

Touch Play Pool: 정전용량형 터치 센서를 이용한 휴대 단말용 사용자 인터페이스

Touch Play Pool: A Mobile User Interface Using Capacitive Touch Sensors

장욱, 조성일, 소병석, 이현정, 박준아

{wook.chang, sicho513, bssoh, hyem.lee, joonah}@samsung.com

삼성종합기술원, Advanced Systems Research Lab, Multimodal Mobile I/O System TF

요약 터치 센서를 이용한 휴대단말용 인터페이스가 다양하게 개발되면서 차세대 인터랙션 수단으로서 많은 주목을 받고 있지만, 버튼 인터페이스에 비해 오동작에 취약하고 터치 인터페이스만의 차별성을 확보하지 못하고 있는 실정이다. 본 논문에서는 터치 드래그 동작을 주요 인터랙션 수단으로 활용한 모바일 기기용 사용자 인터페이스를 개발하여 오동작을 최소화하고 사용자가 조작할 때 즐거움을 줄 수 있는 UI 요소로 활용할 수 있도록 하였다. 본 논문에서 개발한 터치 인터페이스는 기기의 내부/외부 방향의 스크롤 동작을 각각 drag-in/drag-out 동작으로 칭하고 이러한 드래그 동작을 응용 프로그램의 실행과 종료에 할당하였으며 터치 센서를 따라 움직이는 스크롤 동작은 기존과 동일하게 일반적인 스크롤 행위를 지시하는데 적용하였다. 제한한 인터랙션 방법은 터치 센서의 형태에 따라 크게 두 가지 방식으로 구현하였다. 첫 번째 방법은 기기 스크린 외곽에 터치 센서를 2열 배치하여, 터치스크린을 사용하지 않고도 스크린 주변의 터치 센서를 활용한 drag-in/out 동작의 감지가 가능하도록 하였다. 두 번째 구현 방법은 정전용량형 터치 센서 IC를 활용해 터치 스크린 기능과 함께 스크린 주변의 기기 케이스의 일부까지도 사용자의 접촉을 감지할 수 있도록 하였다. 기존 저항 방식 터치 스크린과 달리 본 논문에서 활용한 방식은 스크린과 케이스에서 동시에 터치를 감지할 수 있으며, 다접점 감지 알고리즘의 개발로 두 개의 손가락을 이용한 다양한 터치 인터랙션으로 활용할 수 있다.

핵심어: Touch user interface, Capacitive touch sensor, Mobile device, Input devices

1 서론

디지털 기술의 발전으로 인해 점차적으로 소형 휴대 단말들의 기능은 강화되고 있으며, 이로 인해 많은 기능을 손쉽게 사용할 수 있도록 지원하는 사용자 인터페이스의 개발이 주요한 개발 및 연구 이슈로 대두되고 있다 [1, 2].

본 논문에서는 터치 센서를 이용해 drag-and-drop UI와 多重入力 UI를 유기적으로 결합해 휴대 기기의 사용성을 높이고 사용자에게 흥미를 제공하는 소형 휴대 단말용 인터랙션 방법인 Touch Play Pool (TPP)를 제안한다.

이를 위해 첫 번째는 LCD 주변에서 터치 drag-in/out 동작 및 터치 scroll 동작을 인식하여 여러 응용 프로그램과 기능 실행을 위한 모든 조작을 가능하도록 하였다. 두 번째로는 drag-in/out 동작 및 scroll 동작과 함께 두 개의 손가락을 이용한 다접점 감지를 통해 직관적이고 단순한 인터랙션이 가능하도록 하였다. 그림 1은 TPP 인터랙션 개념을 도시한다.

본 논문에서 제시하는 인터랙션 방법론은 물리적인 그릇에 실행하고자 하는 응용 프로그램 또는 기능을 담은 메타포를 활용한다. 즉, 드래그 동작을 통해 응용 프로그램 또는 기

능을 스크린에 담아 실행하고 종료를 원할 때는 스크린에서 해당 프로그램 또는 기능을 drag-out 동작을 통해 꺼내어 종료하는 것으로 한다.

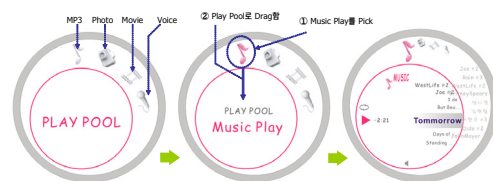


그림 1 Touch Play Pool UI 개념도

기존 터치 UI는 실행은 클릭, file/menu navigation은 스크롤을 사용하는 반면, 본 논문에서 개발한 TPP UI는 短軸方向 스크롤과 長軸方向 스크롤을 분리해 전자는 실행, 후자는 menu/file navigation을 담당하도록 하였다.



그림 2 구현성을 고려한 TPP UI 개념도

제안하는 터치 UI인 TPP를 구현하기 위해 본 논문에서는 두 가지 방법을 사용하였다. 첫 번째 방법은 시스템의 디스플레이 주변에 터치 스위치 센서를 2열 배치하는 것이며 두 번째 방법은 시스템의 디스플레이부와 케이스부에 하나의 투명한 터치 패널을 장착하는 것이다.

전자의 방법은 저가적으로 TPP의 개념을 구현할 수 있으며 사용자의 터치 동작이 디스플레이 외곽 근방에서 이루어지므로 사용 시 발생하는 디스플레이의 오염 문제가 없다. 단점으로는 터치 센서가 외곽에만 배치되어 사용자의 의도가 부정확하게 시스템에 전달될 가능성이 높다. 이러한 문제는 사용자 데이터의 수집과 분석을 통해 강인한 터치 신호의 해석 알고리즘을 개발해 해결하였다.

후자의 방법은 상대적으로 고가의 센서를 활용하나 디스플레이 내부에서도 터치를 감지할 수 있으므로 전자에 비해 다양한 UI를 구성할 수 있는 장점이 있다. 특히 본 논문에서는 더욱 유연한 UI를 구성할 수 있도록 다접점을 감지할 수 있는 알고리즘을 개발해 장착하였다.

기존 터치 UI가 사용자 신체가 우연하게 기기에 접촉할 때 발생하는 오동작에 취약함에 반해 제안한 TPP UI는 단순한 터치 클릭에 비해 더욱 의도성이 강한 스크롤 동작으로 기기의 조작을 하도록 강제함으로 오동작 발생 확률을 낮추도록 하였다.

2 외곽 터치를 활용한 Touch Play Pool

2.1 인터랙션 구조

인터랙션의 기본 원칙은 다음과 같다. 먼저 상단 수평 터치 영역은 최상위 수준 실행 영역으로 정의해 drag 동작을 통해 주요 응용 프로그램과 메뉴의 실행을 담당한다. 우측 수직 터치 영역은 응용 프로그램을 위한 서브 조작 영역으로 정의해 drag-in은 메뉴 실행, drag-out은 이전 단계 이동으로 활용한다. 또한, 스크롤 동작은 재생 중인 아이탬의 접근과 메뉴와 파일 네비게이션에 활용을 한다. 그림 3에서 이를 도시하였으며 그림 4는 개략적인 UI 전개도이다.

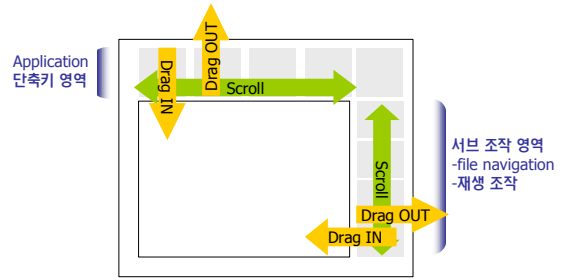


그림 3 TPP UI의 기본 원칙



그림 4 TPP UI 전개 요약

외곽 터치를 활용한 UI 전개를 정리하면 다음과 같다.

- 최상위 메뉴의 실행: Drag-in ↓
- 하부 메뉴의 실행: Drag-in ←
- 하부 메뉴의 실행 취소: Drag-out →

2.2 하드웨어 시스템

시스템 구현을 위해 QRG社의 QT1101 10-KEY QTouch™ 센서 IC를 사용하였다. 그림 5에 전체 시스템 구성도를 제시하고 있다.

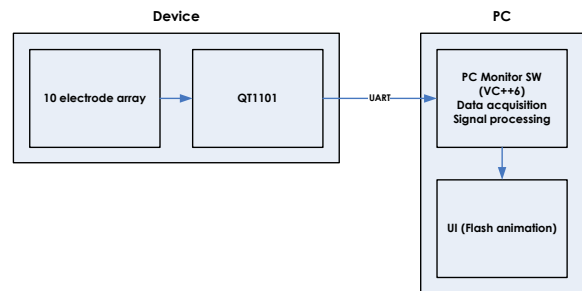


그림 5 시스템 아키텍처

기본 UI는 상단 5개, 우측 4개 영역을 이용해 drag-in/out과 scroll을 감지하도록 구성한다. 이러한 스크롤 (drag-in/out 포함) 터치 액션을 구분하기 위해서는 최소 2개의 터치 센서가 필요하며 더욱 신뢰성있는 판단을 위해서는 3개의 터치 센서가 필요하다. 그러나, 3개의 터치 센서를 활용하는 경우는 LCD 외곽의 센서 점유 면적이 커져 전체적인 외관을 해치게 되고, 터치 센서 IC 1개로는 적용할 수 없는 총 27개 채널이 필요하게 된다. 시스템의 외관과 터치 센서 칩의 HW적 한계를 감안해 터치 센서 전극 배치를 그림 6 (a)와 같이 하여, 이를 통해 drag-in/out 및 스크롤 감지를 1개의 터치 센서 IC로 구성하였다. 그림 7은 기구의 3D 모델링 결과이다.

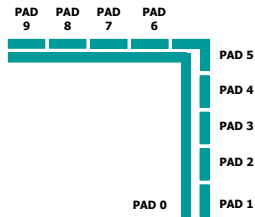


그림 6 터치 센서 배치

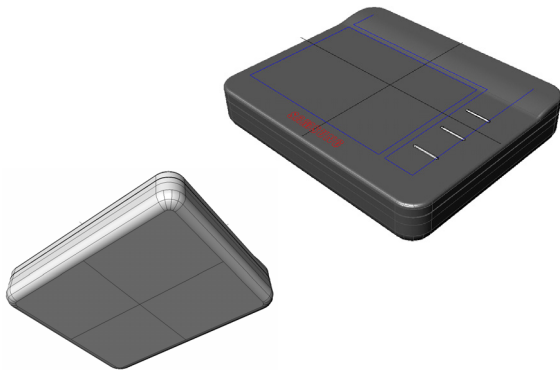


그림 7 외관

2.3 오동작 방지 알고리즘

외곽 터치 센서를 이용하는 경우는 터치 드래그 동작을 정확히 감지하기 위한 센서 감지부의 개수가 부족하므로 사용자의 의도와 다른 동작을 기기가 수행할 수 있는 오동작의 가능성이 있다. 이러한 오동작은 인접한 터치 센서간의 간섭, 터치 센서의 성능 한계로 인해 터치 센서 신호가 검출되지 않거나, 터치 신호 해석의 판단 자체가 불가능한 경우에 발생하게 된다.

이러한 오동작의 문제를 해결하기 위해 본 논문에서는 터치되는 센서의 on/off 여부와 지속 시간을 이용해 강인한 터치 신호 해석 알고리즘을 개발하였다. 그림 8과 같이 기존의 일반적인 터치 신호 해석 알고리즘을 활용하는 경우 터치 drag-in은 89%, 터치 drag-out은 84.22%의 성공률을 보이는 반면, 개선된 터치 신호 해석 알고리즘을 활용하는 경우 터치 drag-in은 94.67%, 터치 drag-out은 92.78%의 성공

률을 보인다. 특히, SW 알고리즘 개선으로 인해 인접 센서 접촉에 의한 판단 성능이 drag-out은 8.33%에서 0.78%로, drag-in은 5.00%에서 0.44%로 월등히 향상되었음을 알 수 있다.

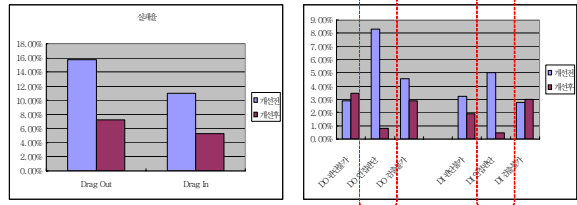


그림 8 실험 결과

3 전면 터치를 활용한 Touch Play Pool

3.1 인터랙션 구조

전면 터치를 활용하는 경우에는 외곽 터치만 활용하는 경우에 비해 상대적으로 UI 설계의 자유도가 높아지고, 전체적인 외관 디자인을 최대한 간결하게 할 수 있는 장점이 있다. 본 연구에서는 외곽 터치 UI와 마찬가지로 drag-in/out 동작을 주요 응용 프로그램과 메뉴의 실행 방식으로 하여 그림 9와 같이 스크린 내/외부 모두에서 드래그 동작을 통한 조작이 가능하도록 하였다.



그림 9 전면 터치 UI (a) 기존 터치 스크린 방식 메뉴 배열 (b) TPP UI 방식의 메뉴 배열 (c) 기존 터치 스크린 방식 목록 배열 (d) TPP UI 방식의 방사형 메뉴 배열

3.2 하드웨어 시스템

전면 터치를 활용하는 시스템은 디스플레이부와 케이스부에서 동시에 터치 신호를 감지할 수 있어야 한다. 현재 소형 기기에 가장 많이 사용되는 터치 스크린은 손가락의 압력을 감지하는 방식이지만, 이 방식은 시스템 외곽에서 터치를 감지할 수 없고, 스크린 영역 내에서만 손가락을 감지할 수 있기 때문에 스크린 내/외부를 오가는 드래그 동작의 구현이 불가능하다. 따라서 본 논문에서는 QRG사에서 개발한 정전용량 감지 방식의 QT802 소형 터치 스크린 IC와 ITO 투명 터치 패널을 활용하여 케이스 부분만 아니라 LCD 디스플레이부까지 터치 센서가 감지될 수 있도록 구성하였다. 그림 10에 이를 도시한다.

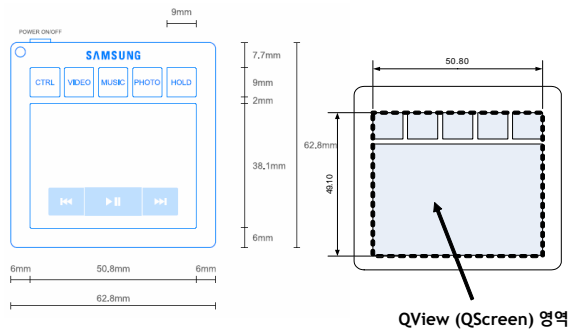


그림 10 기구부

3.3 다접점 감지 알고리즘

현재 상용화되어 있는 대부분의 터치 패드와 터치 스크린에서 활용 중인 터치 센서 시스템은 단일접점 기준의 인터페이스만을 지원하는 것으로, 복수 개의 손가락을 감지하기 위해서는 추가적인 신호처리가 필요하다. 본 연구에서 사용한 QT802도 그림 11과 같은 projective sensor matrix [3] 형태의 터치 패터를 이용하는 2차원 터치 감지 IC로서 본질적으로 단일 접점의 감지를 위한 것이다.

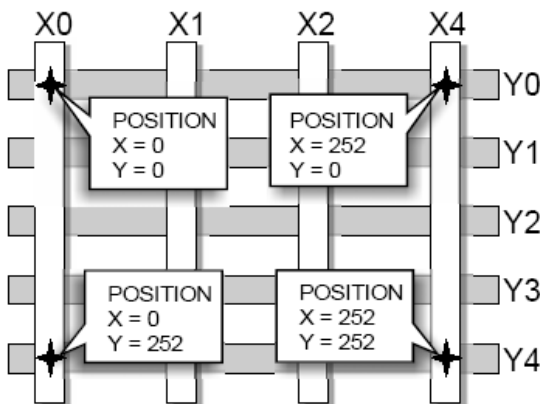


그림 11 QT802를 사용하는 경우 터치 패널의 패턴

이러한 projective sensor matrix에 관한 일반적인 다접점 감지 방식과 본 논문에서 제안하는 특정한 신호 처리 감지 방법에 대해 두 손가락의 경우에 대해 한정하여 설명하고자 한다.

3.3.1 두 손가락이 시간차를 두고 시스템에 접촉하는 경우

먼저, QT802 센서 IC의 원 데이터와 위치 사이에 다음과 같은 관계 수식 (1)이 성립한다고 가정하자.

$$p = f(s, r) = f(s) \quad (1)$$

여기서, p 는 계산된 위치 벡터, s 는 내부 신호 (internal signal), r 는 기준 신호 (reference signal)이다. 위의 가정과 실험 결과를 바탕으로 시간차를 두고 접촉하는 두 손가락의 위치를 판단하는 알고리즘을 상술한다. 먼저 첫 번째 손가락이 시간 t_1 에 접촉되어 고정된 상태에서 두 번째 손가락이 시간 t_2 에 접촉해 이동하는 경우를 고려하자. 이 때, 내부 신호 $s(t)$ 는 두 손가락에 의해 발생하는 신호 강도의 합으로 생각할 수 있다. 따라서, 두 번째 손가락에 의한 신호 강도 s_2 는 다음과 같이 추정 가능하고 위치는 (3)과 같이 계산할 수 있다.

$$s_2(t) = s(t) - s(t_1), t > t_1 \quad (2)$$

$$p_2(t) = f(s_2(t)) \quad (3)$$

지금까지 상술한 바를 그림 12에 도시하였다.

Algorithm 1: Multi-finger touch algorithm for the Case 1

```

Set initial values of  $s_1 = s(0)$ ;
 $k := 1$ ;
if  $s(k) \geq s_{th}$  then
  Compute the position of the second finger;
   $p_2(k) = f(s(k) - s_1)$ 
else
  Compute the position of the first finger;
   $p_1(k) = f(s(k))$ 
  
```

그림 12 다접점 감지 알고리즘

3.3.2 두 손가락이 동시에 접촉해 움직이는 경우

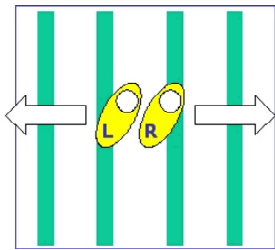
3.3.1절의 알고리즘은 두 손가락의 정확한 위치를 파악할 수 있으나 두 개의 손가락이 순차적으로 접촉하고 이동은 두 번째 손가락만 할 수 있다는 제약이 있다. 이러한 문제를 해결하기 위해 본 절에서는 두 손가락이 동시에 움직이는 경우의 알고리즘을 제시한다.

앞에서 설명하였듯이 QT802 센서의 경우 두 손가락 이상이 동시에 눌리는 경우(Multi-Touch)에는 접촉한 손가락 모두에 의해서 전극의 변화가 발생하게 되어 종래의 방식에 있

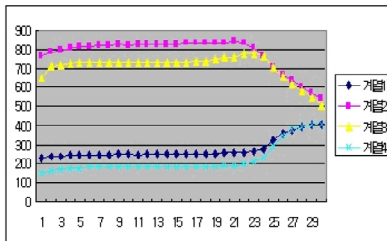
어서는 이에 따른 위치 감지가 불가능하게 된다.

그림 13은 두 손가락이 밖으로 이동하는 경우 (LR Drag Out)에 있어 손가락의 움직임 변화와 이에 따른 전극의 세기 및 세기 변화를 보여주는 도면이다. 접촉한 두 손가락의 동시에 밖으로 이동하는 경우에는 양 사이트에 위치한 전극 (x_1, x_4)으로 손가락이 이동하는 것으로 볼 수 있으며 이때, 두 전극의 세기는 손가락이 이동함에 따라 강도의 세기가 증가하고 중앙에 위치한 전극(x_2, x_3)의 강도의 세기는 점차로 감소하게 된다. 이는 전극 세기의 미분치를 통해 판단 가능하다. 즉, \dot{x}_1 과 \dot{x}_4 는 0보다 크게 되며 \dot{x}_2 과 \dot{x}_3 은 0보다 작게 된다.

본 논문에서 개발한 알고리즘을 이용하는 경우에는 기존의 한 점 감지용 2-D 정전용량 위치 센서를 이용하여 Multi-Touch Drag & Scroll 등 두 손가락 이상이 접촉하는 경우에도 각각 손가락의 움직임 변화를 감지할 수 있도록 확장해서 쓸 수 있다. 이에 따라 한 손가락 기반의 사용자 인터페이스가 아니라 Multi-Finger 기반의 사용자 인터페이스 설계가 가능하다.



(a)



(b)

그림 13 Multi-touch drag의 예: 두 손가락이 스크린 밖으로 이동 (a) 개념도 (b) X축 전극 변화

4 결론

본 논문에서는 소형 다기능 휴대 단말의 사용자 인터페이스로 활용 가능한 터치 인터페이스인 TPP 인터페이스를 제안하였다. TPP 인터페이스는 기능의 실행을 위해 Play Pool에 기능을 대표하는 아이콘을 스크린 Pool 안에 드래그하여 집어 넣고 기능의 중지를 위해서는 Play Pool에서 기능을 드

래그하여 빼는 개념을 사용한다. 이를 통해 직관적이고 단순한 조작이 가능하고 기존의 클릭 방식에 비해 터치 센서의 오동작을 줄일 수 있다. 또한, 외곽 터치 센서를 활용하는 경우에는 LCD 오염 없이 터치 조작을 수행할 수 있는 장점이 있다.

TPP 인터렉션 방법은 두 가지 형태로 구현되었다. 먼저 기기 스크린 외곽에 터치 센서를 2열 배치해 스크린의 오염이 방지되고 값비싼 터치 스크린 대신 저가의 터치 스위치 IC를 사용할 수 있도록 구성하였다. 또한, 첫 번째 방식으로 구현 시 본 논문에서는 저가 센서의 하드웨어적인 한계로 인해 발생 가능한 오동작 확률을 저감하는 알고리즘을 개발하였다. 두 번째 구현 방법은 정전 용량형 터치 센서 IC를 활용해 스크린부와 기기 케이스의 일부가 사용자의 접촉을 감지할 수 있도록 하였다. 기존 저항 방식 터치 스크린과 달리 본 논문에서 활용한 방식은 1개의 터치 센서 IC만을 활용해 스크린과 케이스에서 터치를 감지할 수 있는 장점이 있다. 또한, 이 방법은 감지부가 외부에 노출될 필요가 없어 스크린의 보호에 기존 저항 방식보다 더욱 유리하다. 추가로 단일 접점만 감지 가능한 하드웨어적인 한계를 극복하고 다접점을 감지할 수 있는 알고리즘을 개발하였다.

참고문헌

- [1] H. Benko, A. D. Wilson, and P. Baudisch, "Precise selection techniques for multi-touch screens," presented at CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, Montreal, Quebec, Canada, 2006.
- [2] J. Rekimoto and C. Schwesig, "PreSenseII: bi-directional touch and pressure sensing interactions with tactile feedback," presented at CHI '06: CHI '06 extended abstracts on Human factors in computing systems, Montreal, Quebec, Canada, 2006.
- [3] C. v. Wrede and P. Laskov, "Using classification to determine the number of finger strokes on a multi-touch tactile device," in *ESANN 2004*, 2004.