

Mobiltelefon alkalmazások siket felhasználóknak

FELDHOFFER GERGELY, BÁRDI TAMÁS, JUNG GERGELY, HEGEDŰS IVÁN MIHÁLY
Pázmány Péter Katolikus Egyetem, Információs Technológia Kar
{ flugi, bardi, junge, hegivmi}@digitus.itk.ppke.hu

Cikkünkben egy olyan rendszert írunk le, ami lehetővé teszi siket vagy hallássérült emberek beszédérzékelését olyan esetekben, amikor nincs lehetőségük szájról olvasni. A célunk egy hordozható készülék, lehetőleg mobiltelefon fejlesztése, ami beszédhangból közvetlenül egy modellezett száj vagy fej mozgását tudja előállítani. Ez a cikk a fejlesztés mobiltelefon specifikus részleteire koncentrálna.

1. Bevezető

A ma kapható legfejlettebb mobiltelefon készülékek gyors processzoroknak, nagy tárhelyeknek és multimédiás képességeik eredménye képpen igen sokrétűen használhatóak. A már megszokott hang és üzenetküldő szolgáltatások mellett fényképezőgépként, szövegszerkesztőként, és hamarosan televízióként is működnek. A készülékekben rejlő lehetőségek a már meglévő hasznos és kevésbé hasznos alkalmazásokon túlmutatnak. A gyártók és a szolgáltatók üdvözlnek minden jó elképzelést, amely valóban hasznos és vonzó funkciókkal egészíti ki a telefonjaikat. Ilyen jellegű fejlesztésekre igyekeznek ösztönözni külső cégeket, szervezeteket.

A PPKE ITK-n az elmúlt évben egy kutatás-fejlesztési projektbe kezdtünk, melynek fő célja, hogy hallássérült felhasználók mobiltelefonos kommunikációs lehetőségeit javítsuk, minél jobban kiaknázva e készülékek képességeit. A beszéd teremti talán a legközvetlenebb és hatékonyabb kommunikációs kapcsolatot ember és ember között, ezért nem meglepő, hogy a telefon eredendően, és legnagyobb részben még ma is beszéd átvitelére szolgál. Azonban a hang információ a siketek és nagyothallók számára hozzáférhetetlen.

A projektünk célja, hogy beszédből egy okos telefon által más modalításban megjeleníthető információt képezzünk, amely tényleges segítséget jelenthet a siketek hallókkal való kapcsolat tartásában. Konkrétan egy a telefon grafikus kijelzőjén megjelenő animált beszélő fejre gondoltunk, melynek szájmozgásairól a beszéd leolvasható. [1] A másik lehetőség egy a telefonhoz csatlakoztatható taktilis kijelző, mely a tapintási érzékünkre hat. Mindkét elképzeléshez szükség van valós időben működő beszédhangelemző modulra, ezért első lépésben ezt készítettük el. Jelen cikkünk tárgya ez a Symbian operációs rendszerre fejlesztett program, mely a

beérkező beszédjelből a legtöbb automatikus felismerő rendszerben használatos mel-kepsztrum sajátásvektorokat képezi. Érdemesnek látjuk beszámolni a fejlesztési folyamat mobiltelefonos környezet adta nehézségeiről, sajátosságairól.

2. A mobiltelefon mint platform

Egy átlagos ma kapható okos telefonban 100-200 MHz-es órajelű processzor van, de a leggyorsabbak akár 250 MHz vagy ennél nagyobb órajel frekvenciával működnek, és belső memóriájuk 64 MB is lehet. A 90-es évek elején az első hordozható személyi számítógépeknek ennél jóval kisebb volt a teljesítményük. Például az első kereskedelmi forgalomban megjelent laptop, a NEC Ultralite-nak csak 8 MHz-es órajele volt. Nem meglepő tehát, hogy napjaink telefonjai a legkülönbözőbb alkalmazások futtatására képesek.

A mobiltelefonokra írt alkalmazások gyors terjedésének a másik oka, hogy több rendszerhez is létezik nyilvános fejlesztőkörnyezet. Ilyen a Symbian OS is, de ilyennek tekinthető bármely Java MIDP futtatásra alkalmas készülék is. Létezik ezen kívül a PocketPC család, ami Windows CE vagy Windows Mobile rendszert használ, ezek egyelőre mobiltelefon funkcióval nem terjedtek el, továbbá a PalmOS alapú kézi számítógépekre sem jellemző egyelőre a GSM kommunikáció. A Symbian-os készülékek megjelenése előtt a telefonon futó programokat kizárólag a gyártók készítették, és a felhasználó utólag nem tudott azokra újabbakat telepíteni. A szabványosított nyílt rendszerek ezzel szemben lehetővé teszik, hogy a felhasználó harmadik fél által készített alkalmazásokat telepítsen, így létrejött a telefonon futó programok piaca. [2][3]

2.1. A Symbian mobil operációs rendszer

A Symbian konzorciumot a Nokia, az Ericsson, a Psion és a Motorola alapította 1998-ban. A Symbian-t úgy tervezték, hogy kompakt hordozható eszközök operációs rendszereként működjön. A mobil piacon az első sikert a 2000-ben bemutatott 6.0 verzió aratta. A 7.0-ás verzió egy már kifejezetten mobil telefonokra optimalizált rendszernek számított. Az idén megjelent 9.0-ás változat a 3G mobil hálózatokon is működtethető.

Ahogy a telefonok piaca fejlődik, az alkalmazott keretrendszereknek alkalmazkodniuk kell a növekvő igényekhez. Ez olyan fejlesztési eljárás létrejöttéhez vezetett, ahol az operációs rendszerek szinte készülékről készülékre változnak. Miközben a jó operációs rendszer egyik fontos tulajdonsága a hordozhatóság, jóformán alig találni két egyformán programozható telefonmodellt, mert egymást érik a fejlesztések. Egy konkrét példa erre a Nokia telefonok Series 60-as sorozatának (S60) fejlődése. Az S60-as szabványnak a 2.0 verziója 2003-ban, a 3.0-ás 2005 közepén jelenik meg, és a kettő között háromszor bővült a rendszer olyan szolgáltatásokkal, melyek az előző szabványokban nem szerepelnek. Szerencsére ezek a sorozatok visszafelé kompatibilisek.

2.2. Az alkalmazott készülékek

Két típuson dolgoztunk: Nokia 7610-esen (1. ábra) és Sony Ericsson P910-esen. Mindkettőnek StrongARM processzora van. Ez egy RISC processzor, amit csekély tárigényűre és fogyasztásra kialakítottak ki. A gépi kód mérete ilyen processzorokon töredéke az x86-osokénak. A Nokia 176x208-as, és az Ericsson 208x320-as képfelbontása alacsonynak tűnhet, de ha a kijelző méretét is figyelembe vesszük, DPI-ben ezek a folyadékkristályos kijelzők a modern monitorokéval egy kategóriába esnek.



1. ábra Nokia 7610

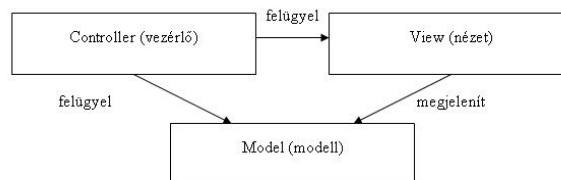
3. Fejlesztés Symbianra

A hagyományos gyakorlat azt várná, hogy a platform hordozza a fejlesztőeszközöket, de nem nehéz belátni, hogy a mobiltelefon nem a legalkalmasabb eszköz programszöveg kezelésére. Ehelyett keresztfordító programokat használnak, amelyek lehetővé teszik, hogy PC-n megírja a fejlesztő a programot, és lefordítsa a készülékre.

A folyamat kényelmesebbé tehető, ha van egy emulátor az eszközhöz, amivel az alkalmazást már a PC-n ellenőrizhető. Az említett gyors piaci fejlődés az operációs rendszerre való fejlesztésben viszont nehézségeket okoz. Az emulátorok általában csak a nagy verzióugrásoknál készülnek el, ezért gyakori, hogy egy alkalmazás csak telefonon fut, az emulátoron viszont nem, mert kihasznál az emulátorból még hiányzó szolgáltatásokat. Az ellenkező eset is fel-lephet, az emulátorban működő program nem feltétlenül fordítható le az eszközre, mert PC-n jóval enyhébb megkötések vannak például a verem méretére. Összességében az emulátorokról azt lehet mondani, hogy segít az apró fejlesztési lépésekben, de nem garantálja a helyes működést.

Az operációs rendszer szolgáltatásait C++ osztály hierarchiába szervezték, támogatva az objektumorientált programozást. Nem ritka, hogy egy szolgáltatás igénybe vételéhez a hiányzó lényegi részeket absztrakt osztálytól kell örökölni, és megvalósítani.

Az operációs rendszer szolgáltatásait ugyanis nem egy egyszerű API, hanem egy teljes fejlesztői könyvtár formájában nyújtják, ahol a programot előre adott szerkezetben érdemes felépíteni. Ez a Model - View - Controller (MVC) architektúra (2. ábra), ahol a program tartalma, ki- és bemeneti lehetőségei, és képe egymástól elkülönül. Ezt a szerkezetet ösztönzi, hogy a Symbian rendszer rendelkezik ösosztályokkal, amikből az MVC komponensek örökölhetők, és így a főbb kontrollfolyamatokat átvállalják a programozótól. Ez nagymértékben hasonló a mai ablakkezelést támogató PC-s fejlesztőkörnyezetekhez.



2. ábra Az MVC modell

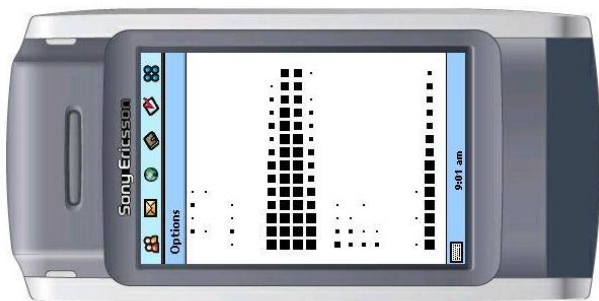
Mindazonáltal a szabványos C++ nyelvnek nem teljes egésze használható a Symbian fejlesztőkörnyezetben. Az STL (Standard Template Library) nem használható, ezért az eredetileg a C++ szabványhoz tartozó könyvtár helyett a Symbian saját

konténerait ajánlják. Az STL hiányát azóta egy a Symbiantól független csoport orvosolta, és elkészült az STL port, ami egyszerűbb rendszereszközökkel valósítja meg az STL szolgáltatásait. Azonban a Symbian még a C++ nyelv szabályait is szigorította, melyek egy része nem feltétlenül látszik szükségesnek. A legfőbb lehetőségek, amiktől a Symbian megfosztja a C++ programozót, a kivételkezelés, a többszörös öröklődés egy része, és több kényelmes lehetőség az objektumok kezelésére. A megszorításokat a Symbian alkotói minden esetben a hardver korlátaival magyarázzák, ami a gyorsan fejlődő eszközök mellett több esetben nehezen érthető.

3.1. A hangkezelés nehézségei

Egy olyan alkalmazásnak, amely hang elemzésével képi és taktilis információkat állít elő, bizonyos alapkövetelményeknek meg kell felelnie. Képesnek kell lennie a hang folytonos feldolgozására, lehetőség szerint minél kisebb átmeneti tárolóval, ügyelni kell arra, hogy a rendszer késleltetése ne növekedjen. A taktilis információk közvetítéséhez a hardverre és a szoftverre egyaránt szükség van. A hang elemzéséhez memória- és processzorkímélő algoritmusok olyan kialakítására van szükség, ami a kritikus követelményeknek megfelel.

A Symbian első verziói a hang kezelésében a felvételre, és a lejátszásra szorítkoztak, ilyenkor egy függvényhívással megkezdődött a rögzítés, egy másik hívással befejeződött. Ez a működés kényelmetlen a folyamatos hangkezeléshez, nincs kettős puffereles, és így a szakasz felvételének befejezése és a következő kezdése közötti idő elvész. Az újabb verziókban ezt a területet kibővítették, és így lehetőség nyílik audio stream használatára, ahol már csak a callback függvényt kell megírni, amivel a megtelt puffer tartalmát feldolgozhatjuk. Az általunk használt telefon rendszere nem tartalmazza ezt, ezért egy hiánypótló MdaAudioInputStream könyvtárat használtunk, ami kettős puffereles nélkül tud folyamatosan hangot kezelni.



3. ábra A spektrum elemző futás közben

A grafika az egyik legkényelmesebben használható terület a Symbian OS programozásában. Mivel a programunk beleágyazódik az Ősosztályokba, a

kirajzolásért felelős programrészek egyszerűen kezelhetők.

A mobiltelefonok egy része szabványos csatolófelülettel rendelkezik, ahol lehetőség van tetszőleges hardver csatlakoztatására. Ez adta az ötletet, hogy a siketek körében sikerrel alkalmazott taktilis kijelzők, mint például a rezgőmotorral ellátott gyűrű, részét képezze a rendszernek. Folyik a kutatás egy olyan vibrátor építésére, ami olyan hangelemző eljárás eredményeképpen rezeg, ami a siketeknek a legjobban feldolgozható információt adja.

3.2. A programmotor elkészítésének folyamata

Az egyik sarkalatos pont a fejlesztési terv kidolgozásakor a hibakeresési módszerek kialakítása. Sajnos a Symbian rendszerek, mint elsősorban grafikus rendszerek, az alfanumerikus kijelzésben a kijelző természete miatt korlátozottak, ráadásul a szöveges és numerikus adatok keverése a konzol alapú környezetekhez képest bonyolult programkódot kíván. Ezért az a döntés született, hogy a külön tesztelhető részeket PC-n készítjük el, és az így kapott programmotort utólag építjük a rendszerbe.

Így készült el az FFT algoritmus is, amit DOS/Windows környezetben készítettünk el, majd a jól ellenőrzött kódot az MVC szerkezetnek megfelelően illesztettük a Symbian rendszerre írt programba. Ez a módszer zökkenőmentesen működött a mel skála programozásánál is. Mivel a hangból képet előállító rendszert még kutatjuk, egyelőre a telefonokra nem készítettünk további elemző részeket.

4. A beszédelemző

A bejövő beszédjelet a programunk 40 ms hosszú átfedés nélküli ablakokra bontja. Hamming ablak alkalmazása után a mel spektrum szerint 16 sávban átlagoljuk az intenzitást. A kiszámított értékeket a kijelzőn a sötét színű négyzetek nagyságával érzékeltetjük, a 3. ábrán látható módon. Számítási szempontból a beszédelemző lelke a gyors Fourier transzformáció. Ehhez a Decimation-in-time Radix-2 FFT algoritmust írtuk át Symbian-ra.

A Radix-2 algoritmus előnye a kis kódméret, ami mobiltelefonon fontos szempont. A Radix-4 és más FFT változatok csak néhány százalékkal gyorsabbak, viszont a kódméretük többszöröse a Radix-2-nek. További előny, hogy 2-hatvány méretű tömbökön tud dolgozni, és nem csak 4-hatvány méretűen.

Az eljárás verzióinkban 32 bites lebegő pontos számábrázolást használ (TReal32), ez a pontosság a céljainkhoz elegendő. A telefonok processzorait jellemzően nem a lebegő pontos műveletek gyors elvégzésére tervezték. Egy fix-pontos számokkal

dolgozó FFT sokkal gyorsabban lefutna, de akkor a számértékek túlcsoordulását megakadályozandó túl sokat kellene engedni a numerikus pontosságból.

A tárhasználat csökkentése érdekében néhány DSP programozásban használatos trükköt vetettünk be. Az egyikre azért van szükség, mert az FFT komplex számokat vár bemenetként, viszont az ablakozott beszédjel valós. A legegyszerűbb megoldás az lenne, hogy az imaginárius részt feltöltjük mindenütt nullákkal, ami viszont a tárgyigényt fölöslegesen megkétszerezné. Inkább a valós tömbünket egy fele annyi elemből álló komplex tömbként értelmezzük, és erre számolunk FFT-t. Ebből lineáris időben dolgozó algoritmussal egyszerűen ki lehet számítani az eredeti valós vektor Fourier transzformáltját. A másik speciális megoldás, hogy az algoritmus „helyben” számol. Ez azt jelenti, hogy az összes műveletet a bemenetként kapott tömbön hajtja végre, ebbe kapjuk a végeredményt is, és nem kell átmeneti tömböket pluszban lefoglalni. Az ilyen helyben dolgozó FFT-k jellemzően bitreverz sorrendben adják ki a Fourier transzformáltat, amit utólag lehet egy egyszerű rutin meghívásával rendes sorrendbe átrendezni. Mindezeket a belső műveleteket egy C++ osztályba csomagoltuk, hogy az objektumot használó programozó számára láthatatlannak maradjanak.

5. Összegzés

A mobiltelefonok mérete, kijelzője, és programozási kapacitása elegendő ahhoz, hogy további hasznos alkalmazásokat lehessen a siketek számára fejleszteni. A további kutatásoktól várjuk a választ arra a kérdésre is, hogy van-e az erőforrásokkal dolgozó algoritmusok között olyan, melynek hibaaránya már elfogadhatóan csekély.

A következő kutatási fázisban egyrészt algoritmusokat fogunk összehasonlítani, pontosság és erőforrás használat szempontjából, másrészt a kijelző hatékonyságát kívánjuk növelni. A pontosságot személyi számítógépeken és mobiltelefonokon futtatott tesztekkel vegyesen, siketekből álló tesztcsoporton fogjuk meghatározni.

Ezzel a technológiával lehetségessé válhat beszédalapú alkalmazások sora, akár beszélőazonosítás, akár felismerési feladatok területén. A folyamat teljesen a telefonkészüléken fut, a szolgáltatótól nem vesz igénybe többlet számítási kapacitást.

Köszönetnyilvánítás

A szerzők hálása az NKTH támogatásáért, a SINOSZ közreműködéséért és hasznos tanácsaiért,

valamint a T-Mobile-nak, hogy a készülékeket rendelkezésünkre bocsátotta. Külön köszönjük Dr. Takács Györgynek az értékes segítségét és a témavezetést.

Irodalom

- [1] Bárdi T., Feldhoffer G., Harczos T., Srancsik B., Szabó G. D.: „Audiovizuális beszéd-adatbázisok és alkalmazásaik”, Híradástechnika, 2005.
- [2] Charaf H., Csúcs G., Forstner B., Marossy K.: Symbian alapú szoftverfejlesztés, Szak, 2004.
- [3] J. Stichbury: Symbian OS Explained, Wiley 2005.