
모바일 장치를 위한 동작 추적형 이미지 브라우징 시스템

Image Browsing in Mobile Devices Using User Motion Tracking

임성훈, Sunghoon Yim*, 황재인, Jane Hwang*, 최승문 Seungmoon Choi*, 김정현 Gerard Jounghyun Kim**

*포항공과대학교 컴퓨터공학과 가상현실 연구실, **고려대학교 컴퓨터학과

요약 현재 대부분의 모바일 장치들에 디지털 카메라가 설치되어 있으며 거대한 양의 이미지 데이터들을 저장할 수 있다. 이렇게 많은 양의 이미지가 저장되어 있을 경우 장치 속의 이미지들에 대한 브라우징을 하기 어려워지며, 그에 걸리는 시간도 증가하게 된다. 특히 모바일 장치의 경우 화면의 크기가 작으며, 일반 컴퓨터와 비교하여 부자연스럽고, 불편한 인터페이스를 갖고 있어 어려움을 더욱 증가시킨다. 우리는 이를 해결하기 위해 3차원 가시화 방법과 모션 센싱을 이용한 인터페이스를 제안하고, 제안된 가시화 방법과 인터페이스의 조합을 통해 모바일 장치에서의 효과적인 이미지 브라우징 방법을 모색해 보았다.

핵심어: HCI, UI, 모바일, 이미지 브라우저, 동작 추적 인터페이스

1. 서론

현재, 모바일 장치들이 발전하고 대중화되면서 이미지들의 취득 및 관리 형태도 새로워지고 있다. 사진을 찍기 위해 제작된 전문적인 디지털 카메라 이외에도 휴대폰, PDA 등의 다양한 장치들에도 디지털 카메라가 설치되면서 언제 어디서나 사진을 찍을 수 있게 되고, 이를 통해 수많은 양의 사진들이 모바일 장치들에 저장되어 있게 된다. 그러나 모바일 장치에 저장되는 사진의 양은 증가되고 있지만, 그를 효과적으로 다루기 위한 브라우징 방식은 그다지 발전하지 못하였다. 모바일 장치는 작은 화면과, 빈약한 인터페이스, 제한된 컴퓨팅 파워의 제한을 갖고 있다. 이 작은 화면과 빈약한 인터페이스로는 방대한 양의 사진을 브라우징 하기 쉽지 않다. 이러한 문제를 해결 하기 위해 본 논문에서 제시하는 방법은 3차원을 이용한 이미지의 배치 및 가시화와 동작인식을 이용한 인터페이스이다. 장치의 움직임에 이용한 인터페이스의 경우 매우 직관적이고 자연스럽게, 사용할 수 있으며, 연속적인 형태의 입력이 가능하다. 특히 현실에서의 움직임과 유사한 메타포를 갖는 인터랙션이 가능하므로, 이의 도움을 얻어 효과적인 브라우징 인터페이스가 될 수 있다고 생각하였다.

2. 관련 연구

현재 적절한 레이아웃과 인터페이스를 이용하여, 작은 화면의 한계를 극복하기 위한 다양한 연구들이 있다. Gutwin[1]은 큰 인터페이스를 작은 화면의 모바일 기기에서 다루기 위한 효과적인 방법을 찾기 위해 다양한 인터페이스와 레이아웃을 비교하였다. Patel[2]은 작은 화면을 가진 모

바일 장치에서의 효과적인 이미지 브라우징을 위해 SDAZ를 응용한 새로운 인터페이스 Auto Zoom과 Gesture Zoom을 제안하였다. Hakala[3]는 spatial cognition을 이용할 수 있는 3차원 가시화 방법을 통해 효과적인 데이터의 전달을 꾀하였다.

동작 추적을 모바일 기기의 인터페이스로 이용하고자 연구 역시 계속되고 있다. Yee[4]는 Peephole display를 제안하였다. Peephole display는 동작 추적을 이용하여, pen interaction시, 작업 공간이 모바일 기기의 화면 크기로 제한되는 것을 극복하였다. 한편 Verplaetse[5]는 외부의 도움 없이 장치 스스로 동작을 추적하기 위해서는 inertial sensing이 필수적이라는 결론을 내리고, inertial sensor를 구현하기 위해 필요한 기술들을 서술하였다. 그에 따르면, self motion sensing을 위한 기술로, 가속도계와 gyroscopes를 들었다. Rekimoto[6]가 기울임 인식을 입력 장치로 이용하는 것을 제안한 이후, 기울임 센서를 이용한 동작 추적을 행한 다양한 연구[7-12]가 있다.

한편 Hwang[13]은 3축 가속도 센서와 비전 알고리즘을 이용한 동작 인식 알고리즘을 이용하여, 4 자유도 동작 인식을 실현하였다. 모바일 기기에서의 가상 현실에 이 알고리즘을 이용하였다.

이미지 브라우저는 여러 개의 이미지로부터 유저가 하나 또는 그 이상의 이미지를 선택할 수 있도록 하는 프로그램이다. 이 이미지 브라우저는 한번에 여러 장의 이미지를 보여줄 수 있어야 하며, 원래의 full resolution image 또한 보여줄 수 있어야 한다[14]. 이 조건을 Zoomable user interface(ZUI)를 이용하면, 쉽게 달성할 수 있다. Zoomable user interface는 information object가, space와 scale로 정

리 되어 있으며, 유저는 interface space에 있는 객체를 zooming 또는 panning을 통하여 직접적으로 상호작용 할 수 있는 형태의 인터페이스이다[15]. Panning은 현재 보이는 information space의 area를 변경 시키고, zooming은 현재 보이는 information space의 scale을 변경시킨다.

3. 동작 추적형 인터페이스

3.1 모바일 장치에서의 동작 추적

모바일 장치에서 동작 인식을 위해 사용하는 센서들은 external reference 없이 self-contained system이어야 할 필요가 있다. 이 조건을 충족하여 모바일 기기에서 사용할 수 있는 센서들을 표1에 정리하였다.

표 1. 자체 모션 인식이 가능한 센서들

센서의 종류	Translation	Rotation
3 DOF Accelerometer	Relative X,Y,Z	
3 DOF Accelerometer (as tilt sensor)		Absolute roll, pitch
Camera (Hwang[16]'s method)	Relative Z	Relative yaw, pitch
Angular-rate-sensing Gyroscope		Relative yaw, pitch, roll

가속도 계의 경우 3축의 가속도를 얻을 수 있으므로, 이를 적분해서, 속력과 위치를 얻을 수 있다. 하지만, 적분을 통해서 얻어야 하므로 오류가 많고, 짧은 시간의 체크치 정도만 인식이 가능하다. 대부분 모바일 기기에서 가속도 계를 사용할 경우엔, 가속도 계가 중력 가속도를 기준으로 삼는다는 점을 이용하여, 기울기 센서로 사용하였다.

카메라는 optical flow와 같은 비전 기술을 이용, 얻어진 영상을 처리하여 상대적인 위치를 센싱 할 수 있다. Hwang의 방법에서는 특징 점의 상대적인 위치를 이용하여, 전 후 움직임도 인식할 수 있었다.

자이로스코프의 경우엔 가속도 계와 카메라 보다 상대적으로 비싸지만 3 축의 회전을 전부 얻어 낼 수 있는 장점을 갖고 있다. 다만 가속도 계를 기울기 센서로 사용할 때와 달리 상대적으로 기울임을 인식하므로, 에러가 누적될 수 있다.

각각의 센서와 알고리즘은 서로 장, 단점을 갖고 있고, 인식할 수 있는 범위가 조금씩 차이가 있다. 때문에 가속도 계와 카메라를 병용하는 방법[13]을 이용하였다. 이 방법의 경우 3 차원이 인식 가능한 4 자유도의 움직임을 인식할 수 있

기 때문이다.

3.2 모바일 장치의 위치 추적을 이용한 인터페이스(Position Tracking Interface)

Hwang[13]의 위치 추적 방법을 사용할 경우 pitch, roll, yaw와 1D translation의 4 자유도 움직임을 얻을 수 있다. 이 4 자유도 움직임은 모바일 장치의 3 차원 공간에서의 상대적인 위치를 얻어 내기에 충분하다. 사용자가 화면을 수직으로 바라보면서 움직인다고 가정하면, 사용자의 움직임은 그림 1과 같이 유추 할 수 있다.

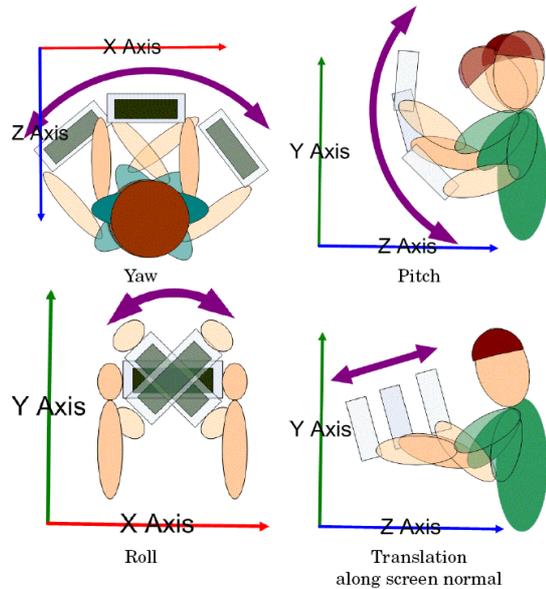


그림 1. 사용자의 4 자유도 움직임

이렇게 얻어진 사용자의 움직임을 이미지 브라우저의 기능과 대응 시킬 수 있다. 일반적인 2D 이미지 브라우저라면 yaw를 수직 방향의 panning에 pitch를 수평 방향의 panning에 대응시킬 수 있다. 한편 얻어진 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, pitch의 경우엔 사용자를 축으로 하고 사용자와 스크린 사이의 거리를 반지름으로 하는 지면과 수직인 호를 그리는 움직임으로 대응할 수 있다. 스크린 방향의 움직임은 각각 zoom 또는 스크린 방향의 움직임 자체로 대응시킬 수 있다. 표2에 다시 정리하였다.

이렇게 현재의 추적된 상대 위치와 가상의 상대 위치를 그대로 대응시켜 3D 브라우저를 하면, 유저의 움직임에 그대로 따르므로, 자연스럽게 직관적인 인터페이스가 될 수 있다고 생각하였다. 특히, 이미지의 위치를 실제 공간과 연관 지어서 spatial cognition과 proprioception의 도움[17]을 얻을 수 있을 것이라 예상하였다.

표 2. 모바일 장치의 위치 추적을 이용한 인터페이스의 사용자의 움직임과 브라우저 기능의 대응

입력	2D 브라우저 기능	3D 네비게이션 기능
Yaw	1D 수평 panning	1D 수평 호
Pitch	1D 수직 panning	1D 수직 호
Translation along screen normal	Zooming	Translation along view direction

하지만 이 경우엔 현실의 workspace가 사용자의 팔 길이에 제한되는 문제가 있다. 브라우징 해야 하는 이미지의 수에 따라 가상의 workspace 크기는 무제한에 가깝게 증가할 수 있으므로, 가상의 넓은 workspace와 현실의 제한된 움직임 사이의 불일치가 발생한다. 이 경우 현실의 움직임과 가상의 움직임 사이의 gain을 증가시키게 되는데, gain의 값은 사용자의 움직임의 해상도에 제한이 된다. 이를 해결 할 수 있는 방법 중에 clutching이 있다.

Clutching은 일반적인 2D mouse의 움직임과 같은 방식이다. 마우스를 사용하다가, workspace의 끝에 다다르면, 마우스를 들어서(declutching) workspace의 가운데로 옮긴 다음에, 내려 놓는(clutching) 것과 같이, 입력을 잠시 declutching하고 workspace를 옮긴 다음 clutching하는 방법이다. 하지만 이 방법에는 clutching으로 인한 시간의 소모가 있으므로 workspace의 크기에 제한을 받지 않는 velocity control 방식으로 변경하는 것이 가장 근본적인 해결책이 된다. 대신 velocity control 방식일 때는 에러의 영향이 position control 보다 증폭 되므로, 에러가 누적되지 않는 인식 방법을 사용하는 것이 효과적이다.

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

모바일 장치의 기울기 추적을 이용한 인터페이스는 가속도계의 절대적인 인식을 이용하였다. 중립 위치로부터의 기울기에 비례하여, 브라우저의 panning velocity를 조정하는 형태가 된다. 일반적인 2D 이미지 브라우저의 경우엔 중립 위치로부터의 roll의 차이는 수평 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 제한 된다면, 실질적으로 무한대의 workspace를 제어할 수 있게 된다. 표 3에 인터페이스를 정리하였다.

구현 시엔 기울기 차이의 제공에 비례하여 속력을 변화시켰다. Pilot test의 결과 workspace의 제한을 극복할 수 있었지만, 세밀한 제어에서 문제를 일으켰으며, 사용자의 spatial cognition을 이용할 수 없었다.

표 3. 모바일 장치의 기울기 추적을 이용한 인터페이스의 사용자의 움직임과 브라우저 기능의 대응

입력	2D 브라우저 기능	3D 네비게이션 기능
중심 값으로부터 Pitch의 차이	1D 수직 panning의 속도	1D 수직 호를 따르는 움직임의 속도
중심 값으로부터 Roll의 차이	1D 수평 panning의 속도	1D 수평 호를 따르는 움직임의 속도
Translation along screen normal	Zooming	Translation along view direction

3.4 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 경우엔 문제가 발생하였다. 이 둘의 장점을 혼합할 수 있다면, 좀더 효과적인 인터페이스가 될 수 있다.

'모바일 장치의 위치 추적을 이용한 인터페이스'를 이용할 때는 roll을 사용하지 않는다. 한편 사용자의 움직임의 경우에도 장치의 무게 때문에, 좌우보다 상하 기울임의 정도가 제한되는 경향이 있었다. 한편, 현재의 카메라를 이용한 pitch sensing의 경우엔 translation과 rotation을 정확히 구분 못하는 특성이 있다. 따라서 실제로 많이 기울이지 않더라도 pitch값을 입력 받을 수 있다.

표 4. 위치, 기울기 혼성 추적 인터페이스의 사용자의 움직임과 브라우저 기능의 대응

입력	2D 브라우저 기능	3D 네비게이션 기능
Yaw	1D 수평 panning	1D 수평 호
Pitch	1D 수직 panning	1D 수직 호
중심 값으로부터 Pitch의 차이	1D 수직 panning의 속도	1D 수직 호를 따르는 움직임의 속도
중심 값으로부터 Roll의 차이	1D 수평 panning의 속도	1D 수평 호를 따르는 움직임의 속도
Translation along screen normal	Zooming	Translation along view direction

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 경

우엔 위치 인터페이스와 같은 방식으로 작동하고, roll 또는 pitch가 threshold값을 넘었을 경우에는 위치 추적을 중단하고, 기울기 추적 방식으로 작동한다.

4. 모바일 기기에서의 다수의 이미지 배치

4.1 격자 원기둥 레이아웃(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 view를 사용하는 3D navigation을 이용할 경우에는 zoom이 가능한 2D image browser와 기능적으로 차이가 없다. 그림 1에서 볼 수 있는 모션 인터페이스의 움직임을 고려해 볼 때, 비교적 효과적인 이미지 배치는 사용자들 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 pitch의 변화는 크지 않으므로, 구 대신 원기둥을 사용함으로써 발생하는 왜곡 역시 크지 않다. 그림은 인터페이스의 움직임과 원기둥 배치 사이의 관계를 보여 준다.

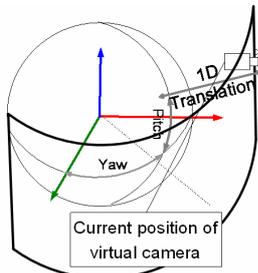


그림 2. 사용자의 움직임과 원기둥 배치

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



그림 3. 격자 원기둥 배치의 삽화(좌)와 실제 실행화면(우)

4.2 피쉬아이 원기둥 레이아웃 (Fisheye Cylindrical Layout)

Gutwin[1]의 결과에 따르면, 작은 화면을 가진 모바일 기기에서 큰 문서를 사용하는 경우 fisheye view의 사용은 상당히 효과적이었다. 이것이 다수의 이미지를 사용하는 경우

에도 적용이 가능할 것이라 생각되었다. Fisheye view는 이미지를 왜곡 시켜, 가운데는 강조가 되어 크게 보이고, 주변부는 작게 보이는 형태의 view이다. 이를 레이아웃에 적용을 한다면, 현재의 view에서 중심에 가장 가까운 이미지는 크기를 증가 시키고, 주변부로 갈수록 이미지의 크기를 감소시키는 형태가 된다. 또한 가상의 카메라의 view가 업데이트 될 때 마다, 레이아웃도 업데이트 되어야 한다.

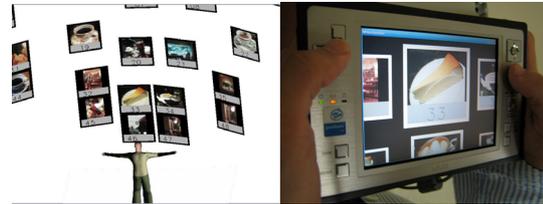


그림 4. 피쉬아이(fisheye) 원기둥 배치의 삽화(좌)와 실제 실행화면(우)

5. 인터페이스와 레이아웃 간의 사용성 실험

5.1 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.

5.2 실험 설계

5.2.1 피실험자

24 명의 피실험자가 실험에 참여하였다. 피실험자 중 18명은 남성, 6명은 여성이었다. 피실험자는 모두 학생이었으며, 나이 분포는 만 19세에서 25 세 사이였다. 피실험자 중, 모바일 기기를 이용하여 이미지를 관리하는 인원은 15 명, 모션 인터페이스를 사용한 경험이 있는 인원은 6 명이었다.

5.2.2 독립 변수

실험의 독립 변수는 버튼 인터페이스, 기울기 추적 인터페이스, 음성 추적 인터페이스의 중 하나의 인터페이스 적용 여부와, 격자 원기둥 레이아웃과 피쉬아이 원기둥 레이아웃 중 하나의 레이아웃의 적용 여부이다. 표5에 독립 변수들이 정리되어 있다. 위치 추적형 인터페이스는 clutching 문제로 인해, 음성 추적 인터페이스보다 좋지 못한 성능을 pilot experiment에서 보여주었으므로, 실험에 사용되는 변수 중에서는 제거되었다. 실험의 전체적인 디자인은 3 또는 2 level을 갖는 2 factor within-subject design으로, 피실험자는 표6 와 같은 6 가지의 실험 세션을 수행하게 된다. 각각의 실험 세션은 latin square를 통해서 6 명 마다 다른 순서로 설정된다.

5.2.2 종속 변수

실험의 종속 변수는 실행 시간, 사용하기 쉬움, 선호도, 직관적, 자연스러움, 재미이다. 실행 시간을 제외한 나머지 수치는 설문지를 통하여 얻었다.

표 5. 독립 변수

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

표 6. 인터페이스와 레이아웃의 조합

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

5.2.3 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 가능한 한 배제하고 인터페이스와 레이아웃의 효과만을 검증할 수 있다. 그리고 후자의 수행 작업을 통해 인지적인 부담을 요구하는 작업에서의 인터페이스와 레이아웃의 효과를 검증할 수 있다.

5.2.4 실험 절차

피 실험자는 실험 전에 기본적인 인적 정보와 모바일 기기를 이용한 이미지 관리 여부, 모션 인터페이스 사용 여부에 대한 설문을 하고 인터페이스의 사용법에 대해서 설명을 듣는다. 모든 피실험자가 모션 인터페이스에 익숙하지 않으므로, 모션 인터페이스의 기본적인 동작에 대한 트레이닝 세션을 두었다. 트레이닝 세션이 끝난 뒤에는 각각의 조합에 대한 수행 작업을 행하였다.

수행 작업의 순서는 사진의 위치를 알려주고, 사진을 찾는 수행작업을 실행한 후에, 사진의 위치를 알려주지 않고 사진

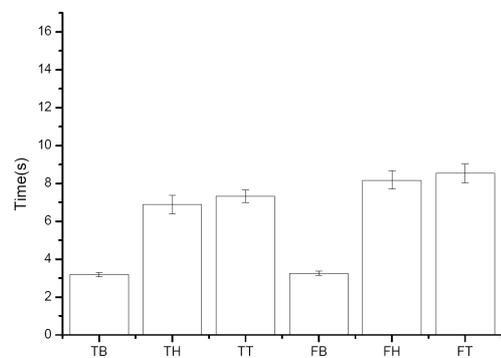
을 찾는 수행작업을 실시한다. 매 수행 작업은 24번의 trial로 이루어져 있고, 이 중 처음 4 번의 trial은 트레이닝 세션으로 사용한다. 이 두 수행작업을 6 가지의 실험 세션마다 반복하게 되므로, 피실험자는 12 번의 수행작업을 하게 된다.

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

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

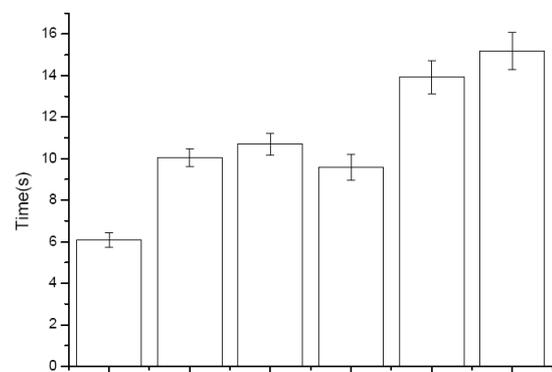
5.4 실험 결과 및 토론

5.4.1 평균 수행 시간



T: Tiled Cylindrical Layout F: Fisheye Cylindrical Layout
B: Button Interface, H: Hybrid Motion Interface, T: Tilt Motion Interface
그림 5. 위치를 알려준 이미지의 탐색 시 평균 수행 시간

위치 알려준 이미지의 탐색 시 버튼 인터페이스가 혼성 추적 인터페이스와 기울기 추적 인터페이스보다 유의하게 짧은 평균 수행 시간($F_{2,46} = 84.10$ $p < 0.0001$)을 기록하였다. 또한 격자 원기둥 레이아웃이 피쉬아이 원기둥 레이아웃보다 유의하게 짧은 수행 시간($F_{1,23} = 11.53$ $p = 0.0025$)을 기록하였다.



T: Tiled Cylindrical Layout F: Fisheye Cylindrical Layout
B: Button Interface, H: Hybrid Motion Interface, T: Tilt Motion Interface
그림 6. 위치를 알려주지 않은 이미지의 탐색 시 평균 수행 시간

위치를 알려주지 않은 이미지의 탐색의 경우도 마찬가지로 버튼 인터페이스가 혼성 추적 인터페이스와 기울기 추적 인터페이스보다 유의하게 짧은 평균 수행 시간($F_{2,46} = 48.61$, $p < 0.0001$)을 기록하였다. 또한 layout의 경우도 위치를 알려준 경우와 같이, 격자 레이아웃 레이아웃이 피쉬아이 레이아웃보다 유의하게 짧은 수행 시간($F_{1,23} = 70.51$, $p < 0.0001$)을 기록하였다.

5.4.2 설문 수행 결과

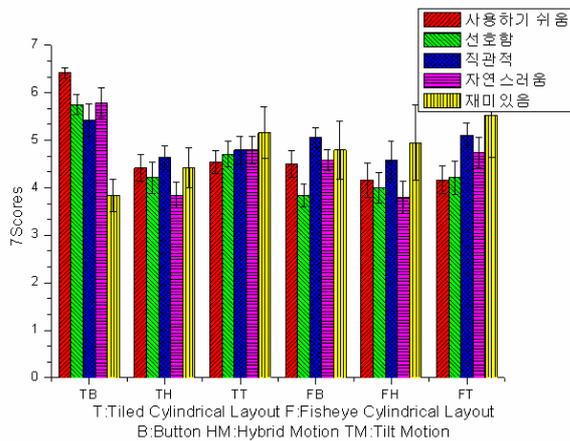


그림 7. 설문 결과

설문의 수행 결과는 그림 7의 그래프와 같다. 사용하기 쉬움에 있어서는 버튼 인터페이스와 격자 레이아웃이 가장 사용하기 쉬운 것으로 밝혀졌으며, 이 조합을 제외한 다른 조합들의 경우는 서로 간에 통계적으로 유의한 결과는 발견하지 못하였다. 선호도의 경우도 버튼 인터페이스와 격자 레이아웃을 가장 선호하였고, 다른 조합들은 서로 간에 통계적으로 유의한 결과를 얻지 못하였다. 직관적의 경우엔 모든 조합들이 서로 유의한 차이를 보이지 못 하였지만, 자연스러움의 경우에도 버튼 인터페이스와 격자 레이아웃이 가장 자연스럽다는 결과를 얻었다. 재미의 경우에만, 격자 레이아웃보다 피쉬아이 레이아웃이 버튼 인터페이스보다는 두 동작 추적 인터페이스가 더 재미있다는 결론을 얻었다.

5.4.3 실험 결과 및 토의

실험 결과 버튼 인터페이스와 격자 레이아웃이 이미지 브라우징에 효과적이었으며, 사용자로부터도 좋은 평가를 받았다.

움직임 추적 인터페이스들은 효과적이지 못하였는데, 사용자는 충분히 빨리 움직이지 못하였고, 장비 자체의 무게에 대한 불만이 존재하였다. 또한 버튼 인터페이스에 비해 신뢰성과 해상도에서도 한계를 보였다.

또 혼성 추적 인터페이스가 기울기 추적 인터페이스보다 더 부자연스러운 결과를 얻었는데, 사용자가 모드 변환에 익숙하지 못 하였고, 실제 공간을 움직이면서 브라우징을 하기 보단 작은 움직임으로 브라우징을 하는 쪽을 선호하였다.

피쉬아이 레이아웃 역시 효과적이지 못하였는데, 우선 사

용자가 더 많은 이미지가 보이는 것 보다 세부를 관찰할 수 있는 쪽을 선호하였다. 하지만 피쉬아이 레이아웃의 경우에는 parameter가 최적화되지 않았을 가능성 역시 존재한다.

6. 결론 및 향후 계획

논문에서 제안한 동작 추적 인터페이스는 이미지 브라우징에 부적합하였다. 이 결과와 관련한 유력한 원인 중에 하나로 고정된 이미지가 움직이는 이미지 보다 효과적으로 인지된다는 연구결과[18]가 있다. 버튼 인터페이스의 경우 이미지가 단속적으로 제시되는데 반해, 동작 추적 모션 인터페이스는 움직이는 이미지로 제시하므로 사용자가 효과적으로 인지하지 못할 수 있다. 이는 동작 추적 모션 인터페이스 자체가 버튼 인터페이스에 비해 태생적으로 부적합할 가능성을 시사한다.

그리고 기울기 인터페이스보다 혼합형 인터페이스가 더 부자연스러운 결과를 보여준 것으로 보아 현실과 같은 메타포 역시 이미지 브라우징에는 그다지 효과적이지 못할 가능성이 있다.

또한 구현된 인터페이스가 해상도와 신뢰성이 충분하지 못하였고, 무게 등의 이유로 사용하기 어려웠다. 이 불안정한 모션 인터페이스 역시 결과에 영향을 끼칠 수 있다고 본다.

레이아웃의 경우엔 격자 레이아웃이 다수의 이미지를 위한 2D 브라우징엔 더 효과적이었다. 이는 하나의 큰 이미지를 브라우징 하는 경우와는 달리 전체적인 부분을 확인할 필요가 없었기 때문으로 생각된다. 하지만 피쉬아이 레이아웃의 경우에는 parameter가 최적화되지 않았을 가능성이 존재하므로, 더 깊은 연구가 필요하다고 생각한다.

향후엔 좀더 신뢰성이 높은 동작 추적 인터페이스를 개발할 필요성이 있다. IR range sensor와 angular-rate-gyroscope를 조합한다면 더 신뢰성 있는 결과를 얻을 수 있을 것이라 생각한다. 또한 고정된 이미지가 움직이는 이미지보다 효과적으로 인지된다는 연구결과에 따라, 동작 추적의 경우에도 고정된 형태로 이미지를 제시할 수 있는 방법을 고려해볼 필요가 있다고 생각한다.

감사의 글

본 연구는 한국 정보통신부의 ITRC 프로그램의 지원을 받아 수행되었다.

참고문헌

[1] C. Gutwin and C. Fedak, "Interacting with Big Interfaces on Small Screens: a Comparison of Fisheye, Zoom, and Panning Techniques," presented at Proceedings of Graphics Interface, 2004.

[2] D. Patel, G. Marsden, S. Jones, and M. Jones, "An Evaluation of Techniques for Browsing Photograph Collections on Small Displays,"

- presented at MobileHCI, Glasgow, Scotland, 2004.
- [3] T. Hakala, J. Lehtikainen, and A. Aaltonen, "Spatial Interactive Visualization on Small Screen," presented at MobileHCI, 2005.
- [4] K.-P. Yee, "Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers," presented at CHI, 2003.
- [5] C. Verplaatse, "Inertial proprioceptive devices: Self-motion sensing toys and tools," *IBM SYSTEMS JOURNAL*, vol. 35, 1996.
- [6] J. Rekimoto, "Tilting Operations for Small Screen Interfaces," presented at ACM Symposium on User Interface Software and Technology, 1996.
- [7] B. L. Harrison, K. P. Fishkin, A. Gujar, C. Mochon, and R. Want, "Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces," presented at CHI, 1998.
- [8] J. F. Bartlett, "Rock' n' Scroll Is Here to Stay," *Computer Graphics and Applications*, IEEE, vol. 20, pp. 40-45, 2000.
- [9] K. Hinckley, J. Pierce, M. Sinclair, and E. Horvitz, "Sensing techniques for mobile interaction," presented at ACM Symposium on User Interface Software and Technology, 2000.
- [10] A. Crossan and R. Murray-Smith, "Variability in Wrist-Tilt Accelerometer Based Gesture Interfaces," presented at Mobile HCI, 2004.
- [11] P. Eslambolchilar and R. Murray-Smith, "Tilt-Based Automatic Zooming and Scaling in Mobile Devices - A State-Space Implementation," presented at Mobile HCI, 2004.
- [12] G. Marsden and N. Tip, "Navigation Control for Mobile Virtual Environments," presented at MobileHCI, 2005.
- [13] J. Hwang, J. Jung, and G. J. Kim, "Hand-held Virtual Reality: A Feasibility Study," presented at ACM Symposium on Virtual Reality Software and Technology, 2006.
- [14] T. T. A. Combs and B. B. Bederson, "Does Zooming Improve Image Browsing?," presented at International Conference on Digital Libraries, 1999.
- [15] K. Hornbæk, B. B. Bederson, and C. Plaisant, "Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview," *ACM Transactions on Computer-Human Interaction*, vol. 9, pp. 362-389, 2002.
- [16] J. Hwang, G. J. Kim, and N. Kim, "Camera based Relative Motion Tracking for Hand-held Virtual Reality," presented at NICOGRAPH International 2006.
- [17] M. R. Mine, F. P. Brooks, and C. H. Sequin, "Moving objects in space: exploiting proprioception in virtual-environment interaction," presented at International Conference on Computer Graphics and Interactive Techniques, 1997.
- [18] K. Cooper, O. d. Bruijn, R. Spence, and M. Witkowski, "A Comparison of Static and Moving Presentation Modes for Image Collections," presented at Advanced Visual Interfaces, 2006.