노라마 이미지 생성

파노라마는 광각 또는 360도 방향의 모든 시야(view)를 담아내는 기법으로 일반 이미지와 달리 사용자에게

실감형(immersive) 경험을 제공한다. 파노라마 이미지는 중첩되는 영역이 있는 이미지들에 대해 이미지 스티칭(stitching) 기법을 이용하여 생성된다. 이를 위해 scale-invariant 특징점(feature) 검출 기법과 homograph 모델을 이용한 방식이 가장 많이 사용된다[6].

파노라마 이미지의 일반적인 생성 과정은 그림 3과 같다. 파노라마 생성을 위해 이미지 취득 단계(Image Acquisition)에서 입력되는 이미지는 스티칭이 원활하게 진행되도록 중첩되는(overlapping) 부분이 있어야 한다. 특징점 추출과 매칭 단계(Feature Detection and Matching)는 이미지에서 식별이 용이한 지점을 찾아 이미지 간 대응되는 지점을 구분할 수 있도록 하여 영상 간 매칭을 가능하게 한다. 특징점은 크기, 위치, 시점, 조명 등이 변해도 쉽게 식별이 가능해야 한다. 특징점을 검출하기 위해 가장 많이 사용되는 알고리즘으로 SIFT(Scale Invariant Feature Transform)와 SURF(Speeded Up Robust Features)가 있다[7, 8]. 추출된 특징점들의 정보는 이미지 매칭 단계(Image Matching)에서 사용된다. 서로 다른 이미지에서 동일한 지점에 해당되는 특징점을 매칭하기 위해 컨센서스(consensus)가 최대인 모델을 뽑아내는 RANSAC(RANdom SAmples Consensus) 알고리즘이 사용된다[9]. Global Refinement 단계는 사용된 영상들에 대한 카메라 파라미터들을 보정하여 이미지 간 불균형을 최소화시킨다. Composition과 Blending 단계에서는 영상들을 조합(blending)하여 특정 표면에 투영(projection)하여 파노라마 이미지를 최종적으로 생성한다.



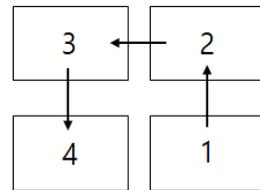
(그림 3) 파노라마 이미지 스티칭 과정
(Figure 3) Panoramic Image Stitching Procedure

4. MAC 오프로딩 기반 이미지 스티칭

4.1 이미지 획득 및 정렬 방법

스마트 디바이스에서 파노라마에 사용될 이미지들을 기존 방식처럼 순차적 단방향으로 획득하지 않고 상하좌우 랜덤하게 획득하여 처리할 경우 이미지 분류 및 비교 처리시간이 많이 소요되기 때문에, 입력 이미지들 간 상관관계 정보를 유지 및 관리해야 한다. 이를 위해 스마트 기기에서 제공되는 가속도 센서를 이용하여 처음 획득한 이미지를 기준으로 이후 획득되는 이미지들의 x, y 축으로의 상대적 이동지점을 알아낸다. 상대적 이동 지점의 좌표를 참조하여 현재 획득한 이미지가 파노라마의 어떤 부분에 해당되게 되는지 파악하고 mxn 크기의 이미지 처리 배열에 획득한 이미지 정보를 저장한다.

그림 4는 이미지 획득 및 정렬 방법에 대해 예시를 보여준다. 최초로 획득한 1이란 이미지의 정보는 이미지 처리 배열의 (0, 0) 위치에 저장된다. 가속도 센서의 정보를 이용하여 이미지 2는 이미지 1의 상위 위치(y 축 변화)에서 획득하였음을 알 수 있다. 새로 획득된 이미지 2의 정보는 배열 내 (0, 0) 위치에 저장되고, 기존 이미지 1의 정보는 (0, 1)의 위치로 이동되어 배열의 정렬 자체가 이미지들의 상대적 위치를 나타내게 된다. 마찬가지로, 가속도 센서 정보를 이용하여 이미지 3은 이미지 2의 왼쪽에서 획득되었음을 알 수 있다(x 축 변화). 획득된 이미지 3의 정보는 배열 내 (0, 0) 위치에 저장되고, 기존 이미지 1과 2의 정보는 각각 (1, 1), (1, 0)의 위치로 이동되어 배열 내에서 이미지들의 상대적 위치가 업데이트 된다.



이미지 획득 순서

	0	1				
0	1		2		3	2
1			1			1

이미지 처리 배열 내 정렬 순서

(그림 4) 이미지 획득 및 정렬 예시
(Figure 4) Example Image Acquisition and Sorting

4.2 애드혹 네트워크 생성

스마트 기기 간의 연결을 위한 애드혹 네트워크 생성은 Wi-Fi Direct를 사용한다[10]. 안드로이드 운영체제의 경우 버전 4.0(API 14)부터 Wi-Fi Direct 기능이 기본적으로 제공되고 있다. Wi-Fi Direct는 Wi-Fi Alliance에서 제정한 Wi-Fi P2P(Peer-to-Peer) 규격에 정의되어 있다. Wi-Fi Direct는 기설정된 AP(Access Point) 없이 Wi-Fi를 사용하여 단말들은 상호간 연결할 수 있다.

오프로딩 하고자 하는 기기가 그룹 소유자(group owner) 역할하며 소프트웨어적으로 AP를 생성한다(Soft AP). 인근 기기들은 생성된 Soft AP에 일반 AP에 접속하는 것과 마찬가지로 접속한다. Wi-Fi Direct는 최대 200미터까지의 통신반경을 가지며, 기기 간 전송률은 최대 250Mbit/s이다.

애드혹 네트워크 형성 과정에서 기기 발견 단계(discovery)에서 네트워크를 구성하는 단말들의 오프로딩에 필요한 하드웨어 규격 정보를 교환한다.

4.3 오프로딩 결정 프로세스

오프로딩을 결정할 때 사용되는 조건은 다음과 같다.

$$t_d < t_g + Q/v$$

여기서 t_d 는 연산을 완료해야 할 시간제한이며, t_g 는 태스크 생성 시간, Q 는 단말이 수행해야 하는 연산량, v 는 CPU 처리 속도이다.

MAC을 구성하는 단말에게 2개의 이미지에 대한 스티칭 처리를 오프로딩 했을 때의 소요시간 T_{OFLD} 은 디바이스에서 MAC 단말로 데이터를 전송하는 시간(T_{UL}), MAC 단말에서 스티칭 처리 시간(T_s), MAC 단말에서 디바이스로 결과 데이터를 전송하는 시간(T_{DL})등으로 구성되며 아래와 같은 식으로 표현할 수 있다[11].

$$T_{OFLD} = T_{UL} + T_s + T_{DL} \\ = \frac{F_{UL}}{d_{UL}R_{UL}} + \frac{1}{d_s v_s} L_s + \frac{F_{DL}}{d_{DL}R_{DL}}$$

여기서 L_s 는 이미지 스티칭 관련 태스크이며, v_s 는 MAC 단말의 처리 속도, d_s 는 해당 지연요소이다. 무선 애드혹 네트워크에서의 전송속도는 전송 데이터의 크기(F_{UL} , F_{DL})을 전송속도(R_{UL} , R_{DL})로 나눈 값이며 전송 지연 요소

(d_{UL} , d_{DL})가 영향을 미칠 수 있다.

다수의 MAC 단말들에 대한 전체 오프로딩 처리 시간은 각 단말에 해당되는 T_{UL} , T_{DL} 값들을 더해주고, 스티칭 처리는 오프로딩 후 동시에 병렬 처리됨으로 가장 큰 값의 T_s 를 더하면 된다.

파노라마 이미지를 수행하고자 하는 디바이스는 입력 이미지 수와 MAC을 구성하는 단말 수를 고려하여 자기 자신이 모든 연산을 처리했을 때 소요되는 시간과 발견된 MAC 단말들에게 오프로딩 했을 때의 처리시간을 비교하여 보다 빠른 쪽을 선택한다.

MAC 단말로 오프로딩할 이미지 데이터를 선택할 때는 이미지 처리 배열에 저장된 순서를 참조하여 관련성이 있는 이미지들을 선별하여 전송한다.

4.4 원격 이미지 스티칭

이미지를 전송받은 원격 기기들은 각 이미지에서 특징점을 추출하고, 추출된 특징점들에 대해 k-d tree를 사용하여 k nearest-neighbor를 찾는다. RANSAC을 이용하여 두 이미지 간 homography를 추측한다. 각도와 초점거리를 수정하고 multi-band blending을 사용하여 최종 파노라마를 생성한다.

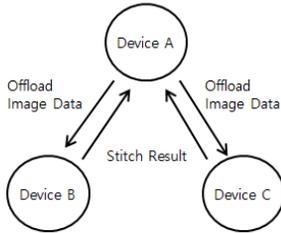
5. 구현 및 결과

모바일 애드혹 클라우드를 이용하여 파노라마 이미지를 생성하는 방법을 실험하기 위해 안드로이드 운영체제 기반 스마트폰을 3대를 사용하였다. 빠른 구현을 위해 이미지 스티칭 라이브러리는 OpenCV를 사용하였다[12]. 테스트 환경 구성은 그림 5와 같다. 기본적으로 모든 기기가 오프로딩 처리에 참여할 의사가 있고 자원공유를 한다고 가정한다.

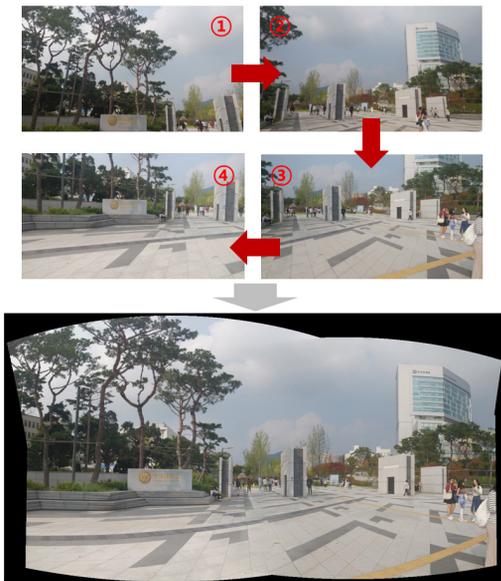
파노라마를 생성하고자 하는 Device A가 Soft AP를 생성하고 Wi-Fi Direct를 통해 인근에 위치하고 자원공유가 가능한 Device B와 C를 발견한다. Device A는 파노라마를 구성하는 4개의 사진을 찍고 다른 2개의 기기에 이미지 스티칭을 진행할 사진 2개씩을 전송한다. Device B와 C는 각각 전송받은 사진을 스티칭하여 Device A로 전송한다. Device A는 전송받은 이미지들을 다시 스티칭하여 최종 파노라마를 완성시키게 된다.

그림 6은 이와 같은 과정을 통해 생성된 파노라마 결과물을 보여준다. 즉, Device A는 ①, ②, ③, ④ 4개의 사진을 입력으로 찍고, ①과 ④는 Device B로, ②와 ③은

Device C로 전송한다. Device B와 C는 각각 전송받은 이미지에 대해서 이미지 스티칭을 진행하고 스티칭된 결과물을 Device A로 보낸다. Device A는 ①+④, ②+③ 영상을 스티칭하여 최종 파노라마를 구성하게 된다.



(그림 5) 기본 테스트 환경 구성
(Figure 5) Basic Test Topology



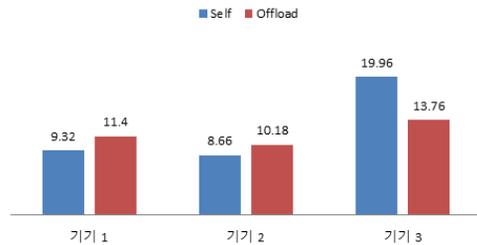
(그림 6) 이미지 스티칭 결과
(Figure 6) Image Stitching Result

표 1은 사용된 스마트폰의 기본 사양을 보여주고 그림 7은 측정결과를 보여준다. 기기 1, 2, 3은 각각 Device A의 역할은 수행하고 측정한 결과를 보여준다. 기기 1과 기기 2는 오프로딩 결정 프로세스에 따라 원칙적으로 오프로딩을 수행하지 않지만 (오프로딩 하는 것이 처리시간이 더 많이 소요됨), 결과값 비교를 위해 강제적으로 오프로딩을 수행하도록 하였다.

파노라마 생성을 위해 사용된 이미지의 크기는 1280x960 픽셀이다. 사진마다 추출되고 처리되는 특징점의 수가 달라 최대한 동일한 시점에서 이미지를 캡처하여 처리하였다. 그림 7에서 Self는 4개의 이미지에 대한 스티칭을 기기 자체적으로 처리할 때 걸린 시간을 초단위로 나타낸다. Offload는 기기 1는 기기 2, 3에게 오프로딩 하고, 기기 2는 기기 1, 3에게 오프로딩을 하고, 기기 3은 기기 1, 3에게 오프로딩을 해서 이미지 스티칭을 처리했을 때의 결과를 보여준다. 기기3의 성능이 낮기 때문에 기기1과 2가 오프로딩 했을 경우 처리시간은 자체 처리하는 것보다 느리게 나왔다. 기기3의 경우 오프로딩함으로써 처리시간이 6.2초가 단축되어, 자체처리 대비 30%에 해당하는 성능향상 효과를 볼 수 있다.

(표 1) 디바이스 사양
(Table 1) Device Specification

항목	기기 1	기기 2	기기 3
코어	Quadcore	Quadcore	Quadcore
BogoMIPS	38.40	38	13.5
Android Version	6.0.1	7.1.2	4.4.4



(그림 7) 자체처리 및 오프로딩 성능 비교
(Figure 7) Performance Comparison: Self-Processing vs. Offloaded

6. 결 론

본 논문에서는 모바일 스마트 단말에서 파노라마 이미지 생성 프로세스를 단축하기 위해 모바일 애드혹 클라우드를 활용하는 방법을 제안하였다. 파노라마 이미지를 생성하고자 하는 단말은 Wi-Fi Direct 기반 애드혹 네트워크를 형성하여 자원공유가 가능한 단말들을 인근에서 찾는다. 작업 분산 처리를 고려한 이미지 획득 및 정렬 방법에 의해 분류된 이미지 데이터를 제안된 오프로딩 알고

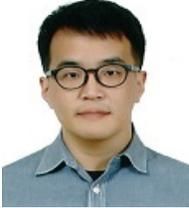
리즘을 기반으로 MAC을 구성하는 단말들에게 분배하여 처리한다. 구현을 통해 MAC을 활용한 방법은 파노라마 이미지 처리 시간을 단축함을 확인하였다.

현 연구단계에서는 오프로딩 파라미터로 소비전력이 고려되지 않았지만 향후 사용자의 파워 프로파일에 따른 오프로딩을 지원할 예정이다. 또한 본 구성을 360 가상현실(VR, Virtual Reality) 애플리케이션에 적용하여 특수한 광각 카메라 장치 없이도 스마트 기기의 기본 카메라와 MAC의 자원지원을 받아 관련 콘텐츠 생성이 가능하도록 확장할 예정이다.

참고문헌(Reference)

- [1] P. Burt and E. Adelson, "A Multiresolution Spline with Application to Image Mosaics", *ACM Transactions on Graphics*, vol. 2, no. 4, pp. 217-236, Oct. 1983.
<http://dx.doi.org/10.1145/245.247>
- [2] Q. Wang, F. Reimeier, and K. Wolter, "Efficient Image Stitching Through Mobile Offloading", *Electronic Notes in Theoretical Computer Science*, vol. 327, pp. 125-146, Oct. 2016.
<http://dx.doi.org/10.1016/j.entcs.2016.09.027>
- [3] R. Tron and R. Vidal, "Distributed Computer Vision Algorithms", *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 32-45, Apr. 2011.
<http://dx.doi.org/10.1109/MSP.2011.940399>
- [4] M. Chen, Y. Hao, Y. Li, C. Lai, and D. Wu, "On the Computation Offloading at Ad-hoc Cloudlet: Architecture and Service Modes", *IEEE Communications Magazine*, vol. 53, no. 6, pp. 18-24, June 2015.
<http://dx.doi.org/10.1109/MCOM.2015.7120041>
- [5] I. Yaqoob, E. Ahmed, A. Gani, S. Mokhtar, M. Imran, and S. Guizani, "Mobile Ad hoc Cloud: A Survey", *Wireless Communications and Mobile Computing*, vol. 16, no. 16, pp. 2572-2589, July 2016.
<http://dx.doi.org/10.1002/wcm.2709>
- [6] M. Brown and D. Lowe, "Automatic Panoramic Image Stitching Using Invariant Features", *Int. J. Computer Vision*, vol. 74, no. 1, pp. 59-73, Aug. 2007.
<http://dx.doi.org/10.1007/s11263-006-0002-3>
- [7] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *Int. J. Computer Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004.
<http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>
- [8] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)", *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, June 2008.
<http://dx.doi.org/10.1016/j.cviu.2007.09.014>
- [9] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Comm. of the ACM*, vol. 24, no. 6, pp. 381-395, June 1981.
<http://dx.doi.org/10.1145/358669.358692>
- [10] Wi-Fi Alliance. Wi-Fi Direct. Accessed Sep. 2017.
<https://www.wi-fi.org/discover-wi-fi/wi-fi-direct>
- [11] J.-M. Chung, Y.-S. Park, J.-H. Park, and H. Cho, "Adaptive Cloud Offloading of Augmented Reality Applications on Smart Devices for Minimum Energy Consumption," *KSH Trans. on Intern. and Info. Sys.*, vol. 9, no. 8, pp. 3099-3111, Aug. 2015.
<http://dx.doi.org/110.3837/tiis.2015.08.020>
- [12] OpenCV homepage. Accessed Sep. 2017.
<http://opencv.org>

◎ 저 자 소 개 ◎



박 용 석(Yong-Suk Park)

1997년 Carnegie Mellon University, Dept. of Electrical & Computer Engineering (공학사)
1998년 Carnegie Mellon University, Dept. of Electrical & Computer Engineering (공학석사)
2008년~현재 연세대학교 대학원 전기전자공학부 박사과정
1998년~2000년 (주)에스원 기술연구소 주임연구원
2000년~2003년 아이앤씨테크놀로지 기술연구소 주임연구원
2003년~현재 전자부품연구원 콘텐츠응용연구센터 책임연구원
관심분야 : M2M/IoT, Cloud Computing, Mixed Reality, Biometric Authentication.
E-mail : yspark@keti.re.kr



김 현 식(Hyun-Sik Kim)

2002년 인하대학교 정보통신공학과 (공학사)
2004년 인하대학교 대학원 정보통신공학과 (공학석사)
2017년 연세대학교 대학원 전기전자공학부 (이학박사)
2004년~현재 전자부품연구원 콘텐츠응용연구센터 책임연구원
관심분야 : 콘텐츠 공유, Wearable Computing, Human-Computer Interaction, M2M/IoT.
E-mail : hskim@keti.re.kr



정 종 문(Jong-Moon Chung)

1992년 연세대학교 전자공학과 (공학사)
1994년 연세대학교 대학원 전자공학과 (공학석사)
1999년 Pennsylvania State University, Dept. of Electrical Engineering (공학박사)
1997년~1999년 Assistant Professor & Instructor, Pennsylvania State University, Dept. of Electrical Engineering
2000년~2005년 Director, Advanced Communication Systems Engineering Laboratory (ACSEL)
2000년~2005년 Director, Oklahoma Communication Laboratory for Networking & Bioengineering (OCLNB)
2000년~2005년 Associate Professor (Tenured), Oklahoma State University, School of Electrical & Computer Engineering
2005년~현재 연세대학교 전기전자공학과 교수
관심분야 : IoT & Smartphones, Smart Cars & ITS, Military Communications, NFV / ICN / SDN, LTE-A / 5G Networks, Cloud Computing & Big Data, Public Safety AR Simulators and Networking.
E-mail : jmc@yonsei.ac.kr