ISSN 1598-0170 (Print)
ISSN 2287-1136 (Online)
http://www.jksii.or.kr

2D-Touch 스마트 디바이스에서 사용자 행동 패턴 분석을 통한 가상 3D-Touch 구현을 위한 방법☆

An Approach to implement Virtual 3D-Touch using 2D-Touch based Smart Device through User Force Input Behavior Pattern

남 춘 성[1*]
ChoonSung Nam

요 약

3D-Touch 인터페이스의 등장은 사용자로 하여금 새로운 인터랙션을 제공할 수 있는 기반을 마련해 주었다. 하지만 일부 스마트폰에서만 3D-Touch가 제공되고 있고, 대부분 2D-Touch 스마트폰에서 사용되는 인터랙션 방법은 새로운 방식의 3D-Touch의 인터랙션을 제대로 적용할 수 있는 방안은 마련되지 않고 있다. 이는 3D-Touch와 2D-Touch 인터랙션에서의 차이가 발생하는 주요 원인이다. 이에 따라 본 연구에서는, 2D-Touch 기반 디바이스에서도 3D-Touch의 소프트웨어적 기능과 인터페이스를 활용할 수 있게 해주는 가상 포스 터치 (Virtual Force Touch) 방식을 이용해서 3D-Touch 인터페이스를 활용할 수 있는 방안에 대하여 제안한다. 제안한 방안은 2D-Touch를 제공하는 스마트폰에서 사용자가 포스터치 입력을 발생하였을 때 스마트폰에 차지하는 손가락 면적을 분석하여 적절한 가상 포스 터치 구현하기 위한 것이다.

☞ 주제어 : 3D-Touch, 포스 터치, 가상 포스 터치, 스마트 디바이스, 포스 임계값

ABSTRACT

The appearance of 3D-Touch interface provided the basis of a new interaction method between the users and the mob ile interface. However, only a few smartphones provide 3D-Touch features, and most of the 2D-Touch devices does not provide any means of applying the 3D-Touch interactions. This results in different user experiences between the two interaction methods. Thus, this research proposes the Virtual Force Touch method, which allows the users to utilize the 3D-Touch Interface on 2D-Touch based smart devices. This paper propose the suitable virtual force touch mechanism that is possible to realize users' inputs by calculating and analysis the force touch area of users' finger. This proposal is designed on customized smartphone device which has 2D-Touch sensors.

☞ keyword : 3D-Touch, Force Touch, Virtual 3D-Touch, Smart device, force threshold

1. 서 론

스마트폰을 중심으로 한 스마트 디바이스의 생활변화는 사용자로 하여금 언제 어디서나 인터넷을 통한 정보를 능동적 혹은 수동적으로 접할 수 있게 만들었다. 정보를 획득하기 위한 과정에서 스마트 디바이스와 사용자와의 의사소통은 인터랙션(interaction)인 사용자 인터페이스(UI: User Interface)를 통해서 이루어진다. 즉, UI는 사용자와 디바이스 간에 상호작용을 발생시킬 수 있는 프로그램(program)으로 정의 될 수 있다[1]. 스마트폰의 등장 이후로 사용자 경험(UX: User eXperience) 중심의 직관적이고 인간친화적인 인터페이스가 적용되고 있다. 특히, 터치(touch) 형식의 입력 방식은 대표적인 스마트폰의 인터페이스 장치로 쓰이고 있다[2].

2D-Touch 인터페이스는 2차원 평면 좌표값을 이용하여 스크린에 나타난 화면과 연동하여 입력을 수행하는 방식이다. 터치 인터페이스는 사용자의 손가락의 위치와 개수를 이용해 싱글터치(single touch), 멀티터치(multi-touch), 롱터치(long touch), 스와이프(swipe), 드래그(drag) 등과 같은 다양한 입력 방식을 사용해 사용자와의 인터페이스를 제공하고 있다. 지금까지의 2D-Touch 인터랙션

[1] IT Research and Lab, College of Information and Communication Engineering, Sungkyunkwan University., Suwon, 440-746, Korea
* Corresponding author (namgun99@gmail.com)
[Received 25 April 2016, Reviewed 19 May 2016, Accepted 4 October 2016]
☆ 이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2016R1A6A3A11932892)

방안은 화면상에서의 위치 정보만을 활용하고 있고, 손가락 제스처(gesture)가 취할 수 있는 다른 물리적 속성들은 반영하지 못했다[3]. 즉, 사용자는 손가락으로 입력할 때 같은 위치를 지정하더라도 힘의 조절을 통해 다른 결과를 가져올 수 있지만, 2D-Touch를 활용한 인터랙션은 스크린에 힘(force)라는 속성을 반영하지 못하기 때문에 힘의 양과 상관없이 항상 같은 결과만을 가져올 수밖에 없다. 또한, 이러한 2D-Touch 의 인터랙션 방안은 웨어러블(wearable) 디바이스와 같이 소형화된 기기에서 공간 제약 문제 때문에 기존의 인터랙션 방식인 멀티터치, 스와이프, 드래그 등과 같이 공간 활용이 많은 인터랙션을 활용하기에 쉽지 않다.

이러한 2D-Touch가 지닌 한계를 극복하기 위한 연구들이 3D-Touch를 통해 지속적으로 진행되어 왔다[4,5,6,7]. 상용화 제품으로 애플사에서는 스크린에 가해지는 힘의 입력이 가능한 3D-Touch 인터페이스를 발표하였다[8]. 3D-Touch는 스크린의 z축 정보를 기반으로 사용자가 가하는 힘을 인식하는 인터랙션 방안이다. 화면의 공간을 한 차원 더 확장시켜 활용할 수 있다는 점에서 최소한의 입력으로 더 많은 아웃풋(output)을 낼 수 있도록 도와준다. 가령, 기존의 2D-Touch에서는 특정한 정보를 접근하기 위해 여러 번의 터치를 통해 반복적인 작업을 시행해야 했다면, 3D-Touch를 활용한다면 힘의 조절을 통해 한 번의 터치로 원하는 정보를 접근할 수 있다. 즉, 특정한 정보를 접근하기 위해 사용자가 인지 할 수 있는 아이콘의 배치와 같은 숏컷(shortcut) 기능을 설정하기 위해서 활용될 수 있다. 이렇듯 3D-Touch는 기존의 2D-Touch가 반영하지 못했던 사용자 인터랙션을 활용해 기존의 모바일 디바이스에서 활용하지 못 한 인터랙션을 제공하고 뿐만 아니라 화면의 공간이 제한적인 웨어러블 디바이스 (wearable device) 등과 같은 기기에서도 활용할 수 있다.

다양한 3D-Touch 연구와 활용 가능성에도 불구하고, 하드웨어 구현의 어려움으로 인해 3D-Touch가 대중화되기 까지는 오랜 시간이 걸리고 있다. 이는 현재까지 3D-Touch가 적용된 스마트 디바이스가 아이폰 6s, 6s Plus[8]와 화웨이의 mate s[9]등 일부 기종에서만 한정되어 있기 때문이다. 즉, 3D-Touch의 장점을 활용한 소프트웨어가 출시되고 있음에도 불구하고 2D-Touch 스마트폰이 대부분이기 때문에 이러한 새로운 방식의3D-Touch의 인터랙션을 제대로 적용할 수 있는 방안이 적다.

이에 따라 본 연구에서는, 2D-Touch 기능만을 지니고 있는 모바일 디바이스에서도 3D-Touch의 소프트웨어적 기능과 인터페이스를 활용할 수 있게 해주는 가상 포스 터치 (Virtual Force Touch) 방식을 이용해서 3D-Touch 인터페이스가 내장되어 있지 않은 2D-Touch에서도 활용할 수 있는 방안에 대하여 제안한다. 가상 포스 터치를 이용하기 위해서 사용자의 손가락의 넓이 변화를 통해서 3D-Touch의 힘을 구현한다. 이를 위해 넓이를 사용자 어떻게 인식하는지를 실험을 통해 증명하여 모든 사용자가 이용할 수 있는 가상 포스 터치에 대한 기준을 제시한다.

2장에서는 2D-Touch 인터페이스와 3D-Touch 인터페이스에 대해서 알아보고 2D-Touch 인터페이스에서 극복할 수 있는 방안을 알아본다. 3장에서는 사용자가 설정할 수 있는 면적에 따른 임의 구역을 구분할 수 있는지를 실험하고, 4장에서는 위와 같은 실험을 통해 사용자가 구분 가능한 방법 및 이를 증명한다. 마지막으로 5장에서는 결론을 맺는다.

(표 1) 2D-Touch기반의 제스처
(Table 1) Gesture of 2D-Touch

종류	내 용
탭 (Tap)	화면을 한 번 터치한 후 바로 들어 올리는 방법
더블탭 (Double Tap)	화면을 연속으로 두 번 터치한 후 들어올리는 방법
프레스 (Press)	화면 위를 일정 시간 동안 터치하는 방법
드래그 (Drag)	화면 터치 후 정해진 방향으로 손가락을 움직이고 떼는 방법
플릭 (Flick)	스와이프 보다 좀 더 빠르게 한 방향으로 미는 방법
로테이드 (Rotate)	두 손가락을 화면에 터치 후 각 손가락을 회전시키는 방법
스프레드 (Spread)	두 손가락을 화면에 터치 후 두 손가락 사이를 넓히는 방법
핀치 (Pinch)	두 손가락을 화면에 터치 후 두 손가락 사이를 좁히는 방법
프레스 및 탭 (Press & Tap)	하나의 손가락으로 화면을 누르고 다른 손가락으로 다른 면을 터치하는 방법

2. 관련 연구

2.1 2D-Touch 인터페이스

일반적인 터치스크린(touch screen) 기반 스마트 디바이스들은 2D-Touch 인터페이스를 이용하여 디바이스의 입력을 사용한다. 2D-Touch인터페이스는 터치와 같이 손

가락이 화면에 닿은 스크린상의 위치(x축과 y축) 입력을 받는 방식이다. 손가락 입력은 다수의 손가락을 인식할 수 있는 멀티터치, 손가락이 스크린상에 닿은 시간, 화면에서 손가락의 움직임에 따른 다양한 데이터를 수집할 수 있다. 특히, 손가락의 위치와 더불어 손가락의 움직임 즉, 제스처[10]에 따라서 터치 인터페이스는 다양한 방식으로 입력 방법으로 나뉠 수 있고, 사용자와의 인터랙션을 제공할 수 있다. 이와 같은 제스처의 종류는 다음 표 1과 같이 다양하게 나타날 수 있다.

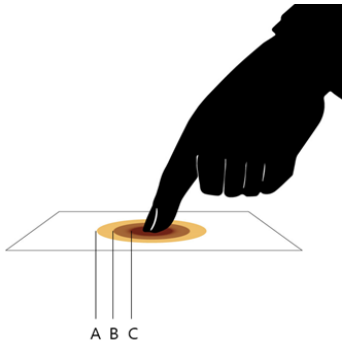
표 1에서와 같이 2D-Touch의 제스처를 이용한 인터랙션은 손가락이 닿는 위치와 제스처가 같다하더라도 힘의 조절에 따라 다른 결과를 가져올 수 있는 방안이 제시되어 있지 않다. 이는 2D-Touch를 활용한 인터랙션은 스크린에 힘라는 속성을 반영하지 못하기 때문에 항상 위치에 대한 결과만을 가져올 수밖에 없다.

2.2 3D-Touch 인터페이스

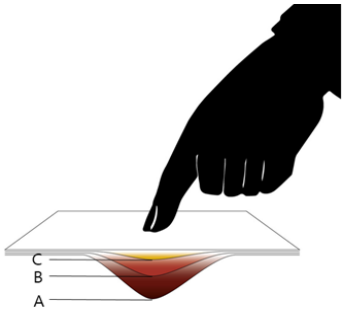
3D-Touch 인터페이스는 스마트 디바이스의 스크린에 손가락이 닿는 지점의 좌표값(x축과 y축)과 사용자가 누르는 힘의 크기(z축)를 함께 인식할 수 있는 입력장치이다. 일반적인 2D-Touch 인터페이스와는 다르게 사용자가 누르는 힘의 크기가 일정한 임계값(threshold)을 구분 지어 입력의 단계를 나누는 방법이다. 스마트폰 iPhone 6S의 경우, 3D-Touch 인터페이스는 스크린을 누르는 힘의 크기에 따라 터치의 종류를 2단계 3D-Touch인터페이스와 2D-Touch 인터페이스로 구분한다. 힘의 세기에 약한 힘은 Peek을 강한 힘은 Pop을 통해 사용자와의 인터랙션을 부여하고 있다[8]. 이와 같은 방식은 화웨이(Hwawei) mate s도 같은 방식으로 사용되고 있다. 화웨이의 경우 사진의 확대를 위해 사용자의 힘에 따른 연속적인 값을 화면의 비율로 변환하여 사진의 크기를 조절할 수 있다[9].

이와 같은 3D-Touch의 장점은 하나 이상의 입력을 받아야 하는 인터랙션, 즉, 멀티터치, 드래그, 스와이프, 플릭, 핀치 인/아웃, 로테이트(roate)와 같은 제스처를 표현하는 2D-Touch에서 터치와 제스처를 통해 표현하는 방안을 터치와 힘만을 이용해서 이를 대체할 수 있다. 이는 웨어러블 디바이스인 스마트 와치와 같이 제한적인 크기를 갖는 디바이스 등에 유용하게 적용될 수 있다. 또한, 3D-Touch를 통한 연속적이고 단계적인 임계값을 활용한다면 보다 다양한 입력을 동시에 제공할 수 있고, 새로운 인터랙션 방안을 제시 할 수 있다. 나아가 2D 터치에서 사용되었던 반복적인 입력을 방지할 수 있고, ‘Shortcut’를 통한 특정 App의 위치 혹은 버튼 클릭을 유도할 수 있어 UI를 제작하는데 도움을 수 있다.

기존의 2D 터치 기반의 스마트 디바이스는 위와 같은 3D-Touch를 지원하지 않기 때문에 3D-Touch 사용자와 같은 인터랙션을 공유할 수 없다. 따라서 본 논문에서는 위와 2D-Touch 기반 사용자에게 3D-Touch 인터랙션과 같은 인터랙션을 지원하기 위해 가상의 3D-Touch를 제공할 수 있는 방안을 제안한다.



(그림 1) 넓이 변화에 따른 z축 입력 방안
(Figure 1) z-dimension input by width of a finger



(그림 2) 힘 변화에 따른 z축 입력 방안
(Figure 2) z-dimensiion input by force value of a finger

2.3 넓이 변화를 통한 z축 입력 방법

사용자의 손가락이 스마트폰에 닿는 면적에 따라 구분지을 수 있는 입력값은 그림 1과 같이 나타낼 수 있다. 이는 그림 2와 같이 3D-Touch 인터페이스에서 사용자의 힘에 의해서 z축의 값을 얻는 것과 유사하다. 또한, 일반적으로 사용자가 스마트폰의 화면을 터치할 때 힘을 적게

가할 때와 힘을 많이 가할 때 스마트폰에 닿는 손가락의 면적의 변화가 나타난다. 즉, 힘을 적게 가할 때는 적은 면적이 힘을 많이 가할 때는 넓은 면적이 나타난다. 따라서 사용자가 힘을 가해 얻는 값과 면적의 변화에서 얻는 값을 일치시킨다면 사용자가 z축을 입력하기 위한 방법으로 사용될 수 있다. 따라서 면적의 변화에 따른 값에 의해서 3D-Touch에서의 Peak & Pop 과 같은 인터랙션 또한 구현이 가능하다.

3. 사용자 면적기반 3D-Touch 분석

3.1 사용자 손가락 면적에 따른 포스 구분 방안

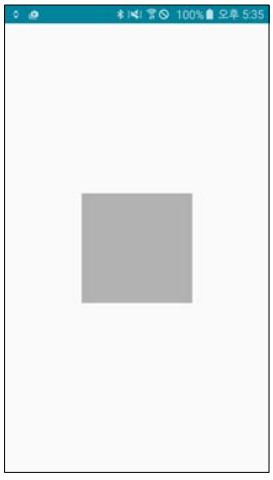
사용자의 손가락 면적에 따른 가상 포스 터치를 적용하기 위해서는 각 사용자가 조정할 수 있는 터치 면적의 크기를 스스로 단계로 나눌 수 있는 지 먼저 분석해야만 한다. 이를 위해 사용자가 일반적으로 스크린에 손가락을 눌렀을 때 차지하는 면적을 분석해야만 한다. 즉, 어떠한 사용자도 3D-Touch 에서 가능한 입력(약한 포스터치(3D-Weak), 강한 포스터치(3D-strong), 일반적인 터치 2D-Touch)에 대한 면적이 일정한 지를 분석해야만 한다. 이를 통해 일정한 손가락 면적을 구할 수 있다면 각 면적을 구분하는 값을 통해 가상의 3D-Touch를 구할 수 있을 것이다. 하지만, 일반적으로 사용자들의 손가락 면적은 각기 다르기 때문에 위와 같은 방법으로 일정한 구역을 나눌수는 없을 것이다. 따라서 자신이 강한 포스터치(3D-strong)를 눌렀을 때의 면적(손가락 입력 시 가장 넓은 면적)을 기준으로 이를 단계별로 면적을 나눌 수 있다면 사용자가 자신의 손가락 면적을 구분할 수 있다. 이를 통해 가상의 3D-Touch를 구현할 수 있다.

즉, 본 논문에서는 실험 A에서 사용자 별 손가락 입력 의한 손가락 면적의 최대치를 기준으로 사용자가 면적을 3단계(Touch, Peak(Weak), Pop(Strong)) 로 구분할 수 있는 지를 실험한다. 또한, 실험 B는 3D-Touch를 Pop(strong)으로 눌렀을 때 나타나는 사용자의 손가락 면적을 기준으로 사용자가 면적을 실험 A와 같이 3단계로 구분하여 입력이 가능한지를 살펴보는 실험이다. 이 두 가지 실험은 사용자에게 특별한 면적에 대한 기준이 정해지지 않은 상태에서 자신만의 손가락 면적을 스스로 설정한 값(면적)에 대한 인식과 조정이 가능한지에 대해서 분석한다.

만약, 위 두 가지 실험과 같이 스스로 조절이 가능하다면 이를 통한 가상 3D-Touch를 구현이 가능할 것이다.

(표 2) 개발 환경
(Table 2) Development environment

	Smart device (Galaxy S3)	Development Computer
CPU(AP)	Exynos 7420 (2.1GHz+1.5GHz)	Intel Core i7 2.3GHz
Memory	3GB	8GB
OS	Android 5.0.2	Windows 10 pro
Language	Java 1.8	Java 1.8
Display	QHD Super AMOLED Dual Edge 577ppi	-
Touch Type	Capacitive Sensing	-



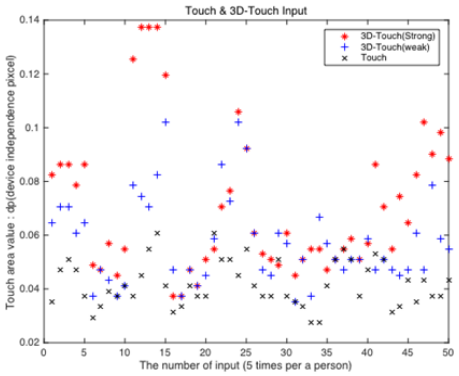
(그림 3) 안드로이드 가상 포스 터치 측정을 위한 실험 화면
(Figure 3) Android app for measuring virtual 3D-Touch

3.2 실험 환경 및 실험 방법

위 두 실험을 위해 다음과 같이 실험을 설계하였다. 먼저 사용자의 포스터치와 터치에서의 면적 변화를 측정하기 위해서 10명의 실험자를 남자 5명, 여자 5명으로 설정하였다. 사용자는 총 5번의 같은 task를 실험하도록 하였다. 실험을 위해 갤럭시 s6를 사용하였고, 개발환경은 표 1과 같다.

3.3 실험 A: 사용자별 손가락 기준 면적 구분

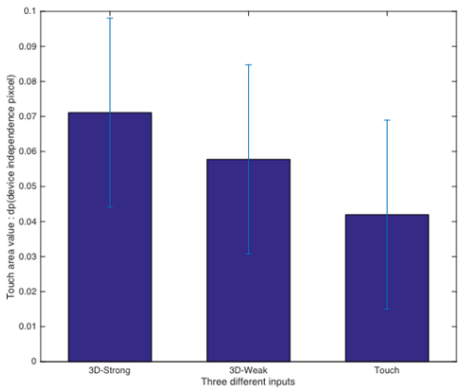
사용자가 누르는 입력 중에서 힘을 가하지 않는 Touch, 약한 힘을 가하는 3D-Weak, 강한 힘을 가하는 3D-Strong의 면적을 측정하여 사용자가 임의로 누르는 값(면적)이 사용자 스스로 구분할 수 있는지를 파악해야 한다. 만약 이 값(면적)들이 각 사용자 자신들뿐 아니라 실험에 참가한 각 사용자들도 비슷한 힘으로 힘의 영역이 구분될 수 있다면 사용자는 임의 값으로도 면적에 비례하여 포스에 대한 입력 값을 구분시킬 수 있다. 이 실험을 위해서 그림 3과 같이 사용자에게 아무런 입력 기준을 제시하지 않고 버튼에 3가지 입력(Touch, 3D-Weak, 3D-Strong)을 수행하도록 하였다.



(그림 4) 사용자의 3가지 입력(Touch, 3D-Weak, 3D-Strong)에 따른 dp 분포도
(Figure 4) dp distribution on user three input (Touch, 3D-Weak, 3D-Strong)

측정된 데이터의 결과는 그림 4와 같이 사용자별 3개 입력값을 DP(Device independence Pixel)로 측정하여 표현하였다. DP값으로 비교한 이유는 사용자 디바이스별로 해상도가 다르기 때문에 픽셀(pixel)값으로 표현하게 되면 각 디바이스마다 사용자 입력 영역이 달라지기 때문이다. DP값으로 설정하면 어떠한 디바이스에서든 같은 크기의 면적을 표현할 수 있다. 따라서 측정 단위를 DP값으로 변환해서 표현하였다.

그림 4 에서 사용자는 5번씩 반복 실험하였기 때문에 0~5까지 비슷한 값의 경향을 보인다. 하지만 사용자들끼리 비교를 하면 Touch, 3D-Weak, 3D-Strong영역이 서로 겹치는 것을 볼 수 있다. 비록 터치 영역이 0.027~0.061 정도로 어느 정도 값의 영역이 유지되지만, 3D-Weak과 3D-Strong 영역은 서로 간에 구분을 할 수 없을 정도로 값이 겹치기 때문에 사용자가 임의로 설정한 입력으로는 구분될 수 없음을 알 수 있다. 이를 위한 정확한 차이는 다음 그림 5를 통해서 알 수 있다. 3D-Weak의 영역은 0.0352pd ~ 0.10196pd이고, 3D-Strong는 0.0372dp ~ 0.1372dp로 상당부분 서로간의 영역을 구분하기 힘들다. 이는 그림 5의 각 입력값 평균에 대한 표준편차를 통해 극명하게 나타낼 수 있다. 3D-Strong은 평균 0.0711dp로 편차가 0.02698dp로 나타낸다. 3D-Weak값은 평균 0.0577dp이고 편차는 0.0164dp로 나타낸다. 이는 3D-Weak와 3D-Strong을 비교해봤을 때 사용자가 이 임의로 설정한 면적으로는 이 두 값을 구분하기 어려운 것을 나타낸다. 또한 Touch는 평균 0.0419dp이고 편차가 0.0084dp로 편차의 값은 적으나 이 역시 3D-Weak와 구분이 어렵고 심지어 3D-Strong과도 구분하기 힘들다. 따라서 사용자가 임의로 입력한 3개의 값으로는 면적을 구분하기 어려운 결과를 보인다.



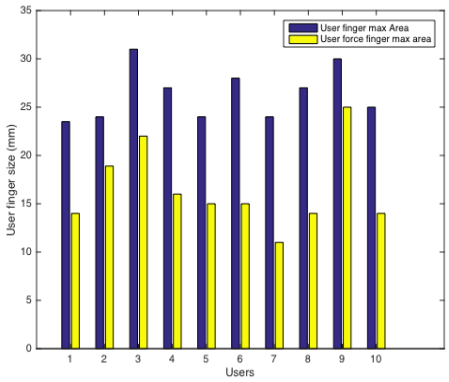
(그림 5) 사용자의 3가지 입력에 따른 면적 분포비교
(Figure 5) Comparison of user's finger area by three input

3.4 실험 B:사용자별 3D-Strong 기준 면적 구분

사용자 별로 3D-Strong으로 최대 손가락 면적을 기준으로 면적을 구분할 수 있는가를 알아보기 이전에 3D-Strong 시에 사용자 별 비슷한 혹은 같은 손가락 면적을 가질 수 있는지 먼저 살펴봐야만 한다. 만약 사용자들이 3D-Strong 시 비슷한 면적을 가진다면 이를 기준으로 면적을 구분하기 위한 기준 면적을 잡을 수 있기 때문이다. 따라서 본 실험에 앞서 사용자가 누를 수 있는 손가락의 최대 면적과 3D-Strong에서 사용자가 스크린에 누르는

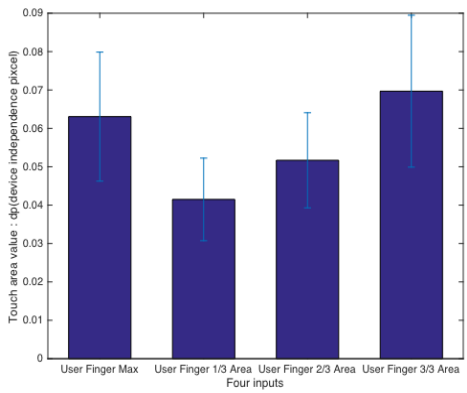
면적을 비교하였다.

사용자들의 손가락 최대 면적과 3D-Strong를 했을 경우에 나타나는 손가락의 면적 변화를 나타낸다. 그림에서 각 실험자(Y1 ~ Y10)의 왼쪽 이미지는 자신이 누를 수 있는 최대 면적의 손가락 면적이고, 오른편은 3D-Strong시의 면적을 나타낸다. 각 사용자의 면적의 크기를 수치로 비교한 것은 그림 6과 같다. 사용자가 3D-Strong시 손가락 면적의 평균 넓이는 16.49mm이다. 하지만 그림 6에서와 같이 사용자별로 가질 수 있는 3D-Strong면적이 최대 22.5mm에서 최소 11mm로 그 편차가 크기 때문에, 이를 기준 면적을 지정하여 모든 사용자의 최대 면적을 만족하기 위한 평균값을 설정하는 것은 무리가 있다. 이는 측정된 손가락의 평균값을 기준으로 최대 면적을 설정한다 하더라도 평균에서 +-10%내의 오차를 갖는 기준 면적을 만족하는 사용자는 Y4, Y5, Y6로 단지 30%만이 면적의 기준을 만족하기 때문에, 사용자가 누르는 임의의 3D-Strong면적을 기준으로는 가상 포스 터치를 위한 시스템에서 이를 반영하는 것은 적절하지 못 하다.



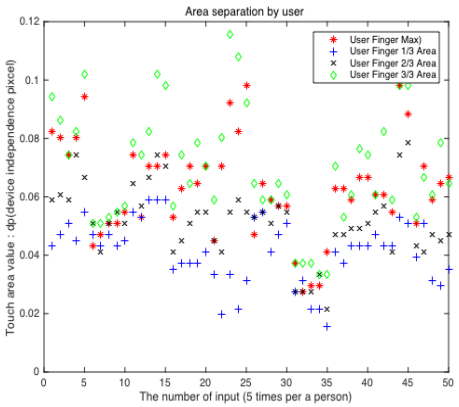
(그림 6) 사용자 손가락 면적비교(사용자 최대 손가락 면적과 3D-Strong 면적)
(Figure 6) Comparison with maximum user's finger area and maximum user's 3D-Strong area

위 결과에 따라 3D-Strong시의 명확한 기준 값을 이용해서 3D-Touch를 인식할 수 없기에 사용자가 자신이 정한 3D-Strong 면적을 기준으로 면적을 단계적으로 나눌 수 있는지에 대해서 실험하였다. 이는 실제 3D-Touch를 사용할 때의 면적을 사용자가 가상 포스 터치에 사용할 수 있는지 알아보기 위함이다. 사용자가 3D-Strong을 했을 경우 면적을 최대 면적으로 정하고 이를 task1으로 측정하였다. task 1의 면적을 기준으로 1/3의 면적(task2), 2/3 면적(task3), 그리고 다시 처음 정했던 면적 3/3 (task4) 을 사용자에게 입력하게 하여 총 네 번의 작업을 수행하도록 하였다. 그림 7에서 task 1의 과 task 4의 결과가 어느 정도 일치하는 결과를 보여준다. 그림 7에서와 같이 대체적으로 처음 설정한 값(task1) 보다는 단계별로 구분하여 원래의 면적에서 다시 실행했을 경우에 면적 값이 크게 나타난다. 이는 사용자가 자신이 처음 누른 3D-Strong면적을 정확하게 측정하지 못 한다는 것을 보여준다. 즉, 사용자는 자신이 설정한 3D-Strong시의 면적을 인지하지 못 한다는 것을 반영한다.



(그림 7) Task별 평균값기준 표준편차
(Figure 7) Standard deviation by tasks

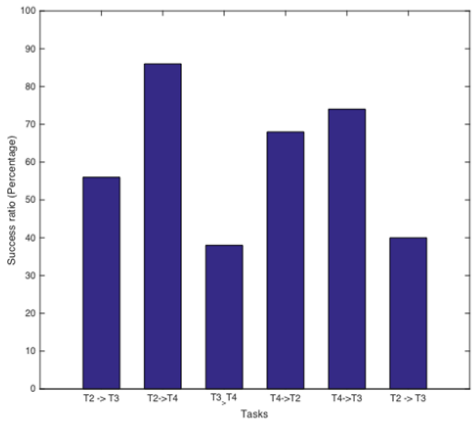
그림 8에서와 같이 task2(Touch), task3(3D-Weak), task4(3D-Strong)를 구분하기란 쉽지 않다. 하지만 총 실험 횟수 50회 중 주어진 task들을 구별하지 못 한 경우는 단 5회 만을 기록하였다. 즉, 사용자가 자신의 3D-Strong을 구분하기 위해 면적을 조절하는 것은 90%이상의 성공률을 보인다. 이 결과로 각 task들을 구분할 수 있다고 판단할 수 있으나 각 task들을 구분하기 위한 차이 값이 매우 적기에 이를 구분하기란 어렵다. task2와 task3를 구분하기 위한 평균 dp 값은 0.0107dp이고 task3와 task4를 구분하기 위한 평균 dp값은 0.0177dp이다. 이는 매우 작은 값으로 사용자가 의도적으로 값을 구분하여 인식하기엔 무리가 있다. 또한, 본 실험에서 사용자가 단계를 면적으로 구분하기 위해서는 task2와 task3 그리고 task4의 각 단계를 구분할 수 있느냐가 중요한데 위와 같은 값으로는 사용자가 면적의 값을 조절할 수 없다.



(그림 8) 사용자별 4개의 서로 다른 값 입력(3D-Strong, 1/3 면적, 2/3면적, 3/3면적)
(Figure 8) Comparison of four inputs (3D-Strong, 1/3 area, 2/3 area, 3/3 area)

그림 7에서 task별 평균 표준 편차 값을 보면, task2와 task3의 값은 상당부분이 겹쳐져 있고 심지어 task2와 task4와도 값이 겹쳐져 있다. 또한 task3를 기준으로 구분해도 task2와 task4는 겹쳐져 있음을 알 수 있다. 따라서 이 겹쳐진 값들은 사용자의 손가락 평균을 기준으로 값을 설정한다 하더라도 사용자별 특성에 의한 크기 변화를 단계로 규정지을 수 없고, 이는 곧 가상 포스 터치를 위해 단계별로 입력 값을 제시할 수 없다. 이는 그림 9에서와 같이 편차 범위에서 각 task별 입력 시 성공률을 보면 확실히 알 수 있다.

그림 9는 task별 입력 성공률을 보여준다. x축은 각 task별로 입력을 요구할 경우 이 값이 다른 task에 영향을 미칠 수 있는 경우를 보여준다. 즉 T2->T3는 task2를 입력받기 원할 경우 이 값이 task3 영향을 주는 지를 판단한다. 그림 9에서 task2를 누를 때 task3의 면적과 겹치는 확률이 44%로 매우 높고, task4일 경우에도 14%나 된다. 따라서 성공률이 각각 56%, 86%로 나타난다. 즉, task2는 task4와는 확연히 구분이 되지만 task3와는 구분되기 힘들다. 즉, 사용자가 터치(task2)를 누르는 면적과 3D-Strong (task4)면적은 구별은 된다는 것이다. 반면에 task3와 task2와의 관계(task3를 누를 때 task2영역과 겹치는 영역)에서는 단지 38%의 성공률을 보이고 task3와 task4는 단지 68%의 성공률만 보인다. 이는 사용자는 작은 면적보다는 큰 면적 구별이 쉽다는 것을 보여준다. 이는 3D-Strong시의 면적이 워낙 작기 때문에 크기가 클수록 구분하기 쉽고, 일반적으로 작은 면적보다는 큰 면적을 구분하기가 쉽기 때문이다. task4의 기준 성공률은 task2 대비 74%이고, task3대비 40%로 task2와 task4와의 성공률은 대부분은 높다. 즉, 사용자 자신이 임의로 정한 영역일 지라도 단계를 2단계, Touch와 3D-Touch만을 구별할 수 있는 근거를 마련할 수 있다. 하지만 기존의 3D-Touch에서 사용되는 3D-Weak와 3D-Strong 그리고 Touch와 같이 3단계로 구분되기 때문이 이 실험과 같이 2단계로 구분하는 것은 기존 3D-Touch 인터랙션 방식과 다르다.



(그림 9) Task별 단계 입력 성공률
(Figure 9) Comparison of task success ratio

4. 면적 제시에 따른 가상 포스 터치 단계 구분 및 구현 방안

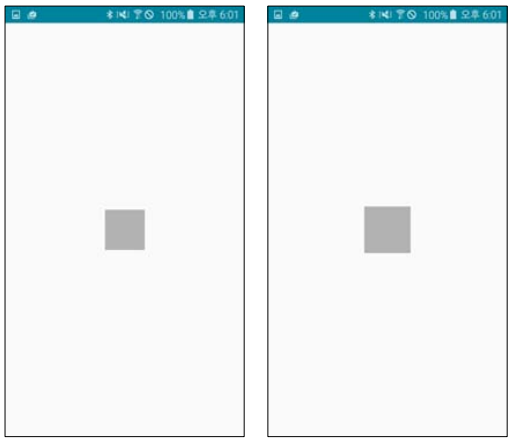
4.1 단계구분을 위한 최소 면적 제시

위 실험 A, B를 통해 사용자가 임의로 설정한 면적을 구분하지 못 하는 이유는 먼저 자신의 손가락이 스크린에 닿는 면적을 확인할 수 없고, 면적을 추측하기가 어렵기 때문이다. 따라서 본 논문에서는 사용자가 인지할 수 있는 면적을 제시하고 이 제시된 면적으로 사용자가 3D-Touch와 같이 3단계(Touch, 3D-Weak, 3D-Strong)를 구분할 수 있는지를 실험한다. 이를 통해 면적을 구분할 수 있는 임계값(threshold)을 찾아내어 가상 포스 터치를 위해 활용할 수 있는 기준 값을 찾는다.

위 두 실험에서 사용자가 주어진 기준 면적이 없을 경우라도 74%이상의 두 단계를 구분하기 위한 가능성을 보

였다. 따라서 주어진 면적의 경우 사용자가 면적에 대한 기준을 더 확실하게 인지할 수 있을 것이다. 주어진 면적에 대한 기준은 아이폰에서 사용하는 아이콘의 크기 9.5mm와 11mm를 기준으로 설정하였다. 이는 기존의 실험[11]을 통해 사용자가 아이콘에 정확하게 터치 할 수 있는 최소 크기보다 큰 아이콘의 크기이기 때문에 아이콘을 정확히 터치할 수 있을 만한 크기이다.

실험 환경은 3장과 같은 환경으로 설정하였고, 그림 10과 같이 9.5mm의 정사각형 버튼과 11mm의 정사각형 버튼을 생성하여 사용자에게 각 5회식 task1(1/3 면적), task2(1/2면적), task3(2/3 면적), task4(3/3 면적)을 입력하도록 하였다. 사용자가 자신이 입력한 면적을 알지 못 하게 아무런 피드백(feedback)도 설정하지 않고, 단지 사용자가 버튼의 면적만을 크기에 따라 입력을 하도록 하였다.



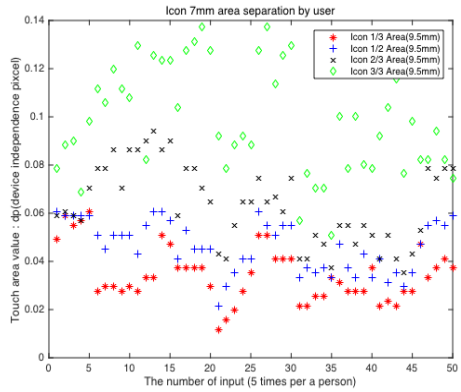
(a) 9.5mm (b) 11mm

(그림 10) 아이콘 크기 별 실험

(Figure 10) Android app for measuring user's finger area when (a) 9.5mm and (b) 11mm

4.2 단계 구분을 위한 최소 면적 비교 및 분석

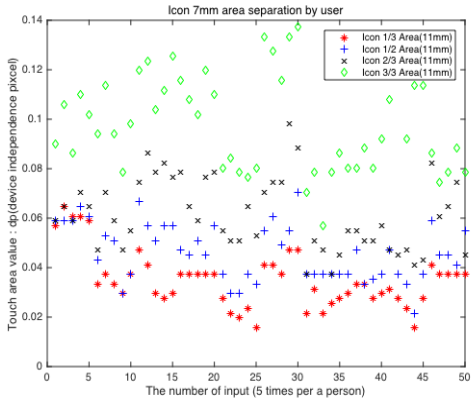
그림 11은 버튼을 9.5mm로 설정하였을 경우 각 task별 평균에 대한 표준 편차 값을 나타낸다. 사용자가 임의로 설정한 방법인 그림 4와 그림 8과 비교하여 각 task별로 다른 task영역에 겹치는 부분이 적다. 즉, 구분이 명확해짐을 알 수 있다. 특히, task2(1/3 영역)과 task3(2/3 영역), task4(3/3)의 면적이 이전에 비해 서로 영역이 겹치는 부분이 줄어듦을 알 수 있다.



(그림 11) 9.5mm 아이콘에서 사용자 task별 입력 분포도

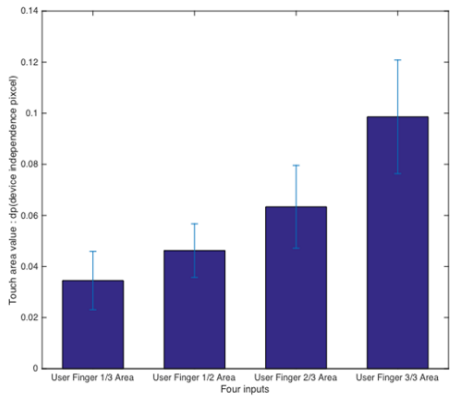
(Figure 13) Task input distribution on user's input in 9.5mm icon

그림 12와 같이 버튼 11mm로 설정하였을 경우에도 더욱 두드러지게 면적의 겹치는 범위가 명확히 두드러진다. 두 그림에서 task1과 task3에서 서로 간섭하는 영역이 현저히 줄어들었고, 특히 task2과 task4는 완벽히 구분됨을 알 수 있다. 즉, 사용자가 지정된 그림영역에서 면적을 통한 입력을 구분하는 것이 2단계 이상을 넘어 3단계 3D-Touch와 같이 나타낼 수 있음을 보여준다. 하지만, task1과 task2는 task2와 task3는 면적의 변화량이 적어 여전히 면적을 구분하는 것은 어렵다.



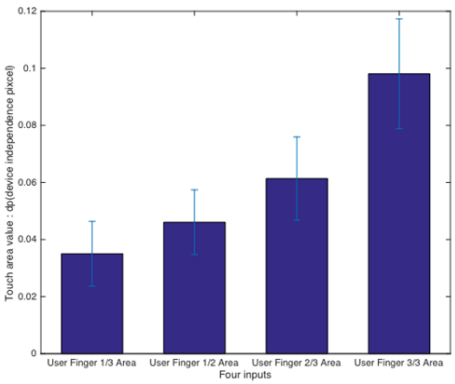
(그림 12) 11mm 아이콘에서 사용자 task별 입력 분포도

(Figure 14) Task input distribution on user's input in 11mm icon



(그림 13) 9.5mm 버튼에서 면적 단계 구분
(Figure 11) Area division step in 9.5mm icon

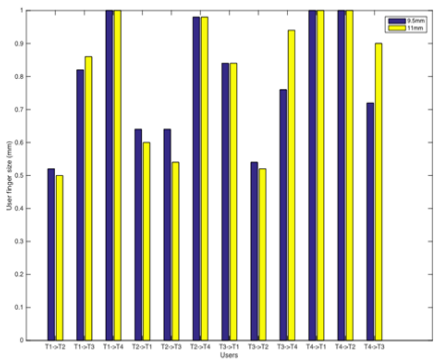
9.5mm아이콘을 주어졌을 경우가 11mm아이콘을 주어졌을 경우보다 평균에 대한 표준편차도 작아진다. 이는 보다 넓은 면적에서 면적을 구분하기가 용이하다는 것을 알려준다. 또한, 사용자가 임의로 설정한 면적을 기준으로 면적을 통한 단계를 나누는 것 보다는 제시된 크기를 사용자가 나눌 수 있고, 보다 정확한 면적 구분을 할 수 있다는 것을 보여준다. 이는 그림 14과 15에서와 같이 사용자의 task별 평균값에 대한 표준편차를 통해 아이콘 9.5mm와 11mm에서 task들의 영역을 확연히 구분될 수 있음을 보여준다.



(그림 14) 11mm 버튼에서 면적 단계 구분
(Figure 12) Area division step in 11mm

그림 5와 7을 비교해서 그림 13와 14에서 각 task별 면적의 분포도를 보면 주어진 면적에 대한 영역이 task별로 더 명확히 구분이 되어 있다. 그림 13의 9.5mm 아이콘을 제시했을 경우와 그림 14의 11mm 아이콘을 제시했을 경우에 11mm의 경우가 더욱 명확한 구분된 영역을 task별로 보임을 알 수 있다. 특히, task2를 제외했을 경우를 보면 task1, task3, task4만을 구분하면 이러한 성향은 더욱 명확해 진다.

9.5mm에서 DP 값을 기준으로 task1은 최소 0.0222dp에서 최대 0.0459dp 값을 보이고 task3에서는 최소 0.0454dp에서 최대 0.0798dp 값을 보여 두 영이 간섭하기 위한 영역은 단지 0.0005dp만을 보일 수 있다. 반면 task4에서는 최소 0.0719dp에서 최대 0.1219dp를 보여 task3와 0.0079dp의 영역이 겹치는 것을 볼 수 있다. 11mm에서 DP 값을 기준으로 task1은 최소 0.02229dp에서 최대 0.0463dp 값을 보이고, task3에서는 최소 0.0479dp에서 최대 0.07591dp이다. 이에 두 영역이 겹치는 부분은 단지 0.0015dp로 매우 적은 영역을 간섭한다. task4에서는 최소 0.0754dp에서 최대 0.1179dp로 영역을 차지하고, 매우 적은 0.0004dp 만큼의 영역을 간섭하다. 즉, 이는 11mm에서는 task1, task3, task4와 서로 구분될 수 있음을 보여준다.



(그림 15) 9.5mm와 11mm 별 성공률
(Figure 15) Comparison with 9.5mm and 11mm success ratio

그림 15는 그림 9와 같이 task별 사용자의 입력 성공률을 보인다. 즉, task별로 타 task에 간섭되지 않는 비율을 보여준다. 먼저 task1을 누를 경우에 task2, task3 및 task4에게 간섭되지 않을 성공률은 9.5mm에서는 task2와는 약 53%, task3일 경우에는 약 83%, task4는 100% 로 나타난다. 즉, task1와 task2는 서로 구분하기 힘들다는 것을 보여주고 task1과 task4는 완벽히 구분됨을 보여준다. task2

의 경우에는 task1과 task2 둘 다 약 62%의 성공률을 보이고, 오히려 11mm인 경우에 60%이하의 성공률을 보인다. 이는 task2에서 task1과 task2를 구분하기가 쉽지 않음을 뜻한다. task3의 경우 역시나 task2와 구분하기가 어렵지만, task3와는 9.5mm에서는 76%, 11mm에서는 92%의 성공률을 보인다. 이는 task4의 경우에도 task3에서 11mm가 약 90%의 성공률을 보이고 9.4mm일 경우 약 70%의 성공률을 보이는 것과 같이 사용자가 아이콘의 크기가 커질수록 작은 면적보다는 큰 면적을 구분하기가 쉽다는 것을 입증한다. 따라서 위 결과를 기반으로 3D-Touch에서 사용하는 3가지 입력을 가지는 가상의 포스 터치를 구현하기 위해서는 11mm이상 면적을 설정할 경우 최소 83% 이상의 성공률을 보임을 알 수 있다.

5. 결 론

본 연구에서는 가상 포스 터치의 기반을 마련하기 위해 사용자가 실질적으로 3D-Touch를 사용할 때 나타나는 행동패턴을 분석하였다. 2D-Touch와 3D-Touch를 사용할 때 사용자의 스크린 닿는 터치 접점의 면적 변화 비교와 사용자가 직접 이를 인지할 수 있는지를 분석해 2D-Touch에서도 사용자가 3D-Touch와 같은 입력을 할 수 있는 방안을 제시하고 결과를 도출해 내었다. 최소한 83%이상의 성공률을 위해서 사용자가 인식할 수 있는 면적을 11mm의 아이콘으로 제시해야만 한다. 본 연구의 기반을 통해 가상 포스 터치가 3D-Touch를 사용할 수 없는 디바이스에 적용될 것이라 예상한다. 이를 통해 3D-Touch의 인터랙션 효과 및 다양한 기능에도 불구하고 하드웨어적 한계로 3D-Touch를 이용한 사용 경험을 전달해 줄 수 있다. 또한, 실질적인 하드웨어의 개발이 필요하지 않기 때문에 물질적인 비용을 최소화시키면서도 부가적인 인터랙션을 할 수 있다는데 의의가 있으며, 소프트웨어 제공자들이 3D-Touch 기기 보유 상황에 관계없이 다양한 어플리케이션에 3D-Touch 기능을 개발할 수 있도록 기반을 마련해 줄 것이다.

참 고 문 헌 (Reference)

[1] Shneideman, Ben. “Designing the user interface : strategies for effective human-computer interaction”, Addison-Wesley, 1992.

[2] Choonmo Ahn, ″Studies of Current Status and Future Prospect of UI″, Innovation studies vol.6, number 2, pp.109-135, 2011.
http://www.ksoi.or.kr/sub/sub04_01_01.php?category=7

[3] S.K Heo, G.H Lee, “Force Gestures : Augmenting Touch Screen Gestures with Normal and Tangential Forces”, UIST’11, pp.621-626, USA, 2011.
http://dx.doi.org/10.1145/2047196.2047278

[4] Craig Stewart, Michael Rohs, Seven Kratz, Georg Essl, “Characteristic of Pressure-Based Input for Mobile Devices”, CHI 2010, USA, 2010.

[5] B.R. Lee, H.J Lee, S.C Lim, H.G. Lee, S.J. Han, J.H. Park, “Evaluation of Human Tangential Force Input Performance,” CHI 2012, USA, May, 2012.
http://dx.doi.org/10.1145/2207676.2208727

[6] S.K. Heo, G.H, Lee, “Force Gestures: Augmenting Touch Screen Gestures Using Normal and Tangential Force”, CHI 2011, pp.1909-1014, Canada, 2011.
http://dx.doi.org/10.1145/1753326.1753444

[7] James Scoot, Lorna M. Brown, Mike Molloy, “Mobile Device Interaction with Force Sensing”, 7th International Conference, Pervasive, vol. 5538, pp. 133-150, Japan, 2009.
http://link.springer.com/chapter/10.1007/978-3-642-01516-8_10

[8] 3D Touch. [Online] available:
http://www.apple.com/iphone-6s/3d-touch/

[9] Smart Pressure-Sensitive Screen. [Online] available:
http://consumer.huawei.com/minisite/worldwide/mateS/touch.htm

[10] Craig Villamor, Dan Willis, Luke Wroblewski, “Touch Gesture preference guide”, [Online] available:
http://static.lukew.com/TouchGestureGuide.pdf

[11] Pekka Parhi, Amy K. Karlson, Benjamin B. Bederson, “Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices”, MobileHCI’06, pp.203-210, Finland, 2006.
http://dx.doi.org/10.1145/1152215.1152260

◐ 저 자 소 개 ◑

남 춘 성 (ChoonSung Nam)
2005년 상명대학교 소프트웨어학과 졸업(학사)
2007년 숭실대학교 대학원 컴퓨터학과 졸업(석사)
2011년 성균관대학교 대학원 전자전기컴퓨터학과 졸업(박사)
2016~현재 성균관대학교 컨버전스연구소 선임연구원
관심분야 : VANET, IoT, UAV & Force Touch interaction, etc
E-mail : namgun99@gmail.com