Single view metrology

Juhász Réka, Papp László, Pintér Csaba, Soponyai György

2005. december 8.

Kivonat

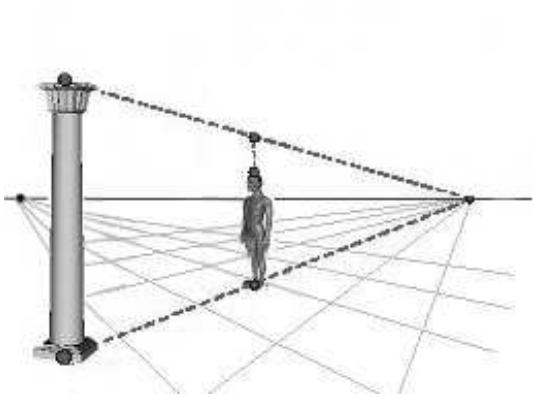
Jelen munka az SZTE TTK-n meghirdetett Számítógépes Látás gyakorlati kurzus 4-es számú beadandó programjának, a *single view metrology*-nak végső dokumentációját és kezelésének leírását tartalmazza.

A projekt tagjai

név	**évfolyam**	**email cím**
Juhász Réka	PTM V.	juhasz.reka@stud.u-szeged.hu
Papp László	PTM V.	papp.laszlo.4@stud.u-szeged.hu
Pintér Csaba	PTM V.	pinter.csaba@gmail.com
Soponyai György	PTM V.	soponyai.gyorgy@stud.u-szeged.hu

A megoldandó probléma

A *single view metrology* a számítógépes látás és képfeldolgozás egyik problémája, ahol egy perspektivikus fotó által ábrázolt jelenet térbeli mértékeit szeretnénk meghatározni. Ehhez szükséges megadni a kép egy referenciaobjektumának pontos méretét (pl. az alábbi képen az oszlop magasságát) és ezen ismert adat birtokában meghatározható a kép más objektumainak mérete (az ember magassága).



A *SVM*-rendszer működése

1. A bemeneti képet *(input image)* paraméterként megkapja egyrészt a C++ nyelven megírt grafikus felülettel ellátott kezelőprogram *(GUI+I/O)*, másrészt a MatLab.

2. A MatLab egy algoritmus-szkriptet értelmezve meghatározza a kép vanishing point-jait, melyek koordinátáit egy szöveges fájlban *(input data)* elmentve paraméterként megkapja a kezelőprogram.

3. A kezelőprogram irányítja az input/output műveleteket, a felhasználó számára interaktívan biztosítja a képen való objektumkijelöléseket valamint az esetleges input adatok – referenciaobjektum mérete – bevitelét.

4. Az *input data*-t feldolgozza a program, majd egy listában megjeleníti a kép vanishing point-jait, melyből a felhasználó kijelölheti azt a kettőt, melyek együttese megadja a mérés alapjául szolgáló vanishing line-t.

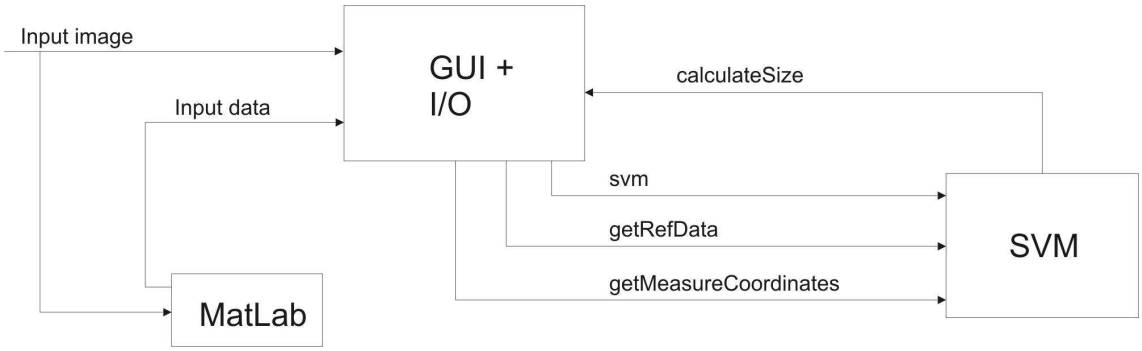
5. Amint a két vanishing point ki van jelölve, a *GUI+I/O* létrehoz egy *SVM* objektumot az `svm` konstruktor meghívásával.

6. A képen kijelölünk egy referenciaobjektumot, megadjuk méreteit majd az adatokat átadjuk az *SVM* objektumnak a `getRefData` metódus segítségével.

7. Tetszőleges számban kijelöljük a mérendő objektumokat azok tető–, és talppontjaik megadásával, majd átadjuk az adatokat az *SVM* objektumnak a `getMeasureCoordinates` metódussal.

8. Az *SVM* objektum kiszámolja a mérendő objektum méretét. Az eredményt a `calculateSize` metódus segítségével szolgáltatja a *GUI+I/O* rendszernek, amely megjeleníti az eredményt.[1]

Architektúra-terv



MatLab szkript

A programunk által használt MatLab szkript internetes forrásból származik[2]. Feladata egy inputként megadott kép vanishing point-jainak detektálása és a *GUI+I/O* modul számára ezen adatok biztosítása egy fájlban eltárolva. A *VP*-detektálás lépései a következők:

1: Az input kép Canny–éldetektálása (`findlines.m`)
2: Az él-képre Hough–transzformáció végrehajtása
3: A Hough-trafó által generált egyenesek metszéspontjai közül a potenciális vanishing point-ok kigyűjtése (`findvanishingpoints.m`)

A teljes algoritmus az *(általunk implementált)* `vanishing_point.m` szkript elindításával futtatható, amely a következő 4 paraméterrel rendelkezik:

1. A beolvasandó kép neve

2. Az output fájl; a detektálandó vanishing pointok tárolási helye *(a fájl specifikációját lásd lejjebb)*

[1] Az utolsó két lépés tetszőleges sokszor elvégezhető
[2] http://www.dia.uniroma3.it/ iarusso/Seminari%202004 2005/TesinePrededenti/aa0405

3. { **true** színes kép esetén
   false szürkeárnyalatos kép esetén
   a paraméter feltüntetése azért szükséges, mert színes *(RGB)* képek esetén el kell végeznie a programnak egy szürkeárnyalatosra konvertálást...

4. { **true** ha a kép által ábrázolt jelenet zárt térben (egy szobában) fekszik
   false „szabadtéri" képek esetében

A szkript egy lehetséges paraméterezése a következő:

```
vanishing_point('fenykep1.jpg', 'fenykep1.dat', false, false);
```

Előállhat olyan eset, amikor a vanishing point-ok nem jó pozíciókban helyezkednek el a detektálás pontatlansága miatt. Ekkor a felhasználónak lehetősége adódik arra, hogy a vanishing line-t meghatározó pontokat kézzel, egérkattintással helyezze el.

Az előállított fájl formátumának specifikációja

A generált fájl egész számokat tartalmaz legalább 2 sorban. Minden sora két egész számot tartalmaz szóköz-karakterrel elválasztva. Az i-edik sor a kép i-edik vanishing point-jának pixeles koordinátáit tartalmazza[3].
Egy példa adatfájl (A mellékelt `Krakko20.jpg` kép vanishing point-jai):

```
234 456
-1 356
120 765
```

Koordináta-geometriai alapok

A két ponton átmenő egyenes egyenlete $(y-y_1)(x_2-x_1) = (x-x_1)(y_2-y_1)$. Mivel egy egyenes megadható az $y = Mx + b$ általános alakban, a két összefüggés alapján meghatározható a következő két tulajdonság:

$$M = \frac{(y_2 - y_1)}{(x_2 - x_1)} \qquad b = \frac{y_1(x_2 - x_1) - x_1(y_2 - y_1)}{x_2 - x_1}$$

Függőleges egyenesek esetében a módszer ilyen formában nem alkalmazható a 0-nevező miatt. Ekkor vektoros vagy polárkoordinátás reprezentációt használva a probléma megoldódik. Legyen két, egymással nem párhuzamos egyenes egyenlete a következő:

$$y = M_1 x + b_1 \qquad y = M_2 x + b_2$$

Ezek alapján meghatározható a két egyenes (x_0, y_0) metszéspontja, ahol...

$$y_0 = \frac{M_1 b_2 - M_2 b_1}{M_1 - M_2} \qquad x_0 = \frac{y_0 - b_1}{M_1}$$

Mérés

A fenti koordináta-geometriai tulajdonságokat felhasználva meghatározható a referencia objektum paramétereinek ismeretében a mérendő objektum magassága. Legyen...

- *meretMerendo* a mérendő objektum mérete pixelben,
- *meretMerendoVanish* a mérendő objektum talppontja és az (x_{MV}, y_{MV}) pont távolsága pixelben,

[3] Az x és y koordináták lehetnek bal alsó origó módszerrel (x balra, y fel) vagy mátrix (x. sor, y. oszlop) módszerrel generáltak, de mindenképp a GUI-nak legmegfelelőbb módon kell legenerálni őket. A jelölésmódot a GUI fejlesztője határozza meg az input data fájl generálója számára.

- *refsize* az általunk megadott referenciaobjektum mérete.

Ekkor a mérő algoritmus a következő:

1: húzzunk egy egyenest a mérendő objektum és a referenciaobjektum talppontjain át
2: **if** az egyenes iránytangense megegyezik a vanishing line iránytangensével **then**
3: egyszerű arányszámítással meghatározzuk a méretet
4: **else**
5: meghatározzuk a vanishing line és az egyenes metszéspontját $\rightarrow (x_t, y_t)$[4]
6: (x_t, y_t)-n és a referenciaobjektum tetején keresztül húzzunk egy egyenest
7: meghatározzuk ezen egyenes és a mérendő objektum egyenesének metszéspontját $\rightarrow (x_{MV}, y_{MV})$
8: Arányszámítással meghatározzuk a mérendő objektum méretét: $\frac{\text{refsize} * \text{meretMerendo}}{\text{meretMerendoVanish}}$
9: **end if**
10: **return** a mérendő objektum metrikus mérete

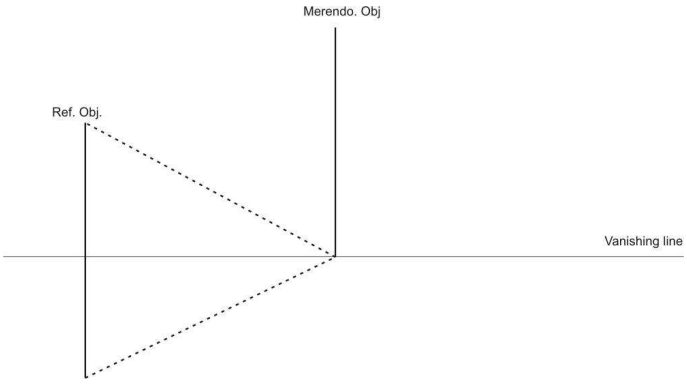
Abnormális eredmények generálódhatnak „majdnem párhuzamos" egyenesek mérésekor. Vegyük azt a speciális esetet, amikor a mérendő–, és a referenciaobjektum talppontjainak y koordinátái között csupán 1 pixelnyi eltérés van. Ekkor – kellően nagy méretű kép esetén – előfordulhat, hogy a talppontokat összekötő egyenes és a vanishing line metszéspontjának x koordinátája nagyobb (kisebb) lesz, mint az adott C++ fordító által ábrázolható legnagyobb (legkisebb) szám.

Ilyenkor szintén feltesszük, hogy a két objektum egymás mellett áll. Az elfogadott metszéspont értékét `MIN_LONGINT` és `MAX_LONGINT` között határoztuk meg, így biztos minden fordítóval generált kód helyesen fut.

A program hiányosságai

„Nem kellően ideális" képek esetén a MatLab szkript túl sok hamis vanishing point-ot generál, ami nehézkessé teszi a program használatát.

Abban az esetben is probléma adódik, ha referenciaobjektum a vanishing line-on „áll", vagyis talppontja megegyezik (x_t, y_t)-vel[5]. Ekkor a mérendő objektum magassága mindig $+\infty$ lesz a referenciaobjektum méretétől függetlenül:



Továbbfejlesztési lehetőségek

A program több irányba is továbbfejleszthető:

- A program jelen változata a vanishing line-ra merőleges irányt veszi „függőleges"-nek, a számításokban nem veszi figyelembe a függőleges vanishing point elhelyezkedéséből adódó perspektivikus torzítást.

[4] a „t"-index a „talp"-szóból származik
[5] Ugyanilyen helyzet áll elő, ha a mérendő objektum áll a vanishing line-on.

- MatLab szkript automatikus elindítása a C++ programból. Így a felhasználónak csak *egy* felületet kell kezelnie.

Az *SVM* osztály specifikációja

```
class svm {
  private:
    int vp1_x;     //1. vanishing point x koordinátája
    int vp1_y;     //1. vanishing point y koordinátája

    int vp2_x;     //2. vanishing point x koordinátája
    int vp2_y;     //2. vanishing point y koordinátája

    int reftop_x;  //ref.obj. tetejének x koordinátája
    int reftop_y;  //ref.obj. tetejének y koordinátája

    int refdown_x; //ref.obj. aljának x koordinátája
    int refdown_y; //ref.obj. aljának y koordinátája

    float refsize; //referencia objektum mérete

  public:
    svm(int w, int h, int vp1_x, int vp1_y, int vp2_x, int vp2_y);

    void getRefData( int reftop_x, int reftop_y,
                     int refdown_x, int refdown_y, float refsize);

    void getMeasureCoordinates( int mtop_x,    //mérendő tetejének x koord.
                                int mtop_y,    //mérendő tetejének y koord.
                                int mdown_x,   //mérendő aljának x koord.
                                int mdown_y);  //mérendő aljának y koord.

    float calculateSize(); //mérendő objektum méretét meghatározó metódus

};
```