

PanoGL projekt specifikáció

Polcz Péter

Leírás

Az általam készített panoráma alkalmazás képes fényképeket készíteni úgy, hogy a már elkészült fényképeket a képernyőn átlátszó módon megjeleníti és a szenzor adatok alapján az elkészített képeket a képernyőn a készülék forgatásával, mozgásával megfelelő módon pozicionálja. Eredményképpen a képernyőn a már elkészített képek olyan szögben és olyan pozícióban láthatóak majd, mint a valós 3D térben. Ideális esetben, ha csak a már elkészült fotókat nézzük, azok a képernyőt csak akkor látszanak teljesen, ha a készüléket abba a szögbe irányítjuk ahonnan az illető képeket készítettük. Ezt a műveletet 360 fokban el lehet végezni és eredményül egy gömb alakú képet (képek sokaságát) kapunk, amit ha a gomb középpontjából nézünk (és forgatjuk) olyan mintha az eredeti helyszínen szemlélőnk a valóságot. A képek kallibrációs adatait (szög, pozíció) elmenti, így a képek panorámaként visszanézhetőek egy forgatható OpenGL felületen.

Felhasználó interfész

de lehetővé szeretném tenni, hogy API level 8 on is fusson.

Mindemellett előfordulhatnak olyan kiegészítések, amelyek az API level 11 ben bevezetett feature-öket használja.

Fényképező activity (PanoGLCameraActivity)

Ahogy a neve is mondja, ebben az activityben lehet fényképeket készíteni.

A activity layoutja a következő rétegekből tevődik össze:

- fénykép előnézet (legalul)
- átlátszó módon a már elkészített képek egy átlátszó OpenGL felületen
- a fényképező activity gombjai

Az átlátszó OpenGL felületen a már elkészített képek átlátszó textúraként jelennek meg.

A fényképező gomb (középen) lenyomásával az a fényképelőnézet rövid időre megáll.

Galléria activity (GalleryListActivity)

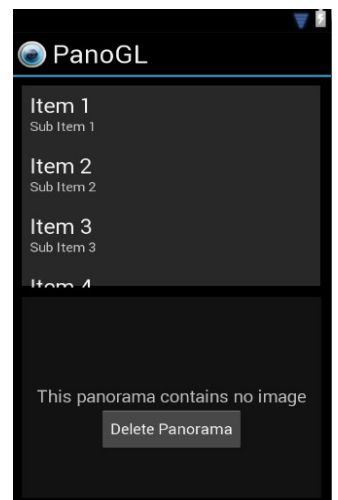
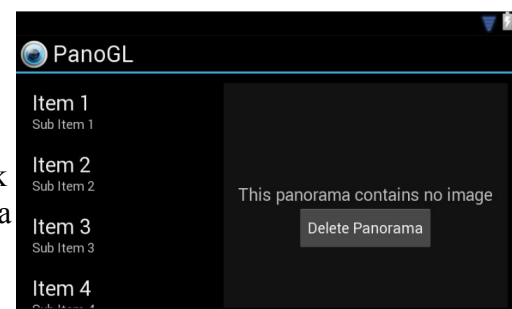
Itt láthatóak a már elkészített panorámák és az őket alkotó képek.

Minden egyes panorám egy elem a listában, melyet megnyomva az alul (balra) található ViewPager előhívja az őt alkotó képeket, melyek között jobbra-balra húzással lehet navigálni. Amennyiben a panoráma nem tartalmaz képeket egy külön layout jelenik meg a ViewPager helyén (lásd ábra), melyben az alkalmazás felajánlja a panoráma törlését. Hosszan nyomva egy listaelemet megjelenik egy menü, melyben a következők közül lehet választani:

- képek megtekintése teljes képernyős nézetben
- panoráma megtekintése (OpenGL felületen)
- panoráma törlése

Ha nincs egy panoráma se, akkor egy külön layout jelenik meg, melyben az alkalmazás felajánlja egy új panoráma készítését.

Az “action bar”-ban található egy fényképező icon melyet megnyomva elindul a fényképező activity, de előbb egy dialógus ablak ugrik fel, kérve az új panoráma nevét.



Megjegyzés: egy elkészített panorámához több kép már nem készíthető, mivel kevés esély van arra, hogy a felhasználó ugyanabban a pozícióban, található a fényképezés során mint amikor előzőleg már elkészítette a panorámát, így az új képek teljesen el fognak ütni a régiektől.

Activity a fényképek megtekintésére teljes képernyős nézetben (GalleryFullScreenViewActivity)

Ebben egy ViewPager látható ahol a jobbra-balra navigálva nézhetőek a képek, és két újjal nagyíthatóak.

Panoráma visszanező activity (PanoGLViewActivity)

A panorámát alkotó képek visszanezhetőek egy 3D forgatható OpenGL felületen, ahol minden egyes kép OpenGL textúraként jelenik meg abban a szögben és pozícióban amelyben készült.

Fényképek tárolása

Minden egyes panorama képsorozatra külön mappát hozok létre, amelyben tárolom a képeket is és a hozzájuk tartozó kalibrációs adatokat is. Továbbá a panorámák nevét, fényképezés dátumát és a mappa nevét SQLite adatbázisban tárolom.

Sensor adatok lekérése

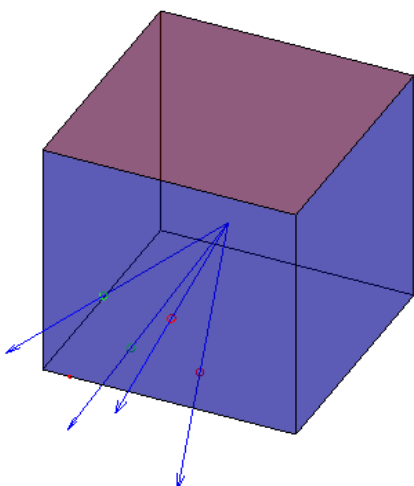
Egy külön csomagot csak a szenzor adatok lekérése és transzformációjára fordítottam. A szenzor zajok csökkentésére egy aluláteresztő szűrőt (integrátort) használtam, így fényképezés közben az OpenGL felület modellje nem oszcillál folyamatosan.

A készüléknek amelyen az alkalmazást feljesztettem, nem rendelkezett se giroszkóp, se mágneses szenzorral, így készülék azimuth irányait nem volt lehetőségem lekérni. A probléma ellensúlyozásaképpen lehetővé tettem, hogy fényképezés közben azimuth (észak-dél-kelet-nyugat) irányokban kézzel lehet forgatni a modellt annak érdekében, hogy illeszkedjenek az elkészített képek a fényképnézethez.

Képek összeillesztése

Első megközelítésben a képeket illesztés nélkül oda helyeztem, amilyen szögben készült. Ezt a következő képpen valósítottam meg:

1. szenzor adatok segítségével forgattam az OpenGL modellt, ha a fényképezőt az ég felő irányítom, akkor az OpenGL felületen is a gömb (kocka) felső fele látható.
2. Kiszámolom a projekciós és a modell-nézet transzformációs mátrixokat, melyeket invertálva megkapom a képernyő négy sarkának origó körüli szögét, majd a kapott vektort metszem a kocka oldalaival és a metszéspontok fogják megadni a textúra sarkait.
3. A kapott 3D négyszögre textúrázom a fényképet.



Megjegyzés: OpenCV segítségével megvalósítható, képeket Affine SIFT (Scale Invariant Feature Transform) algoritmus segítségével egymáshoz regisztráljuk és egy nagy képek készítsünk belőlük, melyeket egy gömbre textúrázva azt a hatást érhetjük el mint ami az egyik demó activityben látható.

Demók

A menüből elérhető két demó:

- PanoView Demo: szemlélteti, hogy milyen lesz a panoráma, ha majd sikerül összeilleszteni a képeket egy nagy (2:1 méretarányú) panoráma képpé
- CameraActivity Demo: szemlélteti a Fényképező activity felületének rétegződéseit, amelyeket a fentiekben már említettem. Vegyük észre, hogy ha a készüléket landscape-portrait irányokban forgatom akkor az android icon továbbra is függőleges állásban marad.

Tesztelés

Az alkalmazást HTC Desire 500 készüléken, Android 4.1.2 rendszeren fejlesztettem és teszteltem.

Az osztályok állapotainak vizsgálatára és az elő-, utó feltételek teljesítésének ellenőrzésére "assert"-eket használtam.

Minimum és cél SDK

Az alkalmazást célirányosan 18 as szintű API-ra fejlesztettem, és minden olyan feature-t, amely 8 as szintű API-ban nincs meg, azt egyéb alternatívákkal helyettesítettem.

Eredeti specifikációtól való eltérés

- A panoráma képet nem gömbre, hanem egy kockára textúráztam, azonban ez megfelelő transzformációk segítségével látvány szempontjából egyenértékű a gömbi textúrával.
- Időhiány miatt a képeket nem sikerült egy nagy panorámába összefésülni, ehelyett a képeket egyenként textúráztam a megfelelő négyszögre. Nincs megoldva a képek közti fokozatos átmenet, de a képek panoráma jellege így is előjön.
- Nem csináltam külön tablet layout-ot és az activity-k nincsenek fragmentekbe rendezve, mert a 3D panorámát mindenképpen teljes képernyős nézetben akartam megvalósítani. Továbbá így az activity-k közötti navigálás is egyszerűbben megvalósítható volt.
- Azimuth irányú automatikus forgatás nincs a rendelkezésemre álló készülék hiányosságai miatt.

További fejlesztési, tesztelési lehetőségek (becsült fejlesztési idő)

- Alacsony API szintű (8 - 10) Samsung készülékeken az alkalmazás Fényképező funkciói nem működnek. Ezt korrigálni szeretném. (1 hét).
- Giroszkóp és/vagy mágnesesség szenzorral felszerelt eszköz segítségével lekérni és megfelelő alakra transzformálni és ezt a Fényképező activitybe beépíteni (1 nap).
- ADB monkey teszt végzése
- Képek panorámába való összeillesztése OpenCV, NDK segítségével (1 hónap)

Felhasznált kódok

- (1) Teljes képernyős képnézegető
info.androidhive.imageslider.helper
<http://www.androidhive.info/2013/09/android-fullscreen-image-slider-with-swipe-and-pinch-zoom-gestures/>
- (2) OpenGL ES panoráma nézegető
org.openpanodroid
<https://github.com/duerrfk/OpenPanodroid>
Saját kontribúciók, kiegészítések:
 - GLSurfaceView átlátszósága
 - forgatás z-tengely irányában
 - szenzor irányította forgatás
 - átlátszó textúrák
 - runtime kiegészülő textúrák fényképezés közben
 - transzformációs mátrixok inverzének kinyerése

Segítségemre voltak

- (1) de.onyxbits.sensorreadout
<https://github.com/onyxbits/sensorreadout>
- (2) com.kviation.android.sample.orientation
<https://github.com/kplatfoot/android-rotation-sensor-sample>
- (3) 3D Graphics with OpenGL ES
http://www.ntu.edu.sg/home/ehchua/programming/android/Android_3D.html