

모바일 장치를 위한 동작 추적형 이미지 브라우저 시스템

2007 임성훈



석사학위논문

모바일 장치를 위한 동작 추적형
이미지 브라우징 시스템

임성훈(任成勳)

정보통신학과(가상현실전공)

포항공과대학교 대학원

2007



모바일 장치를 위한 동작 추적형
이미지 브라우징 시스템

**Image Browsing in Mobile Devices
Using User Motion Tracking**



Image Browsing in Mobile Devices Using User Motion Tracking

by

Sunghoon Yim

Department of Computer and Communications
Engineering

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

A thesis submitted to the faculty of Pohang University of
Science and Technology in partial fulfillment of the require-
ments for the degree of Master of Science in the Department
of Computer and Communications Engineering

Pohang, Korea

2006. 12 . 27 .

Approved by

Major Advisor: Seungmoon Choi



모바일 장치를 위한 동작 추적형 이미지 브라우징 시스템

임성훈

위 논문은 포항공대 대학원 석사 학위 논문으로 학위 논문 심사
위원회를 통과하였음을 인정합니다.

2006년 12 월 27 일

학위논문 심사위원회 위원장	최승문	(인)
위 원	김정현	(인)
위 원	한성호	(인)



MIT 임성훈, Sunghoon Yim, 모바일 장치를 위한 동작 추적형 이미지
20052620 브라우징 시스템, Image Browsing in Mobile Devices Using User
Motion Tracking, Department of Computer and Communica-
tions Engineering, 2007, 85 p, Advisor: 최승문 (Seungmoon
Choi). Text in Korean

Abstract

One of the recent trends in the use of digital technology is the proliferation of sensing, communication and storage devices (mostly due to low cost), either separately (e.g. USB memories, digital cameras) or in integrated form such as a cell phone. One side effect is that we get ourselves flooded with information, manifested in the very high number of “files” we have to manage. For instance, it is often the case that hundreds (even thousands) of multimedia data are stored in one’s PC or mobile devices in an unstructured manner. This in turn makes the browsing and manipulation of such high amounts of data difficult and time consuming. Moreover, the basic hand-held device interfaces are 2D and discrete input based with a very limited display size, and thus is not fit for such a demanding task.

A possible solution is to adopt a 3D and continuous input based interface such as a motion based interaction. Previous researches have shown that a motion based interaction on hand-held devices produces high task performance (e.g. object selection and navigation), and is intuitive and natural to use as well. To further enhance the effectiveness of the motion based interaction, we also explore the use of particular data layouts, tiled cylindrical and fish-eye cylindrical are implemented and tested for their usability and effectiveness in combination with different styles of data layouts, vs. the traditional button based interface



The motion based interface on the hand-held device is implemented using the camera and a 3 axis accelerometer and can track motions in 4 degrees of freedom (e.g. forward-backward translation and rotations in roll, pitch, and yaw). Three types of motion based interfaces (position tracking, tilt tracking and hybrid tracking) are implemented.

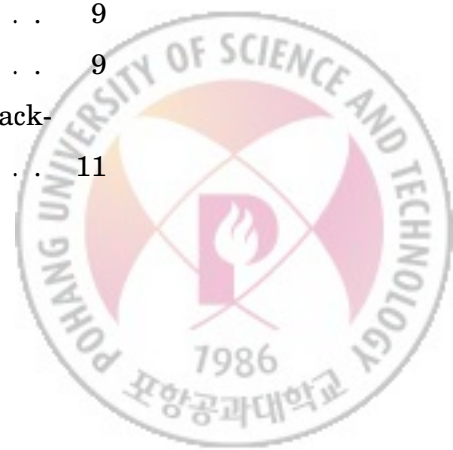
Contrary to our expectation, the motion based interface did not exhibit any particular advantages over the button based one in our evaluation study. This may be due to the nature of the task, image browsing, that requires a minimum amount of scanning time for effective browsing. Motion based interface, on the other hand, seems to be fit for fast scanning of data. In addition, the motion based interface may be more appropriate for 3D task as well. The study suggests that a careful interface design is required for hand held devices for different tasks and applications.



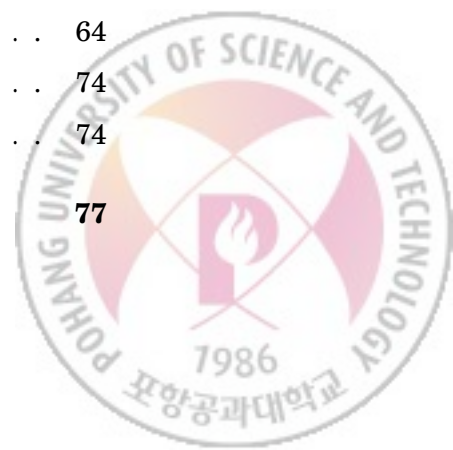


목 차

제 1 장 서론	1
제 1 절 연구 배경	1
제 2 절 연구 목적	2
제 3 절 논문의 구성	3
제 2 장 관련 연구	4
제 1 절 모바일 장치의 효과적인 데이터 전달을 위한 레이아웃 및 인터페이스에 대한 연구	4
제 2 절 모션 인식을 이용한 모바일 인터페이스에 대한 연구	5
제 3 장 모바일 이미지 브라우저의 구현	8
제 1 절 이미지 브라우저의 요건	8
제 2 절 모션을 이용한 인터페이스	9
2.1 모바일 장치에서의 모션 인식	9
2.2 모바일 장치의 위치 추적을 이용한 인터페이스(Position Tracking Motion Interface)	11



2.3	모바일 장치의 기울기 추적을 이용한 인터페이스(Tilt Motion Interface)	14
2.4	위치, 기울기 혼성 추적 인터페이스(Hybrid Motion Interface)	16
제 3 절	모바일 기기에서 다수의 이미지 배치	17
3.1	Tiled Cylindrical Layout	17
3.2	Fisheye Cylindrical Layout	18
제 4 절	이미지 브라우저의 구현	19
4.1	인터랙션 설계	19
4.2	구현	24
제 4 장	인터페이스와 레이아웃 간의 사용성 실험	26
제 1 절	실험 목적	26
제 2 절	실험 설계	26
2.1	피실험자	26
2.2	독립 변수	27
2.3	종속 변수	28
2.4	구현	28
2.5	실험 절차	30
제 3 절	실험 결과 및 토론	34
3.1	위치를 알려준 사진에 대한 브라우징 태스크 수행 결과	34
3.2	위치를 알려주지 않는 사진에 대한 브라우징 태스크 수행 결과	39
3.3	인터페이스에 대한 설문 결과	43
3.4	레이아웃에 대한 설문 결과	56
3.5	인터페이스와 레이아웃의 조합에 대한 설문 결과	64
3.6	수행 시간에 대한 선형 회귀 분석	74
3.7	토론	74
제 5 장	결론 및 향후 계획	77



목 차

iii

부록 A 실험에 사용한 설문지

79

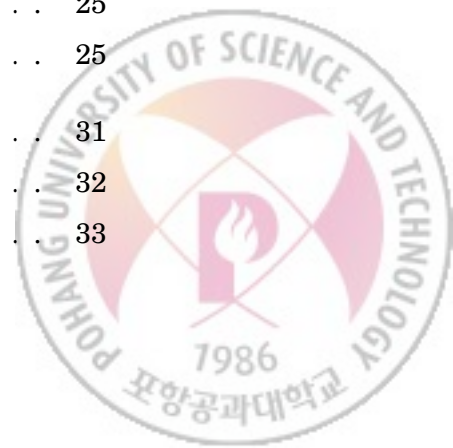
참고 문헌

85



그림 목차

3.1 Hwang[1]의 3 DOF Linear Accelerometer와 카메라 이미지를 병용 한 4DOF 위치 추적	11
3.2 사용자의 4 DOF 움직임	12
3.3 유저의 추적된 움직임에 기반한 3D Navigation	13
3.4 인터페이스의 움직임과 원기둥 배치	17
3.5 Tiled Cylindrical Layout	18
3.6 Fisheye Layout의 이미지 크기 결정	19
3.7 Fisheye Cylindrical Layout	19
3.8 인터랙션의 State Diagram	23
3.9 이미지의 선택	24
3.10 이미지의 이동	24
3.11 한 이미지만 보기	25
3.12 이미지의 복사	25
4.1 트레이닝 세션	31
4.2 위치를 알려준 사진의 브라우징 수행 작업	32
4.3 위치를 알려주지 않은 사진의 브라우징 수행 작업	33



4.4 위치를 알려준 사진에 대한 브라우징 수행시 실행 시간	35
4.5 위치를 알려준 사진에 대한 브라우징 태스크 수행시 이동 궤적의 총 길이	37
4.6 위치를 알려주지 않은 사진에 대한 브라우징 수행시 실행 시간	39
4.7 위치를 알려주지 않은 사진에 대한 브라우징 태스크 수행시 이동 궤적의 총 길이	41
4.8 인터페이스를 사용하기 쉬운 정도	43
4.9 인터페이스를 배우기 쉬운 정도	45
4.10 인터페이스를 선호하는 정도	47
4.11 인터페이스의 직관적인 정도	49
4.12 인터페이스의 자연스러운 정도	51
4.13 인터페이스의 재미있는 정도	54
4.14 레이아웃을 선호하는 정도	56
4.15 레이아웃의 직관적인 정도	58
4.16 레이아웃의 자연스러운 정도 그래프	60
4.17 레이아웃의 재미있는 정도	62
4.18 인터페이스와 레이아웃의 조합이 사용하기 쉬운 정도	64
4.19 인터페이스와 레이아웃의 조합을 선호하는 정도 그래프	66
4.20 인터페이스와 레이아웃의 조합이 직관적인 정도 그래프	68
4.21 인터페이스와 레이아웃의 조합이 자연스러운 정도 그래프	70
4.22 인터페이스와 레이아웃의 조합이 재미있는 정도 그래프	72
A.1 시작 설문	80
A.2 매 조건 실험 후 설문1	81
A.3 매 조건 실험 후 설문2	82
A.4 매 조건 실험 후 설문3	83
A.5 전체 실험 후 설문	84

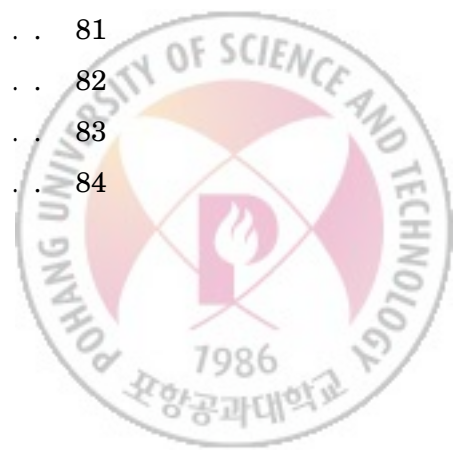
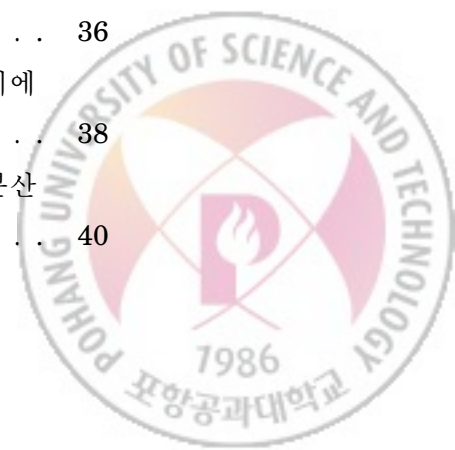


표 목차

3.1	자체 모션 인식이 가능한 센서들	10
3.2	모바일 장치의 위치 추적을 이용한 인터페이스의 사용자의 움직임과 Browser Operation과의 매칭	14
3.3	모바일 장치의 기울기 추적을 이용한 인터페이스의 사용자의 움직임 과 Browser Operation과의 매칭	15
3.4	위치, 기울기 혼성 추적 인터페이스의 사용자의 움직임과 Browser Operation과의 매칭	16
3.5	Motion Interface에서 사용하는 버튼	21
4.1	독립 변수	27
4.2	6 개의 인터페이스와 레이아웃의 조합	27
4.3	종속 변수	28
4.4	위치를 알려준 사진에 대한 브라우징 수행 시간에 대한 분산 분석 . .	36
4.5	위치를 알려준 사진에 대한 브라우징 수행시 이동 궤적의 총 길이에 대한 분산 분석	38
4.6	위치를 알려주지 않은 사진에 대한 브라우징 수행 시간에 대한 분산 분석	40



4.7 위치를 알려주지 않은 사진에 대한 브라우징 태스크 수행시 이동 궤 적의 총 길이의 분산 분석	42
4.8 인터페이스를 사용하기 쉬운 정도에 대한 분산 분석	44
4.9 인터페이스를 배우기 쉬운 정도에 대한 분산 분석	46
4.10 인터페이스를 선호하는 정도에 대한 분산 분석	48
4.11 인터페이스의 직관적인 정도에 대한 분산 분석	50
4.12 인터페이스의 자연스러운 정도에 대한 분산 분석	53
4.13 인터페이스의 재미있는 정도에 대한 분산 분석	55
4.14 레이아웃을 선호하는 정도에 대한 분산 분석	57
4.15 레이아웃의 직관적인 정도에 대한 분산 분석	59
4.16 레이아웃의 자연스러운 정도에 대한 분산 분석	61
4.17 레이아웃의 재미있는 정도에 대한 분산 분석	63
4.18 인터페이스와 레이아웃의 조합이 사용하기 쉬운 정도에 대한 분산 분석	65
4.19 인터페이스와 레이아웃의 조합을 선호하는 정도에 대한 분산 분석 .	67
4.20 인터페이스와 레이아웃의 조합이 직관적인 정도에 대한 분산 분석 .	69
4.21 인터페이스와 레이아웃의 조합이 자연스러운 정도에 대한 분산 분석	71
4.22 레이아웃의 자연스러운 정도에 대한 분산 분석	73
4.23 통계적으로 유의하게($p < 0.05$) 선형 경향성을 가진 실험자의 수 . . .	74



제 1 장

서론

제 1 절 연구 배경

현재 수많은 사람들에게 모바일 기기는 삶의 필수적인 부분이 되었다. 많은 유럽 국가들에서는 100%에 가까운 진입률을 보여 주고 있고, 모바일 기기의 수는 퍼스널 컴퓨터의 수를 추월하고 있다. 1993년에는 전세계적으로 가입자의 수가 10 만명 보다 적었지만, 2003 년 중반에는 중국 만으로 그 10 배 수의 가입자 수를 기록하였다[2]. 이렇게 널리 보급된 모바일 폰은 단순한 통신기기를 넘어서 언제 어느 때나 사진을 찍고 멀티미디어를 재생할 수 있는 디지털 미디어 재생의 핵심이 되어 있다.

또한 디지털 카메라 기술이 발전하면서, 사진을 취득하는 방식 역시 변화하고 있다. 사진을 저장하는 방식이 디지털화 되면서 사진을 찍는데 필요한 비용은 거의 없어졌다. 따라서 매우 짧은 시간 동안에 대량의 디지털 이미지를 취득할 수 있다. 더 나아가, 디지털 카메라가 모바일 폰과 결합하면서, 언제 어느때나 이미지를 얻는 것이 일상이 되어 가고 있다. 이러한 변화는 모바일 기기 상에서 대량의 이미지를 브라우징 해야할 필요성이 증대됨을 의미한다.

하지만 모바일 기기의 한계는 여전히 존재하고 있고, 이 때문에 대량의 이미지를 브라우징하는 것은 쉽지 않은 일이다. 물론 모바일 기기는 소프트웨어와 하드웨어



양면으로 발전하고 있지만, 디스플레이의 크기 만큼은 작은 크기로 제한 될 수 밖에 없다[3]. 따라서 제한된 크기의 디스플레이를 가진 장치 안에서 이미지를 효과적으로 브라우징 해야할 필요성이 있다. 또한 작은 크기의 디스플레이는 일반적으로 모바일 기기에서 사용할 수 있는 터치 패드의 작업 영역이 제한 된다는 것 또한 의미한다. 이를 위해 이미지 또는 문서를 브라우징 하기 위한 새로운 인터페이스 종류의 인터페이스나 인터랙션을 개발하는 움직임이 크게 일고 있다. [4][5][6] 또한 작은 화면에서도, 큰 이미지나 문서를 브라우징 할 수 있는 새로운 레이아웃이나 가시화 방법을 적용하고자 하는 움직임 또한 존재한다[7][8][9]. 새로운 인터페이스와 가시화 방법은 서로 충돌하지 않는 한 조합에 따라 좀더 효과적인 결과를 얻을 수 있다.

한편 모바일 기기에 다양한 센싱 기술을 통합함으로써, 인터랙션을 단순화 하고 좀더 다양한 인터랙션을 가능하게 할려고 하는 움직임이 있다[10][11][12][13][14][?] Mantyjarvi06 특히, 센싱 기술 중에 효과적으로 사용될 수 있는 기술로 기울기 센서 등을 이용한 모션인식이 있다. 이 모션 인식을 이용한 방법은 workspace의 제한이나, 화면 크기로 인한 FOV의 제한을 효과적으로 극복하는데 도움이 되었다[6] [1]. 또한 그 자체로 직관적이고 자연스러운 인터페이스라는 장점이 있다.

적절한 인터페이스와 레이아웃의 조합은 모바일 기기에서 많은 양의 이미지를 다루는데 도움이 되리라고 생각된다.

제 2 절 연구 목적

직관적이고 자연스러우면서 연속적인 입력을 받을 수 있는 모션 인터페이스는 모바일 기기에서 사용할 수 있는 매우 효과적인 인터페이스가 될 가능성이 있다. 모션 인터페이스 경우 현실에 가까운 메타포[11]의 인터랙션이 가능하며 또한 direct manipulation이 가능하여, proprioception의 도움[40] 또한 얻을 수 있을 것으로 생각하였다. 특히 모바일 장치와 모션 인터페이스의 결합이 3 차원 인터페이스로서 효과적임이 증명된 바도 있다[1]. 이 모션 인터페이스가 모바일 기기에서 다량의 브라우징 하기 위한 목적으로 적합한지, 만약 가능성이 있다면 어떠한 형태의 모션 인터페이스가 적합한지를 검증해 보고자 한다.

또한 모바일 기기의 한계를 극복하기 위해 다양한 레이아웃이 제안되었고, 그들 레이아웃이 일정 정도의 효과를 거둔 바 있다[7]. 이들 레이아웃이 레이아웃들은 모



바일 기기에서 대량의 이미지를 브라우징하기 위한 브라우저에도 적용이 가능한지 시험해 본다.

한편 이들 레이아웃과 인터페이스를 조합하였을 때, 어떤 형태의 장.단점이 있는지, 어떠한 레이아웃과 인터페이스의 조합이 가장 효과적일 수 있는지 검증해 본다.

제 3 절 논문의 구성

2 장에서는 현재까지의 모바일 기기에서 작은 화면의 한계를 해결하고자 하는 시도들과, 모션에 기반한 인터페이스들에 대한 연구를 소개하였다. 3 장에서는 이미지 브라우저의 요건과, 그에 적합한 레이아웃, 그리고 모션을 이용한 인터페이스에 대하여 설명하고, 이를 이용하여 구현한 이미지 브라우저에 대하여 설명하였다. 4 장에서는 다양한 레이아웃과 기존의 버튼 인터페이스, 그리고 이번에 제안한 모션 인터페이스 사이의 조합들의 사용성을 실험을 통해 검증하였다. 5 장에서는 이번 연구를 통해 얻어진 결론과, 그에 따른 향후 연구할 사항에 대하여 이야기 하였다.



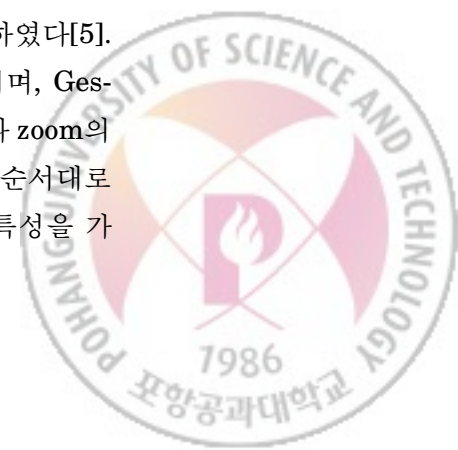
제 2 장

관련 연구

제 1 절 모바일 장치의 효과적인 데이터 전달을 위한 레이아웃 및 인터페이스에 대한 연구

Gutwin[7]은 일반적인 데스크탑을 위해 만들어진 큰 인터페이스를 작은 화면을 가진 모바일 장치에서 다루기 위한 효과적인 방법을 찾기 위해, **panning system**, **2-level zoom system**, **fish-eye view**를 서로 비교하였다. 모든 시스템이 일반적인 큰 화면을 사용하는 시스템보다는 좋지 않는 결과를 보였다. 결과적으로 **fish-eye view**와 **two-level zoom system**이 네비게이션을 위한 인터페이스로 효과적인 것으로 나타났다. **fish-eye view**가 네비게이션 태스크에서 좀더 빠른 결과를 보였지만, 사용자들은 **2-level zoom system**을 더 선호하였다.

Patel[15]은 작은 화면을 가진 모바일 장치에서의 효과적인 이미지 브라우징을 위해 SDAZ[4]응용한 **Auto Zoom technique**와 **Gesture Zoom**을 제안하였다[5]. **Auto Zoom**은 **scrolling speed**와 **zoom**의 크기를 반 비례 시킨 테크닉이며, **GestureZoom**은 수직 방향의 스크롤과 상관 없이 포인팅 장치의 좌우 움직임과 **zoom**의 크기를 연동 시킨 방식이다. 이 테크닉들에 대한 사용성 평가를 통해, 시간 순서대로 정렬된 사진들의 모임에서 특정 이벤트들의 사진을 검색하거나, 특정한 특성을 가



진 사진들을 검색하는데, 일반적인 **Discrete Zoom technique**에 비해 **Auto Zoom**과 **Gesture Zoom**이 효과적임을 보여 주었으며, 이후 같은 **technique**에 대한 좀더 다양한 분석을 하였다[5].

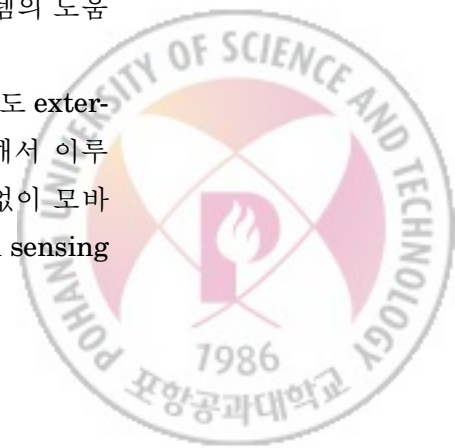
Hakala[8]는 **spatial cognition**을 이용 할 수 있는 3 차원 가시화를 통해, 효과적인 데이터 전달을 꾀했다. 모바일 폰의 파일 구조를 일반적인 리스트가 아니라 3 차원의 정보를 더 제공할 수 있는 방식으로 변경하였다. 이를 통해, 각각의 파일 폴더들은 단순한 구조가 아니라, 공간에서의 위치로도 나타 낼 수 있었다. 그 결과 3 차원 가시화를 이용한 브라우저는 빠르게 배울 수 있고, 모바일 폰의 콘텐츠들을 쉽게 인지할 수 있었다.

제 2 절 모션 인식을 이용한 모바일 인터페이스에 대한 연구

Rekimoto[10]는 **small screen** 장치를 위해 **tilting operation**을 입력으로 사용하는 것을 제안하고, **LCD TV**와 **FASTRAK** 트랙커를 이용하여 그 프로토타입을 구현하였다. 프로토타입 시스템을 통하여 **tilt**를 이용한 메뉴 시스템과 큰 **2D** 스페이스 네비게이터, 그리고 **3D object** 뷰어 시스템을 구현하고 제안하였다. 모바일 시스템 자체에서 모든 부분을 완결 시키지는 못하였지만, 모바일 기기에서 모션 인식을 사용할 수 있는 가능성을 보여 주었다.

Yee[6]은 **Peephole display**를 제안하였다. **Peephole display**는 일반적인 모바일 기기에서 사용하는 **pen interaction**시, 모바일 기기의 화면 크기가 작은 것에서 기인하는 좁은 **workspace**의 문제를 위치 인식을 이용하여 해결하였다. 이를 통해 모바일 기기의 화면보다 큰 가상의 화면에서 다양한 인터랙션을 수행할 수 있었다. 한편 모션 인식을 위해서는 **optical mouse tracking**, **ultrasonic tracking**, **two-tethered tracking** 그리고 **magnetic tracker**를 이용한 **tracking** 등을 이용하였다. 그러나 모두 장치 자체에 모션 센서를 설치하는 형태가 아니라, 외부에 설치된 시스템의 도움을 받아야 하는 것으로 완전하게 휴대 가능한 형태가 되지는 못했다.

한편 모바일 기기가 진정한 모바일 기기로 작용하기 위해서는 모션 인식도 **external reference**를 필요로 하지 않는 **self-contained sensing system**에 의해서 이루어질 필요성이 있다. 모바일 기기의 성능이 발전하여, 외부의 처리 장치 없이 모바일 장치의 프로세싱 파워를 이용할 수 있게 되면서, **self-contained motion sensing**



system을 이용할 수 있게 되었다.

Verplaetse[16]는 inertial sensing이 self motion sensing system에 필수적이라는 결론을 얻고, inertial sensor를 구현하기 위해 필요한 기술들을 서술하였다. 그에 따르면, self motion sensing을 위한 기술로, accelerometer와 gyroscopes를 들었다. accelerometer는 그 자체로 2 축의 절대적인 기울기를 인식하는 tilt sensor로 사용이 가능하기 때문에, 다양한 연구에 적용 되었다. 단 3 자유도 이상의 움직임을 인식할 수 없으므로 3 차원 환경을 위한 인터페이스로 사용하기에는 부족함이 있다.

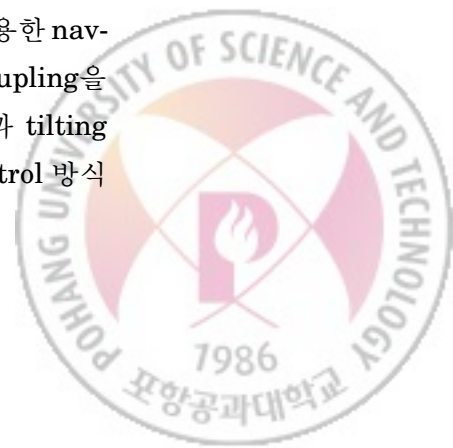
Harrison[11]은 한 문서나 책에서의 네비게이션, 길고 연속된 리스트의 네비게이션, 문서에 주석 표시 등의 인터랙션을 tilting sensor와 pressure sensor를 이용해 실제 세계의 메타포와 비슷한 형태로 구현하였다.

Bartlett[12]은 tilting을 이용한 scrolling technique와 버튼이나 터치패드의 도움 없이 다양한 유저 입력을 얻을 수 있는 tilting gesture를 제시하였다.

Hinckley[13]는 모바일 기기의 인터랙션을 위한 sensing technique을 제안하였다. IR proximity sensor, touch sensor, tilt sensor 등의 다양한 센서를 통해 background-sensing을 한 뒤 이를 이용한 인터랙션을 제안하였다. 논문에서 제안된 인터랙션 방법 중엔 2-axis linear accelerometer를 기반으로 한 tilt sensor를 이용한 portrait/landscape display mode 감지와 결합된 tilt sensor를 이용한 scrolling을 제안하였다.

Crossan[17]은 3 degree of freedom accelerometer와 PDA를 이용하여 한 손으로 모바일 장치를 들고 있는 상태에서 손목 기울임을 이용한 네비게이션에 대한 분석을 하였다. 결과에 따르면 중심에 아래쪽을 향할 때보다 위쪽을 향할 때, variability가 증가하였으며, X축 모션이 Y축 모션 보다 variability가 크다는 결론을 얻었다. 이 결과는 이후 tilting sensor를 이용한 연구에 이용될 수 있다.

Eslambolchilar[18]는 3 degree-of-freedom linear accelerometer를 이용한 tilting sensing과 speed-dependent automatic zooming(SDAZ) [4]방식을 이용한 navigation technique을 결합하고, 그 SDAZ technique에 포함된 dynamic coupling을 dynamic system으로 표현하였다. 한편 그와 병행한 사용자 평가의 결과 tilting navigation이 장점이 존재함에도 불구하고, 사용자들은 이전의 stylus-control 방식을 더 편안하게 여긴다는 결과를 얻었다.



Marsden[19]은 가상환경을 네비게이션하기 위하여 위한 **tilting sensor**를 이용한 모션 인터페이스를 도입하면서, 순수하게 모션을 이용한 것과, 모션과 버튼을 혼용한 두 가지 형태의 인터페이스를 비교하였다. 결과적으로 모션과 버튼을 혼용한 형태의 인터페이스가 만족도, 네비게이션에 걸리는 시간, 배우기 쉬운 정도, 실수를 적게 하는 정도 모두에 있어서 더 효과적이라는 결론을 얻었다.

한편 Hwang[1]은 **3-axis linear accelormeter**와 **vision technique**을 결합한 형태의 모션 인식 방법을 구현하였다. 이 혼합형 모션 인식 방법은 기존의 단순한 2축의 기울임만을 인식하는 것이 아니라, 4 자유도의 움직임을 인식할 수 있다. 이 인터페이스는 3차원의 가상 환경에서의 인터랙션에 적용되었으며, 사용자 평가 결과에 따르면, 모션을 이용한 인터페이스가 **task perfacence**와 **usability** 모두 버튼 인터페이스 보다 좋은 결과를 얻었다.



제 3 장

모바일 이미지 브라우저의 구현

제 1 절 이미지 브라우저의 요건

이미지 브라우저는 여러 개의 이미지로 부터 유저가 하나 또는 그 이상의 이미지를 선택할 수 있도록 하는 프로그램이다. 이 이미지 브라우저는 한번에 여러 장의 이미지를 보여 줄 수 있어야 하며, 원래의 **full resolution image** 또한 보여 줄 수 있어야 한다.[20] 이 조건을 **Zoomable user interface(ZUI)**를 이용하면, 쉽게 달성할 수 있다. **Zoomable user interface**는 **information object**가, **space**와 **scale**로 정리 되어 있으며, 유저는 **interface space**에 있는 객체를 **zooming**또는 **panning**을 통하여 직접적으로 상호작용 할 수 있는 형태의 인터페이스이다[21]. **Panning**은 현재 보이는 **information space**의 **area**를 변경 시키고, **zooming**은 현재 보이는 **information space**의 **scale**을 변경시킨다.

한편 모바일 장치에서 가장 제한 되는 부분은 화면의 크기이다. 일반적인 데스크탑의 화면보다 크기가 작음에 따라, 한 화면에 들어갈 수 있는 이미지의 수나 크기에서 상당히 제한이 있게 된다. 특히 이미지의 크기를 너무 작게 한다면, 사용자가 보고 인지 할 수 없으므로, 어느 정도의 크기를 유지할 수 밖에 없다. 이를 위해 **single-view browsers**로 설계하였다. **Single view browser**는 화면의 전부를 하나



의 view을 위해 모든 screen space를 사용하는 형태의 브라우저 이다. 이 방법은 display space가 부족할 경우에 일반적으로 이용되는 형태이다[22].

Plaisant[22]는 single view browser를 detail only, zoom-and-replace, fish-eye로 구분하였다. Detail only는 zoom이 존재하지 않는 브라우저 시스템으로 화면보다 이미지가 작을 경우나, 언제나 디테일만을 고려해야 하는 경우에 효과적이다. 화면의 크기가 작은 모바일 장치에서는 detail only는 그리 효과적이지 못하다. 한편 zoom-and-replace는 전체 이미지와 세부 사이의 크기 차이가 클 경우에 효과적인 방식이다. Fish-eye는 세부와 전체적인 부분을 single view에서 동시에 보여 줄 수 있는 방법이지만, 보여주는 이미지에 심각한 왜곡이 발생하게 된다. 이 왜곡은 크기나 위치가 중요한 경우엔 큰 문제가 될 수 있다. 모바일 이미지 브라우저가 선택할 수 있는 방식은 zoom-and-replace와 fisheye single view browser가 된다.

이미지 브라우저를 위해 선택되는 모션 인터페이스는 zooming과 panning에 적합할 필요성이 있다. 이를 위해선 최소한 2DOF 이상의 인터페이스가 필요하며, panning과 zooming을 동시에 하기 위해서는 3DOF 이상의 인터페이스를 사용하는 쪽이 적합하다고 본다.

제 2 절 모션을 이용한 인터페이스

2.1 모바일 장치에서의 모션 인식

모바일 장치에서 모션 인식을 위해 사용되는 센서들은 external reference 없이 self-contained system이어야 할 필요가 있다. 이 조건을 충족하여 모바일 기기에서 사용할 수 있는 센서들을 표3.1에 정리하였다.

3 DOF linear accelerometer는 3축의 가속도 값을 센싱하므로, 이를 적분하면, 속도와 위치를 감지 할 수 있다. 하지만, 위치를 알기 위해서는 적분을 두 번해야 하므로, 에러의 크기가 커지고, 심하게 에러가 누적되어 짧은 순간의 gesture를 인식하는 정도의 이상으로 활용하긴 쉽지 않다. 실제로 3 DOF linear accelerometer를 모바일 장치에서 이용한 연구들은 대부분 3 DOF linear accelerometer를 tilt sensor로 이용하는 접근을 하였다. 이 경우엔 2 축 기울기 값을 에러가 누적되지 않는 절대 값으로 얻을 수 있다. 단, 2 축을 자유롭게 선택할 수 있는 것이 아니라, 중력



센서의 종류	Translation	Rotation
3 DOF Accelerometer	Relative X Relative Y Relative Z	
3 DOF Accelerometer (as tilt sensor)		Absolute roll Absolute pitch
Camera(optical flow)	Relative X Relative Y	Relative roll
Camera(Hwang[23]'s method)	Relative Z	Relative yaw Relative pitch
Angular-rate-sensing Gyroscope		Relative yaw Relative pitch Relative roll

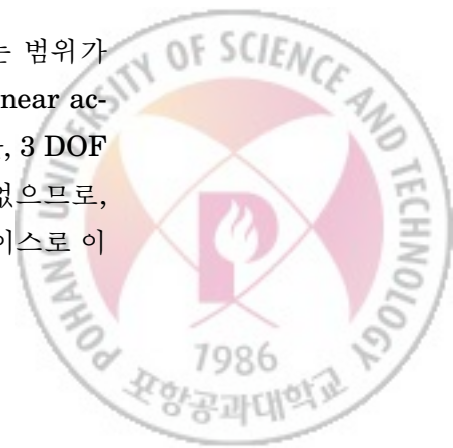
표 3.1 자체 모션 인식이 가능한 센서들

을 기준으로 하기 때문에 얻을 수 있는 기울기 값은 pitch와 roll 뿐이다.

Camera는 영상을 이용하여 정보를 얻어내는 vision 기술을 이용하여, 위치를 센싱한다. Optical flow image processing 방법을 이용하면, 카메라로 부터 얻어진 연속된 이미지로부터 2축의 translation과 1 축의 로테이션 값을 얻을 수 있다. 한편 Hwang[23]이 제안한 형태의 모션 인식 방법은 카메라 이미지로부터 1 축의 translation과 2 축의 로테이션 값을 얻어 낼 수 있다. camera를 이용한 센싱의 문제점은 에러가 누적되고, 강력한 computing power를 필요로 하며, 환경에 영향을 받는다는 점이다.

Angular-rate-sensing gyroscope는 3 DOF linear accelerometer나 camera보다 비싸지만, 3 축의 회전을 전부 얻어낼 수 있는 장점을 갖고 있다. 다만 상대적으로 모션을 인식하므로, 에러가 누적되는 현상을 여전히 갖고 있다.

각각의 센서나 알고리즘들은 서로 장,단점을 갖고 있고, 인식할 수 있는 범위가 조금씩 차이가 있다. 2 장 2절에 소개된 대부분의 연구들에서는 3 DOF linear accelerometer를 tilt sensor로 이용하여 2D interface로 이용하였다. 하지만, 3 DOF linear accelerometer만을 이용할 경우엔 2D 이상의 움직임을 인식할 수 없으므로, 3 DOF linear accelerometer와 카메라를 병용하는 방법[1]을 주된 인터페이스로 이



용하였다. 그 이유는 2D panning과 1D zooming을 통합할 수 있는 인터페이스이기 때문이다.

2.2 모바일 장치의 위치 추적을 이용한 인터페이스(Position Tracking Motion Interface)

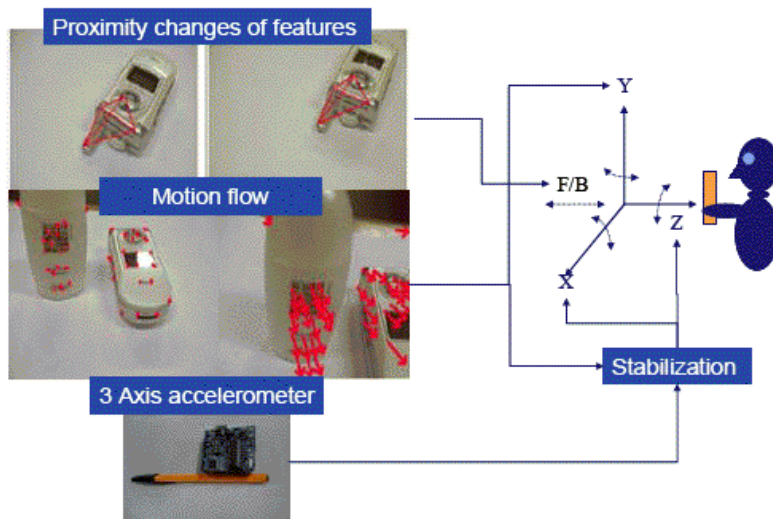


그림 3.1 Hwang[1]의 3 DOF Linear Accelerometer와 카메라 이미지를 병용한 4DOF 위치 추적

Hwang[1]의 위치 추적 방법을 사용할 경우 pitch,roll,yaw와 1D translation의 4 DOF의 데이터를 얻을 수 있다. 이 4 DOF 데이터는 모바일 장치의 3 차원 공간에서의 상대적인 위치를 얻어 내기에 충분하다. 한편, 동시에 몇 가지 가정을 부여한다면, 사용자의 움직임도 어느 정도 유추가 가능하다. 만약 사용자가 장치의 디스플레이를 수직으로 바라보면서 움직인다면, 장치로부터 얻어지는 4 DOF 데이터를 통해, 사용자의 움직임을 그림 3.2와 같이 유추할 수 있다.



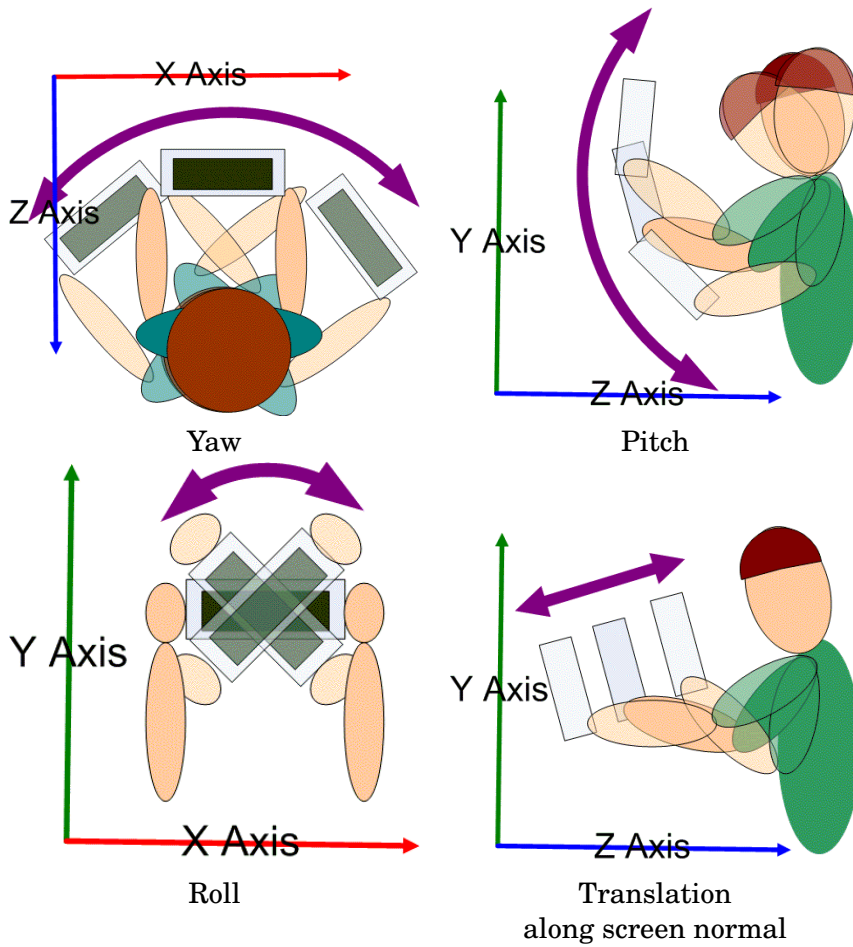


그림 3.2 사용자의 4 DOF 움직임



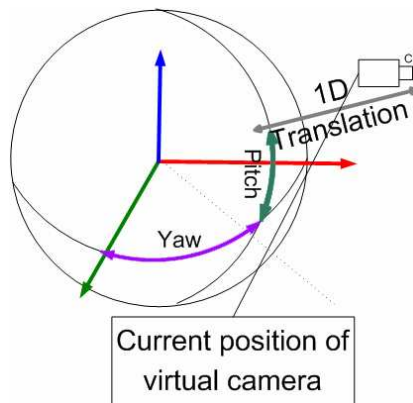


그림 3.3 유저의 추적된 움직임에 기반한 3D Navigation

이렇게 얻어진 사용자의 움직임을 이미지 브라우저의 **manual operation**과 매칭할 수 있다. 일반적인 2D 이미지 브라우저라면, yaw엔 1D horizontal panning을 pitch엔 1D vertical panning을 스크린 방향의 translation은 zoom in out을 대응할 수 있다. 한편 이 사용자의 움직임은 3D 움직임이므로 그 움직임을 3D 환경에서의 움직임으로 그대로 대응시킬 수 있다. yaw의 경우엔 사용자를 축으로 하고, 사용자와 스크린 사이의 거리를 반지름으로 하는, 지면과 평행한 호를 그리는 움직임으로 대응할 수 있다. pitch의 경우에도 yaw와 마찬가지로 사용자를 축으로 하고, 사용자와 스크린 사이의 거리를 반지름으로 하는 지면과 수직인 호를 그리는 움직임으로 대응할 수 있다. 스크린 방향의 움직임은 그대로 screen 방향의 translation이다. screen 방향은 사용자와 수직하다고 가정하였으므로, 사용자와 스크린 사이의 거리를 변화하는 일이 된다. 즉 그림 3.3과 같은 형태의 움직임이 된다. 이들 사용자의 움직임과 browser operation과의 대응을 표3.2에 정리하였다.

이렇게 현재의 추적된 상대 위치와 가상의 상대 위치를 그대로 맵핑하여 3D 브라우저를 한다면, 유저의 움직임 자체가 브라우징 하는 움직임이므로, 자연스럽게 직관적인 인터페이스가 될 수 있다. 하지만 이 경우엔 현실의 workspace는 사용자의 팔 길이에 제한이 된다. 브라우징 해야 하는 이미지의 수에 따라 가상의 workspace의 크기는 무제한에 가깝게 증가할 수 있으므로, 가상의 넓은 workspace와 현실의 제한된 workspace 사이에 불일치가 발생한다. 이 경우 현실의 움직임과 가



	2D Browsing operation	3D Navigation operation
Yaw	1D Horizontal panning	1D Horizontal arc
Pitch	1D Vertical panning	1D Vertical arc
Roll		
Translation along screen normal	Zooming	1D Translation view direction

표 3.2 모바일 장치의 위치 추적을 이용한 인터페이스의 사용자의 움직임과 Browser Operation과의 매칭

상의 움직임 사이의 gain을 증가시키게 되는데, 이 gain이 굉장히 클 경우엔 비효율적이 되는 단점이 있다. 특히, motion interface의 불안정성과 충분하지 못한 해상도 때문에, 실질적으로 가상의 workspace와 현실의 workspace의 사이의 크기 차이가 한계 이상 일 때는 사용할 수 없다. 또한 충분히 interface의 해상도가 높아도, 사용자의 움직임의 해상도가 그에 따르지 못 할 수도 있다. 따라서, 가상의 workspace와 현실의 workspace의 차이를 해결 할 수 있는 방법이 필요하다. 이 방법 중에 하나로 clatching[24]이 있다. Clatching은 일반적인 2D mouse의 움직임과 같은 방식이다. 마우스를 사용하다가, workspace의 끝에 다다르면, 마우스를 들어서(declutching) workspace의 가운데로 옮긴 다음에, 내려 놓는(clutching) 것과 같다.

간단한 pilot test에 따르면 browsing time의 10%가 clatching time으로 소모되었다. Clatching으로 인한 시간 소모가 무시할 수 있는 크기가 아니었으므로, 이를 해결 할 수 있는 방법을 고려하였다. 가장 큰 문제는 workspace크기의 제한이 있었으므로, 현재의 workspace크기의 제한을 받는 position control 방식이 아니라, workspace의 크기에 제한을 받지 않는 velocity control 방식으로 변경하는 것이 가장 근본적인 해결책이 된다. 대신 velocity control 방식일 때는 에러가 미치는 영향이 position control 보다 증폭되므로, 에러가 누적 되는 camera를 이용한 추적 방식 보다는 기울기 만을 이용한 추적 방식이 좀더 효과적이었다.

2.3 모바일 장치의 기울기 추적을 이용한 인터페이스(Tilt Motion Interface)

모바일 장치의 기울기 추적을 이용한 인터페이스는 3 DOF linear accelerometer의 absolute sensing을 이용하였다. 중립 위치로부터의 기울기에 비례하여, 브라우저



	2D Browsing operation	3D Navigation operation
Yaw difference from center position		
Pitch difference from center position	Velocity of 1D vertical panning	Velocity of 1D vertical panning
Roll difference from center position	Velocity of 1D horizontal panning	Velocity of 1D horizontal panning
Translation along screen normal	Zooming	1D Translation along view direction

표 3.3 모바일 장치의 기울기 추적을 이용한 인터페이스의 사용자의 움직임과 Browser Operation과의 매칭

의 panning velocity를 조정하는 형태가 된다. 일반적인 2D 이미지 브라우저의 경우엔 중립 위치로부터의 roll의 차이는 horizontal panning의 속도를 제어하며, 중립 위치로부터의 pitch의 차이는 vertical panning의 속도를 제어한다. 한편 3 DOF linear accelerometer는 2 축만 적용 가능하다는 단점이 여전히 존재하므로, zooming은 여전히 카메라를 이용한 상대적인 position control 방식을 이용하여 조절한다. 이 인터페이스 역시 ‘모바일 장치의 위치 추적을 이용한 인터페이스’와 같이 3차원 환경의 네비게이션을 위해 사용할 수 있다. Screen normal 방향의 움직임이 제한된다면, 실질적으로 무한대의 workspace를 제어할 수 있게 된다.

실질적인 구현의 경우엔 기울기 차이의 제공에 비례하여 속도를 변화 시켰다.

$$Velocity\ of\ panning = \frac{A \times \text{abs}(\Theta - \text{basis}\Theta) \times (\Theta - \text{basis}\Theta)}{\text{samplingTime}} \quad (3.1)$$

식의 Θ 값의 단위는 radian, samplingTime 값의 단위는 second가 된다. 따라서 $Velocity\ of\ panning$ 은 rad/s 의 단위를 갖는다. Accelerometer가 인식할 수 있는 기울기의 최대 값은 $\pm 1.53rad$ 이지만, 장비에 설치된 형태에 따라 실제 얻을 수 있는 최대 값은 이보다 적은 값이 된다. 실제 사용한 값은 4장, 2.4 소절의 실험을 위한 구현 부분에 설명하였다.

Pilot test결과 workspace의 제한은 극복할 수 있었지만, 세밀한 제어에서 문제를 보였다. 한편 모바일 장치의 roll을 이용하기 때문에, 화면이 기울어져 왜곡이 발생하게 된다. 이를 해결하기 위해 virtual camera역시 roll시켜서, 사용자가 보기에



	2D Browsing operation	3D Navigation operation
Pitch difference from center position	Velocity of 1D vertical panning	Velocity of 1D vertical panning
Roll difference from center position	Velocity of 1D horizontal panning	Velocity of 1D horizontal panning
Translation along screen normal	Zooming	1D Translation along view direction
Translation along vertical direction of screen	1D Vertical panning	1D Vertical translation
Yaw	1D Horizontal panning	1D Horizontal arc

표 3.4 위치, 기울기 혼성 추적 인터페이스의 사용자의 움직임과 Browser Operation과의 매칭

는 똑바로 보이도록 수정하였다[25].

2.4 위치, 기울기 혼성 추적 인터페이스(Hybrid Motion Interface)

‘모바일 장치의 위치 추적을 이용한 인터페이스’의 약점은 real workspace와 virtual workspace사이의 차이가 클 경우 clutching으로 인한 성능 문제가 발생한다는 것이다. 한편 ‘모바일 장치의 기울기 추적을 이용한 인터페이스’는 velocity control을 이용하기 때문에, 세밀한 동작의 경우에 문제가 발생하였다. 이들의 장점을 혼합할 수 있다면, 좀더 효과적인 인터페이스가 될 수 있을 것이라고 예상하였다. 표3.2와 표3.3를 관찰하면, 1모바일 장치의 위치 추적을 이용한 인터페이스’를 이용할 때는 roll을 사용하지 않는다. 한편 사용자의 움직임의 경우에도 장치의 무게 때문에, 좌우 보다 상하 기울임의 정도가 제한되는 경향이 있었다. 한편, 현재의 카메라를 이용한 pitch sensing의 경우엔 translation과 rotation을 정확히 구분을 못하는 특성이 있다. 따라서 실제로 많이 기울이지 않더라도 pitch값을 입력 받을 수 있다.

이에 따라, yaw와 pitch의 경우엔 위치 인터페이스를 사용하고, roll과 threshold이상의 pitch의 경우엔 기울기 추적 인터페이스를 하는 형태로 두 개의 인터페이스를 서로 결합할 수 있었다. 일반적인 yaw 움직임 또는 pitch 움직임의 경우엔 위치 인터페이스와 같은 방식으로 작동하고, roll 또는 pitch가 threshold값을 넘었을 경우에는 위치 추적을 중단하고, 기울기 추적 방식으로 작동한다.

간단한 pilot test에 따르면, clutching의 필요성은 없었다.



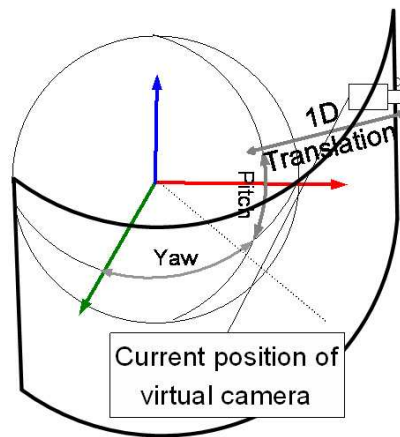


그림 3.4 인터페이스의 움직임과 원기둥 배치

제 3 절 모바일 기기에서 다수의 이미지 배치

3.1 Tiled Cylindrical Layout

모션을 이용한 인터페이스를 사용하므로, 이미지의 배치 역시 모션을 사용한 인터페이스에 적합하도록 해야 할 필요성이 있다. 현재의 인터페이스는 3 차원 모션을 추적할 수 있으므로, 이미지의 배치를 3 차원 적으로 해도 문제는 없다. 특히 *perspective view*를 사용하는 3D navigation을 이용할 경우에는 zoom이 가능한 2D image browser와 기능적으로 차이가 없다. 그림 3.3의 모션 인터페이스의 움직임을 고려해 볼 때, 비교적 효과적인 이미지 배치는 사용자를 중심으로 한 원기둥 형태이다. 모션 인터페이스의 움직임에 맞추어 사용자를 중심으로 하는 구 형의 배치를 할 경우 이미지의 왜곡을 가져 오게 된다. 또한, 실제로 사용자가 사용하는 모션의 범위에서 *pitch*의 변화는 크지 않으므로, 구 대신 원기둥을 사용함으로써 발생하는 왜곡 역시 크지 않다. 그림 3.4은 인터페이스의 움직임과 원기둥 배치 사이의 관계를 보여 준다.

한편 원기둥 형태로 배치할 때, 이미지 사이의 간격과 이미지와 사용자 사이의 거리를 일정하게 배치하여, 일반적인 2D 이미지 브라우저의 tiled 배치와 같게 하였다.





Illustration



Snap Shot

그림 3.5 Tiled Cylindrical Layout

$$LayoutRadius = \frac{0.5ImageWidth}{\tan(0.5SpanAngle)} \quad (3.2)$$

3.2 Fisheye Cylindrical Layout

Gutwin[7]의 결과에 따르면, 작은 화면을 가진 모바일 기기에서 큰 문서를 사용하는 경우 fisheye view의 사용은 상당히 효과적이었다. 이것이 다수의 이미지를 사용하는 경우에도 적용이 가능할 것이라 생각되었다. Fisheye view는 이미지를 왜곡시켜, 가운데는 강조가 되어 크게 보이고, 주변부는 작게 보이는 형태의 view이다. 이를 레아아웃에 적용을 한다면, 현재의 view에서 중심에 가장 가까운 이미지는 크기를 증가시키고, 주변부로 갈수록 이미지의 크기를 감소시키는 형태가 된다. 또한 가상의 카메라의 view가 업데이트 될 때 마다, 레이아웃도 업데이트 되어야 한다. 한편, fisheye cylindrical layout의 한 이미지의 크기는 식3.3와 같다. 현재의 가상의 카메라 위치에서 이미지의 위치를 뺀, 벡터와 현재 가상의 카메라가 바라보는 방향 벡터를 dot product한 뒤 노멀라이즈 한다. 그 뒤, fisheye view에서의 중심부와 주변부의 이미지 크기 차이를 설정함에 따라서 적절한, A,B 상수 값을 정해준다. 실제로 구현에 사용한 A,B 상수 값은 4장,2.4 소절의 실험을 위한 구현 부분에 설명하였다.



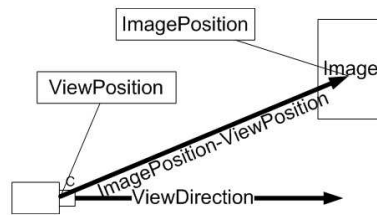


그림 3.6 Fisheye Layout의 이미지 크기 결정

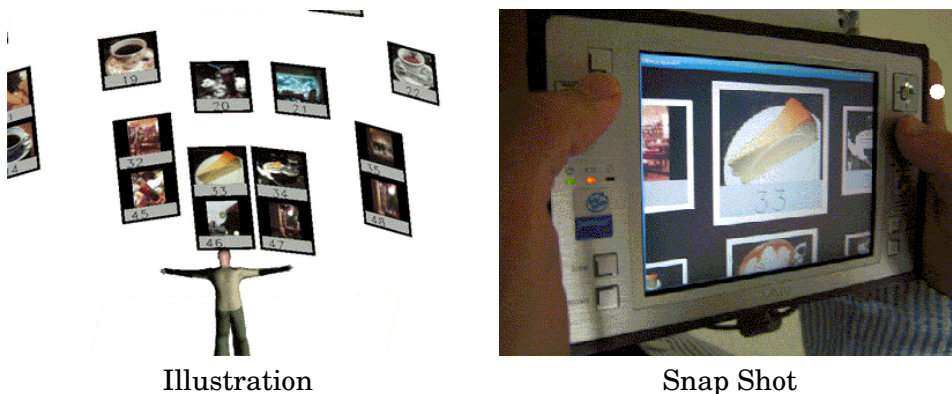


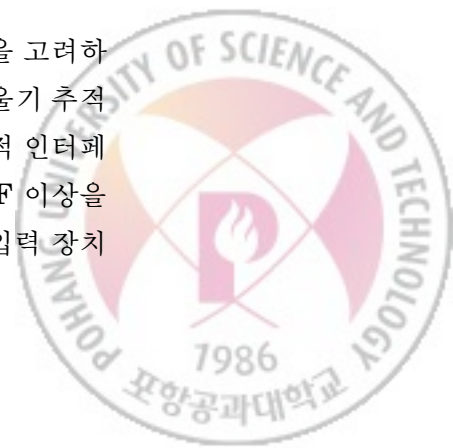
그림 3.7 Fisheye Cylindrical Layout

$$Image\ size = A \cdot \left| \left(\frac{ViewDirection \cdot (ImagePosition - ViewPosition)}{|ImagePosition| \cdot |ViewPosition|} \right)^B \right| \quad (3.3)$$

제 4 절 이미지 브라우저의 구현

4.1 인터랙션 설계

앞서의 장에서 이미지 브라우저에서 사용할 모션 인터페이스와 레이아웃을 고려하였다. 이미지 브라우저에 사용할 모션 인터페이스의 특성을 살펴 보면, 기울기 추적을 이용한 인터페이스는 absolute input device에, 위치와 기울기 혼성 추적 인터페이스는 relative input device에 속한다. 세 종류의 인터페이스 모두 3DOF 이상을 지원하는 인터페이스이며, 터치 스크린이나 태블릿 처럼, 화면을 그대로 입력 장치



로 사용하지 않으므로, **indirect input device**이다[24]. 덕분에 인터랙션에 여러 개의 버튼 입력과 입력 장치의 입력을 동시에 받을 수 있는 장점이 있다. 한편, 이들 인터페이스와 레이아웃을 이용하여, 이미지 브라우저를 만들기 위해서는 이미지 브라우저의 기능과 그 인터랙션을 정의 하여야 한다.

이미지 브라우저가 가지는 기능으로는 **panning**, **zooming**이 있다. 다수의 이미지의 **thumbnail**을 제시할 수 있어야 하고, 이 **thumbnail** 이미지에서의 **panning**과 **zooming**이 가능해야 한다. **zooming**이 이루어 진다면, **thumbnail** 이미지의 **detail**을 보여 줄 수 있어야 하며, 보고자 하는 이미지의 **original size**의 이미지 역시 보여 줄 수 있어야 한다. 2.2, 2.3, 2.4 에서 제안 또는 소개한 인터페이스는 **panning**과 **zooming**을 모두 지원하는 형태의 인터페이스이므로, 이미지 브라우저가 가져야 하는 기능을 쉽게 지원할 수 있다. 또한 인터페이스 적으로 여러 이미지의 **thumbnail**을 보여 주는 상태에서 곧바로 디테일한 이미지를 보여 줄 수 있지만, 일반적 **two level discrete zoom browser**와 같이 선택에 따라서, **original image**하나만 **original resolution**으로 보여주는 모드가 있을 수 있다.

이미지 브라우저를 구현한 환경은 다음과 같다. Intel M733 CPU, 512MB memory, 5 inch LCD, Windows XP 환경인, SONY VGN-U71P ultra mobile computer를 이용하였으며, Freescale의 MMA7260 3 axis-accelerometer와 Logitech quickcam camera를 센서로 이용하였다. 카메라를 통한 모션 인식이 가능한 프로세싱 파워를 필요로 하였기 때문에, 일반 컴퓨터의 성능을 낼 수 있는 **ultra mobile computer**를 이용하였다.

이미지 브라우저는 동시에 이미지를 정리하고 관리하는 파일 관리자로서의 기능을 함께 가질 수 있다. 이미지를 정리하고 관리하는 기능을 갖는다면, 이미지의 선택, 이동, 복사 등의 기능을 필요로 한다. 모션을 이용한 입력 장치가 **indirect input device**이므로, 버튼을 동시에 사용할 수 있는 점을 이용하여, 이러한 추가적인 인터랙션은 버튼과 메뉴 인터페이스를 이용하여 구현하였다.

구현할 인터랙션에서 사용되는 버튼의 목록은 표 3.5와 같고, **state diagram**은 그림 3.8와 같다.



버튼의 종류	기능
Select button	마우스의 왼쪽 버튼과 같이 선택 기능을 한다.
Menu button	마우스의 오른쪽 버튼과 같이 인터랙션을 위한 메뉴를 출력한다.
Toggle button	현재 하나의 이미지를 선택하는 모드인지, 다수의 이미지를 선택하는 모드인지 결정한다.
Left arrow button	메뉴의 선택에 사용된다.
Right arrow button	메뉴의 선택에 사용된다.
Reset button	센서의 입력 값을 초기화 하고, 현재 위치를 기준으로 calibration 한다.
Clutch button	센서의 입력 값을 무시하여, declutching 을 수행한다.

표 3.5 Motion Interface에서 사용하는 버튼



State diagram에서 표시한 인터렉션에서 사용되는 버튼은 `select`, `menu`, `toggle`, `left arrow`, `right arrow`의 5개이며, 각각 버튼을 누른 순간에 썸 순간으로 나뉜다. 즉 state diagram은 10개의 입력 값을 갖는다. State diagram에서 입력 값이 주어졌을 때, 현재의 state를 그대로 유지하는 입력은 생략하였다.

State diagram은 Idle또는 MIdle state를 기본으로 한다. 이 State는 아무 이미지도 선택되지 않았을 경우를 의미하여, M으로 시작하는 state는 다수의 이미지를 선택할 수 있는 경우를 의미한다. Move interaction의 경우엔 Drag state를 통하여 이루어진다. Drag state는 select 버튼을 누른 상태에서 움직였을 때, 들어가는 state이다. 현재 위치로부터 적절한 거리가 떨어진 뒤에 select 버튼을 떼면, move interaction이 이루어지고, Select또는 MSelect state로 변화하게 된다. Select또는 MSelect state에서 menu button을 선택하면 Select또는 MSelect state와 연관된 menu state로 들어가 open, copy, clear, cancel 등의 인터렉션을 할 수 있다. Clear 메뉴는 현재 선택된 이미지를 다 취소하고, Idle또는 MIdle state로 돌아가는 메뉴이고, cancel의 경우는 이전 상태로 돌아가는 메뉴이다. Open 인터렉션을 선택하였을 경우엔 Open state가 되어 하나의 이미지만을 보여 준다. 이 시점에서 select 또는 menu 버튼을 선택하면, 원래의 Select 또는 MSelect state로 돌아온다. Copy 인터렉션을 선택하면, 현재 선택된 이미지가 복사된 상태에서 브라우징을 할 수 있다. 이 경우에 menu button을 선택하면, paste, clear, cancel 등의 인터렉션을 할 수 있다. Paste 메뉴를 선택하면, 복사한 이미지를 붙여 넣을 수 있다.





그림 3.8 인터랙션의 State Diagram



4.2 구현

이미지 브라우저는 각각 hybrid motion interface와 tilt motion interface와 tiled cylindrical layout, fisheye cylindrical layout의 조합으로 구현되었다. 3D motion에 기반한 2D panning과 zooming이 가능하며, 추가로 선택, 이동, 복사, 한 이미지만 보기, 등의 기능을 구현하였다. 각각 기능의 동작은 그림 3.9,3.10,3.11, 3.12에 나타내었다.

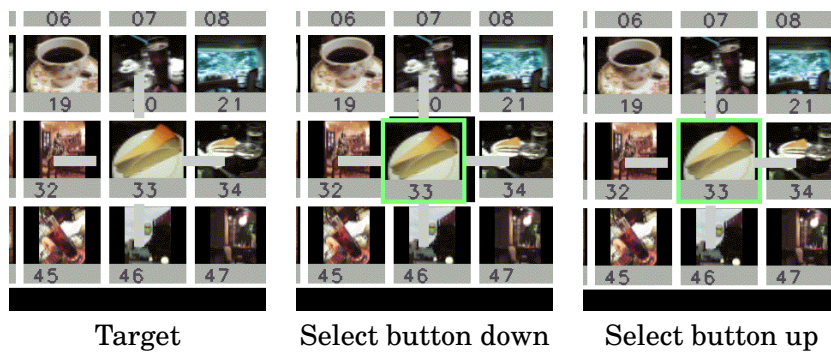


그림 3.9 이미지의 선택

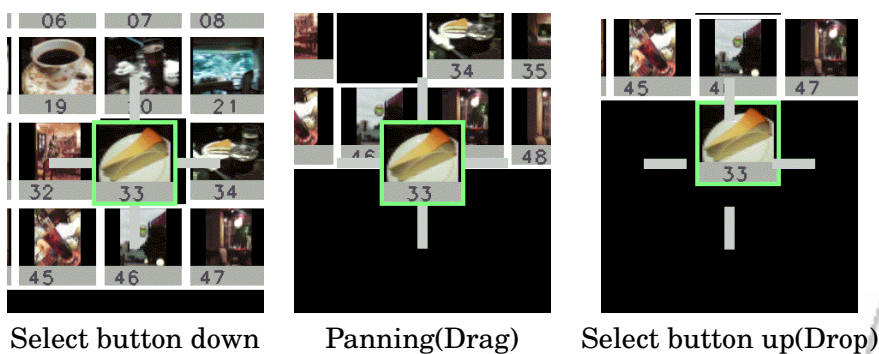


그림 3.10 이미지의 이동



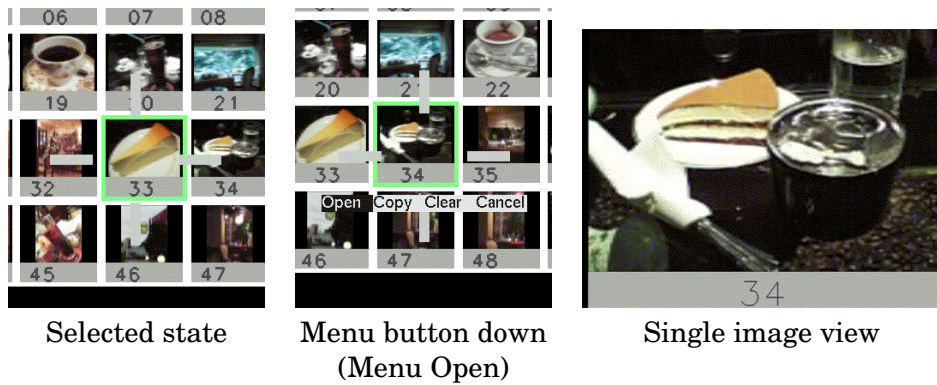


그림 3.11 한 이미지만 보기

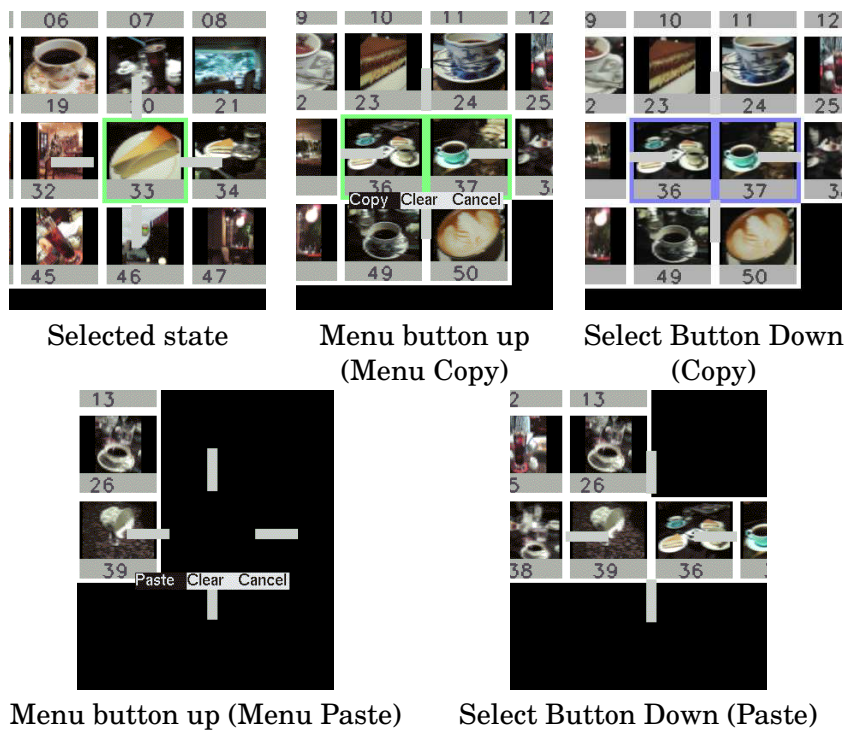


그림 3.12 이미지의 복사



제 4 장

인터페이스와 레이아웃 간의 사용성 실험

제 1 절 실험 목적

이전에 소개하고 구현한 인터페이스들과 레이아웃의 조합들을 직접 이미지 브라우징 과정에 사용해 보면서 그 조합들의 사용성을 평가한다. 이를 통해 각각의 인터페이스나 레이아웃이 기대한 효과를 낼 수 있었는지, 주어진 인터페이스와 레이아웃에 알맞은 조합은 어떤 것인지 확인해 볼 수 있다.

제 2 절 실험 설계

2.1 피실험자

24 명의 피실험자가 실험에 참여하였다. 피실험자 중 18 명은 남성, 6 명은 여성이었다. 피실험자는 모두 학생이었으며, 나이 분포는 만 17 세에서 25 세 사이였다. 피실험자 나이의 평균은 만 21 세이다. 피실험자 중, 모바일 기기를 이용하여 이미지를 관리하는 인원은 15 명, 모션 인터페이스를 사용한 경험이 있는 인원은 6 명이었다.



2.2 독립 변수

실험의 독립 변수는 **button interface**, **tilt motion interface**, **hybrid motion interface**의 중 하나의 인터페이스 적용 여부와, **tiled cylindrical layout**과 **Fisheye cylindrical layout** 중 하나의 레이아웃의 적용 여부이다. 표 4.1에 독립 변수들이 정리되어 있다. **Pilot experiment**시 **position tracking motion interface**는 **clutching** 문제로 인해, **hybrid motion interface**보다 좋지 못한 성능을 보여 주었으므로, 실험에 사용되는 변수 중에서는 제거되었다. 실험의 전체적인 디자인은 3 또는 2 level을 갖는 **2 factor within-subject design**으로, 피실험자는 표 4.2와 같은 6 가지의 실험 세션을 수행하게 된다.

독립 변수	Levels		
Interface	Button interface	Hybrid motion interface	Tilt motion interface
Layout	Tiled cylindrical layout		Fisheye cylindrical layout

표 4.1 독립 변수

Interface	Button interface	Hybrid motion interface	Tilt motion interface
Tiled cylindrical layout	TB	TH	TT
Fisheye cylindrical layout	FB	FH	FT

표 4.2 6 개의 인터페이스와 레이아웃의 조합

한편 각각의 인터페이스들에서는 **zooming** 요소가 제거 되었다. **pilot test**의 결과 사용자들은 **zooming**을 이용하기 보다는 최대한 **zoom out**한 상태에서 실험에 임하였다. 결국 **zoom**과 관련된 결과를 얻을 수 없었으며, 더하여, 모션 인터페이스의 스크린 방향의 **1D translation sensing**능력의 해상도 문제로, 현재 사용자의 팔 길이로 제한 되는 **workspace** 크기 안에서는 적절한 **zooming**이 이루어지지 않았다.



2.3 종속 변수

실험의 종속 변수는 실행 시간, 브라우징을 위한 이동 궤적의 총 길이, 인터페이스의 사용하기 쉬움, 인터페이스의 배우기 쉬움, 인터페이스의 선호도, 인터페이스의 직관적인 정도, 인터페이스의 자연스러운 정도, 인터페이스의 재미있는 정도, 레이아웃의 선호도, 레이아웃의 직관적인 정도, 레이아웃의 자연스러운 정도, 레이아웃의 재미있는 정도, 인터페이스와 레이아웃의 조합을 사용하기 쉬운 정도, 인터페이스와 조합을 선호하는 정도이다. 표4.3에 정리 되어 있다.

종속 변수						
성능	실행 시간			이동 궤적의 총 길이		
인터페이스 관련 설문	사용하기 쉬움	배우기 쉬움	선호도	직관적	자연스러움	재미
레이아웃 관련 설문	선호도		직관적	자연스러움		재미
인터페이스와 레이아웃의 조합 관련 설문	사용하기 쉬움	선호도	직관적	자연스러움	재미	

표 4.3 종속 변수

2.4 구현

구현된 실험 환경에서 이미지의 수는 100장이며 레이아웃의 높이는 5장, 너비는 20장의 이미지를 이용하였다. 레이아웃의 반지름은 1145pixels이며, 이미지와 이미지 사이의 각도는 $5^\circ(0.087\text{rad})$ 이다. 각각의 이미지는 $100 \times 100\text{pixels}$ 이며, 화면의 크기는 $400 \times 300\text{pixels}$ 이다.

실험에서 모션 인터페이스와 비교하기 위한 버튼 인터페이스의 구현이 필요하였다. 버튼 인터페이스는 상,하,좌,우 네 개의 버튼을 이용하였으며, 원하는 방향의 버튼을 누르면, 그 방향의 인접한 이미지의 중심으로 이동하도록 하였다.

실험 환경의 경우 버튼을 한번 누를 때 이동하는 거리는 100 pixels(또는 0.087rad)이다. 이 값은 각각의 이미지의 중심으로 바로 이동 시키기 위해, 이미지를 배치한 레이아웃의 이미지 사이의 각도와 같은 값으로 정해졌다. 이미지를 버튼을 계속 누르고 있는 경우 버튼의 최대 샘플링 레이트는 초당 18번이며, 이에 따라 버튼 입력의 최고 속도는 1800pixels/s 이다.



한편 실험 환경에서 tilt motion의 이동 속도는 horizontal panning의 경우 식3.1의 A 값에 1200pixels/rad을 대입한 값이고, vertical panning의 경우엔 같은 식에 1536pixels/rad을 대입한 값이다. 한편 식3.1의 samplingTime의 값은 초당 8번 sampling하므로, 0.125초이다. Panning속도의 단위가 pixels/s가 되도록 조정 한 식은 4.1이다.

Velocity of horizontal panning

$$= \frac{1200 \times \text{abs}(\text{roll} - \text{basisRoll}) \times (\text{roll} - \text{basisRoll})}{0.125} (\text{pixels/s})$$

Velocity of vertical panning

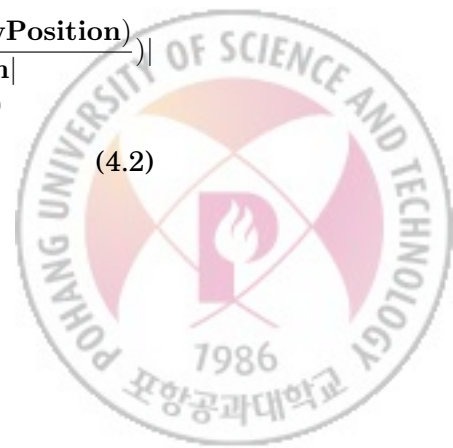
$$= \frac{1536 \times \text{abs}(\text{pitch} - \text{basisPitch}) \times (\text{pitch} - \text{basisPitch})}{0.125} (\text{pixels/s})$$

(4.1)

구현에 사용한 accelerometer는 $\pm 1.53\text{rad}$ 의 값을 인식할 수 있지만 설치한 상태에 따라 실제로 인식할 수 있는 범위는 차이가 있다. 실제 roll 값의 최대, 최소 값은 $\pm 1.21\text{rad}$ 이 된다. Basis는 최대 값과 최소 값의 중간 값을 잡으므로, 0rad으로 한다. 이 값을 식 4.1에 대입하면 vertical panning의 최고 속도는 1757pixels/s가 된다. 이는 버튼 입력의 최고 속도 1800pixels/s와 가까운 값이다. 마찬가지로 실제 pitch값의 최대 값과 최소 값은 각각 $-1.53\text{rad}, 0.72\text{rad}$ 이다. 그 중간 값을 basis로 잡으면, pitch - basisPitch의 값의 최대, 최소 값은 $\pm 1.12\text{rad}$ 가 된다. 이를 식4.1에 대입하면, vertical panning의 최고 속도는 1927pixels/s가 된다. 이 역시 버튼 입력의 최고 속도 1800pixels/s와 가까운 값이다.

Fisheye Layout에서 사용한 이미지 크기의 경우, 식3.3를 식4.2와 같이 변형하였다.

$$\begin{aligned} \text{DotProduct} &= \left(\frac{\mathbf{ViewDirection} \cdot (\mathbf{ImagePosition} - \mathbf{ViewPosition})}{|\mathbf{ImagePosition}| \cdot |\mathbf{ViewPosition}|} \right) \\ \text{ModifiedDotProduct} &= 10 \times (\text{DotProduct} - 0.9) \text{ if } \text{DotProduct} > 0.9 \\ \text{ImageSize} &= \text{ModifiedDotProduct}^{10 \times (\text{ModifiedDotProduct})^{1.5}} \end{aligned} \quad (4.2)$$



2.5 실험 절차

2.5.1 실험 전 설문

피 실험자는 실험 전에 기본적인 인적 정보와 모바일 기기를 이용한 이미지 관리 여부, 모션 인터페이스 사용 경험 여부에 대한 설문을 한다. 이 시점에서 인터페이스의 사용법에 대한 설명을 한다.

2.5.2 모션 인터페이스를 위한 트레이닝 세션

모든 피 실험자가 모션 인터페이스에 익숙하지 않으므로 별도의 트레이닝 세션을 두었다. 트레이닝 세션은 모션 인터페이스의 기본적인 조작을 숙지 시키는 것을 목적으로 하였다. 트레이닝 세션은 ‘자유 조작’, ‘상하 조작’, ‘좌우 조작’, ‘상하좌우 혼합 조작’, ‘자유 조작’의 5 개의 서브 세션으로 나누어져 있고, 각각의 서브 세션은 ‘자유 조작’ 세션을 제외하곤 5개의 트라이얼로 이루어져 있다.

트레이닝 세션의 인터페이스는 그림 4.1에서 볼 수 있듯, 상부에 미니 맵이 존재하여, 찾아야 할 이미지의 위치를 표시하고, 상부 우측엔 찾아야 할 이미지를 작게 표시하여 준다. 또한 찾아야 할 그림의 프레임을 분홍색으로 표시해 준다. 처음의 ‘자유 조작’ 세션에서 기본조작 법을 설명한 다음, 그 뒤의 세션들을 통해 반복 연습 시킨다. 매 트라이얼 마다 주어진 패턴에 따라, 그림을 선택하기 위해선 기본적인 조작법을 따라 움직여야 하는 위치에 그림을 배치하였다.

2.5.3 수행 작업

실험에서 피실험자에게 주어진 수행 작업은 두 가지이다. 전체 100 장의 사진들 중에서 찾아야 할 사진이 있는 위치를 알려준 상태에서 사진 한 장을 찾는 수행 작업과, 전체 100 장의 사진 중에서 찾아야 할 사진이 있는 위치를 알려주지 않은 상태에서 사진 한 장을 찾는 수행작업이다.

전자의 수행 작업을 통해, 사진을 검색하고 인지하는 인지적 부담(cognition load)을 가능한한 배제하고 인터페이스와 레이아웃의 효과만을 검증할 수 있다. 그리고 후자의 수행 작업을 통해 사진을 검색하고 인지하는 인지적 부담을 포함한 상태에서 작업 인터페이스와 레이아웃의 효과를 검증할 수 있다.

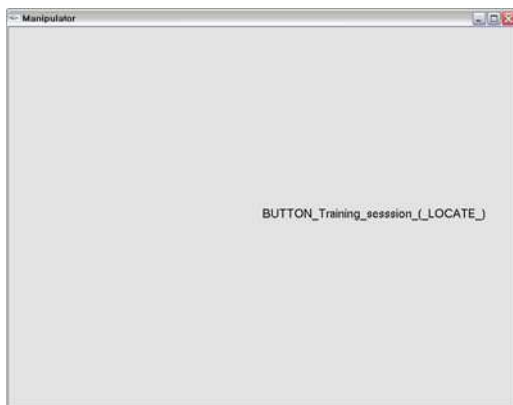




그림 4.1 트레이닝 세션



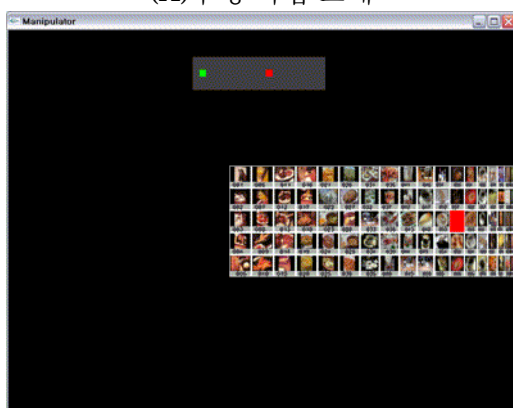
수행 작업의 순서는 사진의 위치를 알려주고, 사진을 찾는 수행작업을 실행한 후에, 사진의 위치를 알려주지 않고 사진을 찾는 수행작업을 실시한다. 매 수행 작업은 24번의 trial로 이루어져 있고, 이 중 처음 4 번의 trial은 트레이닝 세션으로 사용한다. 이 두 수행작업을 6 가지의 실험 세션마다 반복하게 되므로, 피실험자는 12 번의 수행작업을 하게 된다. 6가지 실험 세션의 순서는 Latin Square를 통하여 결정되었다. Trial의 시작 점으로부터 목표 이미지 까지의 거리는 4개의 Trial마다 거리의 합이 같도록 조정되었다.



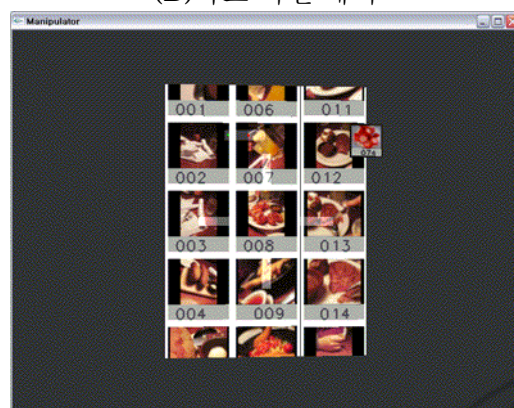
(A)수행 작업 소개



(B)목표 사진 제시



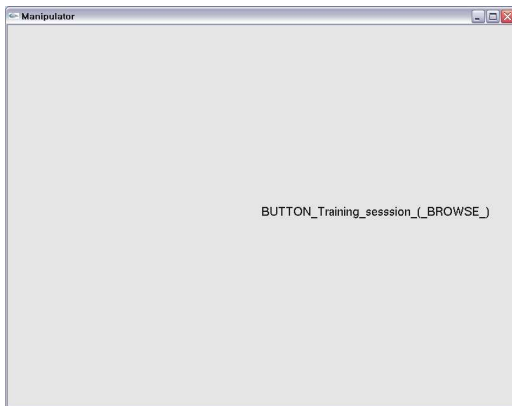
(C)목표 사진의 위치 제시



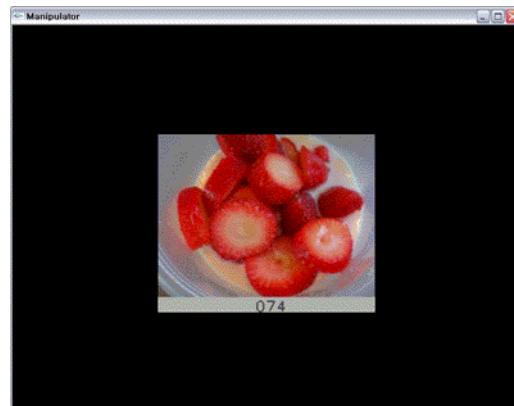
(D)실험 시작

그림 4.2 위치를 알려준 사진의 브라우징 수행 작업

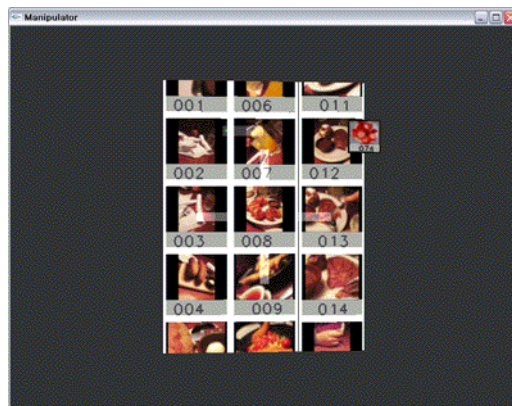




(A)수행 작업 소개



(B)목표 사진 제시



(C)실험 시작

그림 4.3 위치를 알려주지 않은 사진의 브라우징 수행 작업



사진의 위치를 알려주는 수행 작업의 경우엔 화면 상부의 미니 맵에 목표한 사진의 위치가 표시 되고, 화면 상부 우측에 찾아야 할 사진이 작게 표시된다. 그림4.2(D)에 실험 화면이 나타나 있다.

사진의 위치를 알려주지 않는 수행 작업의 경우엔, 그림4.3(C) 에서 볼 수 있듯, 화면 상부의 미니 맵엔 목표한 사진의 위치가 표시 되지 않고, 화면 상부 우측의 미니맵에만 찾아야 할 사진이 표시 된다.

같은 독립 변인 조건으로 두 번의 수행 작업이 끝나면, 현재의 조건에 대한 설문지를 작성한다. 설문지는 인터페이스의 사용하기 쉬움, 인터페이스의 배우기 쉬움, 인터페이스의 선호도, 인터페이스의 직관적인 정도, 인터페이스의 자연스러운 정도, 인터페이스의 재미있는 정도, 레이아웃의 선호도, 레이아웃의 직관적인 정도, 레이아웃의 자연스러운 정도, 레이아웃의 재미있는 정도, 인터페이스와 레이아웃의 조합을 사용하기 쉬운 정도, 인터페이스와 조합을 선호하는 정도에 대하여 질문 하며, 답변은 1~7 까지 7단계로 답하도록 하였다. 또한 매 세션 조건에 대한 전체적인 소감이나 힘들었던 점을 설문지에 작성하도록 하였다.

2.5.4 실험 후 인터뷰 작성

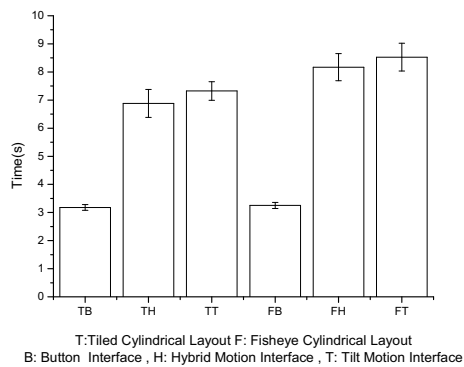
모든 실험을 끝마친 뒤엔 실험에 대한 전체적인 소감이나, 건의하고 싶은 사항을 설문지에 작성하도록 하였다.

제 3 절 실험 결과 및 토론

3.1 위치를 알려준 사진에 대한 브라우징 태스크 수행 결과

3.1.1 작업 시간





Layout	Interface	평균값 (초)	표준편차(초)
Tiled	Button	3.1800106	0.5149285
Cylindrical Layout	Hybrid Motion	6.8812734	2.4389542
	Tilt Motion	7.3238534	1.6244274
Fisheye	Button	3.2524978	0.5342975
Cylindrical Layout	Hybrid Motion	8.1701486	2.3675440
	Tilt Motion	8.5266817	2.4212651

그림 4.4 위치를 알려준 사진에 대한 브라우징 수행시 실행 시간



Source	F value	P Value
Layout	F(1,23) = 11.53	0.0025
Interface	F(2,46) = 84.10	<.0001
Layout*Interface	F(2,46) = 2.92	0.0642

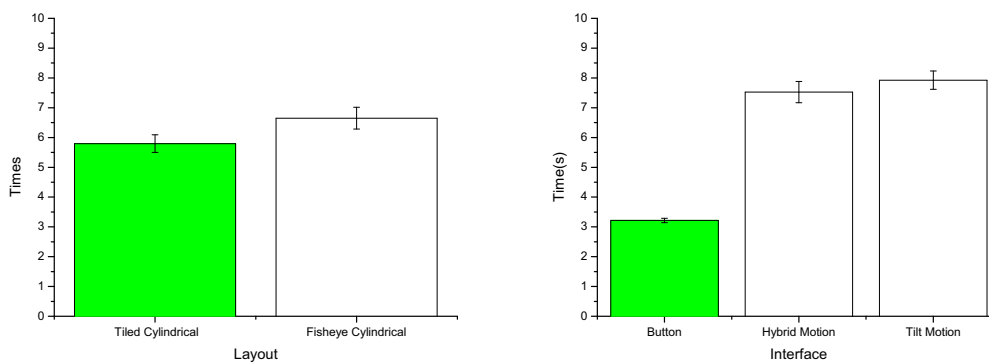
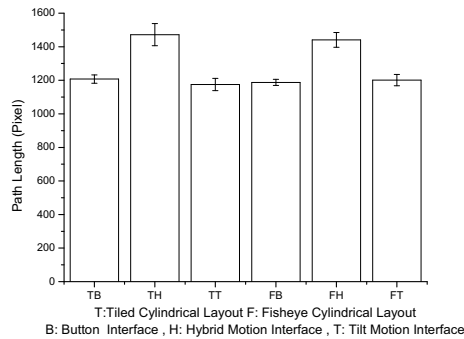
(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$ (B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.4 위치를 알려준 사진에 대한 브라우징 수행 시간에 대한 분산 분석

위치를 알려준 사진에 대한 브라우징 수행 시간의 경우, 표 4.4의 분산 분석에서 볼 수 있듯, button 인터페이스가 hybrid motion 인터페이스나 tilt motion 인터페이스보다 유의 하게 적은 시간이 들었다. 두 모션 인터페이스 사이에는 유의한 차이는 없었다. 또한 레이아웃 역시 브라우징 수행 시간에 유의한 효과를 주었다. Tiled cylindrical layout이 fisheye cylindrical layout보다 더 짧은 시간이 걸렸다. Interface와 layout 간의 interaction은 유의한 수준으로 존재하지는 않았다. 브라우징 수행시 실행 시간의 차이는 레이아웃 보단 인터페이스에 더 영향을 받았다.



3.1.2 이동 궤적의 총 길이



Layout	Interface	평균값 (Pixel)	표준편차(Pixel)
Tiled	Button	1207.42	120.8696591
Cylindrical Layout	Hybrid Motion	1471.88	323.4667048
	Tilt Motion	1175.15	177.8883714
Fisheye	Button	1187.26	89.7946133
Cylindrical Layout	Hybrid Motion	1440.87	214.7745210
	Tilt Motion	1201.07	163.9902333

그림 4.5 위치를 알려준 사진에 대한 브라우징 태스크 수행시 이동 궤적의 총 길이

위치를 알려준 사진에 대한 브라우징 태스크 수행시 이동 궤적의 총길이는 표 4.5 (A)의 분산 분석에서 볼 수 있듯, 인터페이스에만 영향을 받았다. SNK test 결과(표 4.5 (B))에 따르면, hybrid motion interface 만이 다른 인터페이스와 유의한 차이를 보였다. 이는 hybrid motion interface가 기대한 것과는 달리, 섬세한 선택에 효과적이지 못하였다는 것을 의미한다. Hybrid motion interface에 대한 사용자의 설문 결과에 따르면, ‘많이 흔들린다’ 내지는 ‘섬세한 조작을 하는데 불편하다’는 언급이 있었다.



Source	F value	P Value
Layout	$F(1,23) = 0.06$	0.8038
Interface	$F(2,46) = 34.06$	<.0001
Layout*Interface	$F(2,46) = 0.40$	0.6733

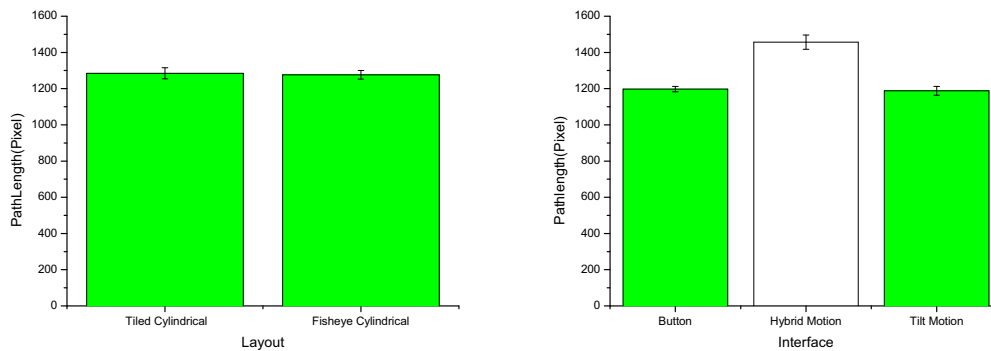
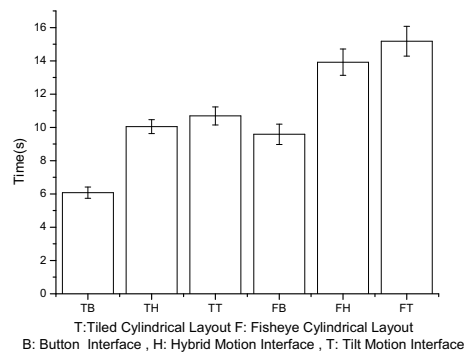
(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$ (B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.5 위치를 알려준 사진에 대한 브라우징 수행시 이동 궤적의 총 길이에 대한 분산 분석



3.2 위치를 알려주지 않는 사진에 대한 브라우징 태스크 수행 결과

3.2.1 작업 시간



Layout	Interface	평균값 (시간)	표준편차 (시간)
Tiled Cylindrical Layout	Button	6.0808370	1.6620614
Tiled Cylindrical Layout	Hybrid Motion	10.0425143	2.0442715
Tiled Cylindrical Layout	Tilt Motion	10.6893285	2.6629527
Fisheye Cylindrical Layout	Button	9.5879515	2.9995676
Fisheye Cylindrical Layout	Hybrid Motion	13.9202137	3.8663998
Fisheye Cylindrical Layout	Tilt Motion	15.1804130	4.3822829

그림 4.6 위치를 알려주지 않는 사진에 대한 브라우징 수행시 실행 시간

위치를 알려 주지 않는 사진에 대한 브라우징 태스크의 결과는 위치를 알려주는 사진과 마찬가지로, 인터페이스와 레이아웃 모두 유의한 영향을 주었다. 버튼 인터페이스를 사용하는 경우의 수행 시간이, 두 모션 인터페이스를 사용하는 경우 보다 유의하게 짧았지만, 두 종류의 모션 인터페이스 사이에는 유의한 차이가 발생하지 않았다. 또한 tiled cylindrical layout이 fisheye cylindrical layout보다 좀더 짧은 수행시간을 보여 주었다. 레이아웃과 인터페이스 사이의 인터랙션은 존재하지 않았다.



Source	F value	P Value
Layout	F(1,23) = 70.51	<.0001
Interface	F(2,46) = 48.61	<.0001
Layout*Interface	F(2,46) = 0.50	0.6117

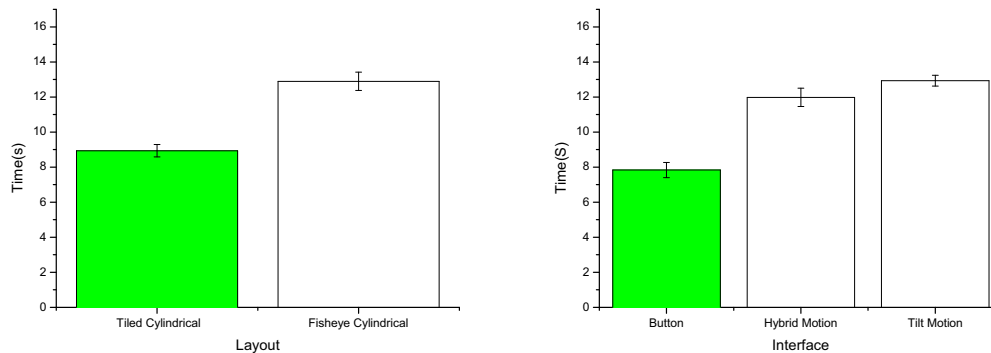
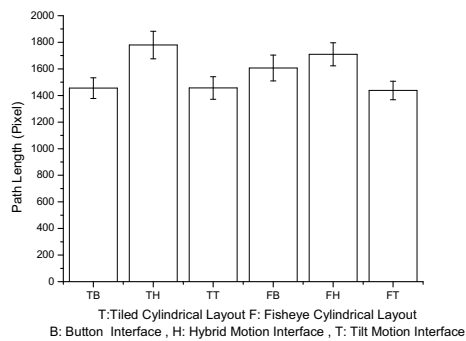
(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$ (B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.6 위치를 알려주지 않은 사진에 대한 브라우징 수행 시간에 대한 분산 분석



3.2.2 이동 궤적의 총 길이



Layout	Interface	평균값 (Pixel)	표준편차 (Pixel)
Tiled Cylindrical Layout	Button	1455.68	381.8858419
	Hybrid Motion	1779.78	507.8030686
	Tilt Motion	1456.48	414.7463430
Fisheye Cylindrical Layout	Button	1607.19	476.0865156
	Hybrid Motion	1709.93	422.6120121
	Tilt Motion	1437.88	341.6922511

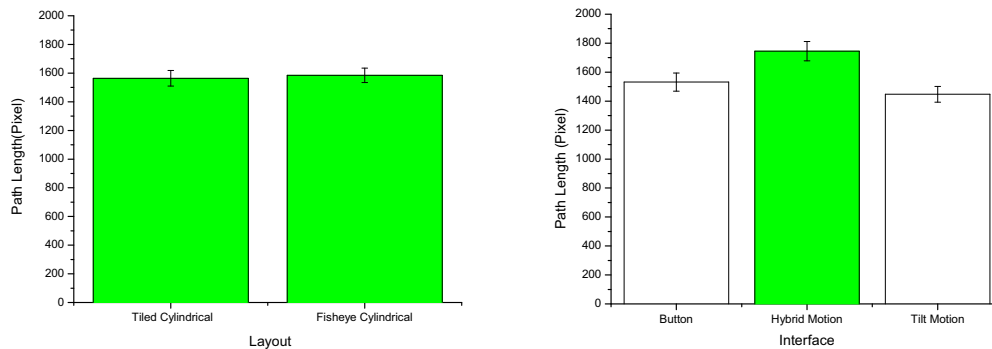
그림 4.7 위치를 알려주지 않은 사진에 대한 브라우징 태스크 수행시 이동 궤적의 총 길이

위치를 알려 주지 않는 경우에도, **hybrid motion** 인터페이스가 다른 인터페이스 보다 유의하게 긴 이동 궤적을 보여 주었다. 그외에 **layout**의 영향은 없었다.



Source	F value	P Value
Layout	$F(1,23) = 0.06$	0.8090
Interface	$F(2,46) = 14.72$	<.0001
Layout*Interface	$F(2,46) = 1.75$	0.1853

(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$



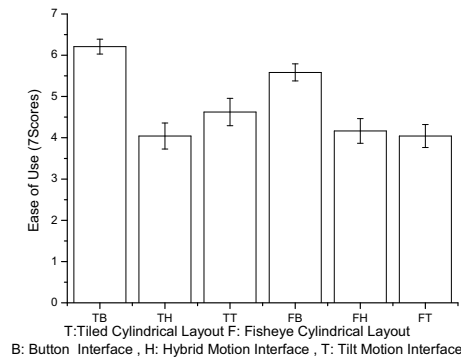
(B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.7 위치를 알려주지 않은 사진에 대한 브라우징 태스크 수행시 이동 궤적의 총 길이의 분산 분석



3.3 인터페이스에 대한 설문 결과

3.3.1 인터페이스를 사용하기 쉬운 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Tiled Cylindrical Layout	Button	6.2083333	0.8836272
	Hybrid Motion	4.0416667	1.5457984
	Tilt Motion	4.6250000	1.6244274
Fisheye Cylindrical Layout	Button	5.5833333	1.0134959
	Hybrid Motion	4.1666667	1.4645571
	Tilt Motion	4.0416667	1.3666578

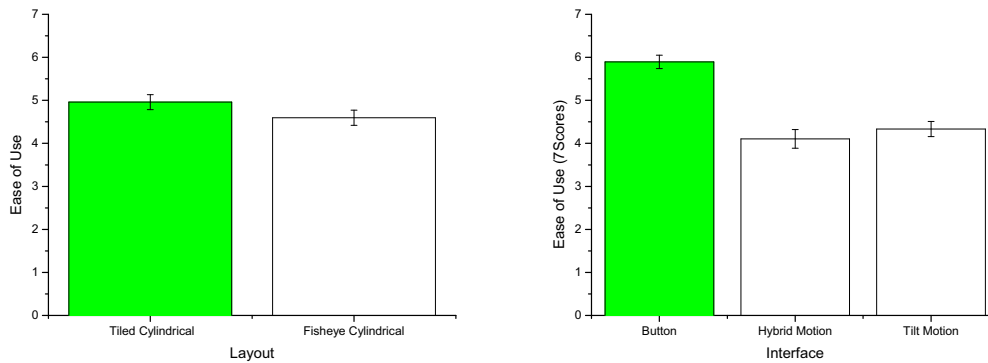
그림 4.8 인터페이스를 사용하기 쉬운 정도

인터페이스를 사용하기 쉬운 정도에 대한 설문 결과 **button** 인터페이스가 두 종류의 모션 인터페이스 보다 사용하기 쉽다는 결론을 얻었다. 인터페이스의 사용하기 쉬운 정도에 레이아웃 또한 유의한 영향을 주었다. 버튼 인터페이스는 이미 사람들에게 익숙하고, 굉장히 많은 훈련을 받은 장치이기 때문인 것으로 생각된다.



Source	F value	P Value
Layout	F(1,23) = 5.79	0.0245
Interface	F(2,46) = 32.76	<.0001
Layout*Interface	F(2,46) = 2.31	0.1104

(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$

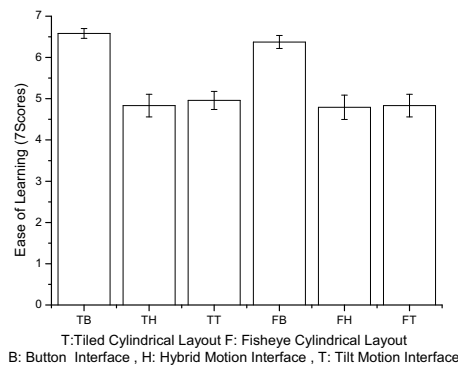


(B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.8 인터페이스를 사용하기 쉬운 정도에 대한 분산 분석



3.3.2 인터페이스를 배우기 쉬운 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Tiled	Button	6.5833333	0.5835921
Cylindrical Layout	Hybrid Motion	4.8333333	1.3405601
	Tilt Motion	4.9583333	1.0826363
Fisheye Cylindrical Layout	Button	6.3750000	0.7696696
	Hybrid Motion	4.7916667	1.4440031
	Tilt Motion	4.8333333	1.3405601

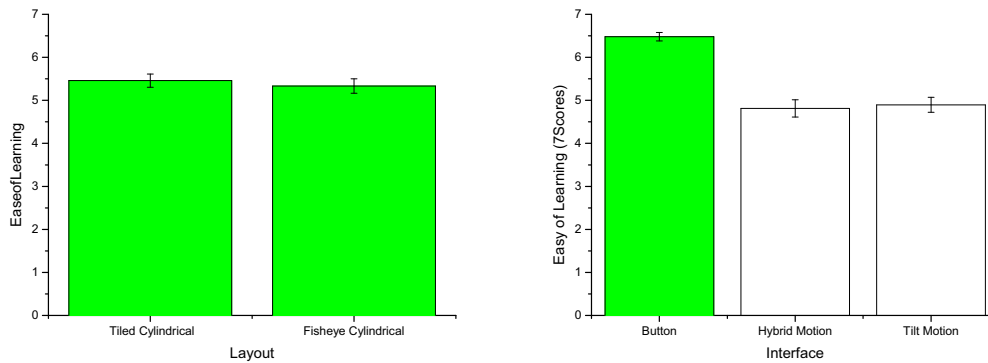
그림 4.9 인터페이스를 배우기 쉬운 정도

인터페이스를 배우기 쉬운 정도 역시 버튼 인터페이스가 가장 배우기 쉽다는 결론을 얻었다. 이미 버튼 인터페이스 자체가 충분히 학습이 된 상태이기 때문에 더 배우기 쉽다고 답변했을 가능성이 크다.



Source	F value	P Value
Layout	$F(1,23) = 1.11$	0.3020
Interface	$F(2,46) = 33.38$	<.0001
Layout*Interface	$F(2,46) = 0.17$	0.8417

(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$

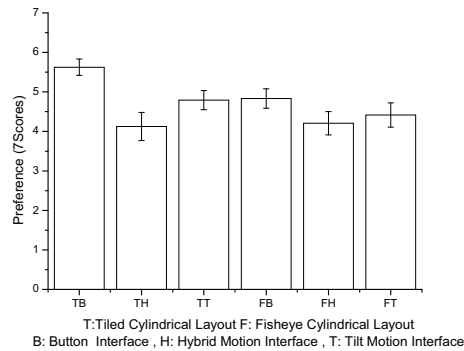


(B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.9 인터페이스를 배우기 쉬운 정도에 대한 분산 분석



3.3.3 인터페이스를 선호하는 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Tiled	Button	5.6250000	1.0134959
Cylindrical Layout	Hybrid Motion	4.1250000	1.7523276
	Tilt Motion	4.7916667	1.1787675
Fisheye	Button	4.8333333	1.2038585
Cylindrical Layout	Hybrid Motion	4.2083333	1.4440031
	Tilt Motion	4.4166667	1.5012072

그림 4.10 인터페이스를 선호하는 정도

인터페이스를 선호하는 정도에 있어서는 레이아웃과 인터페이스 사이의 인터랙션이 발생하였다. Tiled cylindrical layout의 경우에는 button interface, tilt motion interface, hybrid motion interface 순으로 유의한 선호도의 차이가 있었지만, fisheye cylindrical layout의 경우에는 세 인터페이스 사이의 선호도의 차이는 유의하지 않았다. 즉, 평소에 익숙한 레이아웃에서 좀더 선호도의 차이가 큰 것을 볼 수 있다. 특히 버튼 인터페이스가 tiled cylindrical layout 일때와 fisheye cylindrical layout의 사이에서 큰 차이를 보여 주었다.



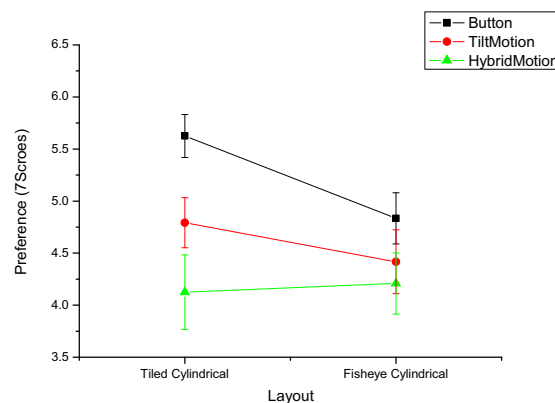
Source	F value	P Value
Layout	$F(1,23) = 3.12$	0.0907
Interface	$F(2,46) = 5.69$	0.0062
Layout*Interface	$F(2,46) = 3.30$	0.0459

(A) 2-Way within subject ANOVA, 유의 수준 $p < 0.05$, Layout*Interface Interaction 존재

Source	Button		Hybrid Motion		Tilt Motion	
	F(1,23)	P Value	F(1,23)	P Value	F(1,23)	P Value
Layout	9.11	0.0061	0.10	0.7531	1.35	0.2577

Source	Tiled	Cylindrical	Fisheye	Cylindrical
	F(2,46)	P Value	F(2,46)	P Value
Interface	8.80	0.0006	1.56	0.2217

(B) 인터페이스 선호도에 대한 simple effect, 1 way-within subject ANOVA, 유의 수준 $p < 0.05$

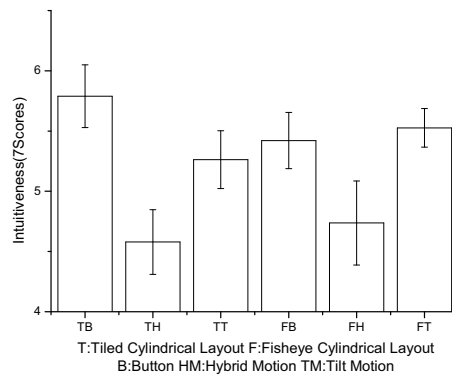


(C) 인터페이스의 선호도에 존재하는 Layout과 Interface 사이의 Interaction

표 4.10 인터페이스를 선호하는 정도에 대한 분산 분석



3.3.4 인터페이스의 직관적인 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Tiled Cylindrical Layout	Button	5.7894737	1.1342617
	Hybrid Motion	4.5789474	1.1697953
	Tilt Motion	5.2631579	1.0457377
Fisheye Cylindrical Layout	Button	5.4210526	1.0173926
	Hybrid Motion	4.7368421	1.5217718
	Tilt Motion	5.5263158	0.6966923

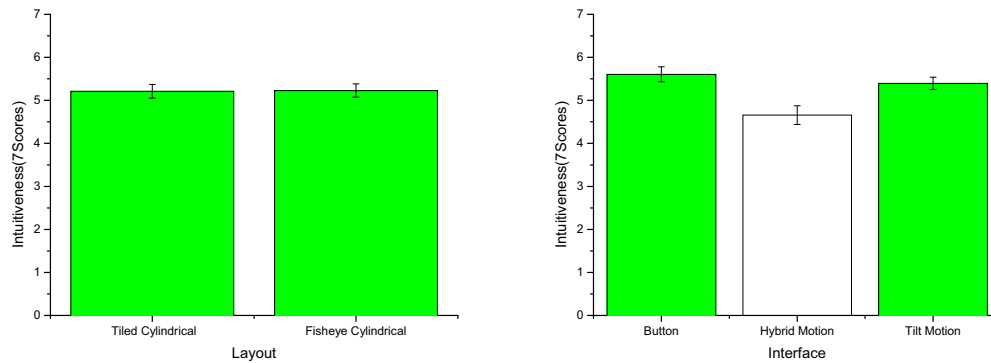
그림 4.11 인터페이스의 직관적인 정도

인터페이스의 직관적인 정도에 대한 설문 결과로는 **button interface**와 **tilt motion interface**가, **hybrid motion interface**보다 직관적인 것으로 나타났다. 이는 **hybrid motion interface**가 두 가지 방식의 인터페이스를 조합하였기 때문에, 직관적이지 못한 것으로 보인다.



Source	F value	P Value
Layout	$F(1,18) = 0.02$	0.8808
Interface	$F(2,36) = 5.19$	0.0104
Layout*Interface	$F(2,36) = 1.74$	0.1898

(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$

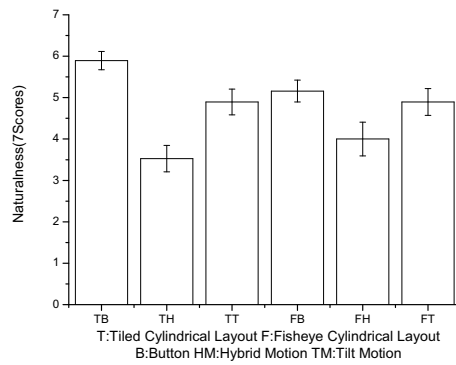


(B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.11 인터페이스의 직관적인 정도에 대한 분산 분석



3.3.5 인터페이스의 자연스러운 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Cylindrical Layout	Tiled Button	5.8947368	0.9365858
	Hybrid Motion	3.5263158	1.3485968
	Tilt Motion	4.8947368	1.3289401
Fisheye Cylindrical Layout	Button	5.1578947	1.1186876
	Hybrid Motion	4.0000000	1.7320508
	Tilt Motion	4.8947368	1.3701069

그림 4.12 인터페이스의 자연스러운 정도



인터페이스의 자연스러운 정도에 대한 설문 조사에서는 layout과 interface사이의 인터랙션이 존재하였다. Tiled cylindrical layout에서는 button interface, tilt motion interface, hybrid motion interface 순으로 자연스럽다는 답변을 얻었지만, fisheye cylindrical layout에선, button interface와 tilt motion interface가 hybrid motion interface보다 자연스럽다는 답변을 얻었다. 이 때, button interface와 tilt motion interface사이에는 유의한 차이가 없었다. 이 결과에서도 tiled cylindrical layout과 button interface의 조합이 높은 점수를 얻었다.



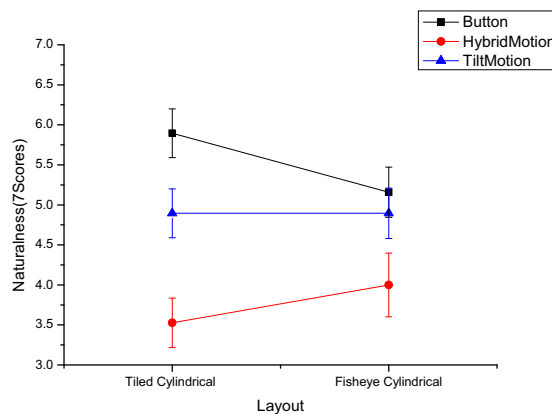
Source	F value	P Value
Layout	F(1,18) = 0.18	0.6765
Interface	F(2,36) = 15.74	<.0.0001
Layout*Interface	F(2,36) = 4.50	0.0180

(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$, Layout*Interface 인터랙션 존재

Source	Button		Hybrid Motion		Tilt Motion	
	F(1,18)	P Value	F(1,18)	P Value	F(1,18)	P Value
Layout	5.20	0.0349	3.70	0.0704	0.00	1.0000

Source	Tiled		Cylindrical	
	F(2,36)	P Value	F(2,36)	P Value
Interface	29.35	<.0001	3.90	0.0294

(B) 인터페이스의 자연스러운 정도에 대한 simple effect, 1 way-within subject ANOVA, (유의 수준 $p < 0.05$)

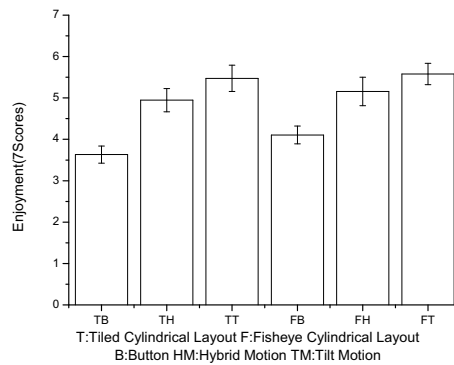


(C) 인터페이스의 자연스러운 정도에 존재하는 Layout과 Interface 사이의 Interaction

표 4.12 인터페이스의 자연스러운 정도에 대한 분산 분석



3.3.6 인터페이스의 재미있는 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Cylindrical Layout	Tiled Button	3.6315789	0.8950808
	Hybrid Motion	4.9473684	1.2235506
	Tilt Motion	5.4736842	1.3891813
Fisheye Cylindrical Layout	Button	4.1052632	0.9365858
	Hybrid Motion	5.1578947	1.5004873
	Tilt Motion	5.5789474	1.1212983

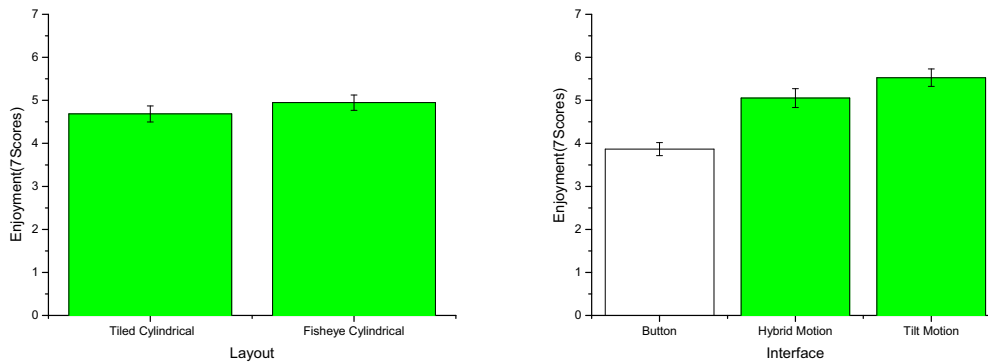
그림 4.13 인터페이스의 재미있는 정도

인터페이스의 재미있는 정도에 대한 설문에서는 모션 인터페이스들이 버튼 인터페이스 보다 재미있다는 결론을 얻었다. 실험자 중엔, ‘게임에 사용하면 좋겠다’ 라는 언급을 하는 사람도 있었다.



Source	F value	P Value
Layout	$F(1,18) = 2.24$	0.1518
Interface	$F(2,36) = 15.21$	<.0001
Layout*Interface	$F(2,36) = 0.59$	0.5612

(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$



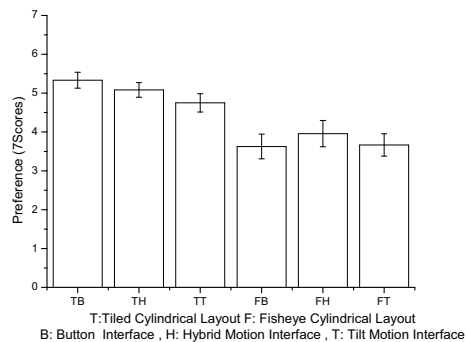
(B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.13 인터페이스의 재미있는 정도에 대한 분산 분석



3.4 레이아웃에 대한 설문 결과

3.4.1 레이아웃을 선호하는 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Tiled Cylindrical Layout	Button	5.333333	1.0072203
	Hybrid Motion	5.083333	0.9286112
	Tilt Motion	4.750000	1.1515585
Fisheye Cylindrical Layout	Button	3.625000	1.5551457
	Hybrid Motion	3.958333	1.6544845
	Tilt Motion	3.666667	1.4039282

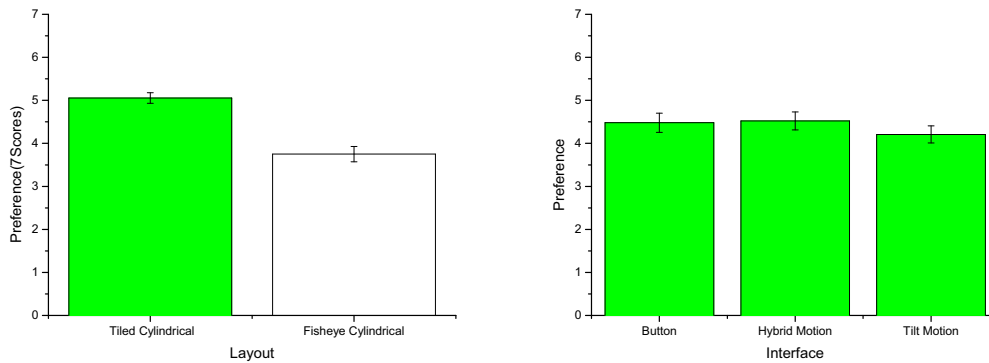
그림 4.14 레이아웃을 선호하는 정도

Tiled cylindrical layout보다 fisheye cylindrical layout을 선호하는 결과를 얻었다. 실험자들은 fisheye cylindrical layout을 ‘어지럽다’, ‘그림의 크기가 너무 작다’라고 언급하였다.



Source	F value	P Value
Layout	F(1,23) = 18.26	0.0003
Interface	F(2,46) = 1.80	0.1761
Layout*Interface	F(2,46) = 1.85	0.1685

(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$

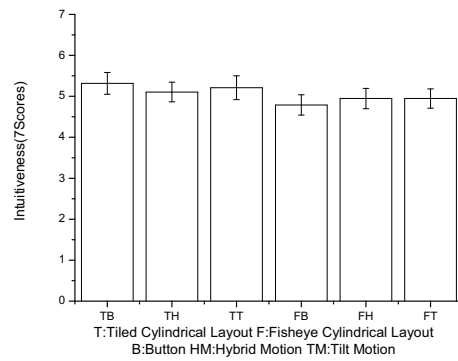


(B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.14 레이아웃을 선호하는 정도에 대한 분산 분석



3.4.2 레이아웃의 직관적인 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Cylindrical Layout	Tiled Button	5.3157895	1.1572300
	Hybrid Motion	5.1052632	1.0485300
	Tilt Motion	5.2105263	1.2727463
Cylindrical Layout	Fisheye Button	4.7894737	1.0841765
	Hybrid Motion	4.9473684	1.0787691
	Tilt Motion	4.9473684	1.0259784

그림 4.15 레이아웃의 직관적인 정도

레이아웃의 직관적인 정도에서는 두 레이아웃 사이에 차이가 없었다.



Source	F value	P Value
Layout	F(1,18) = 1.01	0.3275
Interface	F(2,36) = 0.19	0.8266
Layout*Interface	F(2,18) = 0.82	0.4476

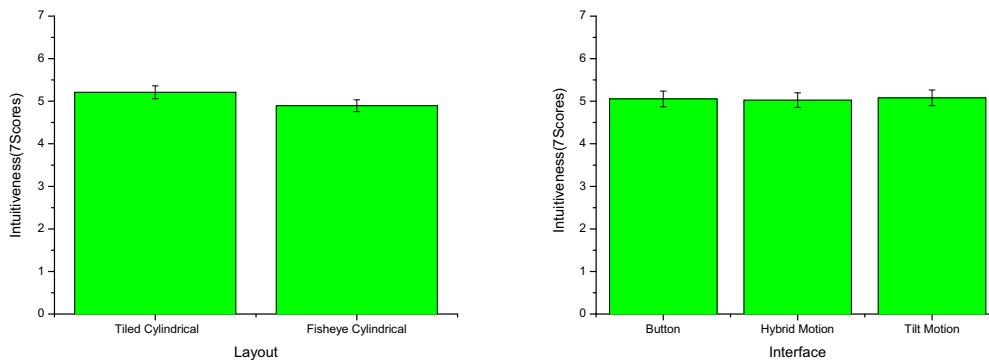
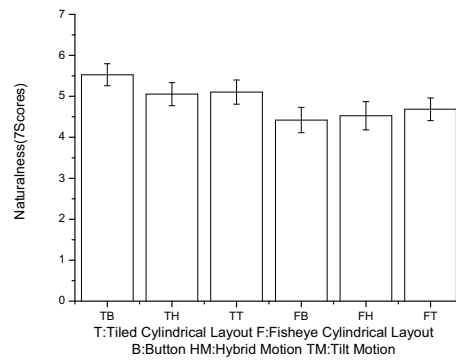
(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$ (B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.15 레이아웃의 직관적인 정도에 대한 분산 분석



3.4.3 레이아웃의 자연스러운 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Tiled Cylindrical Layout	Button	5.5263158	1.1722922
	Hybrid Motion	5.0526316	1.2235506
	Tilt Motion	5.1052632	1.2864567
Fisheye Cylindrical Layout	Button	4.4210526	1.3464269
	Hybrid Motion	4.5263158	1.5043796
	Tilt Motion	4.6842105	1.2042809

그림 4.16 레이아웃의 자연스러운 정도 그래프

레이아웃의 자연스러운 정도에 대한 설문에서도, 두 레이아웃 사이에 차이가 없었다.



Source	F value	P Value
Layout	$F(1,18) = 3.18$	0.0914
Interface	$F(2,36) = 0.97$	0.3879
Layout*Interface	$F(2,36) = 2.10$	0.1370

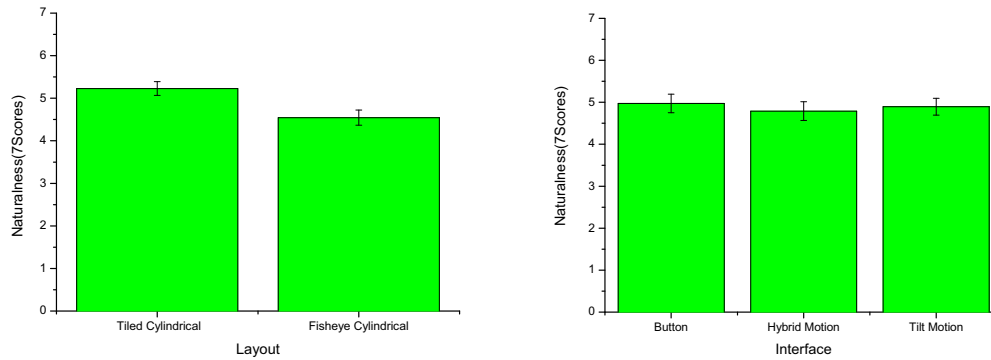
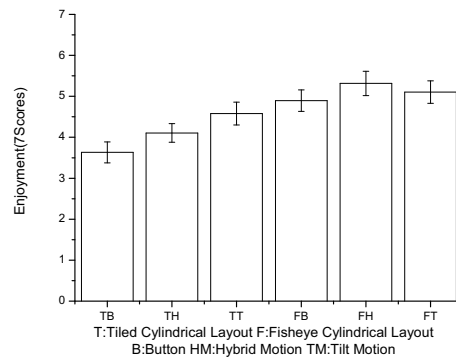
(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$ (B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.16 레이아웃의 자연스러운 정도에 대한 분산 분석



3.4.4 레이아웃의 재미있는 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Tiled	Button	3.6315789	1.1160708
Cylindrical Layout	Hybrid Motion	4.1052632	0.9941348
	Tilt Motion	4.5789474	1.2163602
Fisheye Cylindrical Layout	Button	4.8947368	1.1496249
	Hybrid Motion	5.3157895	1.2932574
	Tilt Motion	5.1052632	1.1969747

그림 4.17 레이아웃의 재미있는 정도

레이아웃의 재미있는 정도에 대한 설문에서는 layout과 interface사이의 인터렉션이 존재하였다. Button interface와 hybrid motion interface에서는 fisheye cylindrical layout이 더 재미있고, tiled motion interface가 덜 재미있는 결과를 보여주었지만, tilt motion interface를 이용할 때는 레이아웃의 영향이 없었다.



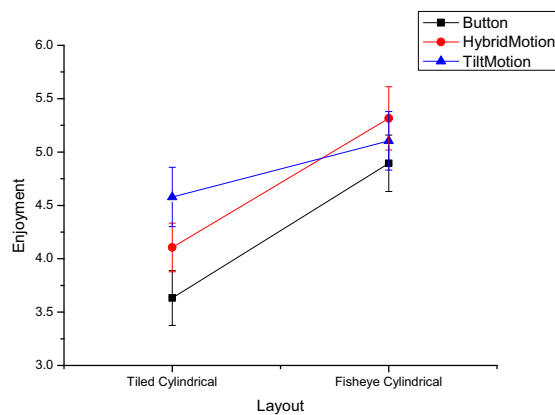
Source	F value	P Value
Layout	F(1,18) = 8.19	0.0104
Interface	F(2,36) = 9.00	0.0007
Layout*Interface	F(2,36) = 4.40	0.0195

(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$, Layout*Interface 인터랙션 존재

Source	Button F(1,18)	P Value	Hybrid Motion F(1,18)	P Value	Tilt Motion F(1,18)	P Value
Layout	12.49	0.0024	12.80	0.0022	1.38	0.2557

Source	Tiled Cylindrical F(2,36)	P Value	Fisheye Cylindrical F(2,36)	P Value
Interface	10.36	0.0003	2.46	0.0995

(B) 레이아웃의 재미있는 정도에 대한 simple effect, 1 way-within subject ANOVA, (유의 수준 $p < 0.05$)



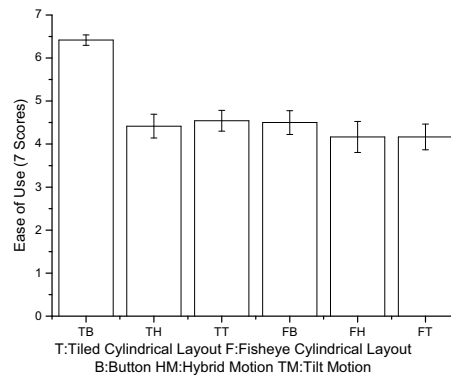
(C) 레이아웃의 재미있는 정도에 존재하는 Layout과 Interface 사이의 Interaction

표 4.17 레이아웃의 재미있는 정도에 대한 분산 분석



3.5 인터페이스와 레이아웃의 조합에 대한 설문 결과

3.5.1 인터페이스와 레이아웃의 조합이 사용하기 쉬운 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Tiled	Button	6.4166667	0.5835921
Cylindrical Layout	Hybrid Motion	4.4166667	1.3486440
	Tilt Motion	4.5416667	1.1787675
	Fisheye	Button	4.5000000
Cylindrical Layout	Hybrid Motion	4.1666667	1.7610932
	Tilt Motion	4.1666667	1.4645571

그림 4.18 인터페이스와 레이아웃의 조합이 사용하기 쉬운 정도

인터페이스와 레이아웃의 조합이 사용하기 쉬운 정도에 대한 분석에서는 tiled cylindrical layout과 button interface의 조합이 가장 사용하기 쉽다는 결론을 얻었다. 다른 모든 조합 보다 button interface와 tiled cylindrical layout의 조합이 가장 사용하기 쉽다는 결과를 얻었다. 나머지 조합들 사이에서는 유의한 차이가 없었다.



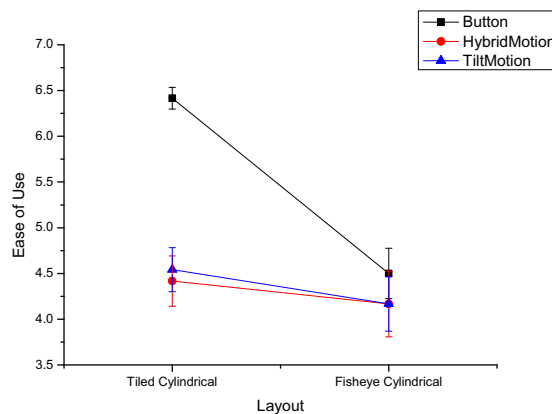
Source	F value	P Value
Layout	F(1,23) = 15.92	0.0006
Interface	F(2,46) = 16.03	<.0001
Layout*Interface	F(2,46) = 10.65	0.0002

(A) 2-Way within subject ANOVA, 유의 수준 $p < 0.05$, Layout*Interface Interaction 존재

Source	Button		Hybrid Motion		Tilt Motion	
	F(1,23)	P Value	F(1,23)	P Value	F(1,23)	P Value
Layout	42.39	<.0001	0.66	0.4259	1.22	0.2808

Source	Tiled Cylindrical		Fisheye Cylindrical	
	F(2,46)	P Value	F(2,46)	P Value
Interface	29.85	<.0001	0.71	0.4967

(B) 인터페이스와 레이아웃의 조합이 사용하기 쉬운 정도에 대한 simple effect, 1 way-within subject ANOVA, 유의 수준 $p < 0.05$

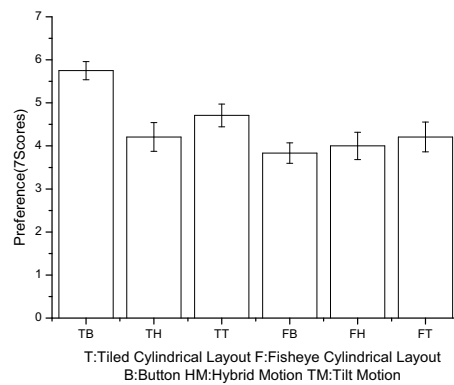


(C) 인터페이스와 레이아웃의 조합이 사용하기 쉬운 정도에 존재하는 Layout과 Interface 사이의 Interaction

표 4.18 인터페이스와 레이아웃의 조합이 사용하기 쉬운 정도에 대한 분산 분석



3.5.2 인터페이스와 레이아웃의 조합을 선호하는 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Tiled	Button	5.7500000	1.0320937
Cylindrical Layout	Hybrid Motion	4.2083333	1.6412924
	Tilt Motion	4.7083333	1.3014763
Fisheye	Button	3.8333333	1.1671842
Cylindrical Layout	Hybrid Motion	4.0000000	1.5603790
	Tilt Motion	4.2083333	1.6934443

그림 4.19 인터페이스와 레이아웃의 조합을 선호하는 정도 그래프

인터페이스와 레이아웃의 조합을 선호하는 정도에 대한 설문에서도 tiled cylindrical layout과 button interface의 조합을 가장 선호하였다. 나머지 조합들 사이에서는 유의한 차이가 없었다.



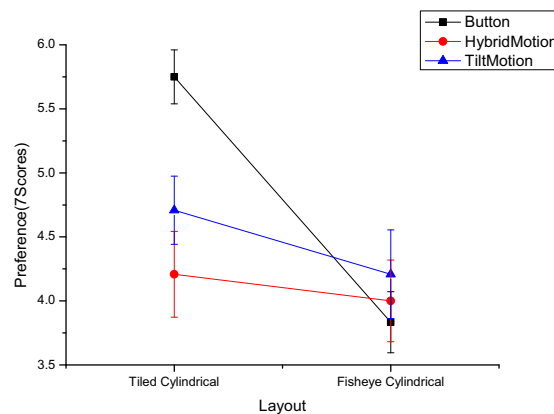
Source	F value	P Value
Layout	F(1,23) = 12.05	0.0021
Interface	F(2,46) = 3.48	0.0391
Layout*Interface	F(2,46) = 10.41	0.0002

(A) 2-Way within subject ANOVA, 유의 수준 $p < 0.05$, Layout*Interface Interaction 존재

Source	Button		Hybrid Motion		Tilt Motion	
	F(1,23)	P Value	F(1,23)	P Value	F(1,23)	P Value
Layout	29.04	<.0001	0.46	0.5039	1.92	0.1795

Source	Tiled Cylindrical		Fisheye Cylindrical	
	F(2,46)	P Value	F(2,46)	P Value
Interface	9.90	0.0003	0.77	0.4668

(B) 인터페이스와 레이아웃의 조합을 선호하는 정도에 대한 simple effect, 1 way-within subject ANOVA, 유의 수준 $p < 0.05$

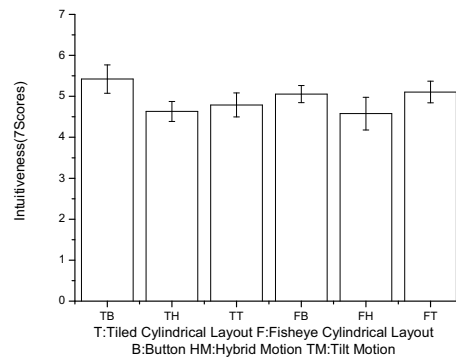


(C) 인터페이스와 레이아웃의 조합을 선호하는 정도에 존재하는 Layout과 Interface 사이의 Interaction

표 4.19 인터페이스와 레이아웃의 조합을 선호하는 정도에 대한 분산 분석



3.5.3 인터페이스와 레이아웃의 조합이 직관적인 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Tiled	Button	5.4210526	1.5024347
Cylindrical Layout	Hybrid Motion	4.6315789	1.0651305
	Tilt Motion	4.7894737	1.2727463
Fisheye	Button	5.0526316	0.9112680
Cylindrical Layout	Hybrid Motion	4.5789474	1.7421503
	Tilt Motion	5.1052632	1.1496249

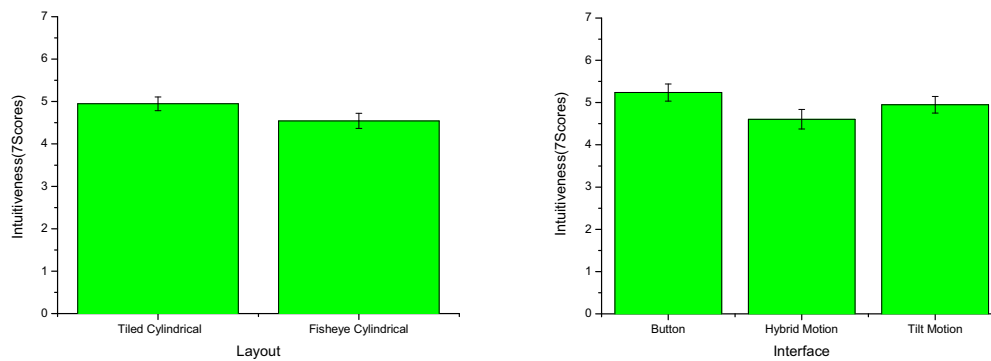
그림 4.20 인터페이스와 레이아웃의 조합이 직관적인 정도 그래프

인터페이스와 레이아웃의 조합이 직관적인 정도에 대한 설문에서는 모든 조합들 사이에서 유의한 차이가 발견되지 않았다.



Source	F value	P Value
Layout	F(1,18) = 0.02	0.8786
Interface	F(2,36) = 2.88	0.0689
Layout*Interface	F(2,36) = 1.28	0.2916

(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$

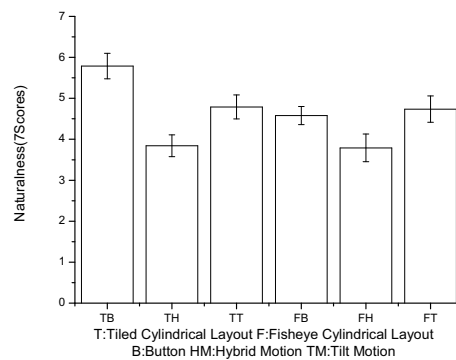


(B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.20 인터페이스와 레이아웃의 조합이 직관적인 정도에 대한 분산 분석



3.5.4 인터페이스와 레이아웃의 조합이 자연스러운 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Tiled	Button	5.7894737	1.3572418
Cylindrical Layout	Hybrid Motion	3.8421053	1.1672931
	Tilt Motion	4.7894737	1.2727463
Fisheye	Button	4.5789474	0.9612370
Cylindrical Layout	Hybrid Motion	3.7894737	1.4749368
	Tilt Motion	4.7368421	1.4079972

그림 4.21 인터페이스와 레이아웃의 조합이 자연스러운 정도 그래프

인터페이스와 레이아웃의 조합이 자연스러운 정도에 대한 설문에서도, tiled cylindrical layout과 button interface의 조합이 가장 자연스럽다는 답을 얻었다. Tiled cylindrical layout에서는 button interface, tilt motion interface, hybrid motion interface 순으로 자연스러운 정도의 차이가 있었으나, fisheye cylindrical layout에서는 button interface와 tilt motion interface가 비슷한 정도의 자연스러움을 보여주었고, hybrid motion interface가 가장 부자연스런 결과를 보여 주었다.



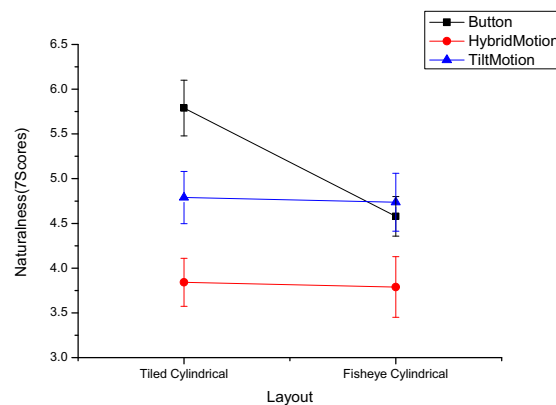
Source	F value	P Value
Layout	F(1,18) = 2.21	0.1546
Interface	F(2,36) = 16.00	<.0001
Layout*Interface	F(2,36) = 7.33	0.0021

(A) 2-Way within subject ANOVA, 유의 수준 $p < 0.05$, Layout*Interface Interaction 존재

Source	Button		Hybrid Motion		Tilt Motion	
	F(1,18)	P Value	F(1,18)	P Value	F(1,18)	P Value
Layout	11.10	0.0037	0.04	0.8534	0.02	0.9005

Source	Tiled Cylindrical		Fisheye Cylindrical	
	F(2,36)	P Value	F(2,36)	P Value
Interface	17.39	<.0001	6.90	0.0029

(B) 인터페이스와 레이아웃의 조합이 자연스러운 정도에 대한 simple effect, 1 way-within subject ANOVA, 유의 수준 $p < 0.05$

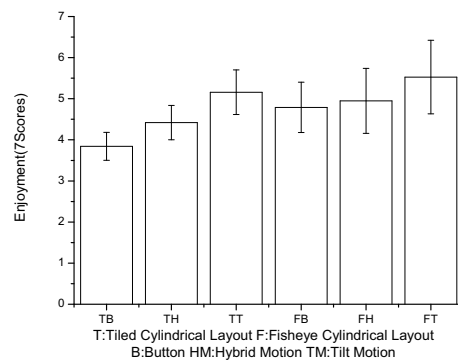


(C) 인터페이스와 레이아웃의 조합이 자연스러운 정도 존재하는 Layout과 Interface 사이의 Interaction

표 4.21 인터페이스와 레이아웃의 조합이 자연스러운 정도에 대한 분산 분석



3.5.5 인터페이스와 레이아웃의 조합이 재미있는 정도



Layout	Interface	평균값 (7scores)	표준편차 (7scores)
Cylindrical Layout	Button	3.8421053	1.2588865
	Hybrid Motion	4.4210526	1.3045131
	Tilt Motion	5.1578947	1.4629938
Cylindrical Layout	Button	4.7894737	1.0316625
	Hybrid Motion	4.9473684	1.4709666
	Tilt Motion	5.5263158	1.389181

그림 4.22 인터페이스와 레이아웃의 조합이 재미있는 정도 그래프

인터페이스와 레이아웃의 조합이 재미있는 정도에 대해서는 tiled cylindrical layout 보다 fisheye cylindrical layout이, button interface보다는 두 모션 인터페이스가 더 재미있다는 결론을 얻었다.



Source	F value	P Value
Layout	F(1,18) = 4.87	0.0406
Interface	F(2,36) = 6.40	0.0042
Layout*Interface	F(2,36) = 0.98	0.3856

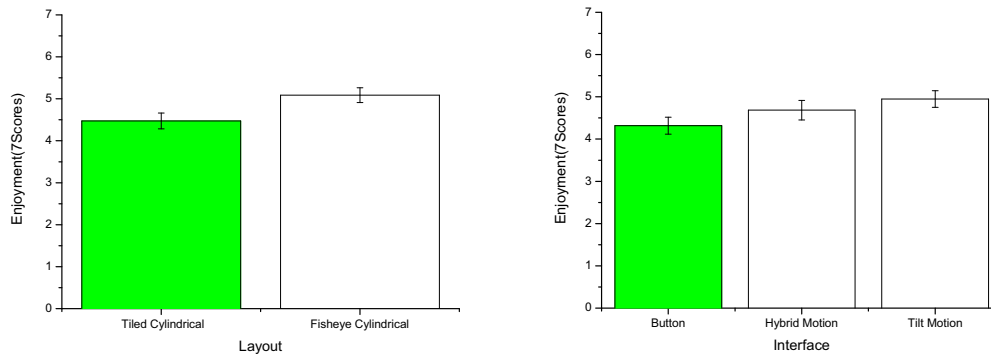
(A) 2-way within subject ANOVA, 유의 수준 $p < 0.05$ (B) SNK Test, 같은 그룹은 같은 색으로 표시, 유의 수준 $p < 0.05$

표 4.22 레이아웃의 자연스러운 정도에 대한 분산 분석



3.6 수행 시간에 대한 선형 회귀 분석

실험 중 트레이닝이 충분하지 못하여 트레이닝 이펙트가 발생했을 가능성에 대비해, 매 실험자의 결과를 4 trial 씩 짝을 지어서, 선형 회귀 분석을 하였다. 각각의 사용자 마다의 결과를 사용하였으며, 유의 수준은 $p < 0.05$ 로 두었을 때, 통계적으로 유의한 선형 경향성을 보인 결과의 수는 표4.23과 같다.

조합	TB	TH	TT	FB	FH	FT
위치를 알려주는 경우	2	6	2	2	1	2
위치를 알려주지 않는 경우	0	0	2	1	2	1

표 4.23 통계적으로 유의하게($p < 0.05$) 선형 경향성을 가진 실험자의 수

표4.23에 따르면, 총 24 명의 사용자 중, tiled cylindrical layout과 hybrid motion interface의 조합을 제외하곤, 2 명 이하의 결과 만이 유의한 선형 경향성을 보였다. 한편 tiled cylindrical layout과 hybrid motion interface의 조합의 경우에도 24 명 중 6 명의 결과 만이 유의한 선형 경향성을 보였다.

결과적으로 사용자의 수행 시간에서 트레이닝 이펙트는 발견되지 않았다. 단 이는 짧은 기간 동안의 트레이닝 이펙트이고, 날짜 단위 이상의 긴 트레이닝 이펙트가 존재할 가능성이 있다.

3.7 토론

결과적으로, button interface의 사용성이 motion interface들 보다 좋았으며, tiled layout이 fisheye layout보다 좋은 사용성의 결과를 얻었다. 많은 경우 실험자들은 button interface에서만 각 인터페이스에서 낼 수 있는 최고 속력에 가까운 이동을 하였고, motion interface에서는 좀더 보수적이고 천천히 움직이는 경향을 보였다. 이 점이 수행 시간에 상당한 영향을 끼쳤다고 본다. 이러한 경향은 motion interface의 해상도와 신뢰성에 기본적으로 문제가 있었기 때문으로 생각된다. 비록 모바일 장치의 휴대성을 위해서는 inertial sensing이 효과적이지만, 신뢰성에는 어느 정도 문제를 발생시키는 원인이 되었다고 본다. 좀더 신뢰성 있는 모션 센싱 기술을 도입할 필요성이 있다. 그 외에도, tilt motion interface의 경우 transfer function을 2nd order로 사용한 것 역시 사용이 어려움에 일조하여, 2nd order transfer



function 중 속력이 증가하지 않은 구간 만을 사용한 것 역시 문제가 되었다고 생각한다.

또한, **button interface**의 사용성과 성능이 더 좋은 결과를 얻은 것에 대한 다른 원인으로 Cooper[26]의 연구 결과에서 밝혀진 것 처럼 이미지를 보여 줄 때, **moving presentation**보다, **fixed presentation**이 더 효과적이기 때문으로 생각된다. **Button interface**의 경우 **fixed presentation**에 가깝지만, 두 종류의 **motion interface**는 **moving presentation**에 가깝다. 따라서 사용자의 이미지 인식이 좀더 쉬운 **button interface**가 더 좋은 결과를 얻을 수 있다. 모션 인터페이스에서도, **fixed presentation**에 가까운 **presentation** 방법을 사용할 수 있다면 좀더 효과적인 이미지 브라우저를 만들 수 있을 것이라고 생각한다.

한편 **fisheye layout**이 더 나쁜 결과를 얻은 것에 있어서는 주어진 태스크가, **overview**와 **detail**을 동시에 요구하는 태스크가 아니었던 것에 문제가 있는 것으로 생각된다. 특히 **tiled layout**에서 구분하기 적절한 정도의 **detail**을 제공하는 데 비해, **fisheye layout**은 중심부엔 **detail**을 많이 제공하였지만, 외곽으로 갈수록 **overview**를 보여주기 위해 과도한 **detail**의 생략을 한 점이 나쁜 사용성에 큰 영향을 준 것으로 보인다. 사용자들의 설문에서도, 외곽의 **detail**에 대한 불만이 발견되었다. **Fisheye layout**에 대한 좋은 결과를 얻었던, Gutwin[7]의 실험이 커다란 문서 하나에 대해서 이루어졌던 것을 고려해 볼 때, 다수의 이미지를 대상으로 한 것 보단, 큰 이미지 하나에 대해서 적용되었을 때 효과적일 수 있었을 것이라 생각된다. 또는 **fisheye layout**의 **parameter**를 좀더 적절하게 조절하여, 사용자에게 보여주는 **overview**와 **detail** 사이의 균형을 잘 조절하였으면, 좋은 결과를 얻을 수 있었을 것이라 생각된다. 또, **fisheye view**를 제공하는 방법에서도, 다른 형태의 알고리즘을 이용하여 제공하였다면 효과적일 가능성이 있다.

Interface와 **layout**사이의 인터랙션이 발견된 항목은 인터페이스의 선호, 인터페이스의 자연스러운 정도, 인터페이스와 레이아웃의 조합이 사용하기 쉬운 정도, 인터페이스와 레이아웃의 조합을 선호하는 정도, 인터페이스와 레이아웃의 조합이 자연스러운 정도의 다섯 항목에 달하였다. 그리고 이렇게 인터랙션이 발생한 경우 **button interface**와 **tiled layout**의 조합이 가장 좋은 점수를 얻는 조합이었다. 이 두 조건은 현재 일반적인 모바일 이미지 브라우징 시스템에서 사용되는 조합으로, 많



은 사용자는 현재의 설정을 더 좋아하는 보수적인 경향을 보였다. 결과들에서 보여지는 경향성 중에 하나는 **tiled layout**이 아닌 경우엔 **button interface**에 대한 선호도는 다른 인터페이스와 별 차이를 보이지 않았다. **Tiled layout**이 아니면서, 좀더 효과적인 **layout**이 존재한다면, **motion interface**가 좀더 선호 받을 수 있는 가능성을 보여 준다.

사용자로 부터 얻은 피드백 중 가장 많은 부분은 인터페이스의 무게에 대한 불만이었다. 대부분의 사용자들이 팔과 어깨 통증을 호소하였고, **motion interface**를 사용할 경우에 그에 대한 문제가 더 컸다. **motion interface**사용 시에 충분한 속력이 나오지 않은 것에는 인터페이스의 문제도 굉장히 크다고 생각된다. 이 부분은 현재의 경우엔 **CPU power**를 요구하는 **vision sensing**을 포기하지 않으면 달성할 수 없으리라 본다. 하지만 **mobile** 기기의 **computing power**는 계속 증가하고 있으므로, 가벼운 일반 핸드폰에서도 충분한 **computing power**를 얻을 수 있으면 가능하리라 본다.



제 5 장

결론 및 향후 계획

결론적으로 현재 다수의 이미지를 브라우징하기 위한 방법으로는 **tiled layout**과 **button interface**의 조합이 다른 인터페이스와 레이아웃의 조합보다 효과적이었다. 시도한 다른 조합들은 조합마다의 장점이 있었지만, **tiled layout**과 **button interface**의 성능과 선호도를 극복하기엔 역부족이었다. 2차원 이미지 브라우징에서는 **motion interface**보다 **button interface**가 더 효과적임을 보여 준다. 이는 모션 인터페이스를 통해 얻을 수 있는 자연스러운 인터랙션 메타포나, **proprioception**의 도움이 없었거나, 이들의 도움이 이미지 브라우징엔 효과적이지 못할 가능성을 시사한다. 특히 기울기를 이용한 인터페이스보다 **spatial**한 속성을 이용할 수 있는, 혼합형 인터페이스가 더 나쁜 결과를 보여준 것이 이들 가능성에 더 힘을 실어준다. 그리고, **tiled layout**이 **fisheye layout**에 비해 이번 실험에서는 나쁜 결과를 보여주었다. 이는 현재의 주어진 태스크가 **overview**보다는 **detail**이 중요한 태스크이기 때문으로 생각된다. 하지만, **fisheye layout**의 적절한 **parameter** 결정이 이루어 지지 않았을 가능성이 있으므로, 이 부분에 대해서는 확실한 결론을 내릴 수는 없다고 생각한다.

한편, 현재의 구현된 이미지 브라우저 및 인터페이스가 3차원 브라우징 환경이지만, 사용성 실험은 **zooming**도 제한된 순수한 2차원 환경에서 실행되었다. 이 부분은 2차원 이미지 브라우징에서는 **motion interface**가 충분히 효과를 발휘하지 못하



있을 가능성을 남겨두고 있다. 현재의 거의 2 차원에 가까운 레이아웃이 아니라, 2.5 내지는 3차원에 가까운 레이아웃에서 3 차원 **motion interface**를 사용한다면, 좀더 효과를 발휘할 가능성이 높다.

Motion interface 자체의 신뢰성과 해상도에도 문제가 있었다. 3 차원에서의 브라우징을 이루기 위해서는 신뢰성과 해상도가 높은 **3D motion interface**가 선행되어야 한다. 가능성이 있는 인터페이스 중에, **IR range sensor**가 있다. **IR range sensor**는 **inertial sensing**이 아니지만, 사용자와 기기 사이의 거리를 쫄 때 사용할 수 있다. 이렇게 얻은 사용자와 기기 사이의 거리는 **1D translation**으로 취급될 수 있다. **IR range sensor**는 신뢰성과 해상도가 상당히 높으므로, 모바일 기기를 위한 훌륭한 **3D motion interface**의 일부로 사용이 가능할 것이라 생각된다. 또한 **gyro sensor**의 경우도 에러가 누적되는 현상을 적절한 소프트웨어 적인 방법이나 **3 axis accelerometer**의 절대 값의 도움을 받아서 조절할 수 있다면, 효과적인 **3D motion interface**로 사용될 가능성이 있다. 이 **IR range sensor**와 **gyro sensor** 등의 새로운 **motion interface**를 이용하여, **3D layout** 환경에서의 사용성 평가를 해볼 필요가 있다.

하지만 이 모든 것에 우선해야 할 것은 이미지를 보여줄 때, 사람이 인지할 수 있는 적절한 **presentation time**을 제공할 수 있도록 하는 방법이다. Cooper[26]의 연구에서 밝혀진 것 처럼 **fixed presentation**이 더 효과적이므로, 이와 같은 효과를 **button interface**뿐만이 아니라 **motion interface**를 이용할 때도 제공할 수 있는 방법을 고려해 보아야 할 것이다.



부록 A

실험에 사용한 설문지



<설문 조사>

실험번호: _____ 실험순서: _____
이름: _____ 성별: _____ 나이(만): _____

- 1. 모바일 기기를 이용하여 이미지를 많이 (50 장 이상) 관리
(1) 한다. (2) 안한다.
- 2. 3 차원 모션을 이용하는 인터페이스를 사용해 본 적이 (예: 트랙커, 건축링 컨트롤러)
(1) 있다. (2) 없다.
- 3. 사용성, 배우기 쉬움, 선호도

여기서부터 는 7단계 중에 하나를 선택하시면 됩니다. 써진 글자 밑에 동그라미를 하시
면 됩니다.

예)



----- 매 실험이 끝날 때 마다 작성해 주세요 -----

그림 A.1 시작 설문



가. 인터페이스

(1) 현재의 인터페이스를 사용하기가



(2) 현재의 인터페이스의 사용법을 배우기가



(3) 현재의 인터페이스를 선호하는 정도가



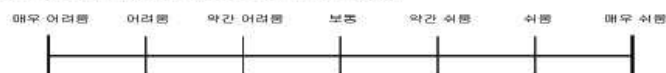
나. 레이아웃

(1) 현재의 레이아웃을 선호하는 정도가



다. 종합

(1) 현재의 인터페이스와 레이아웃의 조합을 사용하기가



(2) 현재의 인터페이스와 레이아웃의 조합을 선호하는 정도가



라. 현재의 조합에 대해서 하고 싶은 말씀

그림 A.2 매 조건 실험 후 설문1



(가) 인터페이스

(1) 현재의 인터페이스를 사용하기가 (____)

매우 직관적이지 못하다.	직관적이지 못하다.	조금 직관적이지 못하다.	그저 그렇다	약간 직관적이다.	직관적이다.	매우 직관적이다.
1	2	3	4	5	6	7

(2) 현재의 인터페이스를 사용하기가 (____)

매우 부자연스럽다.	부자연스럽다.	조금 부자연스럽다.	그저 그렇다.	조금 자연스럽다.	자연스럽다.	매우 자연스럽다.
1	2	3	4	5	6	7

(3) 현재의 인터페이스를 사용하기가 (____)

매우 재미없었다.	재미없었다.	조금 재미없었다.	그저 그렇다.	조금 재미있었다.	재미있었다.	매우 재미있었다.
1	2	3	4	5	6	7

(나) 레이아웃

(1) 현재의 레이아웃이 (____)

매우 직관적이지 못하다.	직관적이지 못하다.	조금 직관적이지 못하다.	그저 그렇다.	약간 직관적이다.	직관적이다.	매우 직관적이다..
1	2	3	4	5	6	7

(2) 현재의 레이아웃이 (____)

매우 부자연스럽다.	부자연스럽다.	조금 부자연스럽다.	그저 그렇다.	조금 자연스럽다.	자연스럽다.	매우 자연스럽다.
1	2	3	4	5	6	7

(3) 현재의 레이아웃 (____)

매우 재미없었다.	재미없었다.	조금 재미없었다.	그저 그렇다.	조금 재미있었다.	재미있었다.	매우 재미있었다.
1	2	3	4	5	6	7

그림 A.3 매 조건 실험 후 설문2



(다)인터페이스와 레이아웃의 조합

(1) 현재의 인터페이스와 레이아웃의 조합을 사용하기가 (_____)

매우 직관 적이지 못 하다.	직관적이지 못하다.	조금 직관 적이지 못 하다.	그저 그렇 다.	약간 직관 적이다.	직관적이 다.	매우 직관 적이다..
1	2	3	4	5	6	7

(2) 현재의 인터페이스와 레이아웃의 조합을 사용하기가 (_____)

매우 부자 연스럽다.	부자연스럽 다.	조금 부자 연스럽다.	그저 그렇 다.	조금 자연 스럽다.	자연스럽 다.	매우 자연 스럽다.
1	2	3	4	5	6	7

(3) 현재의 인터페이스와 레이아웃의 조합을 사용하기가 (_____)

매우 재미 없었다.	재미없었 다.	조금 재미 없었다.	그저 그렇 다.	조금 재미 있었다.	재미있었 다.	매우 재미 있었다.
1	2	3	4	5	6	7

그림 A.4 매 조건 실험 후 설문3



5. 실험에 대하여 하시고 싶은 말씀

감사합니다.

그림 A.5 전체 실험 후 설문



참고 문헌

- [1] Hwang, J., Jung, J., Kim, G.J.: Hand-held virtual reality: A feasibility study. In: ACM Symposium on Virtual Reality Software and Technology, ACM Press (2006)
- [2] Jones, M., Marsden, G.: Mobile Interaction Design. Jon Wiley & Sons, Ltd (2006)
- [3] Luo, J., Singhal, A., Braun, G., Gray, R.T., Seignol, O., Touchard, N.: Displaying images on mobile devices: capabilities, issues, and solutions. In: IEE International Conference on Image Processing. Volume 1., IEEE (2002) 13–16
- [4] Igarashi, T., Hinckley, K.: Speed-dependent automatic zooming for browsing large documents. In: ACM Symposium on User Interface Software and Technology, ACM Press (2000) 139–148
- [5] Patel, D., Marsden, G., Jones, M., Jones, S.: Improving photo searching interfaces for small-screen mobile computers. In: MobileHCI, ACM Press (2006) 149 – 156



- [6] Yee, K.P.: Peephole displays: Pen interaction on spatially aware handheld computers. In: CHI, ACM Press (2003) 1–8
- [7] Gutwin, C., Fedak, C.: Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques. In: Proceedings of Graphics Interface, Canadian Human-Computer Communications Society (2004) 145–152
- [8] Hakala, T., Lehtikoinen, J., Aaltonen, A.: Spatial interactive visualization on small screen. In: MobileHCI, ACM Press (2005) 137–144
- [9] Liu, H., Xie, X., Ma, W.Y., Zhang, H.J.: Automatic browsing of large pictures on mobile devices. In: International Multimedia Conference, ACM Press (2003) 148–155
- [10] Rekimoto, J.: Tilting operations for small screen interfaces. In: ACM Symposium on User Interface Software and Technology, ACM Press (1996) 167–168
- [11] Harrison, B.L., Fishkin, K.P., Gujar, A., Mochon, C., Want, R.: Squeeze me, hold me, tilt me! an exploration of manipulative user interfaces. In: CHI, ACM Press (1998) 17–24
- [12] Bartlett, J.F.: Rock’ n’ scroll is here to stay. *Computer Graphics and Applications*, IEEE **20**(3) (2000) 40–45
- [13] Hinckley, K., Pierce, J., Sinclair, M., Horvitz, E.: Sensing techniques for mobile interaction. In: ACM Symposium on User Interface Software and Technology, ACM Press (2000) 91–100
- [14] Manesis, T., Avouris, N.: Survey of position location techniques in mobile systems. In: MobileHCI, ACM Press (2005) 291–294



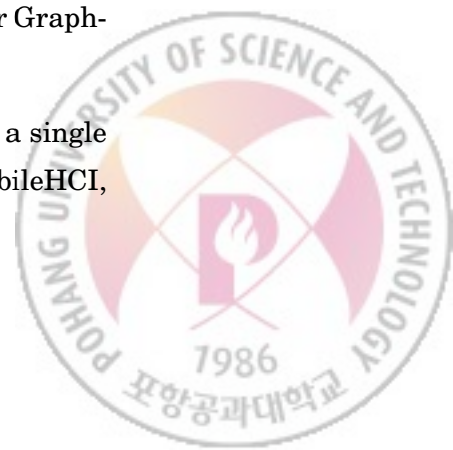
- [15] Patel, D., Marsden, G., Jones, S., Jones, M.: An evaluation of techniques for browsing photograph collections on small displays. In Brewster, S., Dunlop, M., eds.: *MobileHCI*, Glasgow, Scotland, Springer (2004) 132–143
- [16] Verplaetse, C.: Inertial proprioceptive devices: Self-motion sensing toys and tools. *IBM SYSTEMS JOURNAL* **35** (1996)
- [17] Crossan, A., Murray-Smith, R.: Variability in wrist-tilt accelerometer based gesture interfaces. In Brewster, S., Dunlop, M., eds.: *Mobile HCI*, Springer (2004) 144–155
- [18] Eslambolchilar, P., Murray-Smith, R.: Tilt-based automatic zooming and scaling in mobile devices - a state-space implementation. In Brewster, S., Dunlop, M., eds.: *Mobile HCI*, Springer (2004) 120–131
- [19] Marsden, G., Tip, N.: Navigation control for mobile virtual environments. In: *MobileHCI*, ACM Press (2005) 279–282
- [20] Combs, T.T.A., Bederson, B.B.: Does zooming improve image browsing? In: *International Conference on Digital Libraries*, ACM Press (1999) 130–137
- [21] Hornbæk, K., Bederson, B.B., Plaisant, C.: Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Transactions on Computer-Human Interaction* **9**(4) (2002) 362–389
- [22] Plaisant, C., Carr, D., Shneiderman, B.: Image-browser taxonomy and guidelines for designers. *IEEE Software* **12**(2) (1995) 21–322
- [23] Hwang, J., Kim, G.J., Kim, N.: Camera based relative motion tracking for hand-held virtual reality. In: *NICOGRAPH International*. (2006)
- [24] Hinckley, K.: Input technologies and techniques. In Jacko, J.A., Sears, A., eds.: *Handbook of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc. (2002)



- [25] Hwang, J., Jung, J., Yim, S., Cheon, J., Lee, S., Choi, S., Kim, G.J.: Requirements, implementation and applications of hand-held virtual reality. *Journal of Virtual Reality* 5(2) (2006) 59–66
- [26] Cooper, K., Bruijn, O.d., Spence, R., Witkowski, M.: A comparison of static and moving presentation modes for image collections. In: *Advanced Visual Interfaces*, ACM Press (2006) 381–388
- [27] Ballagas, R., Rohs, M., Sheridan, J.G.: Sweep and point & shoot: Phonecam-based interactions for large public displays. In: *CHI*, ACM Press (2005) 1200–1203
- [28] Cockburn, A., McKenzie, B.: Evaluation the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In: *CHI*, ACM Press (2002) 203–210
- [29] Fun, X., Xie, X., Mu, W.Y., Zhnng, H., Zhou, H.Q.: Visual attention based image browsing on mobile devices. In: *ICME*. Volume 1., IEEE (2003) 53–56
- [30] Hachet, M., Pouderoux, J., Guitton, P.: A camera-based interface for interaction with mobile handheld computers. In: *Symposium on Interactive 3D Graphics and Games. I3D*, ACM Press (2005) 65–72
- [31] Herndon, K.P., Dam, A.v., Gleicher, M.: The challenges of 3d interaction. In: *CHI*, ACM Press (1994) 36–43
- [32] Hinckley, K., Pausch, R., Goble, J.C., Kassell, N.F.: A survey of design issues in spatial input. In: *UIST*, ACM Press (1994) 213–222
- [33] Hinckley, K., Tullio, J., Pausch, R.: Usability analysis of 3d rotation techniques. In: *ACM Symposium on User Interface Software and Technology*, ACM Press (1997) 1–10



- [34] Horvitz, E.: Principles of mixed-initiative user interfaces. In: CHI, ACM Press (1999) 159–166
- [35] Jellinek, H.D., Card, S.K.: Powermice and user performance. In: CHI, ACM Press (1990) 213–220
- [36] Karstens, B., Rosenbaum, R., Schumann, H.: Information presentation on mobile handhelds. In: Information Resources Management Association. (2003)
- [37] Kim, W.S., Tendick, F., Ellis, S.R., Stark, L.W.: A comparison of position and rate control for telemanipulations with consideration of manipulator system dynamics. *IEEE JOURNAL OF ROBOTICS AND AUTOMATION* **3**(5) (1987) 426–436
- [38] Kjeldskov, J., Kolbe, N.: Interaction design for handheld computers. In: Asia-Pacific Conference on Computer-Human Interaction, Science Press. (2003)
- [39] Mantyjarvi, J., Patern Chang, F., Salvador, Z., Santoro, C.: Scan and tilt - towards natural interaction for mobile museum guides. In: MobileHCI, ACM Press (2006) 191–194
- [40] Mine, M.R., Brooks, F.P., Sequin, C.H.: Moving objects in space: exploiting proprioception in virtual-environment interaction. In: International Conference on Computer Graphics and Interactive Techniques, ACM Press/Addison-Wesley Publishing Co (1997) 19–26
- [41] Perlin, K., Fox, D.: Pad. In: International Conference on Computer Graphics and Interactive Techniques, ACM Press (1993) 57–64
- [42] Yi, J.S., Choi, Y.S., Jacko, J.A., Sears, A.: Context awareness via a single device-attached accelerometer during mobile computing. In: MobileHCI, ACM Press (2005) 303–306



감사익글

어느덧 대학원에서 지낸 지도 2년이 지났습니다. 2년 전에 기대했던 제 모습에 지금 얼마나 다가갔는지 생각해 보면 많은 아쉬움이 남습니다. 그렇지만, 2년 전의 저와 비교하면, 조금이라도 나아진 모습이 있을 것이라고, 연구실에서 행했던 모든 일이 앞으로의 길에 많은 도움이 되리라고 믿습니다.

먼저 언제나 응원을 아끼지 않으시고, 지금까지 건강하게 키워 주신 부모님께 감사드립니다. 부모님의 응원과 믿음이 있었기에 지금의 제가 있을 수 있을 것입니다.

그리고, 지도 교수님이신 최승문 교수님께 감사드립니다. 제가 행한 수많은 시행착오를 하나 하나 교정해 주시며, 좀 더 나은 방향으로 향하도록 조언을 아끼지 않으신 것에 진심으로 감사드립니다.

비록 먼 서울에 계셨지만 마지막까지 걱정해 주시고 조언을 아끼지 않으신 김정현 교수님께도 감사드립니다. 그리고 제 논문의 미흡한 부분을 쪽집게 처럼 짚어주셔서, 앞으로의 방향을 잡아주신 한성호 교수님 께도 감사드립니다.

2년 동안 함께 했던 연구실 식구들에게도 감사드립니다. 많은 도움과 조언을 주셨던 상윤 형, 진석 형, 남규 형, 성길 형, 진욱 형, 종현 형, 석희 형, 광훈 형, 그리고 재인 형에게 감사 드립니다. 재인 형의 도움이 없었다면, 마지막 까지 논문을 만들어 낼 수 없었을 것입니다. 2년 전에 같이 연구실 생활을 시작한 희노애락을 함께한 동기들, 재훈, 재형, 이제는 옆 그래픽스 랩에 간 용진에게도 감사의 마음을 전합니다. 항상 재미있는 후배 채현 그리고, 앞으로 긴 시간을 함께 볼 후배들, 갑종, 인, 인욱, 건혁에게도 감사의 뜻을 전합니다.

지금은 그저 작은 산을 넘었을 뿐입니다. 앞으로 좀 더 길고 어려운 나날들이 앞에 있을 것입니다. 하지만, 현재까지의 경험을 밑거름 삼아, 만족할 수 있는 삶을 살 수 있다고 믿고, 그 것을 이루도록 노력하겠습니다.



Curriculum Vitae

Name : Sunghoon Yim
Date of Birth : 1981. 09. 14.
Present Address : 강원도 삼척시 성내동
10통 4반 80-24 번지

Education

2000. 3 ~ 2005. 2 : B.S. in 화학, 컴퓨터 공학 복수 전공, 포항공과대학교
2005. 3 ~ 2007. 2 : M.S. in 정보통신학과, 포항공과대학교
Thesis Title :
모바일 장치를 위한 동작 추적형 이미지 브라우징 시스템
(Image Browsing in Mobile Devices Using User Motion Tracking)
Advisor: Prof. 최승문



Publications

- International Journal

1. Hwang, J., Jung, J., **Yim , S.**, Cheon, J., Lee, S., Choi, S., Kim, G.J.:, "Requirements, Implementation and Applications of Hand-held Virtual Reality" *Journal of Virtual Reality* 5(2), 59.66, 2006.

- International Conference

1. Sangyoon Lee, **Sunghoon Yim**, Gerard Kim, Ungyeon Yang:, "Design and Evaluation of a Hybrid Display System for Motion-Following Tasks" *HCI International*, 2007. (Accepted)

- Domestic Conference

1. **임성훈**, 황재인, 최승문, 김정현:, "모바일 장치를 위한 동작 추적형 이미지 브라우징 시스템" *한국 HCI 학술대회*, 2007. (Accepted)

