

BABES-BOLYAI TUDOMÁNYEGYETEM
MATEMATIKA-INFORMATIKA KAR
INFORMATIKA –MODELLEK OPTIMIZÁLÁSA- MESTERI

Tárgykövető robot

IX. Erdélyi Tudományos Diákköri Konferencia - Kolozsvár, 2006. november 25-26

Tudományos vezető

Soós Anna

BABES-BOLYAI TUDOMÁNYEGYETEM
STATISZTIKA ES NUMERIKUS SZAMITAS KAR
ELOADOTANÁR

Magiszteri hallgató

Máté István-Zoltán

Kolozsvár

2006

1.1 Az eszközök leírása:

1.1.1 A párhuzamos port

A számítástechnikában a párhuzamos portot (Parallel Port) interfésznek tekintik a számítógépes rendszer és egy külső hardware eszköz közt. A párhuzamos port lehetővé teszi különböző külső hardware eszközök és a számítógép kommunikációját. A párhuzamos port általában a számítógép hátán van elhelyezve.

A párhuzamos port egy bitet (Bit), közvetít minden egyes szálon, és nem többet egy adott időegységen belül, mint például a soros port (Serial Port).

A párhuzamos porton az adatátviteli bitteken kívül található pár jelzőbit, amelyek célja általában annak az időpillanatnak a meghatározása amikor a külső hardware eszköz készen áll a működtetésre. Manapság az USB (Universal Serial Bus) felváltotta, a párhuzamos portot, és sok ma gyártott számítógépen (2006) már mellőzik a párhuzamos portot, pénzspórlás céljából, és azért mert használata elavultnak tekintett.

Felépítése:

<u>DB-25 Pin</u>	<u>Jel</u>	<u>Irány</u>	<u>Port</u>	<u>Bit</u>
1	Strobe*	ki	#Command	0
2	Data0	ki	#Data	0
3	Data1	ki	#Data	1
4	Data2	ki	#Data	2
5	Data3	ki	#Data	3
6	Data4	ki	#Data	4
7	Data5	ki	#Data	5

8	Data6	ki		#Data	6
9	Data7	ki		#Data	7
10	Ack*	be		#Status	6
11	Busy	be		#Status	7
12	Paper	ki	be	#Status	5
13	Select	ki	be	#Status	4
14	Auto_Feed*	ki		#Command	1
15	Error*	be		#Status	3
16	Init*	ki		#Command	2
17	Select in*	ki		#Command	3
18 to 25	Ground	NA		NA	NA

A párhuzamos portnak, van 8 bit, csak adatátvitelre használt pin, 2 bit, amelyen keresztül a számítógép kommunikál a külső eszközzel, és van 2 bit, amellyen keresztül a külső eszköz hibát jelez a számítógépnek.

1.1.2 A léptetmotrok

A léptetőmotor ahogy neve is mondja, egy olyan motor amely egy bizonyos lépésszöggel képes fordulni jobb, illetve bal irányba. A léptetőmotor abban különbözik a hagyományos motortól, hogy ez kis lépésekben képes fordulni, míg egy hagyományos motor fordulatait percenként mérik (fordulat/perc), nem képes “apró” lépésekre.

A léptető motrok TEAC 14769070-10 típusu 5.25" lemez meghajóból (Floppy Drive) származó motrok.

A motrok 12V-al (Volt) működnek, a motrok vezérlésével egy *interfész* foglalkozik, amely célja a párhuzamos portról érkező jelt a motrok működésére fordítani.

A motrok vezérlése egy 5 szál as adatkábel en keresztül történik, amelyből 1 földelés, és a többi 4 lehetővé teszi a motor bizonyos irányba való elmozdulását.

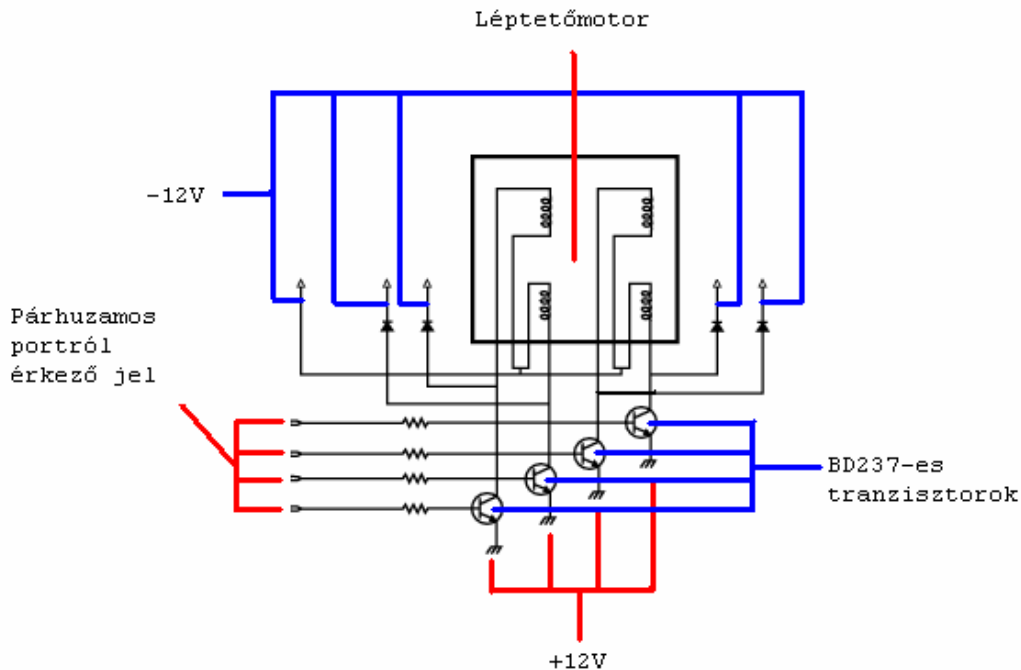
1.1.3 Az interfész:

Az interfész egy olyan elektronikus szerkezet, amely a párhuzamos portról érkező jelt a motrok mozgatására alkalmas jellé alakítja.

Az interfész alapján véve kapcsolóként működik, amely a párhuzamos portról érkező 3.8 V-os feszültséget 12V-ra kapcsolja át.

Felépítésében motronként 4 tranzisztor, 4 ellenállás, és 4 dióda szerepel. Az ellenállás szabályozza a tranzisztorhoz érkező feszültséget, a tranzisztor a párhuzamos portról érkező feszültségkor nyit, és a 12V-os feszültséget átadja a motornak.

1.1.3.1 Az áramkör felépítése:



Amint a fenti ábrán látható, az áramkör gyakorlatilag egy kapcsolóként működik, amikor a párhuzamos portról feszültség érkezik, akkor a tranzisztorok nyitnak, és a 12V-os feszültség a motorokba áramlik. A párhuzamos portról érkező feszültség pár ellenállás segítségével van szabályozva, hogy a tranzisztorok működésének megfelelő legyen. Az áramkörben szereplő 4 dióda szerepe, pedig az, hogy megakadályozzák a rövidzárlat létrejöttét, ugyanis, amikor a tranzisztorok nyitnak, akkor a +12V-os feszültség zárlatot okozna az áramkörben keringő -12V-os feszültséggel. Az ellenállások 1K-osok, az alkalmazásban használt diódák manapság már nemigen használt üvegdiódák.

1.1.4 A kamera:

A projektben használt kamera, USB porton csatlakozó, Logitech Quickcam Express. A kamera 352x288 pixeles (Pixel, színtonkomponens alapegysége a számítógép képernyőjén) elbontásban képes képeket közvetíteni, és 15 fps (Frame Per Second) a frissítése.

Egy lényeges probléma a kamerával, hogy mivel kis felbontású képet közvetít, és egy olcsó kameratípus, teljesítménye nagyon függ a fényviszonyoktól. Nappali fény használatakor az érkezett kép jó minőségű, viszont gyenge fényviszonyok esetén, a képen nagy lesz a zaj, ez gyakorlatilag abban nyilvánul meg, hogy a kép “bolházik”.

1.5 A tárgy felismerése:

1.5.1 Bevezető:

A tárgyak meghatározása a mesterséges intelligencia témakörében a '60-as évektől kezdődően jelentett problémát, amikor a számítógépek fejlődése lehetővé tette az effajta kérdések felvetődését, de egészen a '80-as évekig nem lehetett komolyan foglalkozni ezzel a témával, mert a számítógépek nem tették lehetővé a képfeldolgozást. A tárgyak felismerése, és követése a '90-es évektől kezdődően kihívást jelentett, viszont a hétköznapi számítógépeken történő követés nem volt lehetséges, bár a tárgyfelismerés már lehetséges volt statikus képek esetén. A valós időben történő feldolgozás nehézségei miatt, ami elsősorban a nagy mennyiségű adatfeldolgozást jelentett valós időben, sokáig nem volt lehetséges ezzel a témakörrel foglalkozni. Különálló hardware egységek létrehozásával már lehetett számítógépes tárgykövetést megvalósítani a '80-as évektől, de ez általában csak a nagy cégek, és vállalatok esetében volt lehetséges, mivel egy hardware egység létrehozása rengeteg pénzt és mérnöki munkát igényel. Az első tárgykövetési alkalmazások az amerikai katonaságtól származnak, tárgykövető rakéták alkalmazásaiban fordultak elő a '80-as évek közepétől. A számítógépek egyre nagyobb ütemben való fejlődése lehetővé tette, hogy lehetséges legyen a tárgykövetés hétköznapi felhasználóknak is. Tárgyfelismerés terén rengeteg algoritmus létezik, egy részük a tárgy alakján alapuló tárgyfelismerést alkalmaz. Ez a módszer hátránya a nagy számítási adat feldolgozása, és az algoritmusok nagy számítási igényei, ami az alkalmazás lelassulásához vezet. Ezeket a módszereket általában több számítógépes rendszereken alkalmazzák valós időben való követésre, így érik el a megfelelő teljesítményt. Ezek a módszerek előnye, hogy a tárgyat egyértelműen meghatározzák. Ezt a módszert általában fekete-fehér kamerák esetében alkalmazzák, mert kevesebb számítást kell végezni a tárgy meghatározásához, és mivel nem lehet másképp egyértelműen meghatározni a tárgyakat fekete-fehér színmezőben. Egy másik módszer, amelyet

színes kamerás alkalmazások esetében használnak szín-sűrűségeen alapuló módszerek, de ezen kívül rengeteg más típusú algoritmus van. Az általam alkalmazott módszer alapja a tárgy színhisztogramjának elkészítése.

Ahhoz, hogy megadhassunk egy tárgyat először szükségünk van az adott tárgy tulajdonságaira. Elsődleges szempontok amiket figyelembe kell venni, hogy milyen színű tárgy, milyen színmezőben érhetjük el a színeket, milyen színmezőben akarjuk feldolgozni a képet, RGB, vagy cían színmezőben. A szín tulajdonságon kívül egy tárgyat jellemez a mérete, három dimenziós megjelenése. Mivel egy kameráról érkező kép gyakorlatilag két dimenziós, gyakorlatilag a környezetünkben található világot csak az ember látja három dimenziósnak, ez komoly problémákat jelent. Elsősorban, hogyan lát az ember három dimenziós képeket, hogyan tudja meghatározni egy kép mélység komponensét, és hogyan lehet ezt a számítógéppel is elérni ?

1.5.2 A színhisztogram:

1991-ben Michael J. Swan és Dana H. Ballard “Color Indexing” című publikációjukkal egy olyan módszer alkalmazását mutatták be, amely azóta majdnem tárgyfelismerő alkalmazásban használnak. Ennek a módszernek a mai nap több ezer módosítása is megtalálható, viszont mindegyiknek alapja a színhisztogram.

Ezeket a továbbfejlesztett változatokat egyaránt használnak statikus képekkel dolgozó képfeldolgozó alkalmazásokban, mint a számítógépes látással foglalkozó robotikában. Ezt a módszert azért kellett továbbfejleszteni, mert az egyre fejlődő számítógépes alkalmazások, eddig nem látott valóság-hű modellezést, és feldolgozást tettek lehetővé, és az egyre realisztikusabb helyzetek feldolgozása megkövetelte ezt. A különböző fényviszonyváltozásoktól, fedésektől és más hasonló akadályok leküzdéséért, szükség volt ezt a módszert továbbfejleszteni.

Ezen eljáráson alapuló módszereket használnak több kamerás automata biztonsági berendezések működtetésére, utakon felszerelt biztonsági kamerák, amelyek képesek egy autó számtábláját meghatározni, a nagy városokban felszerelt biztonsági kamerák, amelyek képesek az emberi arcok felismerésére.

1.5.2.1 A színhisztogram alkalmazása:

Legyen a kameráról érkező kép a következő alakú:

$$M = \begin{pmatrix} p_{11} & K & p_{1n} \\ \vdots & \vdots & \vdots \\ p_{m1} & L & p_{mn} \end{pmatrix}$$

ahol p_{ij} egy pixel, és

$i \in [1..n], j \in [1..m]$, n a kameráról érkező kép szélessége, m a kameráról érkező kép magassága.

Ezen M -en belül definiálunk egy

$$N = \begin{pmatrix} p_{i_0, j_0} & K & p_{i_0+w, j_0} \\ \vdots & \vdots & \vdots \\ p_{i_0, j_0+h} & L & p_{i_0+w, j_0+h} \end{pmatrix}$$

mátrixot, ahol

$$i_0 \in [1..n], j_0 \in [1..m], \text{ és } w \in [i_0..n - i_0], h \in [j_0..m - j_0].$$

Ez a figyelő ablak, amely tetszőleges lehet, a kameráról érkezett képen belül. w a figyelő ablak szélessége, h a figyelő ablak magassága.

Legyenek ezek $R_{k,l}$, $G_{k,l}$, $B_{k,l}$ a $p_{k,l}$ pixel komponensei, ahol $k \in [i..i+w]$, $l \in [j..j+h]$.

Abban az esetben, ha 24 bit per pixeles formátumban dolgozunk, akkor ezek értékei:

$$\begin{aligned} R_{k,l} &\in [0..255], \\ G_{k,l} &\in [0..255], \quad \mathbf{(3)} \\ B_{k,l} &\in [0..255]. \end{aligned}$$

A hisztogram matematikailag egy olyan leképezést jelent, amely számolja, hogy hány megfigyelés esik egy adott kategóriába. Ezeket a kategóriákat általában intervallumokkal szemléltetjük.

A színhisztogramon alapuló osztályozása a tárgy színkomponenseinek ennek alapján a következőképpen néz ki:

$$rh_i = \left\lfloor \frac{p_{kl}}{255} \left(R_{kl} \hat{I} \left[255 \frac{n-i}{n}, 255 \frac{n-(i+1)}{n} \right] \right) \right\rfloor_{i_0 \leq k \leq i_0+w, j_0 \leq l \leq j_0+h}$$

$$gh_i = \left\lfloor \frac{p_{kl}}{255} \left(G_{kl} \hat{I} \left[255 \frac{n-i}{n}, 255 \frac{n-(i+1)}{n} \right] \right) \right\rfloor_{i_0 \leq k \leq i_0+w, j_0 \leq l \leq j_0+h}$$

$$bh_i = \left\lfloor \frac{p_{kl}}{255} \left(B_{kl} \hat{I} \left[255 \frac{n-i}{n}, 255 \frac{n-(i+1)}{n} \right] \right) \right\rfloor_{i_0 \leq k \leq i_0+w, j_0 \leq l \leq j_0+h} \quad (5)$$

ahol $i \in [0..n]$.

Ha felépítettünk minden szükséges eszközt arra, hogy tudjuk osztályozni a színeket komponensenként, akkor az algoritmus a következőképpen alakul: minden egyes $a_{k,l}$ pixelre, ahol $k \in [i..n-i]$, $l \in [j..m-j]$ ellenőrizzük, hogy $R_{k,l}$ melyik rh_i intervallumba esik, ahol $i \in [0..n]$, és hasonlóan $G_{k,l}$ estében, melyik gh_i intervallumba esik, és $B_{k,l}$ -re.

1.5.2.2 Színhisztogramok metszete:

Ha adott az M és I hisztogram, akkor valamilyen távolságfüggvényt kell hozzájuk rendelni. Ez gyakorlatban a színkomponensek különbségének abszolút értékéből képezett összeg négyzetgyöke, de használható bármilyen vektornormából származó metrikát is. Például

$$d(M, I) = \sqrt{([Rh^M, Gh^M, Bh^M] - [Rh^I, Gh^I, Bh^I])^t \times ([Rh^M, Gh^M, Bh^M] - [Rh^I, Gh^I, Bh^I])}$$

A használt távolság nem kötelező matematikai értelemben metrika legyen.

1.5.2.2 Megjegyzések:

A színhisztogramos eljárást alkalmazhatjuk a teljes figyelőablakra, vagy a figyelőablakot feloszthatjuk, és a figyelőablak felosztásaira alkalmazzuk. Amennyiben a teljes figyelőablakra alkalmazzuk, fennállhat az a lehetőség, hogy két teljesen különböző kép esetében ugyanazt a színhisztogramot határozza meg a rendszer. Igazából ez a színhisztogram legnagyobb hibája, ugyanis az osztályozás közben elmarad a megfigyelés egyik fontos tulajdonsága, a megfigyelt tárgy alakja, fomája. Ennek a kiküszöbölésére a színhisztogramot csak soronként alkalmazzuk a figyelőablakra, és a színhisztogramok metszetét soronként végezzük, ezáltal nem veszítve szem elől egy fontos tulajdonságot, a tárgy felépítését, és színeinek eloszlását a képen.

1.6 A tárgy követése:

A tárgykövetés annak a problémának a megoldása, amely esetén meg kell becsülni egy tárgy pozícióját, és más releváns információkat amelyek a tárgyhoz kötődnek képsorozatok esetén.

1.6.1 Kalman szűrő:

A szűrő neve a szűrő feltalálójától származik, vagyis Rudolf E. Kalmantól, bár Peter Swerling kidolgozott már egy hasonló szűrőt korábban. Stanley Schmidtet ismerjük mint az első ember aki alkalmazta a Kalman szűrőt. Ez 1958-ban történt amikor Stanley lehetőséget kapott arra, hogy az Apollo programon belül alkalmazást nyerjen. A szűrő papíron Swerling által volt kifejlesztve(1958), majd Kalman által (1960), és Kalman és Bucy által (1961). A Kalman szűrőnek egy nagy sokaságát fejlesztették ki napjainkig, az eredeti Kalman szűrőb'ul, amelyet ma egyszerű Kalman szűrőnek hívunk. Manapság a Kalman szűrőt használják a távközlésben, és gyakorlatilag minden bonyolultabb elektronikai eszközben.

Egy mozgást végző tárgy lehetséges pozíciójának meghatározására alkalmas egyszerű és jól működő módszer a Kalman szűrés (*Kalman Filter*). Ez a módszer egy bizonyos

valószínűséggel adja meg a tárgy helyzetét, és becslést ad a tárgy következő időpillanatbeli pozíciójáról.

Ezeknek a módszereknek az előnye, hogy egy lineáris egyenlet megoldásával relatív gyorsan meg lehet adni a tárgy helyzetét.

A Kalman szűrőt sikeresen alkalmazták különböző becslésen alapuló alkalmazások esetén. A számítógépes látás egyik fontos része a tárgykövetés. Különböző mozgási feltételek és elnyelések akadályozhatják a tárgykövetést. Az alábbiakban a Kalman szűrő használata kerül bemutatásra. A Kalman szűrőt úgy építjük fel, hogy kisebb elnyeléseket is lehetővé tegyen.

Alkalmazható egyaránt statikus illetve nemstatikus környezetekre. A megoldás rekurzív, mivel minden frissített becslési pillanat az előző becslési pillanattól származik. Tehát csak a közvetlenül előtte levő állapot kerül mentésre. Amiatt, hogy a Kalman szűrő nem igényli csak az előző időpillanat információit, sokkal kevesebb számítást igényel, mint ha az összes előző állapotok információiból számolna, az szűrés mindenik pillanatában.

Az alábbiakban be lesz mutatva egy bevezető a Kalman szűrők alkalmazása számítógépes látás esetén.

1.6.2 A Kalman szűrés lépései:

1. Lépés: *Inicializálás*

($k=0$) Ezen lépés esetén a tárgy keresése a teljes képen történik, mivel nem tudjuk a kezdeti pozícióját. Ennek alapján meghatározzuk x_0 -át. A hibát ebben az esetben maximálisnak engedélyezzük, $P_0 = 1$.

2. Lépés: *Becslés*

($k>0$) Ebben az esetben a Kalman szűrő segítségével ténylegesen megbecsüljük a tárgy relatív(mivel abszolút pontosan nem meghatározható) pozícióját. Ebben az esetben \hat{x}_k -t tekintjük a kereső ablak középpontjának.

3. Lépés: *Korrekción*

($k > 0$) Ebben az esetben meghatározzuk a tárgy tényleges pozícióját, amely az \hat{x}_k^- becült pozíció környezetében található. A mérési eredmény segítségével alkalmazzuk a korrekciót, ennek alapján megkapva a tárgy tényleges pozícióját, és meghatározva \hat{x}_k^+ -t.

A 2. és 3.-as lépések a követés alatt vannak meghatározva.

1. Lineáris mozgás esetén:

Egy lineáris mozgást végző tárgyat szeretnénk követni. Ebben az esetben a mozgásegyenlet a következő alakban írható:

$$x_{k+1} = F_{(k+1,k)}(x_k) + w_k$$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ dx_{k+1} \\ dy_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ dx_k \\ dy_k \end{bmatrix} + w_k$$

$$y_{k+1} = H_k(x_k) + v_k$$

$$\begin{bmatrix} x'_k \\ y'_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ dx_k \\ dy_k \end{bmatrix} + v_k$$

A Kalman szűrő előnye, hogy kisebb elnyeléseket tolerál, tehát ha valamilyen oknál fogva a tárgy nem látható, a Kalman szűrő működését nem akadályozza, továbbra működni fog az addig meghatározott paraméterekkel.

1.7 Könyvészet

1. Real-Time Kernel-Based Tracking in Joint Feature-Spatial Spaces -Changjiang Yang, Ramani Duraiswami, Ahmed Elgammal and Larry Davis cs-tr.pdf
2. Object Tracking and Kalman Filtering - Hai Tao lec15.pdf
3. Kalman filter for vision tracking - Erik Cuevas, Daniel Zaldivar and Raul Rojas tr-b-5.pdf
4. Bayesian Filtering and Integral Image for Visual Tracking – Bohyung Han Changjiang, Yang Ramani Duraiswami, Larry Davis wiamis2005.pdf
5. Object Recognition with Color Cooccurrence Histograms - Peng Chang,John Krumm - Microsoft Research ColorConcurrenceCVPR.pdf