A KLT (Kanade-Lucas-Tomasi) Feature Tracker Működése
(jellegzetes pontok választása és követése)

Készítette: Hajder Levente
2008.11.18.

1. Feladat

A rendelkezésre álló videó egy adott képkockájából minél több olyan pontot detektálni, melyeket a következő képkockákon is nyomon lehet követni.

2. Jellegzetes pontok követése

Adott két szomszédos kép, mindkettő szürke intenzitású pixelekkel leírva. Egy tetszőleges ***a*** pozícióban (***a*** kétdimenziós vektor) a megfelelő intenzitásértékeket a két képen *I(**a**) és J(**b**)* jelöli.

Ha kiválasztottunk egy kis *W* területdarabkát, melyet az **a** vektor jelöl ki az első képen, és annak keressük a megfelelőjét a második képen **b** vektor környékén, akkor a következő hibafüggvény kiszámításával kapjuk meg azt a ***d*** kétdimenziós eltolásvektort, amellyel távolabb került a területdarab:

$$\epsilon=\iint_{w}\{J(a+d)-I(b)\}^2\,dp$$

Mindezt felírhatjuk a [2]-es cikkben megfogalmazott módon, ahol picit másképp közelíti meg a szerző az elmozdulást (de ezzel pontosít a numerikus eredményen) :

$$\epsilon=\iint_{w}\{J(a+\frac{d}{2})-I(b-\frac{d}{2})\}^2\,dp$$

Tehát itt ***x*** nem az egyik képen a referenciapont, hanem a referenciapont két előfordulása között van félúton. Ennek gyakorlati haszna van, mert ha a két képet kétdimenziós intenzitás-függvényként képzeljük el, és ezt a függvényt Taylor-sorral közelítjük, akkor pontosabb eredményt kapunk.
A keresendő pontunk két előfordulását a következő módon közelíthetjük:

$$J(a+\frac{d}{2})\approx J(a)+\frac{d_u}{2}\frac{\partial J(a)}{\partial u}+\frac{d_v}{2}\frac{\partial J(a)}{\partial v}$$

$$I(b-\frac{d}{2})\approx I(b)-\frac{d_u}{2}\frac{\partial I(b)}{\partial u}-\frac{d_v}{2}\frac{\partial I(b)}{\partial v}$$

Ha bevezetjük a $g(a,b)=\begin{bmatrix}\frac{\partial}{\partial u}\{J(a)+I(b)\}\\ \frac{\partial}{\partial v}\{J(a)+I(b)\}\end{bmatrix}$ jelölést, továbbá $d=\begin{bmatrix}d_u\\ d_v\end{bmatrix}$, akkor a fentebb definiált hibafüggvény így írható át:

$$\epsilon=\iint_w \{J(a)-I(b)+\frac{1}{2}g^T(a,b)d\}^2\,dp$$

A hibafüggvénynek kell minimumát venni ***d*** eltolás szerint:

$$\frac{\partial\epsilon}{\partial d}=\iint_w \{J(a)-I(b)+\frac{1}{2}g^T(a,b)d\}\,g(a,b)\,dp=0$$

Ebből az összefüggésből ***d***-re nézve lineáris egyenletrendszert kapunk:

$$\iint_w \{J(a)-I(b)\}\,g(a,b)\,dp=-\frac{1}{2}\left[\iint_w \{g^T(a,b)\}\,g(a,b)\,dp\right]d$$

Ez pedig valóban a $Zd=e$ lineáris egyenletrendszer, amennyiben:

$$Z=\iint_w \{g^T(a,b)\,g(a,b)\}\,dp$$

és

$$e=2\iint_w \{I(a)-J(b)\}\,g(a,b)\,dp$$

3. Jellegzetes pontok kiválasztása

Noha a 3-mas pontban bemutatott módszer jellegzetes pontok követésére alkalmas, valójában nem arra használják. A követésre összehasonlításon alapuló 'brute force' módszerek alkalmazhatók, ennél sokkal nehezebb kérdés, hogy mely pontokat próbáljuk meg követni.

A Tomasi-Kanade-Lucas szerzőpáros ötlete a következő: a követésre az előző pontban kidolgozott lineáris egyenletrendszert hívjuk segítségül. Adott egy egyenletrendszer, amelyből a kívánt távolság kiolvasható. Azokat a pontokat használjuk, ahol az elmozdulás (***d***) minél jobban (vagyis: pontosabban) számítható.

Egy lineáris egyenletrendszer akkor számítható pontosan, ha az együtthatómátrix jól kondícionált. Ez esetünkben azt jelenti, hogy **Z** -re igaz, hogy a $\|Z\|\cdot\|Z^{-1}\|$ minél kisebb.

Amennyiben mátrixnormának a legnagyobb szinguláris értéket vesszük (és **Z** esetén a szinguláris értékek a sajátértékek négyzetei), ez azt jelenti, hogy $\frac{\lambda_1}{\lambda_2}$ hányadosnak minél nagyobbnak kell lennie, ha λ_1 és λ_2 jelöli a két sajátértéket (csökkenő értékek szerint).

Ha **Z** mátrixot kifejtjük, méghozzá úgy, hogy csak egyetlen képünk van (azaz *I*=*J*), akkor a következő alakot kapjuk:

$$Z=\begin{bmatrix}\iint_w I_x^2\,dp & \iint_w I_x I_y\,dp\\ \iint_w I_x I_y\,dp & \iint_w I_y^2\,dp\end{bmatrix}$$

feltéve, hogy I_u jelöli a kép *u* szerinti paricális deriváltját, I_u pedig a *v* szerintit. Azaz a második deriváltakat tartalmazza a mátrix.

A jellegzetes pontok kiválasztására hasznos, ha a **Z** minél nagyobb értékekkel rendelkezik, hiszen így numerikusan stabilabb az eredmény. Ezért **Z** legnagyobb sajátértékére is ki szokták kötni, hogy nagy legyen, nem csak a két sajátérték arányának kell kicsinek lennie.

Összevonva a két feltételt, a KLT algoritmus azt adja meg a pontkiválasztás kritériumának, hogy **Z** mátrix sajátértékei közül a kisebbik legyen egy előre megadott küszöbértéknél nagyobb.

A teljes algoritmus:

1. Gradiens számítása a minta minden egyes pixelén
2. Minden egyes mintára :
    3. *Z* számitása
    4. Lekisebb sajátérték meghatározása
    5. Ha az elõre megadott küszöbnél nagyobb ez a sajátérték, megtartjuk a mintát.
6. A kisebbik sajátérték szerint csökkenõ sorrendbe rakjuk a mintákat.
7. Legnagyobbat kiválasztjuk (jó mintának minõsítjük)
8. dobjuk a jó minta körüli mintákat a sorból.
9. Goto 7, amíg van minta

4. Kiterjesztés színes képek esetére

Ha a képet *R, G, B* színösszetevőként képzeljük el, és az intenzitásfüggvények *R(x,y)* , *G(x,y), B(x,y)*, akkor a levezetést végigcsinálva **Z** értéke így módosul:

$$Z=\begin{bmatrix} R_x^2+G_x^2+B_x^2 & R_x R_y+G_x G_y+B_x B_y \\ R_x R_y+G_x G_y+B_x B_y & R_y^2+G_y^2+B_y^2 \end{bmatrix}$$

Ennek a mátrixnak kell a sajátértékeit vizsgálni.

5. A követés kiterjesztése egyéb transzformációkra

A feladatot pontosabban meg tudjuk oldani, ha egyéb transzformációt is megengedünk. Tegyük fel, hogy a mintánk pozícióját (középpontja) a $p=(u.v)^T$ pontban van. Elforgatás és eltolás segítségével így kapjuk meg a pontot:

$$p=\begin{bmatrix} u \\ v \end{bmatrix}=\begin{bmatrix} u_0 \cos p_1 u+v_0 \sin p_1+p_2 \\ -u_0 \sin p_1 u+v_0 \cos p_1+p_3 \end{bmatrix}$$

Az eltoláshoz szükségünk van még Δu és Δv meghatározására. Segítségül hívjuk a parciális deriváltakat:

$$\frac{\Delta u}{\Delta p_i}=\frac{\partial u}{\partial p_i}\rightarrow \Delta u=\Delta p_i \frac{\partial u}{\partial p_i}$$

$$\frac{\Delta v}{\Delta p_i}=\frac{\partial v}{\partial p_i}\rightarrow \Delta v=\Delta p_i \frac{\partial v}{\partial p_i}$$

Ezt vektoros formában is elkészíthetjük:

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}=\begin{bmatrix} \dfrac{\partial u}{\partial p_i} \\ \dfrac{\partial v}{\partial p_i} \end{bmatrix} \Delta p_i$$

Ha több paraméter van (legyen P a paraméterek száma), a hatásokat össze kell adni:

$$\begin{bmatrix}\Delta u\\ \Delta v\end{bmatrix}=\begin{bmatrix}\frac{\partial u}{\partial p_1} & \frac{\partial u}{\partial p_2} & \cdots & \frac{\partial u}{\partial p_P}\\ \frac{\partial v}{\partial p_1} & \frac{\partial v}{\partial p_2} & \cdots & \frac{\partial v}{\partial p_P}\end{bmatrix}\begin{bmatrix}\Delta p_1\\ \Delta p_2\\ \vdots\\ \Delta p_P\end{bmatrix}$$

Röviden mindezt így szokás írni:

$$\begin{bmatrix}\Delta u\\ \Delta v\end{bmatrix}=\frac{\partial W}{\partial p}\Delta p$$

Ezek után a feladatunkat átírhatjuk:

$$\iint_w \left(J(p)+\nabla J(p)\frac{\partial W}{\partial p}\Delta p-I(p)\right)^2 dp$$

A megoldást Δp szerint keressük, ezért deriválni kell Δp szerint, és az eredményt nullával egyenlõvé kell tenni :

$$2\iint_w \left(J(p)+\nabla J(p)\frac{\partial W}{\partial p}\Delta p-I(p)\right)\left(\nabla J(p)\frac{\partial W}{\partial p}\right)^T dp=0$$

A megoldás pedig:

$$\left[\iint_w \left(\nabla J(p)\frac{\partial W}{\partial p}\right)^T\left(\nabla J(p)\frac{\partial W}{\partial p}\right)dp\right]\Delta p=\iint_w \left(J(p)-I(p)\right)\left(\frac{\partial W}{\partial p}\right)^T dp$$

Bevezetve az alábbi jelöléseket:

$$\delta=\iint_w \left(J(p)-I(p)\right)\left(\frac{\partial W}{\partial p}\right)^T dp$$

$$Z=\left[\iint_w \left(\nabla J(p)\frac{\partial W}{\partial p}\right)^T\left(\nabla J(p)\frac{\partial W}{\partial p}\right)dp\right]$$

az egyenlet megoldás $\Delta p=Z^{-1}\delta$ alakra egyszerûsödik.

A teljes algoritmus (az órán kiosztott lapnak megfelelõ lépésekkel):

1. Vágjunk ki a minta feltételezett helyét a képbõl.
2. Hibaképet számoljunk ki *(I-J)*
3. Gradienst számoljunk végig a minta helyére
4. W parciális deriváltjait számítsuk ki az összes pixelre
5. $\nabla J(p)\frac{\partial W}{\partial p}$ számítása.
6. Z számítása
7. δ számítása
8. Δp kiszámítása
9. $p=p+\Delta p$
10. Go to 1. Leállás: $\Delta p<\epsilon$, ahol ϵ egy előre megadott küszöb.
11.

6. Nevezetes transzformációk és parciális deriváltjaik

Az eõzõ sémába tetszõleges transzformációt lehet illeszteni. Itt összefoglaljuk a lehetséges transzformációkat és a paraméterek szerinti parciális deriváltakat:

6.1. Eltolás

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 + p_1 \\ v_0 + p_2 \end{bmatrix}$$

$$\frac{\partial W}{\partial p} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

6.2 Forgatás + eltoláshoz

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos(p_1) u_0 + \sin(p_1) v_0 + p_2 \\ -\sin(p_1) u_0 + \cos(p_1) v_0 + p_3 \end{bmatrix}$$

$$\frac{\partial W}{\partial p} = \begin{bmatrix} -\sin(p_1) u_0 + \cos(p_1) v_0 & 1 & 0 \\ -\cos(p_1) u_0 - \sin(p_1) v_0 & 0 & 1 \end{bmatrix}$$

6.3. Skálázás + forgatás + eltolás

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} p_4 \cos(p_1) u_0 + p_4 \sin(p_1) v_0 + p_2 \\ -p_4 \sin(p_1) u_0 + p_4 \cos(p_1) v_0 + p_3 \end{bmatrix}$$

$$\frac{\partial W}{\partial p} = \begin{bmatrix} -p_4 \sin(p_1) u_0 + p_4 \cos(p_1) v_0 & 1 & 0 & \cos(p_1) u_0 + \sin(p_1) v_0 \\ -p_4 \cos(p_1) u_0 - p_4 \sin(p_1) v_0 & 0 & 1 & -\sin(p_1) u_0 + \cos(p_1) v_0 \end{bmatrix}$$

6.4 Affin transzformáció

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} p_1 u_0 + p_2 v_0 + p_3 \\ p_4 u_0 + p_5 v_0 + p_6 \end{bmatrix}$$

$$\frac{\partial W}{\partial p} = \begin{bmatrix} u_0 & v_0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_0 & v_0 & 1 \end{bmatrix}$$

6.5 Homográfia

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dfrac{p_1 u_0 + p_2 v_0 + p_3}{p_7 u_0 + p_8 v_0 + 1} \\ \dfrac{p_4 u_0 + p_5 v_0 + p_6}{p_7 u_0 + p_8 v_0 + 1} \end{bmatrix}$$

$$\frac{\partial W}{\partial p} = \frac{1}{p_7 u_0 + p_8 v_0 + 1} \begin{bmatrix} u_0 & v_0 & 1 & 0 & 0 & 0 & \dfrac{-u_0(p_1 u_0 + p_2 v_0 + p_3)}{p_7 u_0 + p_8 v_0 + 1} & \dfrac{-v_0(p_1 u_0 + p_2 v_0 + p_3)}{p_7 u_0 + p_8 v_0 + 1} \\ 0 & 0 & 0 & u_0 & v_0 & 1 & \dfrac{u_0(p_4 u_0 + p_5 v_0 + p_6)}{p_7 u_0 + p_8 v_0 + 1} & \dfrac{v_0(p_4 u_0 + p_5 v_0 + p_6)}{p_7 u_0 + p_8 v_0 + 1} \end{bmatrix}$$

4. Hivatkozások

[1] Jianbo Shi, Carlo Tomasi: Good Features to Track, CVPR 1994. pp 593-600

[2] Stan Birchfield: Derivation of Kanade-Lucas-Tomasi Tracking Equation (forrás: http://robotics.stanford.edu/~birch/klt/derivation.ps)

[3] ...20 years on...