

**Technická univerzita v Košiciach**  
**Fakulta elektrotechniky a informatiky**  
**Katedra kybernetiky a umelej inteligencie**

**Detekcia objektu na základe priestorových vlastností – detekcia štvorca**  
(Zadanie predmetu Počítačové videnie)

*Systémová príručka*

1. ročník Ing. štúdia  
Umelá inteligencia  
Letný semester 2011/2012

Michal Puheim

# Obsah

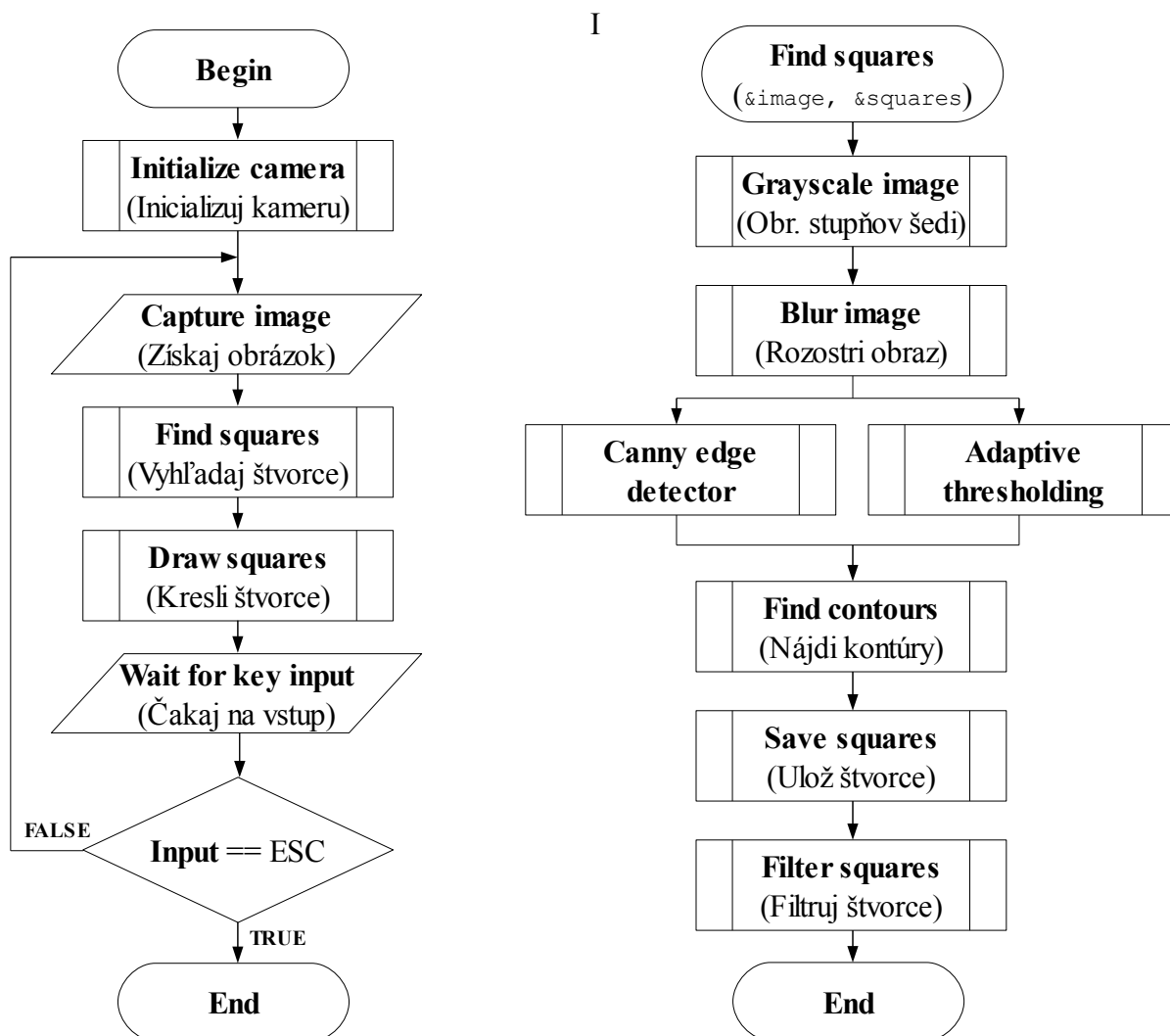
Obsah.....	1
1 Funkcia programu.....	2
2 Návrh riešenia.....	2
3 Implementácia riešenia.....	3
3.1 Konfiguračné parametre.....	3
3.2 Popis funkcií.....	3
4 Kompilácia zdrojového kódu.....	4

# 1 Funkcia programu

Program „square\_detect” umožňuje detekciu štvorcových objektov na živom videu z kamery. Detekovaných objektov môže byť ľubovoľné množstvo, môžu mať rôznu veľkosť, ako aj rotáciu. Predpokladom pre úspešnú detekciu je štvorcový tvar objektu a adekvátne svetelné podmienky. Program môže byť využitý ako demonštrácia schopností otvorenej knižnice počítačového videnia „OpenCV”.

## 2 Návrh riešenia

Program je postavený ako konzolová aplikácia, ktorá na vykreslenie grafického okna používa modul „highgui” knižnice OpenCV. Algoritmus cyklicky načítava obraz z kamery a vyhľadáva na ňom štvorce. Nájdené štvorce spolu s pôvodným obrazom vykresľuje v grafickom okne. Tento cyklus pokračuje až do stlačenia klávesy „Escape”. Realizácia najdôležitejších častí programu je zobrazená vo forme blokovej schémy na Obr. 1:



Obr. 1: Vývojový diagram zobrazuje činnosť hlavného cyklu programu a funkcie určenej na vyhľadávanie štvorcov.

Pri detekcii štvorcov je najprv trojzložkový farebný obraz transformovaný na jednozložkový obraz viacerých stupňov šedi. Tento výsledný obraz sa rozostří (*blur*) a následne sa transformuje na dve binárne čiernobiele obrazy pomocou algoritmov hranového detektora (*canny edge detector*) a adaptívneho prahovania (*adaptive thresholding*). Súčasné použitie dvoch algoritmov pre tvorbu binárneho obrazu je z dôvodu zvýšenia šance na detekciu štvorcových objektov pri rôznych svetelných podmienkach.

Z obrazov získaných oboma prístupmi sa extrahujú kontúry konvexných polygónov. Z vektora kontúr sú následne vybrané tie, ktoré spĺňajú vlastnosti štvorcového polygónu. Nakoniec sa odstraňujú duplicitné štvorce, keďže hranový detektor aj prahovanie, môžu detegovať rovnaké štvorce súčasne.

### 3 Implementácia riešenia

Zdrojové súbory programu sú uložené v priečinku „.\src“. Program je napísaný v jazyku C++ a z knižnice OpenCV používa výhradne API určené pre C++. Dokumentácia knižničných funkcií OpenCV je dostupná na adrese „<http://opencv.itseez.com/index.html>“.

#### 3.1 Konfiguračné parametre

V hlavičkovom súbore „.\src\config.h“ je možné nastaviť niektoré parametre ovplyvňujúce činnosť programu:

- `WND_NAME` (default="Square Detection")  
– názov zobrazovacieho grafického okna.
- `WEBCAM_ID` (default=0)  
– id kamery.
- `AD_THRESH_CONST` (default=8)  
– konštanta odpočítaná od priemeru pri adaptívnom prahovaní.
- `AD_THRESH_BLOCK` (default=13)  
– veľkosť okna použitého pri adaptívnom prahovaní.
- `CANNY_THRESH_LOW` (default=50)  
– prahovacia hodnota hranového detektora použitá pre slabé hrany.
- `CANNY_THRESH_HIGH` (default=200)  
– prahovacia hodnota hranového detektora použitá pre silné hrany.
- `CANNY_THRESH_APP` (default=3)  
– účinná plocha použitá pre Sobelov operátor pri hranovej detekcii.
- `SQUARE_AREA` (default=400)  
– minimálna plocha detegovaného štvorca.
- `SQUARE_TOLERANCE` (default=6)  
– tolerancia nepresnosti voči štvorcovému tvaru.
- `FILTERING_WINDOW` (default=10)  
– okno využívané pri odstraňovaní duplicitných štvorcov.

#### 3.2 Popis funkcií

Vlastný program je realizovaný v súbore „.\src\square\_detect.cpp“. Obsahuje nasledovné funkcie:

- `int main(int /*argc*/, char** /*argv*/)`  
– hlavná funkcia, ktorá realizuje cyklus načítavania obrazov z kamery a ich vykresľovania spolu s detegovanými štvorcami.

- `VideoCapture initializeCamera()`  
– funkcia pre inicializáciu kamery, ktorá navracia smerník na kamerový objekt.
- `void findSquares(const Mat& image, vector<vector<Point> >& squares)`  
– funkcia, ktorá v obraze `image` vyhľadá štvorce a uloží ich do vektora `squares`.
- `void filterDuplicatSquares(vector<vector<Point> >& squares)`  
– funkcia, ktorá z vektora `squares` odstráni duplicitné štvorce.
- `Point getSquareCenter(const vector<Point>& square)`  
– funkcia navracajúca súradnice stredu štvorca.
- `bool squareTest(const vector<Point>& contour)`  
– funkcia určuje či kontúra spĺňa vlastnosti štvorcového polygónu.
- `double ptDistance(Point p1, Point p2)`  
– funkcia určuje vzdialenosť medzi dvoma bodmi, resp. dĺžku čiary.
- `double angle(Point pt1, Point pt2, Point pt0)`  
– funkcia navracia kosínus uhla určeného tromi bodmi.
- `void drawSquares(Mat& image, const vector<vector<Point> >& squares)`  
– funkcia vykresľuje obraz `image` spolu s detegovanými štvorcami vo vektore `squares`.

## 4 Kompilácia zdrojového kódu

MS Windows, Visual Studio:

1. Nainštalujeme a nakonfigurujeme knižnicu OpenCV pre Visual Studio podľa príručky „`..\doc\tutorials_EN\OpenCV - MS Visual Studio.pdf`”.
2. V prostredí Visual Studia vytvoríme nový prázdny projekt konzolovej aplikácie. Pridáme do neho zdrojové súbory z priečinka „`..\src`”. Upravíme nastavenia projektu podľa príručky.
3. V aplikačnom menu vyberieme možnosť *Build->Build all*.

GNU/Linux, g++:

1. Nainštalujeme a nakonfigurujeme knižnicu OpenCV v operačnom systéme podľa príručky „`..\doc\tutorials_EN\OpenCV - Linux.pdf`”.
2. Program skompilujeme príkazom „`g++ `pkg-config opencv --cflags` `pkg-config opencv --libs` square_detect.cpp -o square_detect`”