



Detekcia hrán na binárnom obraze v reálnom čase

Detekcia hrán objektov patrí medzi základné a často realizované metódy predspracovania obrazu. Niektoré metódy si vyžadujú kvalitné hrany objektu so šírkou hrany jeden pixel, napríklad metódy na detekciu rohov. Aj keď existujú metódy na tzv. stenčovanie hrán, z hľadiska strojového času je výhodnejšia metóda detekcie hrán, ktorá priamo poskytne kvalitné hrany s hrúbkou jeden pixel. Tento článok opisuje novú metódu na detekciu hrán na binárnom obraze pracujúcu v reálnom čase.

Úvod

Za posledné desaťročia sa detekcia hrán stala neodmysliteľnou súčasťou veľkého množstva aplikácií od bezpečnostných kamerových systémov na parkoviskách, uliciach či v nákupných centrách až po systémy triedenia kvality alebo meracie systémy dvojrozmerných objektov [1] – [8].



Obr.1 Binárny obraz (vľavo) a jeho detegované hrany po aplikovaní operátora Laplacian (vpravo) spolu so zobrazením detailu jednej hrany

Detekcia hrán na binárnom obraze pomocou metódy masiek je podľa [9] približne 300-krát rýchlejšia ako pomocou štandardných metód na detekciu hrán. Táto metóda vznikla v dôsledku skutočnosti, že počítač ako digitálne zariadenie najrýchlejšie realizuje binárne operácie, napríklad logický súčin (AND), logický súčet (OR) alebo logickú negáciu (NOT).

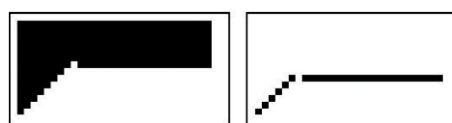
Navrhnutý algoritmus

Detekcia hrán na binárnom obraze pomocou metódy masiek je definovaná podľa [9] takto: bod (pixel 5 v matici 3 x 3) tvoriaci hranu na binárnom obraze je čierny bod, ktorý má medzi 4 susednými bodmi aspoň jeden biely alebo medzi 8 susednými bodmi nanajvýš jeden čierny.

1	2	3
4	5	6
7	8	9

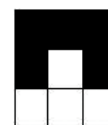
Obr.2 Matica 3 x 3 ako referenčný obraz pre metódu masiek („4 susedné body“ sú pixle označené 2, 4, 6 a 8, medzi 8 susedných bodov patria aj pixle označené 1, 3, 7 a 9)

Uvedené pravidlá nezabezpečia dostatočnú kvalitu hrán (prerušenie, nevyhladené hrany). Na zvýšenie kvality hrán sa definovali iné pravidlá na základe empirických pokusov. Napríklad, ako možno vidieť na obr. 3, ak jeden pixel spôsobil prerušenie hrany podľa. Túto situáciu vieme opísať referenčným obrazom (obr. 4), pomocou ktorého vieme tento prípad detegovať.



Obr.3 Binárny obraz (vľavo) a jeho prerušenie plynulosti hrany (vpravo)

Ak teda pomocou referenčného obrazu zistíme, že nastal tento problém, riešením bude nahradenie tohto bieleho pixla (v strede matice) čiernym. Týmto riešením zabezpečíme plynulosť uvedenej hrany.



Obr.4 Referenčný obraz

Ďalším príkladom môže byť situácia na obr. 5. V takomto prípade použijeme referenčný obraz z obr. 6.



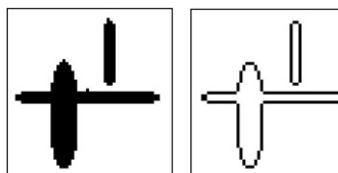
Obr.5 Binárny obraz (vľavo) a jeho prerušenie plynulosti hrany (vpravo)

Riešením bude (rovnako ako v predchádzajúcom prípade) nahradenie tohto bieleho pixla čiernym, čím zabezpečíme plynulosť uvedenej hrany.

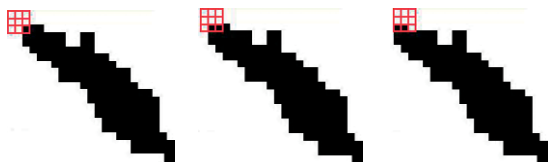


Obr.6 Referenčný obraz

Uvedenými spôsobmi sa našlo až 89 referenčných obrazov, pomocou ktorých detegujeme hrany a zabezpečíme ich vyhladenie, plynulosť ako aj ich hrúbku vo veľkosti jedného pixla.



Obr.7 Detegované hrany (vpravo) pomocou 89 referenčných obrazov sú vyhladené a plynulé, s hrúbkou jeden pixel



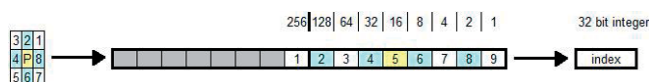
Obr.8 Štandardný spôsob vyhľadávania referenčných obrazov vo vstupnom obraze, príklad v prvom kroku (vľavo), druhom kroku (v strede) a v treťom kroku (vpravo)

Pri uvedenej metóde detekcie hrán pomocou referenčných obrazov je dôležitý spôsob implementácie, pretože pri štandardnom spôsobe pomocou vyhľadávania vo vstupnom obraze je algoritmus časovo náročný (obr. 8). S použitým hardvérom (2,26 GHz mikroprocesorom P8400 a 4 GB pamäťou RAM) a softvérom (operačným systémom Windows Vista SP1) potreboval daný algoritmus na detegovanie hrán na obraze s veľkosťou 348 x 348 pixlov 182 milisekúnd.

Treba pripomenúť, že na presné meranie výkonu musí byť program spustený samostatne a v režime „release“.

Optimalizácia navrhutej metódy

Navrhnutý algoritmus detekcie hrán pomocou referenčných obrazov možno z pohľadu strojového času značne optimalizovať. Referenčné obrazy (ako aj predkladaná vzorka) sú teda matice s veľkosťou 3 x 3 pixle, v ktorých sú hodnoty 0 alebo 1. Počet týchto hodnôt v danej matici je 9. Jednoduchým kopírovaním týchto logických čísel 0 a 1 do 2-bajtového slova, ktoré bolo inicializované pre celočíselný dátový



Obr.9 Referenčný obraz (vľavo) a jeho reprezentácia pomocou 32-bitového čísla (vpravo), spôsob umiestnenia bitov z matice v dvojбайtovom slove

1	2	3	4	5	6	7	8	9	index
1	1	1	1	1	1	0	0	0	504
0	0	0	1	1	1	1	1	1	63
0	1	1	0	1	1	0	1	1	219
1	1	0	1	1	0	1	1	0	438
0	0	1	0	1	1	0	1	1	91
1	1	0	1	1	0	1	0	0	436
0	1	1	0	1	1	0	0	1	217
1	0	0	1	1	0	1	1	0	310
0	0	0	1	1	0	1	1	0	54
0	0	0	0	1	1	0	1	1	27
0	1	1	0	1	1	0	0	0	216
0	1	0	1	1	1	0	0	0	184
1	1	1	0	1	1	0	0	1	473
1	1	1	1	1	0	1	0	0	500
1	0	0	1	1	0	1	1	1	311
0	0	1	0	1	1	1	1	1	95
1	1	1	1	1	1	1	0	0	508

1	2	3	4	5	6	7	8	9	index
1	0	1	0	1	1	0	1	1	347
0	0	1	1	1	0	1	1	1	119
0	0	1	1	1	0	1	1	1	123
0	1	0	1	1	0	1	1	1	183
1	1	0	1	1	1	1	0	0	444
1	1	1	0	1	1	0	1	0	474
1	1	1	1	0	1	0	0	0	488
1	1	1	0	1	0	1	1	0	422
0	0	0	1	0	1	1	1	1	47
0	1	1	0	0	1	0	1	1	203
1	0	1	1	0	1	1	1	1	367
1	1	1	0	0	1	1	1	1	463
1	1	1	1	0	1	1	0	1	493
1	1	1	1	0	0	1	1	1	487
1	1	1	0	0	1	0	1	1	459
1	1	1	1	0	1	1	0	0	492
1	1	0	1	0	0	1	1	1	423

1	2	3	4	5	6	7	8	9	index
1	0	0	1	1	1	1	1	1	319
0	0	1	1	1	1	1	1	1	127
1	1	1	1	1	1	0	0	1	505
1	1	1	1	1	0	1	1	0	502
1	1	0	1	1	0	1	1	1	439
0	1	1	0	1	1	1	1	1	223
1	1	1	0	1	1	0	1	1	475
0	0	0	1	1	0	1	1	1	55
1	1	0	1	1	0	1	0	0	436
1	1	1	0	1	1	0	0	0	472
0	0	1	0	1	1	0	1	1	91
0	0	0	1	1	0	1	1	1	55
0	1	1	1	1	0	0	0	0	496
0	1	1	0	1	1	0	0	1	217
0	0	0	0	1	1	1	1	1	31
1	0	0	1	1	0	1	1	0	310
0	0	0	0	1	0	1	1	1	23

1	2	3	4	5	6	7	8	9	index
0	0	1	1	0	1	1	1	1	111
0	1	1	0	0	1	1	1	1	207
1	0	0	1	0	1	1	1	1	303
1	1	1	1	0	0	1	1	0	486
1	1	1	1	0	1	0	0	1	489
1	1	1	1	0	0	0	1	1	483
0	0	1	1	0	1	1	1	1	111
1	1	0	1	0	0	1	1	1	423
1	1	1	1	0	1	1	0	0	492
0	1	1	1	0	1	1	1	1	239
1	1	0	1	0	1	1	1	1	431
1	1	1	1	0	1	1	1	0	494
1	1	1	1	0	1	0	1	1	491
1	0	1	1	0	1	1	1	1	367
1	1	1	1	0	0	1	1	1	487
1	1	1	1	0	1	1	0	1	493
1	1	1	0	0	1	1	1	1	463

1	2	3	4	5	6	7	8	9	index
1	0	0	1	1	0	1	0	0	308
1	1	1	0	1	0	0	0	0	464
0	0	1	0	1	1	0	0	1	89
0	1	0	1	1	1	1	1	1	191
1	1	0	1	1	1	1	1	0	446
1	1	1	1	1	1	0	1	0	506
0	1	1	1	1	1	0	1	1	251
0	1	0	0	1	1	1	1	1	159
1	0	0	1	1	1	1	1	0	318
1	1	1	1	1	0	0	1	0	498
0	1	1	1	1	1	0	0	1	249
0	1	1	0	1	1	1	0	1	221
1	0	0	1	1	1	0	1	1	315
0	1	1	1	1	0	1	1	0	374
1	1	1	1	1	0	0	0	1	497
1	1	0	1	1	0	1	0	1	437
1	1	1	0	1	1	1	0	0	476

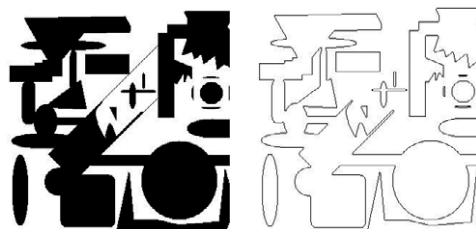
Tab.1 Tabuľky všetkých referenčných obrazov (matic 3 x 3) získaných empirickými pokusmi a ich indexy

typ „integer – 32 bitový“, tak dostaneme číslo reprezentujúce danú maticu.

Výhoda tejto optimalizácie spočíva v príprave jednorozmerného poľa, ktoré sa pripraví raz, a to pred samotným spustením algoritmu. Toto pole bude počas algoritmu v operačnej pamäti, teda jeho používanie bude rýchle. V tab. 1 sú definované všetky referenčné obrazy (logická nula predstavuje čiernu farbu a logická jednotka predstavuje farbu bielu), ako aj ich indexy. To znamená, že pole bude obsahovať logické nuly okrem prvkov, ktorých index sa nachádza v tab. 1.

Predkladanú vzorku obrazu (maticu 3 x 3) pretransformujeme na celočíselného reprezentanta (v jazyku C#) pomocou operácie súčet a rotácie bitov takto:

```
int n = matrix[i - 1, j - 1];
n += matrix[i - 1, j - 0] << 1;
n += matrix[i - 1, j + 1] << 2;
n += matrix[i - 0, j - 1] << 3;
n += matrix[i - 0, j - 0] << 4;
n += matrix[i - 0, j + 1] << 5;
n += matrix[i + 1, j - 1] << 6;
n += matrix[i + 1, j - 0] << 7;
n += matrix[i + 1, j + 1] << 8;
```



Obr.10 Detegované hrany testovacieho obrazca pomocou prezentovanej metódy spolu so zobrazením detailu jednej hrany (hrana má šírku jeden pixel, je plynulá a vyhladená)

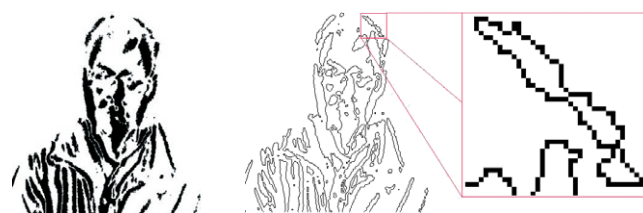
Uvedený celočíselný reprezentant predkladané vzorky je teda indexom do už pripravenej jednorozmerného poľa, v ktorom je uložená informácia, či analyzovaný pixel (v strede matice 3 x 3) tvorí hranu.

S použitým hardvérom (2,26 GHz mikroprocesorom P8400 a 4 GB pamäťou RAM) a softvérom (operačným systémom Windows Vista SP1) potreboval daný algoritmus na detegovanie hrán na obraze s veľkosťou 348 x 348 pixlov len 79 milisekúnd. To znamená, že uvedenou optimalizáciou sme navrhnutý algoritmus zrychlili takmer 2,5-násobne.

Experiment

V jazyku C# sme vytvorili program, do ktorého sme implementovali navrhnutý algoritmus detekcie hrán pomocou referenčných obrazov. Predkladané binárne obrazce sú

nielen testovacie, ale aj reálne, ktoré boli získané web kamerou a algoritmi detekcie pohybu.



Obr.11 Detegované hrany reálneho obrazca pomocou prezentovanej metódy spolu so zobrazením detailu jednej hrany (hrana má šírku jeden pixel, je plynulá a vyhladená)



Záver

Cieľom bolo navrhnúť odolný algoritmus detekcie hrán na binárnom obraze s kvalitnými hranami (t. j. vyhladené hrany bez prerušení so šírkou jeden pixel, zvlášť vhodnou na následnú detekciu rohov) s minimálnymi nárokmi na matematickú náročnosť a schopnosťou pracovať v reálnom čase. Navrhnutý algoritmus okrem prevodu predkladanej vzorky na celočíselnú reprezentáciu pracuje bez jedinej aritmeticko-logickej operácie, čím aritmeticko-logickú jednotku procesora počítača vynecháme. Algoritmus je teda založený len na práci s operačnou pamäťou.

Článok vznikol za podpory grantu VEGA 1/0690/09.

Literatúra

- [1] B. G. BATCHELOR, D. A. HILL and D. C. HODGSON (Eds.), „Automated Visual Inspection“, IFS Publications Ltd., UK, 1985.
- [2] R. T. CHIN, C. A. HARLOW, „Automated Visual Inspection: A survey“, IEEE Trans. Pattern Anal. Machine Intell. Vol. PAMI-4, pp. 557-573, November 1982.
- [3] G. MAITRE, H. HUGLI, F. TIECHE and J. P. AMANN, „Range Image Segmentation Based on Function Approximation“, published at ISPRS90, Zurich, Sept. 1990.
- [4] I. M. ELEWA, H. H. SOLIMAN and A. A. ALSHENNAWY. „Computer vision Methodology for measurement and Inspection: Metrology in Production area“. Mansoura Eng. First conf. Faculty of Eng. Mansoura Univ. , March 28-30, 1995, Pp. 473-444.

[5] A. A. ALSHENNAWY, „Measurement and Inspection of Three Dimensional Objects Using Computer Vision System“, Pd. D thesis, Mansoura University, Egypt, 2003.

[6] H. D. HOFMANN, „Application of Intelligent Measurements with Metrical Image Processing for Quality Control“, presented at the 5th International Conference, PEDAC '92, Alexandria, EGYPT, December 1992.

[7] P. RUMMEL, „GSS – A Fast, Model-Based Gray-Scale Sensor System for Workpiece Recognition“, Proceed. of 8th International Conf. on Pattern Recognition, pp. 18-21, Paris, France, Oct. 27-31, 1986.

[8] A. M. DARWISH and A. K. JAIN, „A Rule Based Approach for Visual Pattern Inspection“, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 10, No. 1, pp. 56-68, January 1988.

[9] ABDEL-MONEIM A. WAHDAM a kol., Edge detection if binary image using the method of masks, Sciencific bulletin Vol. 35, No. 3, Sept 30, 2000.

[10] internet <<http://www.cs.cf.ac.uk/Dave/C/node13.html>>

Ing. Peter Benický
prof. Ing. Ladislav Jurišica, PhD.

Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Ústav riadenia a priemyselnej informatiky
Ilkovičová 3, 812 19 Bratislava
e-mail: peter@benicky.info
ladislav.jurisica@stuba.sk

29