

Matrica kamere - odredjivanje i dekompozicija

Srdjan Vukmirović, decembar 2020.

Napomena : (uz ovaj fajl ide i odgovarajuca fotografija)

Pretpostavimo da je zadat položaj kamere, tj. centar projektovanja C, orijentacija kamere - ortogonalna matrica A, kao i gornje-trougona matrica kalibracije K kamere.

Fukcija Kamera3x4, vraća 3x4 matricu kamere.

```
In[419]:= Kamera3x4[KK_, AA_, CC_] :=  
  Module[{T0 = KK.AA, C0}, C0 = -T0.CC; Transpose[Join[Transpose[T0], {C0}]]]
```

Evo funkcije koja radi obrnuto : za datu 3 x4 matricu kamere T, odredjuje njene parametre K, A, C, tim redom.

```
In[470]:= ParametriKamere[TT_] := Module[{CC, T0, QR},  
  CC = (# / Part[#, 4]) & [({1, -1, 1, -1} Reverse[Minors[TT, 3][[1]]])];  
  T0 = Transpose[Drop[Transpose[TT], -1]];  
  QR = QRDecomposition[Inverse[T0]];  
  {Inverse[QR[[2]]], QR[[1]], Drop[CC, -1]}  
]
```

Naravno, te dve funkcije bi trebalo da su inverzne. Probajmo na primeru (moze i vama da bude test primer).

```
In[471]:= TT =  $\begin{pmatrix} 1 & 5 & 7 & -2 \\ 3 & 0 & 2 & 3 \\ 4 & -1 & 0 & 1 \end{pmatrix};$ 
```

```
In[472]:= rez = ParametriKamere[TT];
```

```
In[475]:= MatrixForm /@ (rez // N)
```

```
Out[475]=  $\left\{ \begin{pmatrix} 2.39317 & 8.31949 & -0.242536 \\ 0. & 2.12824 & 2.91043 \\ 0. & 0. & 4.12311 \end{pmatrix}, \begin{pmatrix} 0.227921 & 0.911685 & -0.341882 \\ 0.0829185 & 0.331674 & 0.939743 \\ 0.970143 & -0.242536 & 0. \end{pmatrix}, \begin{pmatrix} 0.714286 \\ 3.85714 \\ -2.57143 \end{pmatrix} \right\}$ 
```

Ovde vidimo redom matricu kamere K (gornje - trougaona), matricu orijentacije kamere A (ortogonalna) i centar C kamere. Kad od tih elemenata napravimo kameru vidimo da dobijamo originalnu kameru.

```
In[476]:= Kamera3x4[Sequence @@ rez] // N // MatrixForm
```

```
Out[476]//MatrixForm=  
 $\begin{pmatrix} 1. & 5. & 7. & -2. \\ 3. & 0. & 2. & 3. \\ 4. & -1. & 0. & 1. \end{pmatrix}$ 
```

Sada zelimo da od tacaka prostora i njihovih projekcija (najmanje 6 odgovarajucih parova, tj. korespodencija) napravimo matricu kamere. Svaka korespodencija odredjuje 2 jednacone koje su date sledecom funkcijom.

```
In[477]:= CameraEquations[x_, xp_] := N[{Flatten[{0, 0, 0, 0}, -xp[[3]] x, xp[[2]] x}],  
  Flatten[{xp[[3]] x, {0, 0, 0, 0}, -xp[[1]] x}]]];
```

Sada pravimo funkciju koja SVD dekompozicijom za 6 i vise korespodencija odredjuje matricu kamere. Prethodna i naredna funkcija su veoma slicne odredjivanju projektivnog preslikavanja DLP algoritmom. Dodatno, moze da se radi normalizacija i RANSAC.

```
In[478]:= CameraDLP[orig_, images_] := Module[{AA, last, VV},
  If[Length[orig] ≠ Length[images],
    Message[CameraDLP::"Razlicit broj tacaka u originalu u slici"]];
  AA = Flatten[MapThread[CameraEquations, {orig, images}], 1];
  VV = SingularValueDecomposition[AA][[3]];
  last = (Transpose[VV])[12];
  Partition[last, 4]
];
```

Sada na konkretnoj 3 D sceni odredimo 3 D koordinate tacaka (X, Y, Z) u svetskom koordinatnom sistemu. Stavljamo, 4. koordinatu jednaku 1, posto zelimo homogene koordinate. Koordinate su u milimetrima (mereno grubo u centrimetrima), nebitno.

```
In[359]:= M1 = {460, 280, 250, 1}; M2 = {50, 380, 350, 1}; M3 = {470, 500, 100, 1};
M4 = {380, 630, 0, 1}; M5 = {240, 290, 0, 1}; M6 = {580, 0, 130, 1};
```

Sada na fotografiji (camera_resize.jpg) merimo piksel koordinate projekcija tih tacaka (x,y). Videti fotografiju camera_resize_sistemi.jpg. Fotografija je formata 600x800.

```
In[482]:= M1p = {288, 251, 1}; M2p = {79, 510, 1}; M3p = {470, 440, 1};
M4p = {520, 590, 1}; M5p = {365, 388, 1}; M6p = {365, 20, 1};
```

Evo kako izgledaju dve "jednacine", tj. njihovi koeficienti po nepoznatim tij, dobijeni iz prve tacke.

```
In[507]:= CameraEquations[M1, M1p] // MatrixForm
```

```
Out[507]//MatrixForm=
```

$$\begin{pmatrix} 0. & 0. & 0. & 0. & -460. & -280. & -250. & -1. & 115460. & 70280. & 62750. & 251. \\ 460. & 280. & 250. & 1. & 0. & 0. & 0. & 0. & -132480. & -80640. & -72000. & -288. \end{pmatrix}$$

```
In[487]:= TT = CameraDLP[{M1, M2, M3, M4, M5, M6}, {M1p, M2p, M3p, M4p, M5p, M6p}];
```

Evo kako izgleda jedna prosečna matrica kamere :)

```
In[484]:= TT // MatrixForm
```

```
Out[484]//MatrixForm=
```

$$\begin{pmatrix} -0.000346347 & -0.000105716 & 0.00151607 & -0.610881 \\ 0.00118498 & -0.000910219 & 0.000543781 & -0.791715 \\ 1.15447 \times 10^{-6} & 8.04605 \times 10^{-7} & 1.12489 \times 10^{-6} & -0.00249797 \end{pmatrix}$$

Odredimo sada parametre kamere. Vidimo redom matrice K, A i poziciju kamere C.

```
In[494]:= parametri = N[ParametriKamere[TT]]; MatrixForm /@ parametri
```

$$\text{Out[494]} = \left\{ \begin{pmatrix} 0.00140349 & 0.0000287456 & 0.000677477 \\ 0. & 0.00143143 & 0.00069238 \\ 0. & 0. & 1.80155 \times 10^{-6} \end{pmatrix}, \begin{pmatrix} -0.566711 & -0.273462 & 0.777211 \\ 0.517868 & -0.85191 & 0.0778637 \\ 0.640821 & 0.446619 & 0.624404 \end{pmatrix}, \begin{pmatrix} 958.923 \\ 782.787 \\ 676.589 \end{pmatrix} \right\}$$

Pozicija kamere C u svetskom koordinatnom sistemu (poslednji vektor) jako dobro odgovara poziciji mobilnog telefona prilikom fotografisanja. Sve je u milimetrima.

Ortogonalna matrica A ne govori puno. Njena treća kolona predstavlja z - osu kamere i ima smisla. Ta z - osa ide od tacke C prema senzoru, dakle usmerena je "od" scene. Zato ima smisla da su sve koordinate te ose pozitivne.

Gornje trougaona matrica K ima najmanje smisla. Na poziciji (3,3) bi trebao biti broj 1, da bi ona nesto rekla. Normirajmo matricu K. To smemo da uradimo jer to odgovara reskliranju polazne 3x4 matrice kamere TT. Pri reskliranju TT, ne menjaju se A i C.

```
In[497]:= KK = parametri[[1]]; MatrixForm[KK / KK[[3, 3]]]
```

```
Out[497]//MatrixForm=
```

$$\begin{pmatrix} 779.048 & 15.9561 & 376.053 \\ 0. & 794.555 & 384.325 \\ 0. & 0. & 1. \end{pmatrix}$$

Ima logike da na dijagonali budu isti (tj. bliski) brojevi (oni predstavljaju ziznu daljinu). Parametar smicanja je mali (trebao bi biti nula, ali s obzirom na greske merenja i distorziju kamere, to je sasvim OK). Brojevi $x_0 = 376$, $y_0 = 384$, predstavljaju translaciju od centra slike do gornjeg levog coska, a to je otprilike 300 i 400 u pikselima (pola dimenzije fotografije), sto je otprilike tacno. Ono sto vas moze zbuniti je da je zizna daljina oko 780.

Sledi objasnjenje (za frikove samo, ostali ga mogu preskociti).

Tih 780 nije 780 mm vec, 780 piksela u piksel koordinatama, ali kolika je velicina "piksela" u piksel koordinatama. Ceo senzor je prikazan na fotografiji 800 x600 piksela, pa je 780 piksela (u piksel koordinatama) otprilike sirina senzora.

Mobilni je Huawei P30 Pro, slikano glavnom kamerom bez digitalnog zuma. Zna se da je taj senzor oko 4.5 puta manji od senzora formata 36 x 24 mm, tj. fizicke velicine je oko 5.1 x 7.6 mm.

Kako smo rekli, zizna daljina je otprilike jednaka sirini senzora, tj. iznosi oko 7 mm. Ovo ima smisla jer udaljenost od sociva kamere do senzora kamere je malo manja od debljine mobilnog, pa ima smisla da bude oko 7 mm. Naravno, ovo je samo gruba aproksimacija stvari.

Susina gore napisanog je da 800 piksela otprilike odgovara 7mm, pa da bi dobili parametre kamere u milimetrima, piksel koordinate treba podeliti sa $800/7 = 110$. To cemo da uradimo i onda ce matrica kamere imati "mere" u milimetrima.

Sustina gore recenog je da, ako zelimo da dobijemo piksel koordinate u milimetrima, treba da ih podelimo sa, otprilike 110. Matrica A i centar C, ostaju isti, jedino matrica kamere ima mere u milimetrima.

U novim koordinatama racunamo matricu kamere, matricu K i normalizujemo K tako da bude $k_{33} = 1$.

```
In[503]:= Resize[{x_, y_, z_}] := {x / 110, y / 110, z};
```

```
TT = CameraDLP[{M1, M2, M3, M4, M5, M6}, Resize /@ {M1p, M2p, M3p, M4p, M5p, M6p}];
```

```
KK = ParametriKamere[TT][[1]];
```

```
MatrixForm[KK / KK[[3, 3]]]
```

```
Out[506]//MatrixForm=
```

$$\begin{pmatrix} 7.08289 & 0.14512 & 3.41865 \\ 0. & 7.22392 & 3.49378 \\ 0. & 0. & 1. \end{pmatrix}$$

Sada parametri kamere imaju smisao u milimetrima. Jos jednom, parametar smicanja bi trebalo da je 0, ali nije kao posledica raznih aproksimacija. Napomena: ovo je samo matematicka aproksimacija stvari (bazirana na tzv. "pinhole" kameri), stvari sa ziznom daljinom i optikom su mnogo slozenije.