

# Automatizált panoráma fénykép készítő

## Android rendszerre

## Specifikáció

Polcz Péter

October 16, 2013

### Leírás

A felhasználó fényképeket készíthet egy adott pozícióból különböző szögekben, és minden fényképezés után a program összeregisztrálja a képeket a gyroscope adatok alapján. A felhasználó minden fényképezésnél halovány színnel a háttérben láthatja a már elkészített fotókat, úgy, hogy a készülék forgatásával a háttérben megjelenő már elkészült fotók automatikusan haladnak a fényképezendő tényleges objektumokkal együtt. 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 kapnánk, amit ha a középpontjából nézünk (és forgatjuk) olyan mintha az eredeti helyszínen szemlélnénk a valóságot,

### User interface tervek, elképzelések

#### Minimum, cél SDK

Az alkalmazást célirányosan API level 18 ra szeretém fejleszteni, 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

Az alkalmazásnak egy activity-jét csak a fényképezés folyamatára szeretém fordítani, amelyben két SurfaceView-t szeretnék majd megjeleníteni egymáson úgy, hogy egyik takarja a másikat legyen átlatszó.

Az egyik SurfaceView fogja a kamera előzetesét (camera preview) ábrázolni a másik SurfaceView pedig egy OpenGL ES segítségével írt View lesz amelyben a textúrázott gömböt fogjuk látni, melyet a megfelelő szögben felnagyítón. Tehát reményeim szerint a felhasználó a fényképezés alatt nem fog tudomást venni arról, hogy ő tulajdonképpen nem is egy panorámát készít, hanem egy gömb belső felületére vetít képeket.

#### Fényképek tárolása

Minden egyes panoráma képsorozatra külön mappát szeretnék majd létrehozni, amelyben tárolom a képeket is és a hozzájuk tartozó kalibrációs adatokat is. Azt még nem döntöttem el, hogy az OpenGL modellt elmentem-e külső fájlba, ha gyorsan betölti a képek és a kalibrációs adatok alapján, akkor nem.

#### Activity a fényképek és panoráma megtekintésére

A fényképek és a panoráma megtekintésére egy külön activity-t szeretnék létrehozni, amelynek tervrajza lentebb látható. Lehetőséget szeretnék nyújtani mind a különálló képek, mind a panoráma teljes képernyős megtekintésére. Ezt két különálló activity segítségével szeretném megvalósítani.

A panorámát egy OpenGL ES működtette SurfaceView-ba szeretném helyezni, amely 3 touchscreen gesture-re lesz figyelmes:

- horizontális gesture egy ujjal - forgatás a látóhatár mentén
- vertikális gesture egy ujjal - forgatás földtől az ég felé és vissza





- körkörös gesture két ujjal - forgatás a képernyő síkjára merőleges tengely körül

Megjegyezném, hogy a harmadik fajta forgatást a készülék elforgatásával is elérhetjük, így időhiány esetén ezt ki fogom hagyni az alkalmazásból.

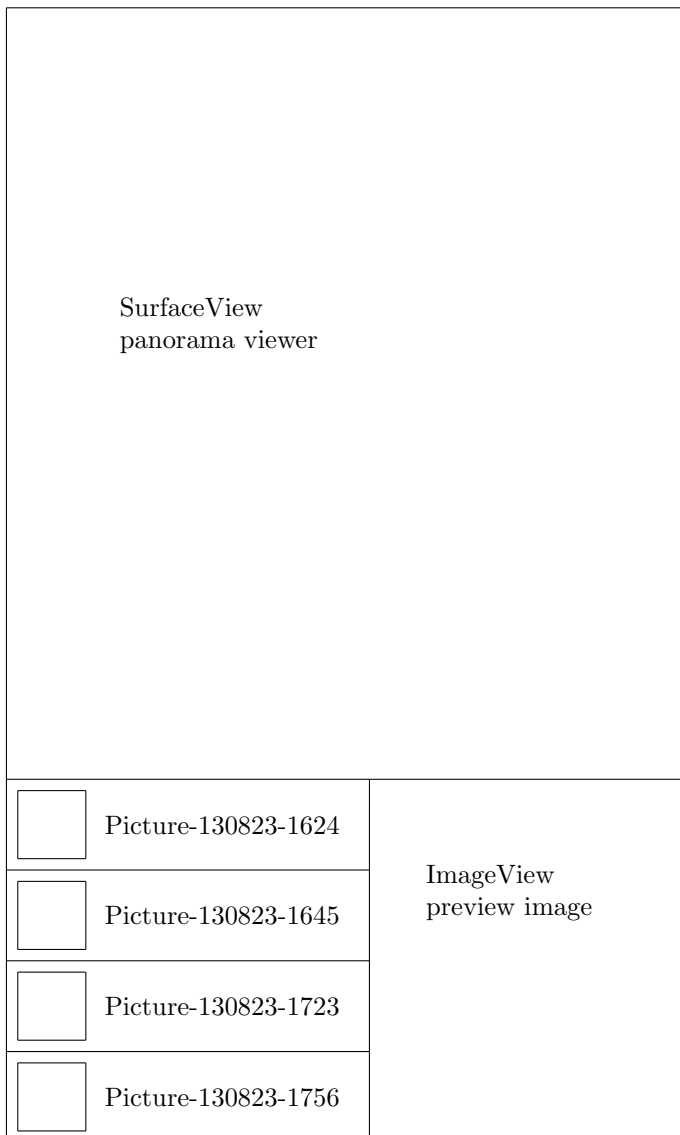
### Képek összeillesztése

Első körben azt szeretném megvalósítani, hogy a rendelkezésre álló gyroscope adatok alapján a képet oda helyezzem, ahova a kalibrációs adatok (pitch, roll, azimuth) szólnak. Átlapolódás esetén homályosításokkal próbálom takarni az esetleges nem egyezést, úgy hogy minnél inkább belóg egy pixel a másik képbe, annál inkább homályos. Ez az eljárás egy folytonos kevésbe zavaros összképet fog eredményezni.

### UI tablet landscape nézetben

 Picture-130823-1624	<div>SurfaceView panorama viewer</div>
 Picture-130823-1645	
 Picture-130823-1723	
 Picture-130823-1756	
<div>ImageView preview image</div>	

## UI tablet portrait nézetben



## UI telefon

Telefon esetén a fragment activity két része (a 3D viewer és a listview-preview) nem egyszerre fog megjelenni, hanem a menüből lehet navigálni egyikből a másikba.

## Fullscreen activity-k

A fényképezés procedúrája, a panoráma és a képek teljes képernyős megjelenítése mind-mind külön activity-kben fognak megjelenni.

## Esetleges extra kiegészítések

Ha az alkalmazást az eddigi specifikáció szerint sikerült megvalósítani, akkor a képek összehasonlítására megpróbálunk egy SIFT (scale invariant feature transformation) algoritmus leimplementálni. Az OpenCV lehetőséget biztosít a jellemző pontok kinyerésére. A feladat két kép feature pontjainak egyeztetése és egymáshoz rendelése. Mivel ez egy CPU igényes művelet, ezt csak háttérben futó szálakon érdemes implementálni.

A feladat komplex és nehéz úgy megvalósítani, hogy közel real-time legyen.