

Vyhľadávanie rohových bodov v obraze

Zsolt Horváth <xhorva03@stud.fit.vutbr.cz>

Alexander Páldy <xpaldy00@stud.fit.vutbr.cz>

13. decembra 2011

1 Úvod

Cieľom projektu bolo implementovať program na vyhľadanie rohových bodov v obraze. Vstupom programu je obraz v ktorom detekujeme rohové body. Výstupom je množina detekovaných bodov. Implementovaná metóda by mala byť štabilná, v takom zmyslu, že v podobných snímkoch danej scény nájde rohové body čo najväčšou spoločnou podmnožinou a čo najmenšími rozdielmi.

Zvolenou metódou je *Harrisov detektor affinných regiónov*, ktorá je založená na extrakcie špecifických vlastností z obrazu. *Harrisov detektor* sa používa na detekciu podobných regiónov medzi obrázkami, ktoré môžu byť rôzne zväčšené, osvetlené, atď. Affino-invariantné metódy sú schopné identifikovať obrázky fotené z rôznych pohľadov, teda môžu byť nejakým spôsobom natočené, skosené, alebo inak transformované. Porovnanie affinných detektorov je možné nájsť v [K. 06], [Gy.03]. Také detektory sa nazývajú invariantné a kovariantné spoločne. Regióny detekujeme invariantne voči transformácii, ale menia sa kovariantne s transformáciou obrázka.

Implementačným jazykom je *C++* a používame framework *OpenCV* vytvorený pre počítačové videnie, ktorý uľahčuje prácu s obrázkami a ich spracovanie.

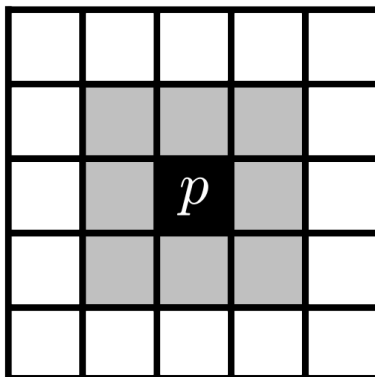
2 Teória

2.1 Harrisov detektor

Harrisov detektor je založený na matice druhých momentov. Nazýva sa aj ako *auto-korrelačná matica* a často sa používa na detekciu určitých vlastností, alebo k popisu lokálnych štruktúr v obraze. Táto matica sa musí prispôbovať tak, aby popísal obraz nezávisle na zmene veľkosti. Matica je definovaná nasledovne:

$$\mu(\mathbf{x}, \sigma_I, \sigma_D) = \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix} = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_x L_y(\mathbf{x}, \sigma_D) \\ L_x L_y(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix} \quad (1)$$

kde \mathbf{x} je bod v obraze, σ_I intergráčná veľkosť, σ_D je diferenčná veľkosť, funkcia $g(\sigma_I)$ je gaussovské rozmazanie a L_a je derivácia obrázka v smere a . Znak $*$ značí operáciu konvolúcie.



Obr. 1: 8 okolie, prevzatý z [Pro11]

Akým spôsobom je možné vypočítať konkrétnu hodnotu z takého nejasného matematického zápisu? Prvým krokom je, že obrázok premeníme na čiernobiely. Na základe čísla σ_D vypočítame hodnoty derivovanej gausovky. Nie je vhodné používať operátor *Sobela*, lebo to by bola len hrubá aproximácia. Tieto hodnoty nám slúžia ktomu, aby sme vedeli vypočítať deriváciu obrázka. Nasleduje klasická konvolúcia deriváciou gaussa (je dobré zvoliť veľkosť konvolučného jadra vo veľkosti 3σ). Po konvolúcii máme hotové dva obrázky (L_x a L_y) derivované v smere x a y . Vytvoríme ďalšie 3 obrázky, L_x^2 ,

L_y^2 a $L_x L_y$. Príklad: pre všetky body zoberieme hodnotu z L_x a vložíme jeho mocninu do L_x^2 . Teraz naša matica je hotová. Predposledným krokom je, že obrázky rozmazáme s *Gaussovým filtrom* a nakoniec ich vynásobíme s číslom σ_D^2 . Tým máme hotovú maticu a môžeme hľadať konečne rohové body.

Ako zistíme ktoré sú rohové body v obraze? Jednoducho, pomocou nasledovnej rovnice, ktorá určuje mieru rohovitosti:

$$cornerness = \det(\mu(\mathbf{x}, \sigma_I, \sigma_D)) - \alpha \text{trace}^2(\mu(\mathbf{x}, \sigma_I, \sigma_D)) \quad (2)$$

kde \det je determinat matice, trace je súčet hodnôt na diagonále a α je empirická konštanta z intervalu $< 0.4, 0.6 >$. Do tejto rovnice dosadíme pozície pixelov (\mathbf{x}) a máme vypočítané hodnoty rohovitosti pre všetky pixely. Tieto hodnoty musíme ešte ďalej prahovať a filtrovať, ak je bod maximum v osem okolí, tak je ponechaný, inak odhodíme.

2.2 Harris-Laplace detektor

Ideou je vybrať charakteristickú veľkosť lokálnej štruktúry [Ton94], pre ktorú daná funkcia nájde maximum medzi rôznymi veľkosťmi. Charakteristická veľkosť [Ton08] je nezávislá na rozlíšení obrazu. Veľkosť súvisí so štruktúrou a nie s rozlíšením v ktorom je reprezentovaná. Experimenty ukazujú, že najvyššie percento korektných charakteristických veľkostí nájde tzv. *Laplacian-of-Gaussians* funkcia, ktorú definujeme nasledovne [Kry01]:

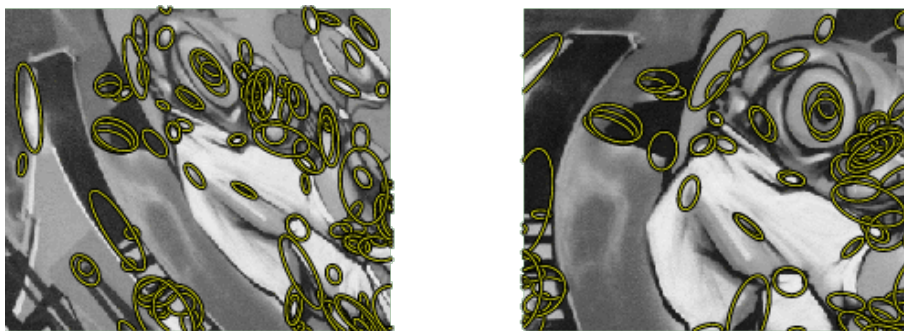
$$|LoG(\mathbf{x}, \sigma_n)| = \sigma_n^2 |L_{xx}(\mathbf{x}, \sigma_n) + L_{yy}(\mathbf{x}, \sigma_n)| \quad (3)$$

kde L_{aa} je druhá derivácia obrazu vo smeru a . V prípade, že veľkosť jadra LoG sa zhoduje s veľkosťou štruktúry, tak odozva funkcie nabýva exterémum.



Obr. 2: Harris-Laplace detektor

Harris-Laplace detektor používa *Harrisovu* funkciu k nájdeniu rohových bodov a ich veľkosti, pod ktorým sú reprezentované. Po detekcie vyberie tie body, ktoré dosahujú maximum pre rôzne veľkosti.



Obr. 3: Harrisov affinný detektor

Budeme teda potrebovať iteratívny algoritmus. Tento algoritmus má dva kroky: detekciu, a iteratívny výber veľkosti a pozície. Podrobnejší popis algoritmu a affinných detektorov je možné nájsť v [Kry03] alebo v [Wik11].

3 Popis riešenia

Riešenie nebolo jednoduché. Dosť náš spomalily matetický zápisy, ktoré na prvý pohľad vyzerajú komplikovane. Potom keď to človek prečíta viackrát, už sa začínajú veci ujasniť. Nakoniec keď sa spustí do implementácie zase zistí, že tomu stále nerozumie. Napriek týmto ťažkostiam sa nám podarilo implementovať *Harrisov detektor* rohových bodov a iteratívnu detekciu veľkosti pod ktorým má daný bod najväčšiu odozvu. Funguje spoľahlivo a detekuje podobné body v obrázkoch s malým rozptylom.

Z časových i z iných dôvodov (nejasne napísaná literatúra a málo skúsenosti s počítačovým videním) sa nám nepodarilo implementovať affinný detektor, aj keď nás to zaujímalo.

Implementovaný algoritmus má niekoľko krokov. Prvým je, že použijeme *Harris-Laplaceovú* funkciu k nájdeniu rohových bodov. Následne prahujeme a filtrujeme nájdené body. Na konci iteratívne vyberieme body pre ktorých *Laplacián* dosahuje maximum pre rôzne veľkosti v určitom okolí.

4 Vyhodnotenie

Tady by mělo být napsané jak to funguje. Protože se jedná o počítačovou grafiku nebo vidění, tak by tady měl být screenshot, ze kterého bude poznat jak to funguje. K tomu by měla být ideálně tabulka s vyhodnocením jak přesně/rychle to funguje.

5 Záver

Projekt nebol náročný z programátorského hľadiska, ale určite pomôže, ak človek už má nejaké skúsenosti s *OpenCV*. Algoritmy nie sú tak komplikované ako sa zdajú na prvý pohľad, len treba naučiť sa v nich orientovať a prečítať relevantnú literatúru o nejasných častiach. Získané skúsenosti s počítačovým videním môžeme ešte uplatniť i v iných projektoch a bude to dobrý základ k ďalším úkolom.

Literatúra

- [Gy.03] Gy. Dorkó, C. Schmid. Selection of Scale-Invariant Parts for Object Class Recognition. online, 2003. [Online; accessed 10-Dec-2011].
- [K. 06] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Van Gool. A Comparison of Affine Region Detectors. online, 2006. [Online; accessed 10-Dec-2011].
- [Kry01] Krystian Mikolajczyk, Cordelia Schmid. Indexing based on scale invariant interest points. online, 2001. [Online; accessed 10-Dec-2011].
- [Kry03] Krystian Mikolajczyk, Cordelia Schmid. Scale & Affine Invariant Interest Point Detectors. online, 2003. [Online; accessed 10-Dec-2011].
- [Pro11] Prof. Dr. Philippe Cattin. Digital Image Fundamentals. online, 2011. [Online; accessed 10-Dec-2011].
- [Ton94] Tony Lindeberg. Scale-Space Theory in Computer Vision. online, 1994. [Online; accessed 10-Dec-2011].

- [Ton08] Tony Lindeberg. Scale-Space. online, 2008. [Online; accessed 10-Dec-2011].
- [Wik11] Wikipedia. Harris affine region detector. online, 2011. [Online; accessed 10-Dec-2011].