

포스 터치 커버를 이용한 사용자 중심적 포스 터치 측정에 관한 연구

A Study on User-Centric Force-Touch Measurement using Force-Touch Cover

남 춘 성¹ 서 민 수² 신 동 렬^{1*}
ChoonSung Nam Min-soo Suh DongRyeol Shin

요 약

터치 인터페이스는 스마트 디바이스에서 널리 사용하고 있는 입력 방식이다. 최근 터치 입력 방식에서 힘 인식을 포함한 새로운 방식의 터치 인터페이스인 포스터치가 등장하였다. 포스터치는 사용자의 입력을 위치 뿐 아니라 압력을 제공함으로써 한 번의 터치에 두 가지 이상의 입력을 제공하여 제한된 공간에서 다양한 입력이 가능한 방식이다. 다양한 입력이 가능하기 때문에 포스터치를 이용하는 디바이스는 기존의 일반적인 터치를 이용하는 디바이스와는 다른 사용자 인터랙션이 요구된다. 이는 기존의 터치 방식이 포스터치에서 이용할 수 있는 입력의 깊이(depth)를 사용하지 못 하기 때문이다. 따라서 본 논문에서는 기존의 포스터치를 제공하지 못 하는 스마트 디바이스에게 포스터치를 위한 방안으로 포스터치 커버를 제작한다. 제작된 포스터치 커버를 통해 압력을 측정하는 센서의 정확성과 압력을 가하는 위치에 따라 변화하는 압력값을 분석한다. 또한, 사용자가 포스터치를 사용할 때 요구하는 피드백(feedback) 시간을 분석하여 사용자 중심적인 포스터치 구현을 위한 방안을 제시한다.

☞ 주제어 : 포스터치, 포스터치 커버, 사용자 중심적 포스 입력 방법

ABSTRACT

Touch interface has been introduced as one of the most common input devices that are widely used in the Smart Device. Recently Force-Touch interface, a new approach of input method, having the power recognition mechanism, has been appeared in Smart industries. Force-Touching determining multiple things (the geographical and pressure values of touching point) in one touching act allows users to provide more than one input methods in a limited environments. Force-Touching Device is required different user communicational interaction than other common Smart devices because it is possible to recognize various inputs in the one act. It means that Force-Touching is only able to understand and to use the pressure sensitive values, not other Smart input methods. So, we built Force-Touch-Cover that makes typical Smart-Device to have Force-Touching interfaces. We analysis the accuracy of the Force-Touching-Cover's sensor and also assessment the changes in pressure values depending on the pressure position. Via this Paper, We propose the implement of user-oriented Force-Touching interface that is based on users' feedback as our conclusion.

☞ keyword : Force-Touch, Force-Touch Cover, User-Centric Force-Touch Measurement

1. 서 론

최근의 퍼스널 컴퓨터, 스마트 디바이스 및 기타 멀티 미디어 기기들은 사용자 친화적인 입력방식을 채택하고 있다. 특히, 스마트 디바이스들은 터치 스크린 혹은 터치 패널을 이용한 입력방식을 사용하여 구동한다. 터치스크

린 혹은 터치 패널은 스크린에 위치에 사람의 손 혹은 물체가 스크린에 접촉하는 경우 스크린의 위치를 검출하여 입력으로 사용하는 사용자 인터페이스이다. 접촉된 위치 정보는 스크린의 X, Y 좌표를 계산하여 이를 입력정보로 사용한다. 다수의 입력을 지원하기 위해서 멀티터치 포인트를 이용한 입력 방식을 사용한다. 이러한 방식은 멀티터치 방식으로 스마트 디바이스의 조작은 줌인, 줌아웃, 화면이동, 게임조작 등에 이용된다[1].

최근 멀티터치 방식의 X,Y 좌표를 통한 입력 방식 외에 Z축 방향의 터치 힘의 크기를 측정하는 새로운 터치 스크린 입력 방식인 포스터치가 이슈화 되고 있다[2]. 접촉된 포인트와 포스를 통한 입력방식은 압력(pressure 혹은 force) 센서를 이용하여 힘의 크기와 위치정보를 연동

¹ College of Information and Communication Engineering, Sungkyunkwan University., Suwon, 16419, Korea

² UX Lab, Graduate School of Information, Yonsei University, Seoul, 03722, Korea

* Corresponding author (drshin@skku.edu)

[Received 2 January 2017, Reviewed 10 January 2017, Accepted 22 March 2017]

하는 새로운 입력방식이다. 펜 기반 포스터치는 터치스크린 위에 스타일러스 펜(stylus pen)으로부터 제공하는 압력 정보를 활용하여 파라미터 조절 정밀도를 변화 시켜 압력에 의해 위젯(widget)이 제어되는 방식이다[3, 4]. 펜과 다르게 손가락을 이용하는 압력을 측정하는 방법은 모바일 기기 아래 압력센서를 추가하는 방법[5]과 압력 센서를 스마트디바이스에 직접 추가하는 방법이 있다.

포스터치 입력을 이용하는 스마트 디바이스는 사용자 반응에 만족할 수 있는 인터랙션(반응)을 제공해야만 한다. 일반적인 터치 입력 방식은 즉각적으로 사용자에게 반응을 제공하면 되지만, 포스터치 입력 방식은 스크린에 접촉되는 시간과 힘이 발생하는 시점이 다르기 때문에 이를 위한 방안이 필요하다. 또한, 사용자가 포스(force)에 대한 강함과 약함에 대해 서로 다르게 인지하기 때문에 이를 일반적인 힘의 세기로 구분하여 사용하기 위해서는 사용자 실험을 통해 일반화된 임계값을 정의해야만 한다. 이러한 임계값을 이용하여 포스터치가 가능한 일반적인 응용 프로그램에 적용할 수 있어야만 한다.

포스터치를 이용한 스마트 디바이스는 아직까지 애플의 아이폰 6s와 7[6], 화웨이사의 mate s[7]와 같이 한정적인 제품에서만 사용되고 있다. 즉, 다양한 포스터치를 활용한 연구와 활용성에 대한 장점에도 불구하고 기존의 포스터치를 지원하지 않는 스마트 디바이스로 인하여 장점을 활용하지 못하고 있다. 따라서 본 논문에서는 포스터치를 지원하지 못 하는 디바이스를 위해 포스터치 측정이 가능한 포스터치 커버를 제작한다. 나아가 이를 사용하여 사용자 중심적인 포스 입력을 이용하기 위한 방법을 제안한다.

2. 관련 연구

2.1 포스터치 인터페이스 방식의 중요성

포스터치 방식이 기존의 터치방식과 비교하여 더 좋은 인터페이스(interface)를 가지는 이유는 사용자가 터치스크린을 사용하여 손가락으로 디바이스를 조절할 수 있지만 실제로 다양한 운동을 조절할 수 있는 방안이 마련되어 있지 않기 때문이다[8]. 기존의 터치 방식은 2차원 공간으로 대량의 정보를 디스플레이 할 때 윈도우가 오버랩(overlap)되거나 정보의 아이콘 표상(iconic representation)이 필요하다. 또한, 2차원 공간은 직관적이지 않고 사용자로 하여금 관련된 과업(task)를 수행하고 유지하는 데 어려움을 가진다[9]. 반면에 3차원 공간의 인터랙션

(interaction)은 조작 방식이 직접적이고, 일반적인 실시간 피드백(feedback)을 포함하기 때문에 가상현실과 증강현실과 같은 환경에서는 3차원을 직접 조작할 수 있는 인터페이스로의 장점을 가진다.

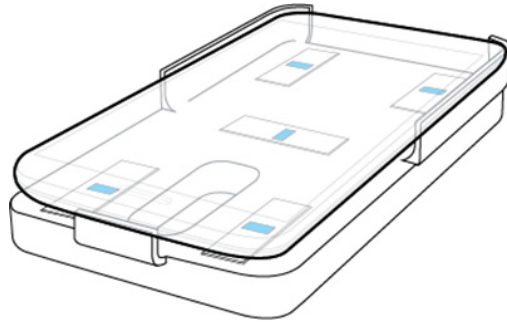
2.2 포스터치 인터페이스를 위한 기존 연구들

포스터치를 측정하기 위한 연구를 위해 이전의 방법들은 스마트 디바이스에 포스터치를 감지할 수 있는 방안이 없었다. 따라서 이를 위해 젤타입의 레이어를 사용하여 젤의 스프링형태로 찌그러지는 형태를 측정하는 방법[10], 스마트 디바이스의 가속도 센서를 이용한 터치 입력을 측정하는 방법[11]과 스마트 디바이스 뒤에 FSR(Force Sensitive Register)을 이용하여 측정하는 방법[12]으로 측정하였다. 레이어를 이용하는 방법은 접촉면에 대한 접선 방향의 힘을 측정하기 위한 방법으로 터치패널 아래 3DOF force sensor를 이용하여 힘을 측정한다. 따라서 젤에 의해서 힘의 크기를 온전히 반영하지 못 하기 때문에 제대로 반영되지 못한다. 가속도를 통한 압력 측정은 강한 터치와 약한 터치를 발생할 때 가속도 센서의 값에 패턴 차이가 존재하는데 이를 이용하여 압력을 측정하는 방법이다. 따라서 압력에 대한 다단계 포스를 구분할 수 없고, 단지 일반적인 터치와 포스터치 값만을 구분할 수 있기 때문에 다양한 압력 값을 이용한 방법에서는 지원할 수 없다. 반면에 FSR 방법은 스마트폰을 누르는 압력을 인지하여 터치된 순간의 압력 값을 측정할 수 있는 장점을 가진다.

하지만 위와 같은 연구들에서 사용자와 디바이스간의 인터랙션을 위한 시간 및 압력 세기에 대한 임계값(threshold)에 대한 연구는 진행되지 않았다. 때문에 개발자 혹은 사용자가 응용프로그램에 포스터치 값을 어떻게 설정 및 사용 할지에 대한 방안이 마련되어 있지 않다. 따라서 본 연구에서는 포스터치 커버를 제작하여 기존의 포스터치를 제공받지 못 하는 스마트 디바이스에게 포스터치 입력을 제공하고, 이에 대한 사용자 반응에 대한 연구를 기반으로 시간 및 압력 세기에 관한 임계값을 찾기 위한 방안을 제시한다.

3. 포스터치 커버 제작

포스터치가 지원되지 않는 스마트 디바이스 실험을 위해서 우선적으로 포스터치 입력을 측정할 수 있는 장치를 개발하였다. 포스터치 입력 장치는 FSR 방식과 같이



(그림 1) 포스터치 측정 커버 디자인 - LG G2용

(Figure 1) Force-touch-measurement cover design for LG-G2 smartphone

스마트폰의 커버 형태로 제작되었고, 5개의 채널을 이용하여 압력의 세기를 각각 측정할 수 있다. 그림 1과 같이 포스터치 측정 커버는 5개의 포스터치 측정 센서(파란색 사각형)와 측정된 포스터치 데이터 전송을 위한 블루투스 장치로 구성되어 있다.

(표 1) 포스터치 커버를 위한 블루투스 데이터 포맷
(Table 1) Bluetooth-data format to measure the force-touch on the cover of the smartphone

채널	측정값
Ch 1 Raw ADC	HEX
Ch 2 Raw ADC	HEX
Ch 3 Raw ADC	HEX
Ch 4 Raw ADC	HEX
Ch 5 Raw ADC	HEX
Ch 1 to Ch 5 Sum ADC	HEX
Force(g : gram)	DEC(50g to 500g)

측정한 포스의 힘은 정역학 이론과 같이 스크린 표면에 가해진 모든 힘의 합과 같다. 따라서 각 센서 채널 5개에서 수집된 데이터 값의 합을 힘의 세기로 지정한다. 포스터치 입력을 측정할 센서는 블루투스 통신을 통해 표 1과 같이 Ch1~Ch5(센서들의 위치에 따른 이름)까지는 측정된 Raw data를 전송한다. 또한, Ch1~Ch5의 Force값과 그것들의 합을 g(gram)으로 환산된 값을 함께 전송한다. 포스의 환산값은 최소 50g에서 최대 500g으로 각 센서가 최소 압력 값 50g에서 최대 500g을 측정할 수 있다. 포스터치 커버와 스마트 디바이스와의 데이터 전송을 위한 블루투스 하드웨어 스펙은 표 2와 같다. 이러한 스마트

커버는 G2 스마트폰을 기준으로 제작되었다. G2의 H/W 스펙과 안드로이드 스펙은 표 3과 같다.

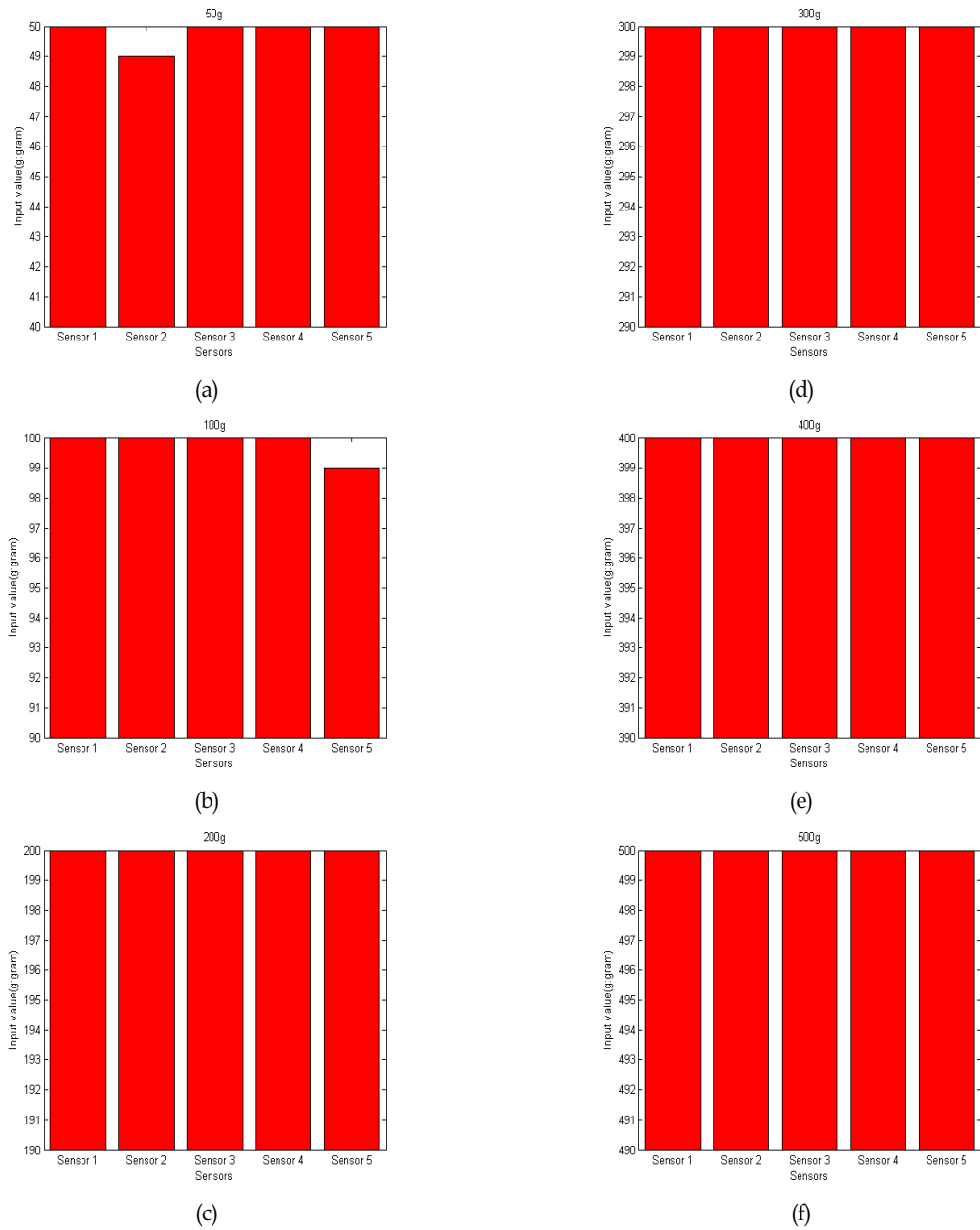
(표 2) 블루투스 장비의 하드웨어 사양
(Table 2) Hardware specifications of Bluetooth device

종류	내용
전송속도	1Mbps to 3Mbps
주파수	2.4GHz
전송세기	1mW
수신거리	10m

(표 3) LG-G2의 하드웨어 사양
(Table 3) Hardware specifications of LG-G2 smartphone

종류	내용
CPU	Snapdragon 800 2.26 GHz Quad-core
OS	Android 5.0
Display	5.2 inch Full HD-IPS
통신방식	LTE
블루투스 버전	4.0

포스터치 측정방법은 포스터치 측정용 커버에 스마트폰을 붙인 다음 고정한다. 스마트 폰에 실행 앱(App)을 실행하여 포스터치 측정 값을 전송받기 위한 블루투스 통신을 먼저 페어링한다. 앱에서 스마트폰을 접촉할 시 발생하는 포스터치 데이터를 커버로부터 수신하여 이를 스마트폰에 저장 혹은 사용한다.

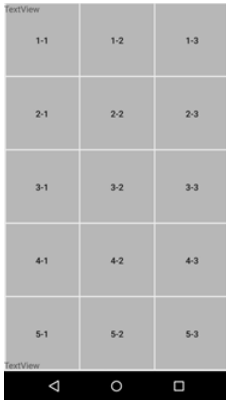


(그림 2) 각 센서 위치별 포스 값

(Figure 1) Force-intension values for each position: (a)50g,(b)100g,(c)200g,(d)300g,(e)400g,(f)500g

4. 포스터치 커버를 이용한 사용자 중심적 포스터치 측정

포스터치 커버를 통해 사용자 중심적 포스터치 인터랙션을 사용하기 위해서는 3가지를 고려하여 실제 앱에 적용해야 한다. 첫 번째, 측정할 값은 사용자가 스크린에 압력을 가하면 각 센서에서 측정되는 압력의 값을 측정한다. 두 번째로는 사용자가 스크린에 힘을 가했을 때 사용자가 포스터치에 대한 피드백을 인식하는 시간을 측정한다. 세 번째로는 사용자가 생각할 수 있는 압력 세기를 터치와 포스터치를 구분하여 입력하였을 때 이를 구분할 수 있는 압력을 측정한다. 이와 같이 3가지 과정과 결과를 통해 응용프로그램에 적용할 수 있는 기준을 마련한다.



(a)



(그림 3) 각 스마트폰 위치별 포스 값 측정을 위한 어플리케이션 (Figure 3) App for measuring the force-intension values for each position: (a) Android-app screen, (b) Test picture

4.1 포스터치 커버의 포스터치 입력 정확성 측정

포스센서에서 측정되는 압력 값을 계산하기 앞서 제작한 포스터치 커버에서 각 센서의 압력 값의 정확성을 먼저 측정한다. 만약 센서의 값에 오차가 발생할 경우 이를 보정해야 한다. 각 센서에서 추 50g, 100g, 200g, 300g, 400g, 500g을 올려놓고 센서 측정값을 검사한다. 그림 2는 각 센서에서 측정된 오차값을 보여준다. 그림 2에서와 50g과 100g의 추를 센서에 올려놓았을 경우에 sensor 2번과 sensor 5번이 1g의 오차를 보인다. 반면에 추가 200g이상일 경우에는 오차값이 발생하지 않고 정확한 무게를 측정하는 것을 보였다. 따라서 100g 이하일 경우에 오차값을 보정하여 압력 값을 측정하였다. 하지만 오차값이 1g이내로 측정되는 되기 때문에 사용자가 가한 실제 압력값에 크게 영향을 받지 않는다.

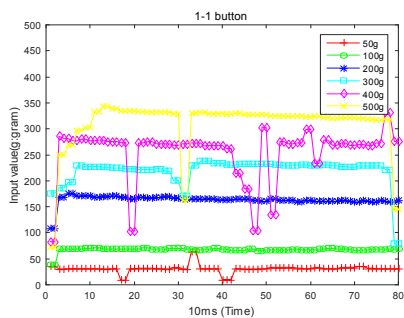
4.2 특정 위치의 포스터치 압력값 측정

포스터치 커버를 이용하여 스마트폰 터치스크린의 위치에 따른 각 위치별 포스터치 값을 측정한다. 스마트폰의 포스터치 커버가 맞닿은 압력센서를 통해 포스터치가 측정되기 때문에 실제 누르는 힘과 측정되는 힘과의 오차가 발생할 수 있다. 따라서 스마트폰의 누르는 위치에 따라서 힘의 분배가 달라질 수 있기 때문에 이러한 측정 실험을 통해서 실제 누르는 힘을 보정해야만 한다. 측정을 위해 그림 3과 같이 앱을 만들어서 위치별 포스터치 값을 측정한다. 그림 3(a)에서 포스터치 압력값 측정 앱은 1-1부터 5-3까지 12개의 버튼으로 구성되어 있다. 버튼 1-1, 1-3, 3-2, 5-1, 5-3은 FSR 센서 중앙에 위치하는 버튼이다. 버튼위치별 정확한 포스터치 값을 입력을 위해서 수식 1과 같이 각 센서에서 수신된 입력값의 합으로 나타낸다.

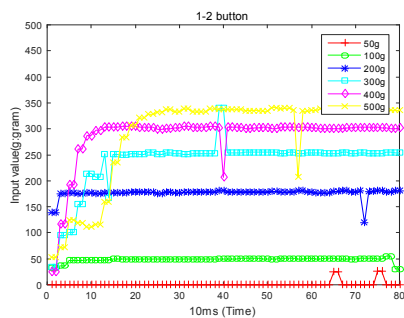
$$F = \sum_{i=1}^5 F_i \quad (1)$$

F는 각 센서들의 압력 값의 합을 나타내고, F_i는 각 센서들의 압력 값이다.

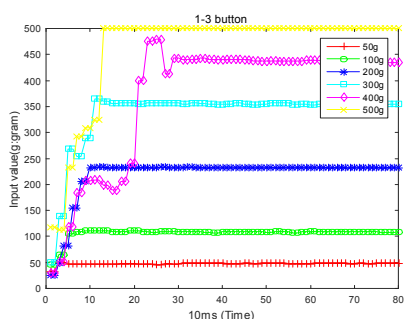
각 위치별 무게 변화(50g ~ 500g)에 따른 압력 값을 측정을 위해 이전 실험과 같이 추를 이용하여 측정되는 압력 값을 파악했다. 이는 그림 4와 같이 무게 변화에 따른 각 버튼 별 압력값 측정 변화를 보여준다. y축은 측정된 무게를 나타내고, x축은 10ms 단위로 측정된 값이다. 즉, 1은 10ms를 뜻하고, 10은 100ms를 나타낸다.



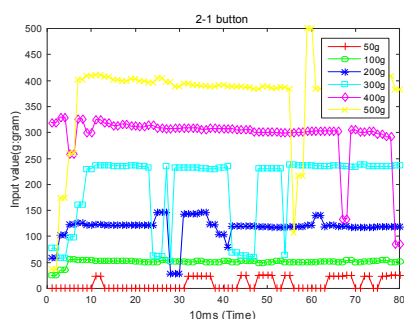
(a)



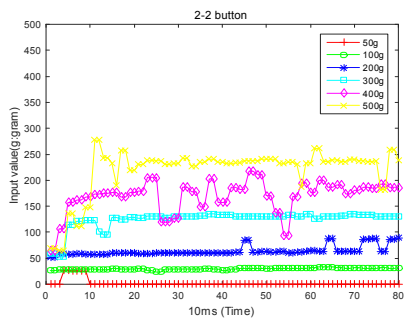
(b)



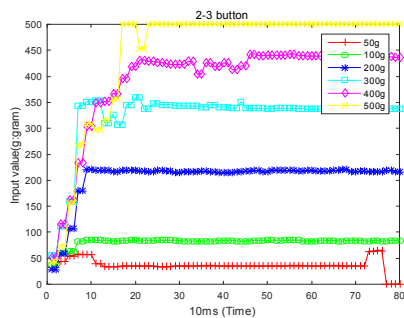
(c)



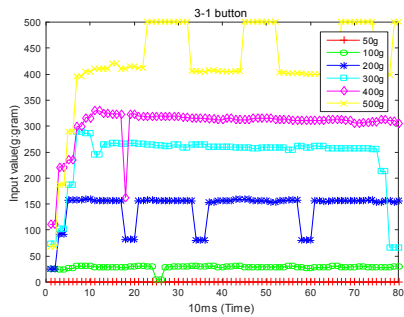
(d)



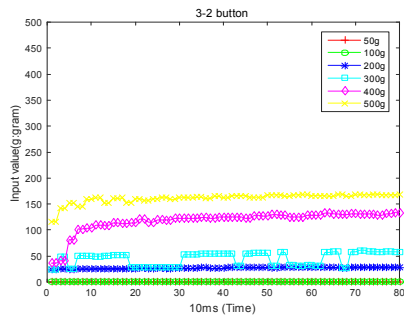
(e)



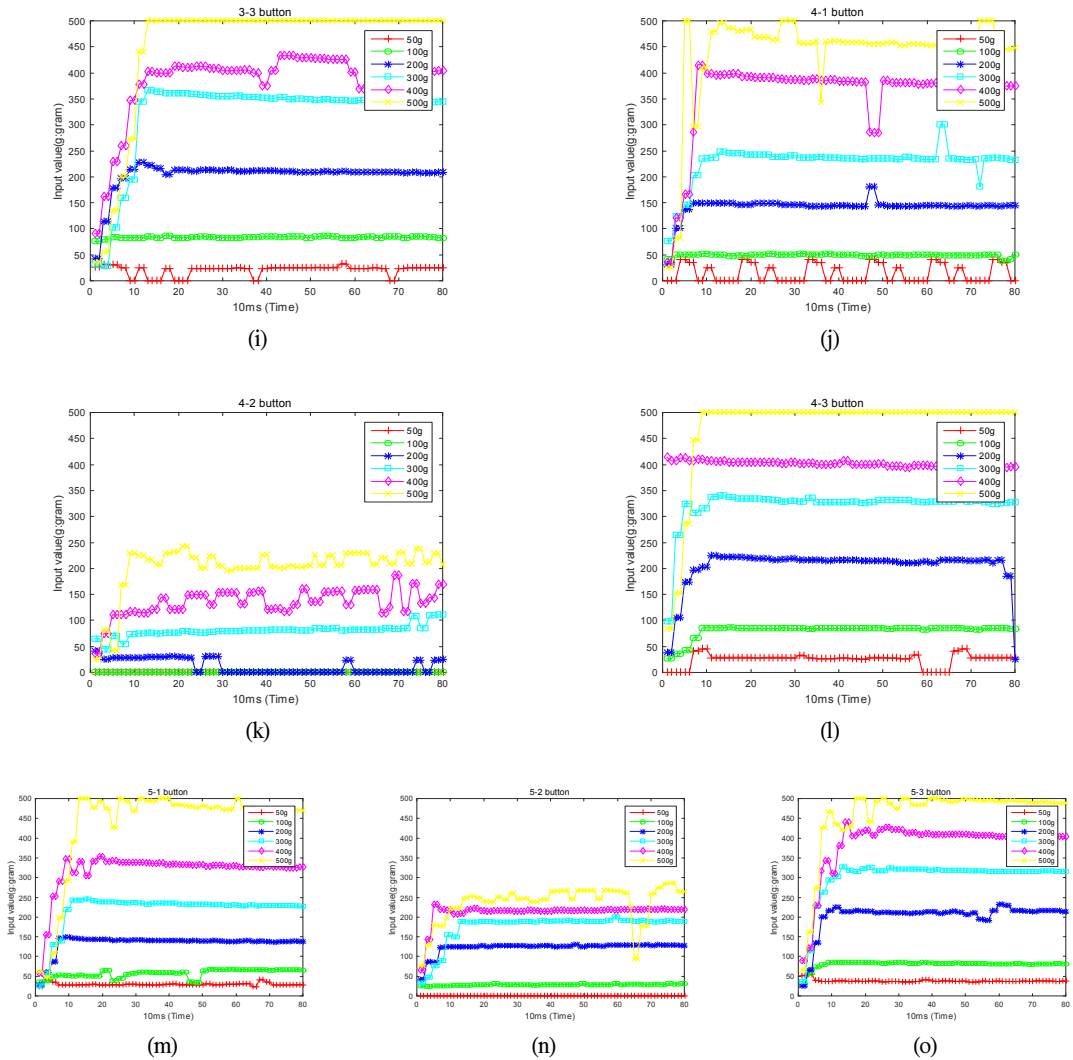
(f)



(g)



(h)



(그림 4) 각 버튼별 포스 값 측정

(Figure 4) Force-intension input values for each position: (a)1-1 button,(b)1-2 button,(c)1-3 button,(d)2-1 button,(e)2-2 button,(f) 2-3 button, (g)3-1 button,(h)3-2 button,(i)3-3 button,(j) 4-1 button,(k)4-2 button,(l)4-3 button,(m)5-1 button, (n)5-3 button,(o)5-3 button

그림 4에서 x축은 포스터치 입력 시간을 나타내고, y축은 포스터치 입력 값을 나타낸다. 위 그림에서 버튼 1-2, 2-1, 2-2, 3-1, 3-2, 4-2, 5-2와 같은 경우 50g일 때 압력의 값이 거의 측정이 되지 않는다. 이는 센서에 가해지는 힘의 값이 스마트폰의 굴곡과 스마트폰 키퍼의 탄성 때문에 발생한다. 특히, 버튼 3-2, 4-2의 위치에 경우에는 200g 이상이 되어야만 힘을 측정할 수 있다. 또한 추가 안

정적으로 압력값이 입력되는 시간은 터치를 한 순간부터 최소 약 100ms에서 최대 200ms의 시간이 소요될 수 있다. 이는 그림 3에서의 실험방법과 같이 고정된 물체의 무게를 측정하기 위한 소요시간으로 사용자가 힘을 가하는 시간과 압력과의 차이가 발생할 수 있다. 즉, 사용자가 스마트폰 스크린에 힘을 가하는 순간에 포스터치 커버가 인지하는 힘의 세기와 압력은 사용자가 입력한 실제 값

(표 4) 각 버튼별 평균 에러의 차이(Mean Margin of error : ME, g)와 안정적인 수신 시간(SRT: Stabilized Receiving Time)

(Table 4) Mean margin of error (ME, g) & stabilized receiving time (SRT, ms) for each button:

button #	50g		100g		200g	
	ME	SRT	ME	SRT	ME	SRT
1-1	12	20	32	20	33	30
1-2	49	-	50	50	23	30
1-3	1	30	2	50	22	100
2-1	42	-	1	50	78	50
2-2	48	-	25	10	147	10
2-3	20	40	21	60	12	90
3-1	-	-	23	10	63	50
3-2	-	-	-	-	25	10
3-3	29	10	27	10	11	100
4-1	32	-	51	10	154	80
4-2	-	-	-	-	178	-
4-3	18	70	13	90	20	110
5-1	22	50	48	60	62	70
5-2	-	-	74	10	73	70
5-3	12	50	29	80	13	90

button #	300g		400g		500g	
	ME	SRT	ME	SRT	ME	SRT
1-1	71	70	112	30	110g	140
1-2	50	150	98	130	162	200
1-3	51	100	23	200	0	120
2-1	68	100	92	30	80	110
2-2	176	60	225	60	273	130
2-3	22	120	24	190	2	160
3-1	48	80	73	120	82	80
3-2	277	40	290	80	349	60
3-3	56	130	26	130	0	120
4-1	52	100	17	100	42	110
4-2	218	90	281	170	276	90
4-3	25	120	12	20	0	90
5-1	56	110	63	100	44	130
5-2	123	130	178	60	251	150
5-3	7	140	9	170	19	180

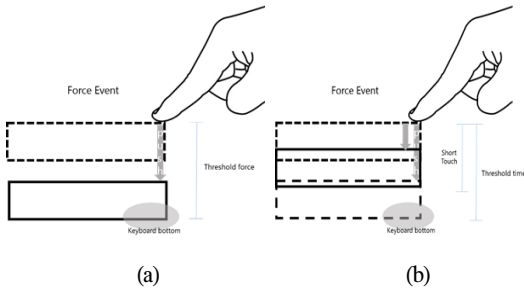
과 다르다는 것을 알 수 있다. 이러한 결과는 표 4를 통해 알 수 있다.

표 4는 버튼별 압력의 평균값을 통해 추의 무게와 비교한 값(ME: Mean margin error)과 추의 값이 일정한 값으로 측정되는 때 까지 걸리는 시간(SRT: Stabilized Receiving Time)을 보여준다. 이 결과에서 압력 값의 변화가 심하게 나타날 수 있는데 이는 스마트폰에서 힘이 가해졌을 때 물체가 스마트 커버에서 미끄러지는 현상과 스마트폰 뒷 커버 자체의 탄성 때문에 발생한다. 즉 각 센서에서 측정되는 값이 변할 수 있다는 것이다. 측정결과 압력이 작은 값을 측정 할수록 측정하기 힘들고, 압력이 클수록 압력 값을 측정하기 쉽다. 결론적으로 위치별 압력 값의 오차를 일정한 압력으로 측정하기 위해서는 압력의 강함이 200g이상이고 300ms 이후의 시간일 때 측정이 가능하다. 실험 결과를 기반으로 본 커버의 입력 값을 미세하게 조정할 수는 없다. 다만, 측정된 값의 비율을 통해서 대략적인 값으로 값을 유추할 수 있다. 이는 각 버튼별로 따로 설정하여 압력 값을 측정한다. 하지만 여전히 사용자에게 단순히 물체를 스마트폰에 올려 측정했을 때의 값으로 실제 스마트폰의 어플리케이션과 사용자 간 인터랙션과의 관계를 설정할 수 없다. 즉, 포스터치 커버가 허용하는 압력값과 시간을 기준으로 사용자가 포스터치 입력을 위한 피드백(feedback)에 허용되는 시간 압력 값을 사용할 수 있는 지를 실험해야 한다. 따라서 포스터치에서의 사

용자 Latency(피드백을 위한 허용시간)와 압력 측정(포스에서의 압력 값)을 통해서 포스터치 어플리케이션에 적용 가능해야만 한다.

4.2 사용자 포스터치 측정 방안

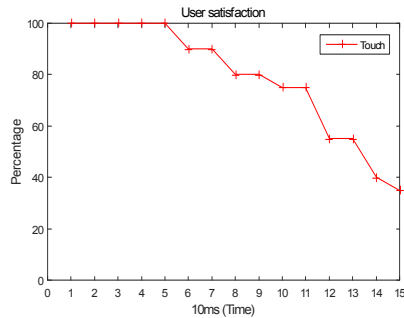
사용자가 포스터치 입력하는 과정은 그림 5에서와 같이 스크린에 손가락을 터치하는 과정, 터치 후에 스크린에 일정한 압력을 가해서 힘을 주는 과정으로 나뉠 수 있다. 그림 5(a)와 같이 사용자가 스크린에 터치를 한 순간부터 특정한 힘의 압력이 발생하는 시점을 포스가 발생한다고 할 수 있다. 따라서 사용자가 포스터치라고 판단하는 힘을 가할 때 포스가 발생하는 최소값을 통해서 일반적인 터치와 구분될 수 있는 $F_{threshold}$ 값을 구할 수 있다. 또한, 그림 5(b)와 같이 사용자는 포스터치 시 일정시간 ($T_{threshold}$) 동안 힘을 가할 때 사용자는 누른 힘에 대한 피드백을 요구한다. 즉, 사용자가 힘을 가하는 시간이 얼마나 걸리는지를 찾아내야 한다. 따라서 본 논문에서는 사용자가 포스터치를 가할 때 요구되는 피드백에 대한 시간과 압력 값을 구한다.



(그림 5) 포스터치시 고려해야할 사항 : 시간, 힘

(Figure 5) Considering Force-Touch event :

(a) force (b) time



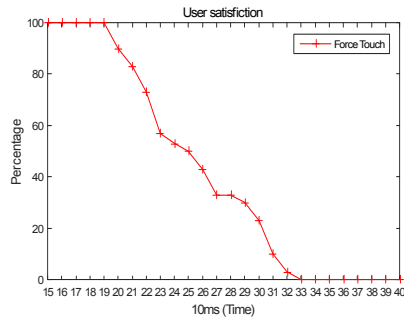
(그림 6) 터치 반응 시간 측정

(Figure 6) Measurement of touch interaction time

4.3 사용자 포스터치 반응시간 측정

본 논문에서는 사용자가 터치와 포스터치 입력 후에 터치와 포스터치의 시간적 구분을 알아야만 한다. 즉, 터치 후 터치에 대한 피드백이 느리다고 생각하는 최소 시간 이후의 시간은 포스터치에 대한 피드백 시간이 가능한 시간이다. 또한, 포스터치 시에 대한 응답이 느린 최소 시간을 찾아 포스터치 피드백에 대한 최소 시간과 최대 시간을 찾아야 한다. 이를 해결하기 위해서 본 논문에서는 버튼을 터치와 포스터치로 눌렀을 때 각각 사용자가 피드백에 대한 반응시간이 적절하게 판단할 수 있는 앱(App)을 구현하였다. 먼저 터치에 대한 반응시간이 적절하지 않는 시간의 최소 시간을 찾는다. 이는 터치와 포스터치를 구분하기 위한 기준점(시간)이다.

앱(App)에서 버튼을 생성한 후에 사용자가 터치를 가 정하여 버튼을 누를 때 버튼의 색을 변경한다. 버튼은 1 단계에서 50단계로 나눈다. 1단계에서는 10ms의 반응시간을 주고, 2단계에서는 20ms로 각 단계가 증가할 때 마다 10ms씩 반응 시간을 늦춘다. 이때 사용자는 터치시 반응시간이 느려지는 단계(시간)를 체크한다. 이러한 실험은 20명을 대상으로 수행하고 각 실험자마다 10번을 반복한다. 그림 6은 사용자가 터치 반응 시간을 측정하였을 경우 터치 반응이 느리지 않다고 판단하는 시간을 보여 준다. 그림에서 터치 시 반응이 느려지는 시점은 실험대상자의 20명의 중에 90%는 50ms 안에 터치반응에 대한 피드백이 100% 느리지 않다고 판단한다. 100ms안에는 약 75%가 느리지 않다고 판단하지만, 이후에는 단 50%만의 사용자가 느리지 않다고 판단한다. 이를 기준으로 했을 경우 터치에 대한 반응은 최소 100ms 안에 이루어 져야 하고, 100ms 이후에 포스터치에 대한 피드백이 이루어 져야 한다.

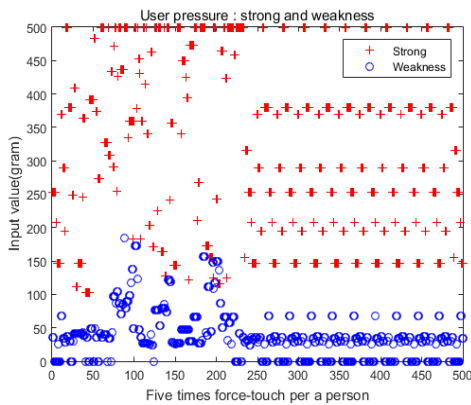


(그림 7) 포스터치 반응 시간 측정

(Figure 7) Measurement of force-touch interaction time

포스터치에 대한 반응 시간에 대한 latency를 구하기 위해서 터치와 같은 방법으로 실험을 진행하였다. 그림 7은 포스터치에 따른 반응시간에 대한 사용자들의 반응시간을 보여준다. 그림 7에서 사용자들은 200ms 이전에 포스터치가 이루어졌을 경우까지는 포스터치에 대한 피드백이 느리지 않다고 판단한다. 210ms에서 90%의 사용자가 피드백이 느리지 않다고 판단하였다면, 220ms까지는 70% 만이 250ms에는 45%만이 느리지 않다고 판단하였다. 따라서 포스터치에 대한 최대 반응시간은 적어도 210ms 정도까지로 조정이 가능하다. 따라서 본 논문에서 제안한 포스터치에 대한 반응 시간은 표 4를 통해 포스터치 값이 안정적으로 들어오는 시간인 200ms를 충족할 수 있음을 보인다.

이는 먼저 사용자의 지각에 대해서 Michotte와 Thines [13]의 연구결과에 따르면 첫 행동이 끝나는 순간과 두 번째 행동이 시작하는 순간 사이에 약 50ms정도의 지연 시간을 사용자가 지각한다고 한다. Scholl과 Nakayama



(그림 8) 포스터치 힘의 값 측정

(Figure 8) Measurement of Force-Touch user pressure

[14]의 실험에서는 사용자는 200m까지의 지연시간을 신뢰할 수 있다는 경향성을 파악하였다. 즉, 200ms 이상의 지연시간이 증가하면 신뢰성이 떨어지는 경향성을 보인 것이다. 시청각 인과 실험에서도 동일한 결과를 유추하였다[15]. 본 연구에서 사용자의 포스터치 행동과 피드백에 대한 결과는 사용자가 두 물체에 대한 충돌(스마트 디바이스와 손가락과 같이)에 대해 200ms까지 지각 할 수 있다는 결과와 일치함을 알 수 있다.

4.4 사용자 포스터치 힘의 값 측정

사용자가 포스터치 커버를 이용하여 포스터치를 단계별(weakness, strong)로 구분해 낼 수 있는지를 측정하기 위해 총 20명의 사용자에게 강함과 약함을 5번씩 실시하게 하였다. 힘을 두 가지로 구분하는 이유는 현재 포스터치가 상용화된 애플의 iPhone6s 이후의 버전에서 인터랙션을 위해 peak과 pop으로 힘을 두 단계로 구분하기 때문이다. 그림 8의 동그라미는 약함을 눌렀을 경우에 생성되는 힘(압력)을 보여주고, 십자가 모양은 강함을 눌렀을 경우의 힘을 보여준다. 약한 힘을 스크린에 가했을 경우에는 최소 0g에서 최대 185g까지 분포되어 있다. 반면에 강한 힘을 가했을 경우에는 약 100g에서 500g으로 분포가 넓어진다. 즉, 강한 힘을 가했을 때 힘의 값은 편차가 커짐을 알 수 있다. 반면에 약한 힘을 가했을 때의 힘의 차이는 적게 나타난다. 이 두 포스터치의 힘을 구분하기 위해서는 포스터치의 약한 힘을 보다 큰 힘을 강한 힘으로 설정할 수 있어야 한다.

그림에서 x축 100번째에서 200번째와 같이 사용자가

힘의 크기를 제대로 제어하지 못 하는 경우에는 약한 힘과 강한 힘을 제대로 구분하기 힘들다. 하지만, x축 250번째 이후에 사용자의 약한 힘 값은 편차가 약 70g이고, 강한 힘은 약 145g이상이므로 힘을 컨트롤 할 수 있다. 특히, 다양한 사용자가 같은 실험환경에서 거의 같은 힘의 크기 가할 수 있음을 보인다. 또한, 강한 힘과 약한 힘의 경계는 약 70g~140g으로 확실하게 구분될 수 있어 힘의 강약을 표현할 수 있는 기준점을 제시할 수 있다. 이를 통해 포스터치 커버를 이용하여 포스터치 힘을 측정할 수 있고, 단계도 기존의 방식과 같은 두 단계를 구현해 낼 수 있다.

5. 결 론

본 논문은 포스터치를 사용하지 못 하는 기존의 일반적인 터치 기반의 스마트 디바이스에 포스터치 인터랙션을 제공할 수 있는 방안에 대한 연구이다. 기존의 포스터치가 제공되지 못 하는 디바이스를 위해 포스터치를 측정할 수 있는 커버를 제작하였다. 제작된 포스터치 커버를 이용하여 힘을 측정할 수 있는 센서의 정확성과 힘을 가하는 위치에 따른 센서의 힘의 변화를 분석하였다. 이를 통해 힘을 측정할 수 있는 최소한의 값을 판단하였다. 또한, 포스터치 커버를 이용하여 사용자가 포스터치를 발생할 경우 요구되는 최대한의 시간을 찾아내어 이를 적용할 수 있는 사용자 중심적인 포스터치 인터랙션이 방안을 제시하였다. 이러한 결과는 사용자 실험을 통해 증명하였다. 이를 통해 사용자의 포스터치 인터랙션에 영향을 미치는 사용자의 입력 반응시간과 힘의 강도를 제시할 수 있는 결과를 도출하였다.

향후 연구 과제로 포스터치 입력에서 사용자가 포스터치를 몇 단계까지 구분할 수 있는가에 대한 연구가 진행할 예정이다. 이를 통해 하나의 입력 방법에서 다양한 명령을 수행할 수 있는 인터랙션방안을 도출 할 것이다.

Acknowledgement

이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2016 R1A6A3A11932892)

참고문헌(Reference)

- [1] Lee, D.W., Lim, J.M., Sunwoo, J., Cho, I.Y.; Lee, C.H. "Actual remote control: A universal remote control using hand motions on a virtual menu.", *IEEE Trans. Consum. Electron.* vol. 55, no. 3, pp.1439-1446, 2009. <http://dx.doi.org/10.1109/TCE.2009.5278011>
- [2] Kim, D.K., Kim, J.H., Kwon, H.J., Kwon, Y.H., "A Touchpad for Force and Location Sensing.", *ETRI Journal*, vol. 32, no. 5, pp.722-728., 2010. <http://dx.doi.org/10.4218/etrij.10.1510.0073>
- [3] Ramos, G., Balakrishnan, R. Zliding. "Fluid Zooming and Sliding for High Precision Parameter Manipulation.", *Proceeding of ACM symposium on User interface software and technology*, pp.143-152, 2005, <http://dx.doi.org/10.1145/1095034.1095059>
- [4] Ramos, G., Boulos, M., Balakrishnan, R., "Pressure widgets.", *Proceeding of CHI '04*, ACM, pp. 487-494, 2004. <http://dx.doi.org/10.1145/985692.985754>
- [5] Miyaki, T.; Rekimoto, J. GraspZoom, "Zooming and Scrolling Control Model for Single-Handed Mobile Interaction." *Proceeding of MobileHCI '09*, ACM , pp. 2009. <http://dx.doi.org/10.1145/1613858.1613872>
- [6] Apple's 3D Touch. <https://developer.apple.com/ios/human-interface-guide/lines/interaction/3d-touch/>
- [7] Smart Pressure-Sensitive Screen. <http://consumer.huawei.com/en/mobile-phones/mateStouch.htm>
- [8] Seongkook Heo, Geehyuk Lee, "Force gestures: augmenting touch screen gestures with normal and tangential forces." *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 621-626, 2011. <http://dx.doi.org/10.1145/2047196.2047278>
- [9] Dumas, C., Saugis, G., Degrande, S., Plénacoste, P., Chaillou, C., & Viaud, M. L. "A 3d interface for cooperative work." *Virtual Reality*, vol.4, no. 1, pp.15-25, 1999. <http://dx.doi.org/10.1007/BF01434991>
- [10] Nakai, Y., Kudo, S., Okazaki, R., Kajimoto, H., Kuribayashi, H. "Detection of tangential force for a touch panel using shear deformation of the gel." *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. pp. 2353-2358, 2014. <http://dx.doi.org/10.1145/2559206.2581149>
- [11] How Smartphones are changing consumers' daily routines around the globe. <http://www.nielsen.com/us/en/insights/news/2014/how-smartphones-are-changing-consumers-daily-routines-a-round-the-globe.html>
- [12] Heo, S.K.; Lee, G.H. "A Method of Augmenting Touch Input on Mobile Devices with Finger-Controlled Touch Screen Using a Built-in Accelerometer." *Proceeding of HCI 2011, HCI Korea*, pp.261-263, 2011.
- [13] Thines, G.; Costall A.; Butterworth, G. Perceived causality. "Michotte's experimental phenomenology of perception", Hillsdale, NJ, Erlbaum, pp.66-87, 1991. ISBN-10: 0898596068
- [14] Scholl, B. J.; Nakayama, K. "Casual capture: contextual effects on the perception of collision events." *Psychol Science*. vol. 13, no. 6, 2002.
- [15] Lewald, J., Ehrenstein, W. H., Guski, R. "Spatio-temporal constraints for auditory-visual integration." *Behavioural Brain Research*, vol. 212, no. 1-2, pp.69-79, 2001. [http://dx.doi.org/10.1016/S0166-4328\(00\)00386-7](http://dx.doi.org/10.1016/S0166-4328(00)00386-7)

● 저 자 소 개 ●



남 춘 성(ChoonSung Nam)

2005년 상명대학교 소프트웨어학과(이학사)
2007년 숭실대학교 대학원 컴퓨터학과(공학석사)
2011년 성균관대학교 대학원 전자전기컴퓨터학과(공학박사)
2014년 연세대학교 IT정책전략연구소 박사후연구원
2016년 성균관대학교 컨버전스연구소 선임연구원
관심분야 : VANET, IoT, UAV & Force Touch Interaction, etc
E-mail : namgun99@gmail.com



서 민 수(Min-Soo Suh)

2016년 연세대학교 컴퓨터과학(공학사)
2016년~현재 연세대학교 정보대학원 UX트랙 석박사통합과정
관심분야 : Force touch, VR etc
E-mail : zeclix1013@gmail.com



신 동 렬(Dong-Ryeol Shin)

1980년 성균관대학교 전자공학과(공학사)
1982년 KAIST 대학원 전기 및 전자공학과(공학석사)
1992년 Georgia Tech, 대학교 전기 및 전자공학과(공학박사)
1994년~현재 성균관대학교 정보통신공학과 교수 .
관심분야 : 유비쿼터스 컴퓨팅, 센서 네트워크 etc.
E-mail : drshin@skku.edu