

Szakmai gyakorlat Koblenzben

Pálfalvi István

Idén tavasszal (nyáron), áprilistól július végéig Koblenzben, az ottani egyetemen voltam szakmai gyakorlaton.

Tavaly már jártam Koblenzben, akkor tanulmányi ösztöndíjjal jutottam ki, és egy projektbe kapcsolódtam be. (Akkori beszámolóim is megtalálható az oldalon, így most inkább a gyakorlat résszel foglalkozok.) A gyakorlati helyre is itt sikerült rátalálnom, megkérdeztem az egyetemen, hogy nem mehetnék-e oda gyakorlatra, és azt a választ kaptam, hogy mehetek, így ismét külföldön töltöttem a félév egy részét.

A gyakorlat dátumát én dönthettem el, mivel viszont a karunkról Bakos Krisztina is Koblenzbe ment (Ő tanulmányi félévre), ezért Neki adott volt az áprilisi kiutazás, így gondoltam én is akkortól kezdem majd meg a 3 hónapomat, amiből végülis 4-et csináltam.

A szálláskereséshez érdemes segítséget kérni kintről, mert onnan sokkal egyszerűbb találni valamit, már csak azért is, mert az itthonról küldött e-mailekre (meg igazából egyáltalán e-mailre) nemigen válaszoltak. (Aki válaszolt, az általában azt válaszolta, hogy már nem aktuális az ajánlat, amit az interneten találtam.¹) Ilyenkor jól jön, ha az ember ellakhat egy ismerősnél egy kis ideig, amíg nem talál szállást.

Megérkezésemkor megkérdezték tőlem, hogy szeretném-e esetleg a tavalyi projekt képfeldolgozás részét folytatni, vagy inkább egy új projektbe csatlakoznék be, én inkább az új projektet választottam, mert szerettem volna mással foglalkozni, mint tavaly.

Így kaptam azt a feladatot, hogy írjak egy programot különböző tárgyak vizsgálatának elvégzésére. A vizsgálat alatt ebben az esetben azt kell érteni, hogy van egy nagyobb doboz, amiben van két fajta fénycső, és egy villanykörte, így azt lehet vizsgálni, hogy különböző megvilágítások mellett milyen színűnek látjuk a tárgyakat. Ezt kiegészítették azzal, hogy beletettek a dobozba egy forgótányért, így a tárgyakat el is tudják forgatni, és több oldalról meg tudják nézni. A tárgyakról a különböző megvilágításokkal, és elforgatásokkal egy kamera segítségével képet készítenek, és ezeket elmentik egy képadatbázisban. Az elmentett képeket a tavaly megismert robot projektjénél is tudják használni, mert most már olyan részén is indulnak a versenyeknek, ahol a cél a mindennapi élet megsegítése a lakásban, így meg lehet mondani a robotnak, hogy mit keres a gazdája, a robot pedig keresni kezdi, és az elforgatott képek alapján jobb eséllyel találja meg, mert a való életben sem biztos, hogy pont előlről, vagy hátulról lehet a

¹ www.studenten-wg.de

tárgyat látni.



1. kép: A doboz és a forgótányér

A programnak vezérelnie kell a forgótányért és a világítást a dobozban, kezelnie kell a kamerát, egy-egy képet venni róla, és az adatbázisban kiválasztott objektumhoz elmenteni a képet, természetesen ezt mind grafikus felület használatával.

Ezt linux alatt kérték tőlem (az egyetemen általában Open SuSE-t használnak, a diákok legtöbbször Ubuntu-t, én is ez utóbbi táborhoz tartoztam, már előtte is), a Qt nevű grafikus eszközkészlet használatával, ami c++-ban íródott, így kézenfekvő volt, hogy a programozás nyelve is a c++ legyen. A programozáshoz használt fejlesztői környezetet nem szabták meg, ők általában a Kdevelopot használják, én inkább az eredetileg java fejlesztésre készített, azóta több más programozási nyelvhez is használható Eclipse c++-os változatát (Eclipse CDT) használtam.

Először össze kellett forrasztanom egy relékből álló egységet, amit egy

mikrokontrolleren keresztül a világítás kapcsolgatásához használtunk. Következő lépésként a mikrokontrollert kellett tudnom vezérelni. Ehhez adtak cd-n windows és linux alatt használható példaprogramokat, és drivereket, ezeket megnézve sikerült nekem is csinálni egy programot, ami tudta vezérelni a mikrokontrolleren lévő LED-eket. A következő lépés az volt, hogy egy olyan változatú programot kellett írnom, ami már grafikus felhasználói felülettel rendelkezik. Ez kicsit több időt vett igénybe nekem, mert először a Qt régebbi változatával próbálkoztam, majd megtudtam, hogy az újabbal kéne, aztán az vett el még sok időt, hogy nem tudtam egyszerűen egy alap felhasználói felülettel rendelkező programot csinálni, amit tudtam volna bővíteni, ahogy kell. Ekkor kezdtem használni az Eclipse-et, amihez volt Qt-s kiegészítés, és tudott nekem olyan új programot csinálni, amit már tudtam könnyen módosítani. (Azóta már a nehezebb módosíthatósággal is elboldogulok; a könnyebb egy felülettervező program, a nehezebb a forráskódból történő változtatás.)

Ezután a forgótányért kellett tudnom irányítani. Kaptam egy c-ben írt programot, amivel irányították régen, de lefordítani nem tudtam, mert sok függősége volt egy másik projektből, amihez hozzáfértem, de az is feladatomból volt, hogy legalább részben, de megszabadítsam tőle (ne kelljen a másik projektet is feltenni azért, hogy ezt lehessen használni). Így néhány fájlt átmásoltam, más részeket, amik nem kellettek kizsedtem, és végre sikerült lefordítani a programot. A program futása közben, amikor kiadott egy parancsot, akkor némi várakozás után egy hibaüzenetet adott, ami alapján megpróbáltam változtatni a program paraméterein, de az eredmény nem lett más. A forgótányért vezérlő kontrollert egy RS232 – USB átalakítóval csatlakoztattam a számítógéphez. Próbáltam megtalálni a hibát, hallottam már, hogy az ilyen átalakítókkal vigyázni kell, mert nem mindegyik működik megfelelően, de esetemben egy terminál programmal a soros portra csatlakozva láttam, hogy a kontroller mindenféle információt kiír, és nem láttam köztük hibaüzenetet. Valamivel később úgy láttam, hogy az adatok csak jönnek a mikrokontroller felől, de felé nem megy semmi. Még néhány sikertelen próbálkozás után beszámoltam a projektvezetőnek erről, akinek megmutattam a programot, azt is, hogy a kontrollerről jönnek adatok, azt is, hogy a program hibát ad. Ő sem értette, hogy miért nem működik, aztán gondolt egyet, kicserélte az átalakítót egy másikra, és onnantól kezdve jól működött minden, lehetett irányítani a forgótányért. Megcsináltam, hogy be lehessen állítani, hogy mekkora lépésenként álljon meg, vagy hogy hány képet akarunk csinálni.

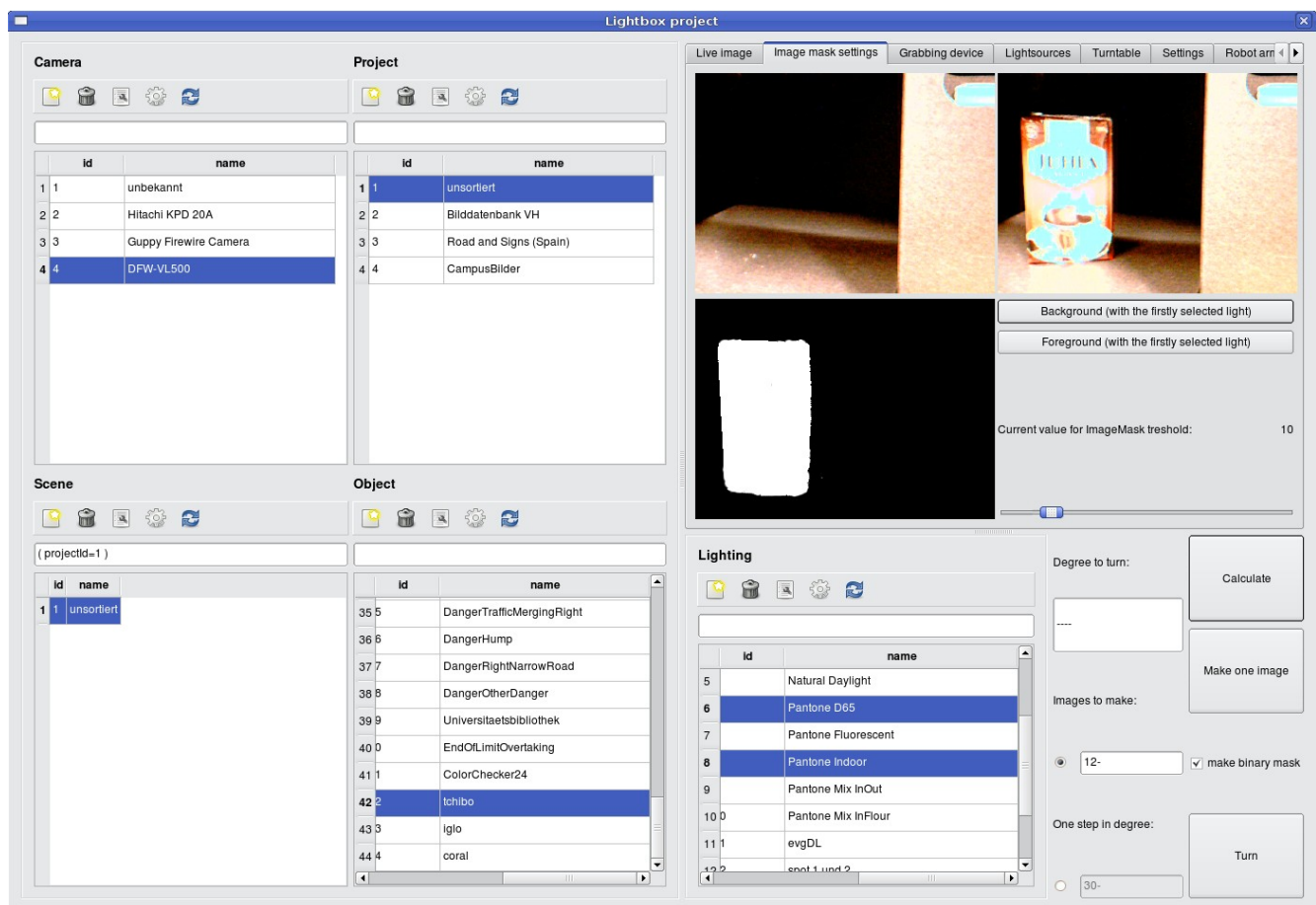
A következő lépés az volt, hogy a program tudjon képet venni egy kameráról. Ehhez először egy firewire csatlakozóval ellátott kamerát használtam. Szerencsére nem kellett nagyon kutakodnom, hogy hogyan is kell ilyesmit csinálni, mert az egyetemen fejlesztenek egy képfeldolgozáshoz használható függvénykönyvtárat (PUMA²), melynek van kamerakezelő része is, kérték, hogy használjam én is ezt. Ez tartalmaz egy egyszerű

2 <http://www.uni-koblenz.de/FB4/Institutes/ICV/AGPaulus/puma>

kamerakezelő példaprogramot is élő képpel, és végül is azt vettem alapul, beépítettem az én programomba.

Ezután megcsináltam, hogy forgás közben mentse le a képet egy fájlba, így tudtam tesztelni, hogy jól működik-e, és észre is vettem egy hibát, ami abból adódott, hogy az élő képet egy időzítő frissítette, viszont fordulás közben nem tudott újra és újra lefutni (ez programból lett megakadályozva, hogy ha valamiért sokáig tart az eljárás (ebben az esetben a forgatás miatt nem tud másik eljárás futni, mert csak egy futási szál van a programban), ne hívja meg újra addig, amíg véget nem ért), így a kívánt kép előtti volt, amit lementett a program. Ezt előbb úgy védtem ki, hogy egy képet fölöslegesen kértem, és a másodikat mentettem le, majd megcsináltam, hogy a forgatás egy másik szálon fusson, így a kép folyamatosan frissülhetett, mindig az aktuálisat tudtam lementeni.

Ezek után kiderült az is, hogy nem csak a képre van szükség, hanem egy maszkra is, ami megmondja, hogy a képen hol helyezkedik el az objektum. Ehhez a képsorozat készítése előtt az egyik világitással (amelyikkel az első sorozatot készítjük) kell csinálni egy képet az objektum nélkül, majd a forgatás közben, minden elkészített képből kivonjuk a háttérrel, és a megmaradt rész lesz a maszk, ezután bekeretezzük a maszkot, és kiszámoljuk a bal felső és jobb alsó koordinátáknak az egész képhez viszonyított százalékos elhelyezkedését.



2. kép: A program futás közben

Feladatom utolsó része a képi adatbázis használatának megoldása volt. Ezt egy ottani diák írja, Ő készítette az adatbázis-komponenseket is, melyeket használtam a programban. Egy példaprogramot is írt, az alapján tudtam beépíteni a programba az adatbázis-kezelést, és kicsit később fejleszteni az egyik részt. El lehet menteni, hogy melyik kamerával, melyik tárgyról, melyik világítással, milyen elfordulás mellett készült a kép, valamint azt is, amit már az előbb kiszámoltunk, hogy az objektum hol helyezkedik el a képen (maszk, és pozíciója).

Összességében nagyon tetszett a projekt, sokat tanultam belőle, c++-ból, Qt-ből, adatbázisokból, és kamerakezelésből is fejlődtem, és láttam egy-két olyan megoldást, ami segíthet abban, hogy szebb, jobb kódot tudjak írni. Ha most kezdeném a projektet, akkor egy-két dolgot már nem pont ugyanígy csinálnék, szívesen újra is írnám a program egyes részeit, hogy most már a követelmények tudatában tervezhessem a működést, és ne csak alakuljon menet közben, mert nem tudtam meg minden részletet az elején, volt, ami csak közben derült ki, vagy közben alakult ki, olyan is volt, amit megkérdeztem, hogy hogyan csináljak, és aztán mégis át kellett írnom a másik megoldásra.



3. kép: Nem kért enni...

A munka mellett azért jutott idő pihenésre is, bár volt olyan időszak, hogy az ottani Erasmusos diákokkal szinte minden hétvégére volt program. Ezek közül a legjobban talán a kölni kirándulás tetszett, nagyjából 12-en mentünk rá, reggel indultunk vonattal, és délután értünk vissza, közben sétáltunk a városban, állatokat etettünk :), megnéztük a kölni dómot, egy-két múzeumot, egy-két templomot, és a Rajna-partot is, mindegyik szép volt. :)

Mindenkit csak bátorítani tudok, hogy menjen ki külföldre néhány hónapra, akár tanulmányi, akár szakmai gyakorlatos ösztöndíjjal, mert sokat lehet tanulni belőle, sok tapasztalatot lehet szerezni, és még az önéletrajzban is jól mutat. :)

Pálfalvi István