

상이한 해상도의 스마트폰의 앱 호환성을 위한 화면 동등성 평가 기법에 대한 실험적 연구*

이정욱, 김태연, 채흥석
부산대학교 컴퓨터공학과
e-mail : leonlly.tykim.hschaee@pusan.ac.kr

An Empirical Study on Method for Discriminating the Screen Equivalence for Evaluating the Compatibility of Smartphone Apps on Different Resolutions.

Lee-Jeong Uk, Kim-Tae Yeon, Chae-Heung Seok
Dept. of Computer Engineering, Pusan National University

요 약

다양한 사용자의 요구를 충족 시키기 위해서 스마트폰 제조사들은 상이한 해상도의 스마트폰을 출시하고 있지만, 이것은 스마트폰 앱의 호환성 문제를 일으킬 수 있다. 호환성 문제는 서로 다른 해상도를 가진 스마트폰에서 동일한 앱을 실행했을 경우에 그 결과가 다른 현상을 말한다. 스마트폰의 해상도 차이에 따른 앱 호환성 문제는 앞으로 더욱 심각해질 것으로 예상된다. 이러한 앱의 호환성을 판별하기 위하여 GUI 기반의 테스트 기법이 고려될 수 있다. 특히 해상도가 다른 두 스마트폰에서 동일한 앱의 실행 결과 화면의 화면 동등성 평가는 매우 어렵다. 본 논문에서는 이 문제를 극복하기 위하여 해상도가 다른 스마트폰의 앱 호환성을 평가하기 위한 화면 동등성을 판별하는 프레임워크를 제안한다. 상이한 해상도의 상용 스마트폰을 대상으로 앱의 실행화면 동등성 평가를 위해 단순 비교 기법과 VDP 기법의 성능 비교를 위한 실험 결과를 소개한다. 실험에서는 상이한 해상도의 스마트폰에서 동일한 앱의 실행 화면을 바탕으로 단순 비교 기법과 VDP 기법의 동등성 판별 성능을 비교한다. 실험 결과 사람의 시각 체계와 유사한 인지 능력을 보이는 VDP 가 화면의 동등성을 비교적 정확하게 판별하였다.

1. 서론

다양한 사용자의 요구를 충족 시키기 위해서 스마트폰 제조사들은 상이한 해상도의 스마트폰을 출시하고 있지만, 이것은 스마트폰 앱의 호환성 문제를 일으킬 수 있다. 호환성 문제는 서로 다른 해상도를 가진 스마트폰에서 동일한 앱을 실행했을 경우에 그 결과가 다른 현상을 말한다. 예를 들어, LG 전자의 320 x 480 의 해상도를 가진 옵티머스 원에서의 페이스북 앱의 동작은 480 x 800 의 해상도를 가진 옵티머스 2X 에서와 동일하게 동작해야 한다.

스마트폰의 해상도 차이에 따른 앱 호환성 문제는 앞으로 더욱 심각해질 것으로 예상된다. 현재에도 많은 종류의 스마트폰이 출시 및 개발되고 있으며, 앞으로 스마트폰 뿐만 아니라 다양한 종류의 태블릿 디바이스도 출시 및 개발 될 예정이다. 위의 LG 의 옵티머스 에에서만 뿐만 아니라 삼성의 갤럭시 S 의 해상도는 480 x 800 이지만 최근 출시한 삼성의 갤럭시 ACE 의 해상도는 320 x 480 으로 동일한 제조사임에도 불구하고 다른 해상도의 제품이 출시되고 있다. 마찬가지로 삼성의 태블릿 디바이스인 갤럭시 탭의 해상도는 1024 x 600 이고, 현재 출시 예정인 갤럭시 탭 2 의 해상도는 1280 x 800 으로 절대적인 해상도뿐만 아니라 가로·세로 비율도 동일하지 않다. 게다가 다른 제조사인 LG 의 옵티머스 패드의 해상도 또한 1280 x 768 으로 절대적인 수치 뿐만 아니라 가로·세로 비율이 다르다.

이러한 앱의 호환성을 판별하기 위하여 GUI 기반의 테스트 기법이 고려될 수 있다[1, 2]. 즉, 동일한 앱을 다양한 스마트폰에서 실행을 하고 그 실행 결과 화면이 동등한 지를 확인함으로써 앱이 해당 스마트폰들에서 호환되는 지를 판단할 수 있다. 그러나 많은 수의 스마트폰을 대상으로 앱의 실행 결과 화면의 동등성을 판단하기 위해서는 많은 노력이 소요된다. 게다가 호환성 테스트 업무는 디바이스 종류 및 플랫폼의 버전이 다양해질수록 동일한 앱을 실행시켜야 하는 스마트폰의 수가 증가하게 되므로 호환성 테스트를 위한 비용은 기하급수적으로 증가될 것이다.

특히 해상도가 다른 두 스마트폰에서 동일한 앱의 실행 결과 화면의 화면 동등성 평가는 매우 어렵다. 이는 해상도의 차이로 인해 화면 구성 요소들의 위치나 크기가 디바이스 마다 다르게

나타날 수 있기 때문이다. 예를 들어, 해상도의 가로 세로 비율이 디바이스 마다 다르기 때문에 개발자가 의도한 화면과는 다르게 화면이 잘리거나, 왜곡되어 사용자에게 보여질 수 있다. 그리고 개발자가 구성 요소의 위치를 화면의 절대적인 값으로 부여할 수도 있고, 상대적인 값으로도 부여 할 수 있기 때문에 해상도에 따라 동일한 화면이 사용자에게 다르게 보여지는 호환성 문제가 발생할 수 있다.

이 문제를 극복하기 위하여 우리는 해상도가 다른 스마트폰의 앱 호환성을 평가하기 위한 화면 동등성을 판별하는 프레임워크를 제안한다. 이 프레임워크는 해상도가 서로 다른 기종의 스마트폰에서 동일한 앱을 실행하고 화면 동등성 판별 알고리즘을 적용하여 그 실행 결과 화면을 비교한다. 화면 동등성 판별 알고리즘으로는 단순 비교 알고리즘과 Visual Differences Predictor(이하 VDP) 알고리즘이 적용된다. 단순 비교 기법은 단순히 두 이미지의 픽셀 값 차이를 산술적으로 비교하는 방법으로서 알고리즘이 단순하여 구현이 쉽고, 판별 속도가 빠르다. VDP 는 인간의 시각 체계를 기반으로 사람이 두 이미지 사이의 차이점을 인지하는 정도를 측정한다.

화면의 동등성을 판별하기 위해서는 두 이미지의 해상도가 동일해야 한다. 그러므로 비교 이미지의 해상도를 원본 이미지의 해상도와 동일하도록 이미지를 수정해야 한다. 하지만 이 과정에서 이미지의 왜곡이 발생하게 되므로 단순히 픽셀 값을 비교하는 방법으로는 정확한 결과를 얻기 어렵다. 반면, VDP 알고리즘은 디지털 이미지의 왜곡에 영향을 덜 받으므로 해상도 변경으로 인한 이미지 왜곡에도 강건하다. 따라서 해상도가 다른 이미지의 화면 동등성 평가에 사용하기 적절하다.

또한, 우리는 해상도가 다른 화면 동등성 평가를 위한 프레임워크를 자동화 할 수 있는 도구(이하 OOSrunner)를 개발하였다. 즉, OOSrunner 는 주어진 임의의 앱을 복수 개의 스마트폰에서 자동으로 실행하고 그 결과 화면을 저장하여 해상도를 일치시키고 단순 비교 기법과 VDP 기법으로 화면 동등성을 판별한다. 본 논문에서는 OOSrunner 를 이용하여 상용 스마트폰을 대상으로 수행한 실험 결과를 소개한다.

본 논문의 기여는 다음과 같다. 첫째, 단순 비교 알고리즘에

* 이 논문 또는 저서는 2010 년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(한국연구재단-2010-20100025361 - 20100613294)

비하여 VDP 알고리즘이 서로 다른 해상도의 차이로 인한 화면 동등성 판별 능력이 뛰어난을 상용 휴대폰을 이용한 실험을 통해 확인하였다. 둘째, 스마트폰 호환성 문제를 극복하기 위한 방법으로 화면 동등성 판별 프레임워크를 제시하고, OOSRunner 도구를 개발하였다.

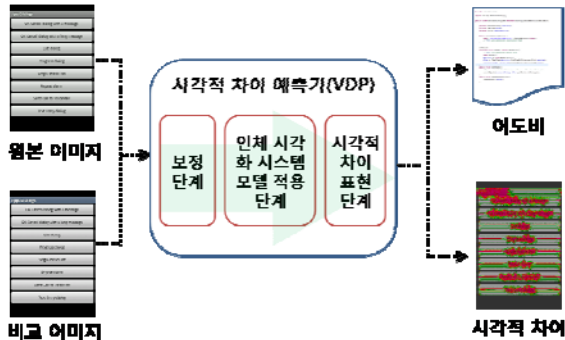
본 논문의 구성은 다음과 같다. 2장에서는 VDP 알고리즘을 설명하고 VDP 알고리즘을 이용하는 영상 처리 관련 연구를 소개한다. 3장에서는 실험 환경 및 실험에 사용된 상용 스마트폰, 실험 대상 앱, 자동화 도구에 대해 설명한다. 4장에서는 단순 비교 기법과 VDP 기법의 화면 동등성의 비교 실험을 진행한다. 마지막으로, 5장에서 결론 및 향후 연구 방향에 대해 기술한다.

2. 배경 지식 및 관련 연구

2장에서는 시각적 차이 예측기인 VDP 를 소개한다. 2.1절에서는 VDP 의 사용 용도와 종류 및 입력과 출력에 대하여 설명하고, 2.2절에서는 VDP 를 활용한 연구 사례와 그 결과에 대하여 설명한다.

2.1. 시각적 차이 예측기(VDP)

S. Daly 가 처음 제안한 VDP 는 인간의 시각적 반응을 표현하는 알고리즘이다[3]. VDP 를 이용하여 원본 이미지를 왜곡시킨 이미지의 정확도를 평가하고, 원본 이미지와 차이나는 부분의 위치 정보를 담은 이미지를 생성할 수 있다. 따라서 VDP 는 이미지 품질에 대한 상대적인 척도를 기술한다. (그림 1)은 VDP 의 입·출력을 나타낸 그림이다. VDP 는 관찰 조건을 조정하는 보정 단계, 인체 시각화 시스템 모델을 적용하는 단계, 인간의 인지적인 관점에서 시각적인 차이를 표시하는 단계로 이루어져 있다. VDP 의 입력은 2 개의 이미지, 관찰 조건과 보정에 대한 매개변수이고 출력은 두 이미지 사이의 시각적인 차이를 보여주는 새로운 이미지이다.



(그림 1) VDP 의 입출력

Mantiuk 등은 고 명암비 이미지를 처리할 수 있도록 VDP 를 확장하여 High Dynamic Range-Visual Differences Predictor(이하 HDR-VDP)를 제안하였다[4]. HDR-VDP 는 LCD 나 LED 디스플레이와 같이 높은 품질의 영상표시 장치에서도 사용할 수 있다. 본 연구에서는 스마트폰의 디스플레이 장치에 적합한 HDR-VDP 를 사용하여 실험을 진행하였다.

2.2. 관련 연구

VDP 는 비디오 및 이미지 영상의 품질 평가, 의학 분야의 Computed Tomography(이하 CT) 이미지 판독 등에서 범용적으로 활용 되었다. <표 1>은 VDP 를 활용한 연구들을 나타낸다. 이 표는 VDP 가 기존의 연구들에서 사람의 시각 체계의 인지적 측면을 정확히 반영하고, 이미지·비디오 영상의 비교 및 품질 측정에 성공적으로 사용되어 왔음을 나타낸다. 특히, 정밀함을 요구하는 의학 연구 분야에서도 VDP 가 사용되었음을 보여준다.

<표 1> VDP 활용 연구 사례

관련연구 (년도)	VDP 활용 방법
Mantiuk 등[5] (2006)	HDR-VDP 를 사용하여 비디오 압축 기법의 품질을 성공적으로 측정
Banterle 등[6] (2007)	VDP 가 신호 대 잡음 비 측정 방법 보다 사람의 인지적인 측면에서 더 정확하다는 결과를 보임

Li 등[7] (2008)	HDR-VDP 를 이용해서 각기 다른 알고리즘으로 생성한 이미지들을 원본 이미지와 비교하여 알고리즘 사이의 상대적인 성능을 측정
Kim 등[8] (2008)	HDR-VDP 를 이용한 CT 이미지 판독 결과가 방사선 전문의의 CT 이미지 판독 능력을 거의 정확하게 재현한다고 주장

3. 실험 환경

3장에서는 실험을 수행하기 위한 환경을 소개한다. 3.1절에서는 실험의 목적에 대해서 설명하고, 3.2절에서는 실험에 사용된 앱인 ApiDemos 와 실험에 사용된 상용 디바이스에 대하여 설명한다. 3.3절에서 화면 동등성 판별 자동화 도구인 OOS Runner 도구에 대해 소개하고, 마지막으로 3.4절에서는 화면 동등성 비교 실험 절차를 단계별로 기술한다.

3.1. 실험 목적

본 실험은 상이한 해상도의 상용 스마트폰에서 수행되는 동일한 앱 실행 화면의 동등성 평가 측면에서 단순 비교 기법과 VDP 기법을 비교한다.

3.2. 실험 대상 디바이스 및 앱

본 실험에서는 대표적인 안드로이드 스마트폰 업체인 LG 전자의 스마트폰을 대상으로 수행되었다. <표 2>는 실험에 사용된 스마트폰의 제품명, 해상도, UI 와 GPU 정보를 요약한다.

<표 2> 실험에 사용된 상용 스마트폰

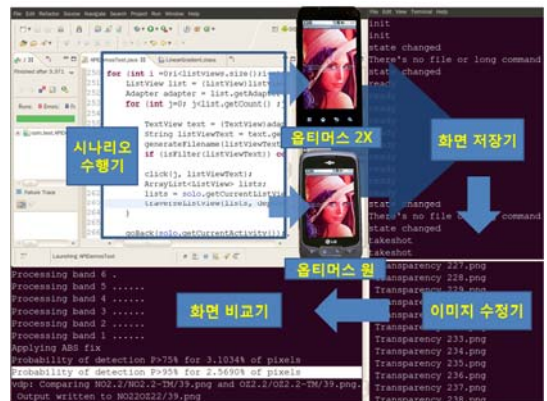
제조사	LG 전자	
제품명	Optimus 2X	Optimus One
해상도	480 x 800	320 x 480
UI	Optimus 1.5	Optimus 1.5
GPU	nVIDIA Tegra2	Adreno 200

본 실험에서는 해상도가 다른 화면의 화면 동등성에 초점을 두므로 그 외의 환경은 동일하게 하기 위해 제조사는 같게 하고, 플랫폼 버전은 Android 2.2 로 통일하였다.

본 실험에서는 화면 동등성 평가를 위하여 Android SDK 에서 샘플로 제공하는 ApiDemos 앱을 선정하였다. ApiDemos 는 안드로이드 앱의 화면을 다양한 메뉴, 버튼 등의 기본적인 요소들로 구성된 화면을 확인할 수 있는 대표적인 앱이다. ApiDemos 에는 이미지 과일과 같은 각종 그래픽 요소와 플랫폼의 UI 요소를 설명하기 위한 프로그램으로 구성되어 있다. 실험에서는 총 75 개의 화면을 사용하였다. 그 중 UI 요소로 구성된 화면이 59 개이고, 그래픽 요소로 구성된 화면이 16 개이다. 전화번호부, 배경 화면 등을 사용하는 화면은 디바이스에 따라 표시되는 화면의 구성 요소가 다르므로 제외 하였고, 자동으로 화면을 저장 하는 동안 리스트 뷰를 이동하면서 저장된 화면도 실험에서 제외하였다.

3.3. 화면 동등성 판별 지원 도구

본 실험을 위해서 화면 동등성 판별을 자동화하기 위하여 도구인 OOSrunner 를 개발하고 실험에 적용하였다. (그림 2)는 OOSrunner 의 실행 화면을 보여 준다.



(그림 2) OOSrunner 실행 화면

OOSrunner 는 주어진 한 쌍의 스마트폰을 대상으로 안드로이드 앱의 테스트 시나리오를 자동으로 수행하고 각 결과 화면을 저장해서 두 개의 이미지를 비교하는 기능을 지원한다. OOSrunner 는 시나리오 수행기, 화면 저장기, 이미지 수정기, 화면 비교기의 네 부분으로 구성되어 있다. 시나리오 수행기는 안드로이드 테스트에 특화된 테스트 프레임워크인 robotium 의 기능 일부를 사용하였다. 그리고 상용 제품의 화면을 저장 하기 위하여 안드로이드 2.3 SDK 에 포함되어 있는 monkeyrunner 테스트 도구의 화면 저장 기능을 사용하였다[9]. 파이썬 이미지 라이브러리를 사용하여 이미지를 수정하였고, 화면 비교기에서는 max planck 연구소에서 제공하는 VDP 라이브러리를 사용하였다[4].

3.4. 실험 절차

해상도가 다른 화면 동등성 비교 실험은 3.3절에서 소개한 OOSrunner 를 이용하여 진행하였다. 다음은 OOSrunner 를 사용하는 화면 동등성 능력 비교 실험 절차이다.

1. 시나리오 수행기는 기준 디바이스와 비교 디바이스에서 동일한 앱을 테스트 시나리오에 따라 실행시킨다. ApiDemos는 리스트 뷰로 구성되어 있으므로, 이것을 자동으로 탐색할 수 있는 테스트 시나리오를 JUnit으로 작성하였다.
2. 시나리오에 따라서 앱이 실행 결과가 화면으로 출력되면 화면 저장기가 스마트폰의 프레임 버퍼에 있는 내용을 읽어서 이미지 파일로 저장한다. 그리고 화면 저장기는 앱의 실행 결과 화면을 정확하게 저장하기 위해 시나리오 수행기와 지속적으로 통신한다.
3. 저장한 두 개의 이미지 파일을 이미지 수정기를 이용하여 해상도를 변환시킨 후 관심 밖 영역을 제외하여 비교 가능한 상태가 되도록 수정한다[10]. 해상도 변환은 서로 다른 두 해상도를 가진 이미지 중 큰 해상도를 가진 이미지의 해상도를 작은 해상도를 가진 이미지의 해상도와 같게 만들어 주는 과정을 의미한다.
4. 수정된 두 이미지를 화면 비교기로 비교한다. 이때, 비교 기법은 단순 비교 기법과 VDP 기법 두 가지를 사용한다. 두 이미지의 비교 작업이 끝나면 단순 비교 기법과 VDP 기법을 사용하여 계산한 결과를 출력한다.

4. 실험 결과 및 해석

4 장에서는 실험 결과를 바탕으로 구성 요소와 위치 변화유무에 따라 화면을 분류하고, 분류된 화면의 측정값들을 해석한다. 4.1절에서는 비교 기법 별로 측정된 실험 결과를 설명하고, 4.2절에서는 측정값을 바탕으로 화면을 분류한다. 4.3절에서는 구성 요소의 위치가 동일한 화면의 측정 결과를 기술하고, 마지막으로 0절에서는 구성 요소의 위치가 다른 화면의 측정 결과를 기술한다.

4.1. 실험 결과

<표 3>은 단순 비교 기법과 VDP 기법을 적용하였을 때의 이도비에 관한 기술 통계량을 보여준다. 이도비는 화면 동등성을 정량적으로 측정하기 위해 전체 화면을 구성하는 픽셀 중 서로 다른 픽셀의 비율을 의미한다.

<표 3> 비교 기법 별 기술 통계량

기법	최대값	75%	50%	25%	최소값	평균	표준 편차
단순 비교	0.95	0.43	0.17	0.14	0.06	0.34	0.31
VDP	0.82	0.17	0.09	0.06	0.03	0.18	0.20

VDP 기법의 이도비의 평균과 표준편차는 단순 비교 기법에 비하여 낮게 나왔다. 특히 평균은 VDP 가 18%, 단순 비교 기법이 34%이므로 약 2 배의 차이가 난다. 본 실험에서는 동일한 앱을 실행하였을 때의 화면을 대상으로 동등성을 비교한 것이므로 낮은 이도비의 값이 합당하다. 그러므로 VDP 기법이 단순 비교 기법에 비하여 해상도가 상이한 디바이스의 화면 동등성을 보다 정확하게

관별하는 것을 의미한다.

4.2. 실험 결과 분석

실험 결과 해상도 변환을 통해 화면 동등성 판단을 할 때에 화면 구성 요소들의 배치 및 크기 표시 방식에 따라서 측정값이 다르게 나타났다. 화면 구성 요소들이 화면 비율에 맞추어 상대적으로 크기나 배치가 설정되어 있다면 해상도 변환하여도 두 화면에서 구성 요소가 동일한 크기와 위치를 가지게 된다. 하지만 구성 요소의 배치나 크기가 절대적인 수치로 설정되어 있는 경우 해상도 변환을 하게 되면 두 화면에서 구성 요소가 서로 다른 크기나 위치를 가지게 됨을 확인 할 수 있었다. 그리고 단순 비교 기법의 경우 화면 구성 요소에 따라서 다른 결과를 나타내었다. 특히 이미지와 같은 그래픽 요소로 이루어진 화면들은 높은 이도비를 나타내었다. (그림 3)은 해상도 변경했을 때 구성 요소의 크기가 다른 이미지이다. (가)와 (나) 모두 동일한 앱을 실행한 화면이지만 해상도의 차이 때문에 구성 요소의 크기와 배치가 다르게 나타났다.



(가) 높은 해상도의 이미지 (나) 낮은 해상도의 이미지
(그림 3) 구성 요소의 크기가 다른 이미지

따라서 효과적인 실험 결과 해석을 위해 각 화면을 다음과 같이 분류하였다.

- 구성 요소 종류: UI, 그래픽
- 구성 요소의 위치 변화: 위치 변화 유무

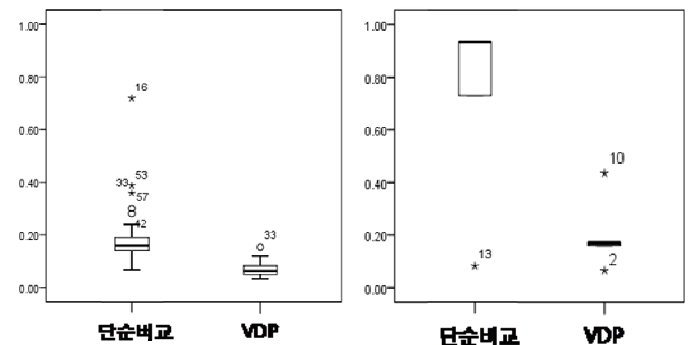
<표 4>는 구성 요소와 구성 요소의 위치 변화에 따른 화면 분류 방법을 나타낸다.

<표 4> 화면 분류 방법

	UI 요소	그래픽 요소
해상도 차이에 따른 위치 변화 없음	UI 로 구성된 화면이며 해상도 차이로 인한 구성 요소의 위치 변화가 없는 화면	그래픽 요소로 구성된 화면이며 해상도 차이로 인한 구성 요소의 위치 변화가 없는 화면
해상도 차이에 따른 위치 변화 있음	UI 로 구성된 화면이며 해상도 차이로 인해 구성 요소의 위치 변화가 발생한 화면	그래픽 요소로 구성된 화면이며 해상도 차이로 인해 구성 요소의 위치 변화가 발생한 화면

4.3. 구성 요소의 위치 변화가 없는 경우

(그림 4)의 상자 도표는 구성 요소의 위치 변화가 없는 경우의 UI 요소와 그래픽 요소로 이루어진 화면의 단순 비교 기법과 VDP 기법의 이도비를 보여준다. 이 그림은 VDP 기법이 화면이 어떤 요소로 구성되어있는지에 상관없이 구성 요소의 위치 변화가 없을 경우 화면은 동일하다고 판단하는 것을 나타낸다.



(가) UI 구성 요소 (나) 그래픽 구성 요소

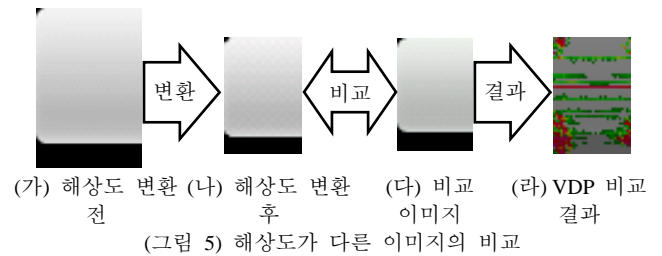
(그림 4) 구성 요소의 위치 변화가 없는 경우의 이도비 상자 도표 동일한 앱이 위치 변화 없이 실행 된 결과이므로 두 화면이 동일하다고 관별하는 것이 합당하다. 하지만 단순 비교 기법은 상대적으로 높은 이도비 측정값이 나타났다. 이러한 결과는 이미지 비교를 위하여 해상도 변환 과정에서 이미지 왜곡이 발생하고, 이로 인해 픽셀 값이 바뀌게 되므로 픽셀 값을 비교하는 단순 비교 기법으로는 정확한 결과를 얻기 힘들기

때문이다. 반대로 VDP 는 단순히 픽셀 값을 비교하는 것이 아니고 사람의 인지적 측면에서 화면의 차이를 측정하는 기법이기 때문에 이미지 왜곡의 변화에 강건하므로 낮은 이도비를 갖는다.

따라서 구성 요소의 위치가 변경 되지 않은 경우에 VDP 기법이 단순 비교 기법에 비하여 해상도가 상이한 화면의 동등성 판단 측면에서 우수함을 보였다.

* 해상도 변환으로 인한 이미지 왜곡 현상

(그림 5)는 이미지 비교가 가능하도록 해상도 변환 과정에서 발생하는 이미지 왜곡 현상을 보여준다. 사용된 이미지는 ApiDemos 의 화면 중 UI 요소인 그래픽 버튼의 일부이다. (가)는 옵티머스 2X 에서 잘라낸 이미지(48 x 72)이고 (다)는 동일한 부분을 옵티머스 원에서 잘라낸 이미지(32 x 48)이다. 즉, 전체 해상도를 1/20 로 줄인 이미지이다. (나)는 화면 동등성 판단을 위하여 (가)의 해상도를 변환하여 (다)의 해상도와 일치시킨 이미지이다. 그리고 (라)는 VDP 기법으로 비교한 두 이미지의 시각적 차이를 나타내는 이미지이다.



<표 5>는 단순 비교 기법과 VDP 기법을 사용하여 (나), (다)의 이도비를 측정한 결과이다. 단순 비교 기법의 측정값인 0.85 는 두 이미지 사이에 85%의 픽셀이 다른 값을 가진다는 뜻이므로 두 이미지가 다른 이미지임을 의미한다. 반면, VDP 의 측정값은 0.05 로 두 이미지를 사람의 인지적 차원에서 봤을 때에는 차이가 거의 없다는 것을 의미한다.

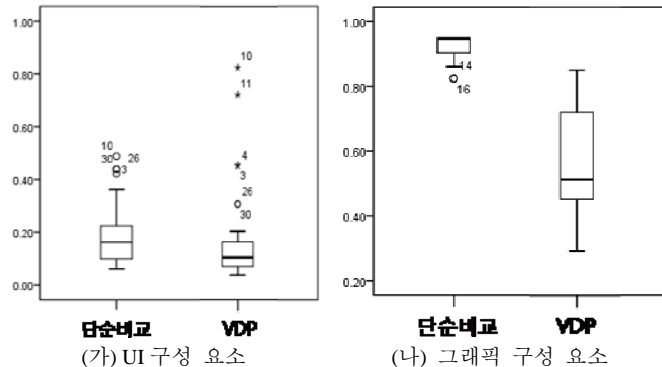
이렇게 (나)와 (다)의 단순 비교 시의 픽셀 값 차이는 (가)를 변환하는 과정에서 생긴 이미지의 기하학적 확대/축소 단계에서 발생하는 픽셀 값의 손실로 기인하는 것으로 판단된다.

<표 5>(그림 5) 이미지의 이도비 측정 결과

	단순 비교 기법	VDP 기법
이도비	0.85	0.05

4.4. 구성요소의 위치가 변환 경우

(그림 6(그림 6)의 상자 도표는 구성 요소의 위치가 변화한 경우의 UI 요소와 그래픽 요소로 이루어진 화면의 단순 비교 기법과 VDP 기법의 이도비를 보여준다.



(그림 6) 구성 요소의 위치가 변화한 경우의 이도비 상자 도표 (가) 에서 구성 요소의 위치가 다른 경우에 단순 비교 기법보다 오히려 VDP 기법을 사용하였을 때에 더 높은 값이 나온 경우가 있다. 이것은 VDP 기법이 화면 구성 요소의 위치가 두 화면에서 다르게 나타난 것을 잘 나타내는 것임을 보여준다. 따라서 테스터나 사용자에게 화면 구성 요소의 위치가 바뀌었음을 알려줄 수 있다. 그러나 단순 비교 기법의 경우에는 위치 변화에 덜 민감하게 반응하는 것을 볼 수 있다. 이것은 화면 구성 요소의 위치가 변화하였음에도 불구하고, 테스터나 사용자에게 화면 구성

요소의 위치 변화에 대한 아무런 정보를 줄 수 없음을 나타낸다. 따라서 구성 요소의 위치가 변경 된 경우에도 VDP 기법이 단순 비교 기법에 비하여 해상도가 상이한 화면의 동등성 판단 측면에서 우수함을 보였다.

4.5 타당도 검증

우리의 실험 결과는 한 쌍의 해상도가 다른 디바이스에 대해서만 진행 되었다는 한계가 있다. 그리고 75 개의 서로 다른 실행 화면을 대상으로 진행되었지만, 실험 대상 앱이 안드로이드 SDK 에 포함된 예제라는 한계도 존재한다.

5. 결론 및 향후 연구

현재 다양한 고객들을 만족 시켜주기 위해서 스마트폰 제조사들은 각기 다른 해상도를 가진 디바이스들을 출시하고 있다. 게다가 앞으로 스마트폰 뿐만 아니라 태블릿 디바이스도 상이한 해상도로 개발 중이거나 출시될 예정이기 때문에 이러한 해상도 차이로 인한 호환성 문제는 더욱 심각해질 것이다.

본 논문에서는 해상도 차이에 기인한 호환성 문제를 극복하기 위해 동일한 앱이 실행될 때 단순 비교 기법과 VDP 기법의 화면 동등성 능력을 분석하기 위한 실험 결과를 소개하였다. 실험 결과 VDP 기법이 단순 비교 기법에 비하여 화면의 동등성을 보다 정확하게 판별하였다. 즉, 디지털 이미지 왜곡에 강건한 VDP 기법이 해상도 변환에 따른 이미지 왜곡에도 구성 요소의 위치의 변화만 없으면 동등한 화면으로 판단하는 반면에 단순 비교 기법은 미세한 차이를 가진 동등한 화면을 이질적인 것으로 판단하였다.

동일한 앱을 실행 하였을 때 해상도의 차이로 사용자에게 다른 화면을 보여주는 것은 스마트폰 제조사나 앱 개발자가 의도하지 않은 소프트웨어의 호환성 오류라고 간주될 수 있다. 그러므로 호환성 테스트를 위해서 화면 동등성을 판단할 때에 VDP 기법이 현장에서 활용되면 스마트폰 앱의 호환성을 높이는데 기여할 수 있다고 생각한다.

향후에는 다양한 해상도를 가진 실제 판매되는 더 많은 디바이스들을 대상으로 화면 동등성을 판단하고, 시장에서 실제로 판매량이 높은 앱을 대상으로 VDP 기법의 판별 능력을 실험할 계획이다.

참고문헌

- [1] Chang, T.-H., T. Yeh, and R.C. Miller, GUI testing using computer vision, in Proceedings of the 28th international conference on Human factors in computing systems2010, ACM: Atlanta, Georgia, USA. p. 1535-1544.
- [2] Jaaskelainen, A., et al. Automatic GUI test generation for smartphone applications - an evaluation. in Software Engineering - Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on. 2009.
- [3] Daly, S., The visible differences predictor: an algorithm for the assessment of image fidelity, in Digital images and human vision, B.W. Andrew, Editor 1993, MIT Press. p. 179-206.
- [4] HDR-VDP. <http://www.mpi-inf.mpg.de/resources/hdr/vdp/index.html>. 2011.
- [5] Mantiuk, R., et al., Backward compatible high dynamic range MPEG video compression. ACM Transactions on Graphics, 2006. 25(3): p. 713-723.
- [6] Banterle, F., et al., A framework for inverse tone mapping. The Visual Computer, 2007. 23(7): p. 467-478.
- [7] Li, X., K.M. Lam, and L. Shen, An adaptive algorithm for the display of high-dynamic range images. Journal of Visual Communication and Image Representation, 2007. 18(5): p. 397-405.
- [8] Kim, B., et al., Prediction of perceptible artifacts in JPEG 2000 compressed chest CT images using mathematical and perceptual quality metrics. American Journal of Roentgenology, 2008. 190(2): p. 328.
- [9] Android. <http://www.android.com/>. 16, Feb, 2011; Available from: <http://developer.android.com/index.html>.
- [10] Harty, J., A Practical Guide to Testing Wireless Smartphone Applications. Synthesis Lectures on Mobile and Pervasive Computing, 2009. 4(1): p. 1-99.