

Aplikácia na rozpoznanie obrazu v jazyku C#

¹Miroslav PETRO, ²Peter PAPCUN, ³Ján JADLOVSKÝ

Katedra kybernetiky a umelej inteligencie, Fakulta elektrotechniky a informatiky,
Technická univerzita v Košiciach, Slovenská Republika

¹miroslav.petro@student.tuke.sk, ²peter.papcun@tuke.sk, ³jan.jadlovsky@tuke.sk

Abstrakt — Úlohou bolo vytvoriť softvér pre kamerový systém flexibilného montážneho podniku v programovacom jazyku C#. Pri navrhovaní softvéru bolo dôležité zabezpečiť, aby bol získaný obraz dostatočne rýchlo a presne spracovaný pre potreby ďalšej práce robotického ramena Mitsubishi RV-2SDB. V tomto článku sa nachádza stručný popis rozpoznania obrazu a pracovného postupu aplikácie. V závere je uvedená úspešnosť vyhotovenej aplikácie zhodnotená na základe testovania v prevádzke.

Kľúčové slová — Spracovanie obrazu, C#, OPC, OpenCV, Kamera, Flexibilný montážny podnik

I. ÚVOD

V laboratóriu V147 v budove V4 Technickej univerzity v Košiciach sa nachádza model flexibilného montážneho podniku, na ktorom je možné vyskladať niekoľko variant výrobu. Cieľom našej práce bola realizácia aplikácie na rozpoznanie obrazu zosnímanom priemyselnou kamerou, ktorá je súčasťou tejto výrobnéj linky. Úlohou aplikácie bolo správne rozpoznať podstavu skladaného výrobku a naviesť robotické rameno Mitsubishi, pomocou ktorého bola táto podstava umiestňovaná na pripravenú paletu. OPC (OLE for process control) komunikácia zabezpečuje odovzdávanie získaných údajov o objekte medzi aplikáciou spustenou na počítači a PLC (Programmable logic controller). Riadiaca jednotka robota komunikuje s PLC pomocou priemyselnej siete ProfiBus.



Obr. 1 Pohľad na model flexibilného montážneho podniku

II. POSTUP PRI ROZPOZNÁVANÍ OBRAZU

Pre dosiahnutie stanovenej úlohy bolo potrebné využiť metódy počítačového videnia. V praxi neexistuje žiadny pevne stanovený postup na spracovanie a rozpoznanie obrazu, preto sme sa pri navrhovaní našej aplikácie rozhodli pre postup, ktorý sa dá všeobecne opísať nasledujúcimi bodmi:

- **Snímanie obrazu:** Prvý krok procesu, kedy je svetlo zo snímaného priestoru pomocou snímacieho zariadenia prevedené na digitálny obsah. Výsledkom je matica údajov popisujúcich obraz.
- **Predspracovanie obrazu:** Cieľom je upraviť obraz tak, aby sa rozpoznanie dialo čo najrýchlejšie a najspoľahlivejšie. Patrí sem potlačenie šumu v obraze, ktorý vznikol pri snímaní alebo prenose informácií. K predspracovaniu sa ešte môže zaradiť úprava jasu a hranové filtre.
- **Segmentácia:** Pri tomto kroku sa obraz delí na jednotlivé segmenty, ktoré reprezentujú rozpoznané objekty. Teda pozorovaný objekt sa oddeľuje od pozadia.
- **Porozumenie získaným informáciám:** Znamená popis a interpretáciu získaných údajov z rozpoznávaných objektov a delenie objektov do vopred známych tried. Vieme určiť rozmery jednotlivých objektov, ich konkávnosť resp. konvexnosť a zároveň dokážeme určiť dĺžky ich hrán a obsah objektov. Vďaka týmto vlastnostiam vieme jednotlivým rozpoznávaným objektom priradiť ich tvar a na základe porovnania so známou polohou kamery, vieme určiť aj reálnu polohu objektu v zosnímanom obraze.

Pre čo najvyššiu rýchlosť a efektívnosť aplikácie sme siahli po programovej knižnici OpenCV. Ide o jednu z najznámejších a najrozšírenejších programových knižníc, ktorá implementuje takmer 2500 optimalizovaných algoritmov z oblasti počítačového videnia a strojového učenia. Pretože bola naša aplikácia vyvíjaná v programovacom jazyku C#, bolo potrebné použiť tzv. wrapper s názvom EmguCV.

III. APLIKÁCIA ROZPOZNÁVANIA OBRAZU

Spracovanie obrazu sa vykonáva v cykle od spustenia rozpoznávania až po jeho ukončenie. Pozostáva z čiastkových úloh, ktoré predstavuje prijatie obrazu z kamery, spracovanie snímku a odoslanie zistených informácií o podstave do PLC. Fragment aplikácie, ktorý je zodpovedný za spracovanie obrazu možno opísať vývojovým diagramom, ktorý môžete vidieť na obr. 2.

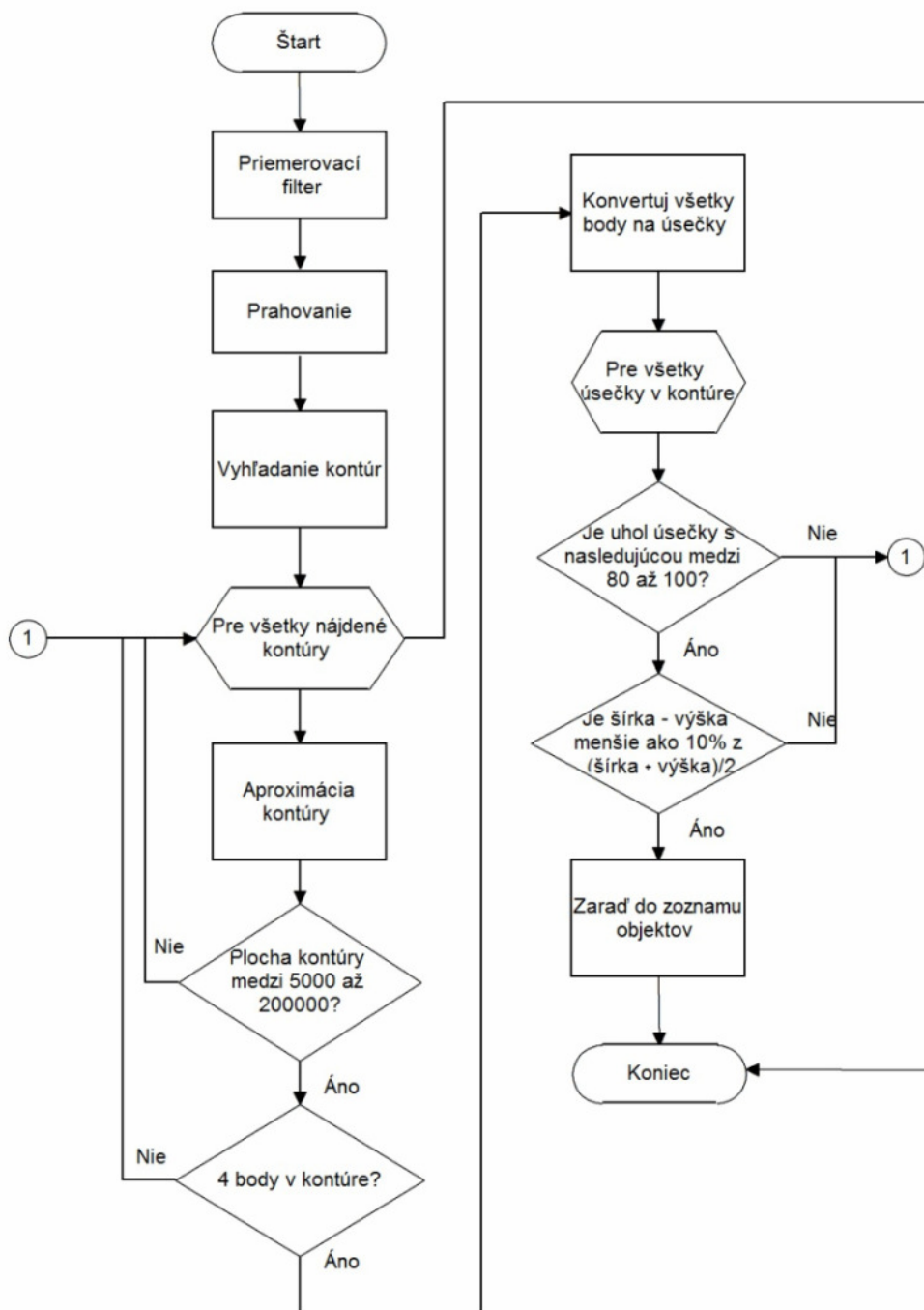
A. Predspracovanie

Ako prvé v rámci celého spracovania obrazu sme použili priemerovací filter, ktorý funguje na princípe nahrádzania jasu pôvodného obrazového bodu aritmetickým priemerom jasových hodnôt jeho okolia, čím dôjde k potlačeniu šumu v obraze. Okolie filtrovaného bodu je určené veľkosťou masky (zvyčajne tvaru štvorca), ktorá sa postupne posúva po obrázku až prejde všetky body. Použili sme masku o veľkosti 3x3, čím dochádza k odstráneniu šumu a zároveň len k minimálnemu rozmazaniu hrán v obraze.

B. Segmentácia a spracovanie obrazu

Na účel segmentácie sme použili výpočtovo nenáročnú metódu jednoduchého prahovania. Prahovanie (thresholding) sa nazýva technika, kedy sa prechádzajú všetky obrazové body $f(x,y)$ a pomocou jedného, alebo niekoľkých prahov zaradíme tieto body do popredia, alebo do pozadia. Prah T predstavuje istú hodnotu, ktorá je hranicou medzi popredím a pozadím. Pri technike segmentácie sa predpokladá, že popredie sa od pozadia líši farbou alebo jasom, preto najlepšie funguje pre jasné a kontrastné obrazy. Výsledkom je nový binárny obraz $g(x, y)$. Jednoduché prahovanie možno popísať nasledujúcim vzťahom:

$$g(x, y) = \begin{cases} 255 & \text{ak } f(x, y) > T \\ 0 & \text{ak } f(x, y) \leq T \end{cases} \quad (1)$$



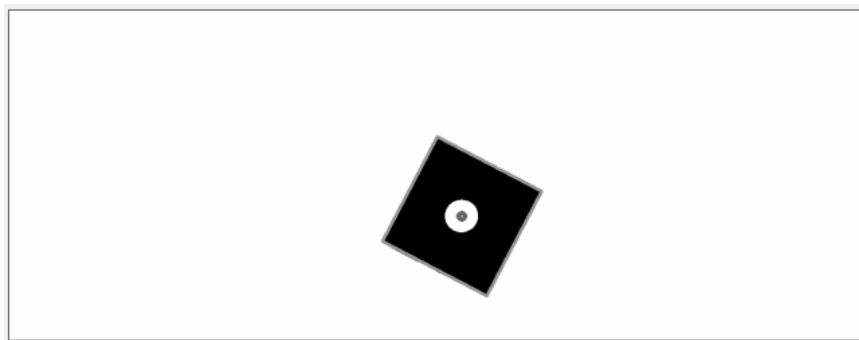
Obr. 2 Vývojový diagram detekcie podstavy

C. Vyhľadanie kontúr

Na takto upravený binárny obraz, kde každý pixel môže mať jednu z dvoch hodnôt označujúcu príslušnosť k objektu alebo pozadiu, sme mohli nasadiť metódu na hľadanie kontúr, ktoré je možné ďalej spracovať na zistenie tvaru a vlastností objektu.

Vďaka metóde *FindContours()* z knižnice *EmguCV.dll* sú nájdené kontúry, ktoré reprezentujú obrysy jednotlivých objektov. Tieto kontúry však predstavujú akýsi „kostrbatý“ mnohoúhelník, pretože princíp akým pracujú kamery, neumožňuje dokonalé hladké zosnímanie hrany. Preto sme na objavené kontúry aplikovali metódu, ktorá implementuje Douglas-Peucker algoritmus. Cieľom tohto algoritmu je vyhladenie nájdených kontúr, teda zníženie počtu vrcholov útvaru, reprezentujúceho kontúru objektu.

Pri vhodnej voľbe presnosti získame skupinu úsečiek reprezentujúcich vyhladenú kontúru objektu.



Obr. 3 Výsledný obrázok po segmentácii a vyhľadani kontúr

D. Plocha

Pri testovaní sme zistili, že doteraz popísané riešenie vyhľadáva aj extrémne veľké a extrémne malé objekty (napríklad tiene), čo zbytočne zaťažuje systémové prostriedky. Preto sme do procesu rozpoznávania vložili podmienku, kde sa testuje, či nájdená kontúra má obsah väčší ako 5000 pixelov a menší ako 200000 pixelov. Podmienka je dostatočne vágna, aby boli detegované podstavy aj pri zmene polohy kamery, kedy sa môže objekt javiť väčší alebo menší. Pri súčasnej inštalácii kamery mala podstava približne obsah 25000 až 31000 pixelov v závislosti od osvetlenia, nastavenia prahu a konkrétnej podstavy.

E. Vlastnosti

Hľadaná podstava v obraze je reprezentovaná približne štvorcovým objektom v binárnom obraze. Aby sme si overili, či sa skutočne jedná o hľadanú podstavu, je potrebné zistiť, či nájdený objekt spĺňa niekoľko vlastností.

Ako prvé sa zisťuje, či je kontúra reprezentovaná štyrmi vrcholovými bodmi. Ak áno, konvertujeme tieto vrcholové body na úsečky reprezentujúce jednotlivé strany objektu.

Určili sme si, že za štvorec budeme považovať objekt, ktorý je reprezentovaný štyrmi úsečkami, vnútorné uhly strán má v rozmedzí 80 až 100 stupňov a absolútna hodnota rozdielu šírky a výšky objektu je menšia, ako 10% z aritmetického priemeru jeho šírky a výšky.

$$\text{abs}(w-h) < 0,1 \cdot \frac{w+h}{2} \quad (2)$$

Kde w – šírka objektu, h – výška objektu a abs – absolútna hodnota.

Ak kontúra vyhovuje podmienkam, ktoré boli uvedené, tak sa môže považovať za reprezentáciu podstavy štvorcového pôdorysu a môže byť zaradená do zoznamu, ktorý obsahuje štruktúry reprezentujúce tento objekt.

IV. PREBERANIE PODSTAVY PRIEMYSELNÝM ROBOTOM

Na modely flexibilného montážneho podniku sa výroba začína vysunutím podstavy zo zásobníka, ktorá sa následne šmykom po naklonenej rovine dostane na pohyblivý dopravný pás. Aplikácia na spracovanie obrazu po rozpoznaní podstavy navedie robotické rameno tak, aby dopravný pás viedol objekt presne pod pripravený manipulátor priemyselného robota. Ak sa podstava dostane pod pripravené rameno robota, aplikácia na rozpoznanie obrazu vyšle príkaz na vyzdvihnutie objektu. Riadiaca jednotka robota príkaz spracuje, rameno vyzdvihne podstavu a umiestni ju na pripravenú paletu.

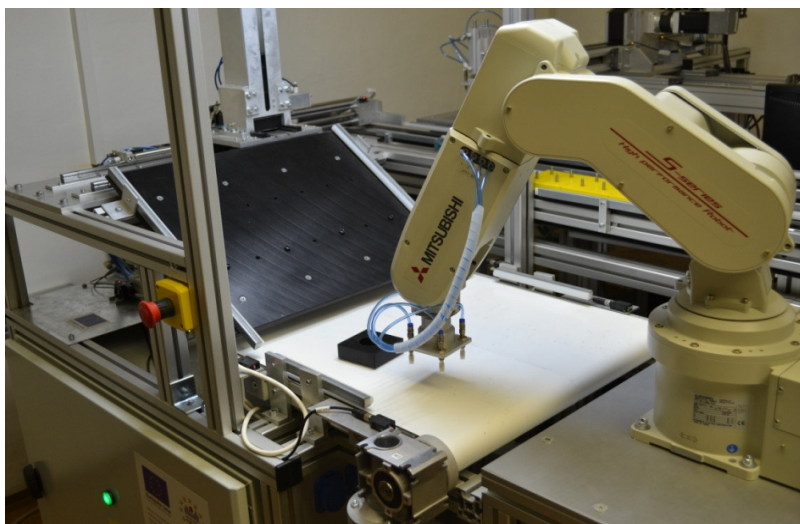
Celý proces spracovania obrazu a vyzdvihnutia podstavy sa začína zosnímaním obrazu a rozpoznáním podstavy, ktoré bolo opísané v predchádzajúcej kapitole. Vďaka zisteniu vzájomnej polohy kamery a robotického ramena a taktiež aj zisteniu pomeru reálnej vzdialenosti a jej zodpovedajúceho počtu obrazových jednotiek, je možné zistiť skutočnú polohu podstavy na páse. Do premenných v PLC automate $P2_osx$ a $P2_uhol$ sa zapíše súradnica X stredu objektu a uhol natočenia. Zároveň sa do premennej $P2_spracovane$ v PLC zapíše hodnota 1. Zmenu hodnoty v premennej $P2_spracovane$ vyhodnotí riadiaca jednotka robota ako príkaz na nastavenie robotického ramena na pozíciu X a Y , kde X (os súbežná so smerom pohybu pásu) je hodnota získaná z premennej $P2_osx$ a Y (os súbežná so smerom pohybu pásu) je vopred určená súradnica, kde bude robot „čakať“ na vyzdvihnutie. Aplikácia

na rozpoznanie obrazu sa ďalej prepne do módu, v ktorom sa sleduje prekrytie podstavy s robotickým ramenom. Ak je podstava tesne pod manipulátorom robota, zapíše do premennej v PLC automate P2_zdvih hodnotu 1, čo robot interpretuje ako príkaz na vyzdvihnutie podstavy. Po úspešnom usadení podstavy na paletu sa hodnoty v premennej P2_zdvih a P2_spracovane vynulujú, čím sa softvér na rozpoznávanie obrazu pripraví na nový cyklus vyzdvihnutia.

Aplikácia zároveň ošetruje malé časové oneskorenie príkazu na vyzdvihnutie, ktoré vzniká pri samotnom spracovaní obrazu, odosielaním údajov a pri vyzdvihovaní. Používateľovi umožňuje regulovať predstih, s ktorým sa príkaz na vyzdvihnutie odošle o istý okamih skôr.

Výhoda tohto riešenia spočíva v tom, že je potrebné rozpoznať veľkosť uhlu natočenia objektu a jeho polohu iba v jednej súradnici. Tým sa znižuje pravdepodobnosť nepresného odhadu súradnice Y. Medzi ďalšie výhody patrí fakt, že robotické rameno bude v čase vyzdvihovania objektu vždy presne na súradnici Y, teda čas oneskorenia pri vyzdvihnutí bude takmer konštantný. Rovnako sa nemusí riadiaca jednotka robota zaoberať aktuálnosťou príkazu na vyzdvihnutie, pretože v aplikácii na rozpoznávanie obrazu je možné nastaviť predstih, teda poslať príkaz na vyzdvihnutie o okamih skôr.

Celý proces vyzdvihovania je zároveň spoľahlivejší, pretože pri zosúladení súradnicových systémov kamery a robota je potrebná presnosť hlavne v osi X, teda nie je potrebné starostlivo dbať aj na synchronizáciu osi Y.



Obr. 4 Robotické rameno čakajúce na okamih, kedy bude podstava pod manipulátorom

V. ZÁVER

Po príprave aplikácie a jej správnej kalibrácii sme vykonali testovanie riešenia v ostrej prevádzke. Nechali sme vyskladať 240 produktov, pričom táto výroba bola rozdelená do 10 fáz z dôvodu kapacity výstupného skladu. Robot vďaka aplikácii rozpoznávania obrazu premiestnil 238 podstáv na prvý krát, zvyšné dva podstavy boli uchopené na druhý krát. V konečnom dôsledku bolo umiestnenie podstavy takmer vždy úspešné. Aplikácia pracuje uspokojivo aj s rôznymi natočeniami objektu a nezaznamenali sme žiadne odchýlky úspešnosti.

Pri zmene intenzity osvetlenia bolo potrebné upraviť prah, ale úspešnosť sa znovu výrazne nezmenila. Otestovali sme fungovanie aplikácie aj s vypnutým osvetlením nad postom. Aplikácia fungovala, ale pretože bolo problematické vhodne nastaviť prah, úspešnosť už nebola tak vysoká.

Celý proces od dopadu podstavy na dopravný pás, rozpoznanie aplikáciou až po prisatie podstavy robotom trvá necelé 3 sekundy. Samotné vyzdvihnutie, presunutie nad paletu a umiestnenie podstavy trvá ďalších 3 až 5 sekúnd v závislosti od presnosti uchopenia. Spolu ide o dostatočne krátky časový úsek na to, aby boli podstavy umiestňované na paletu skôr, ako ostatné posty dokončia úlohu. Nedochádza tak k zbytočnému spomaľovaniu celého procesu vyskladania výrobku. Počas celého testovania nenastala situácia, kedy bola celá výroba zdržiavaná procesom vyzdvihovania podstáv.

POĎAKOVANIE

Tento článok bol podporený vedeckou grantovou agentúrou Slovenskej Republiky v projekte Vega č. 1/0286/11 Dynamické hybridné architektúry v multiagentových sieťových riadiacich systémoch.

REFERENCIE

- [1] *Moderní počítačová grafika*, Brno: Computer Press, 2004, s. 24 – 29, ISBN 80-251-0454-0.
- [2] G. Blázsovits, *Interaktívna učebnica spracovania obrazu*. 2006, ISBN 80-89186-08-4, Dostupné na internete: <<http://dip.sccg.sk>>.
- [3] G. Bradski, A. Kaebler, *Learning OpenCV*. Computer Vision with the OpenCV Library: O'Reilly Media, Inc., 2008, s.1 – 8, ISBN 978-0-596-51613-0.
- [4] P. Papcun, M. Čopík, J. Ilkovič, *Riadenie robota integrovaného v pružnom výrobnom systéme*. Plzeň: ElectroScope, Vol. 2012, no. 2, 2012, s. 1 – 9, ISSN 1802-4564.
- [5] P. Papcun, M. Čopík, J. Jadlovský, *Distributed control of production system*. Praha: 17th International Student Conference on Electrical Engineering: Poster 2013: 16.5.2013, 2013, s. 1 – 5, ISBN 978-80-01-05242-6.
- [6] M. Petro: *Návrh a realizácia aplikácie na rozpoznávanie obrazu v jazyku C#*. Košice: Bakalárska práca, Katedra kybernetiky a umelej inteligencie, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach, 2013.