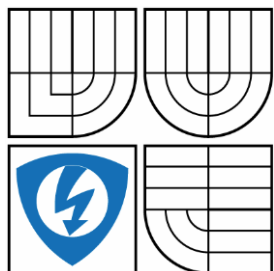


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

VIZUÁLNÍ KONTROLA

VISUAL INSPECTION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

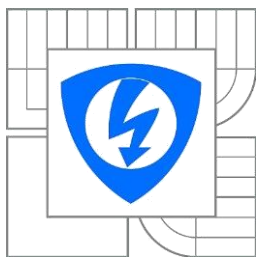
Michal Lancz

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miloslav Richter, Ph.D.

BRNO 2015



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Michal Lancz

Ročník: 3

ID: 146885

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Vizuální kontrola

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvořit aplikaci na hodnocení kvality výrobků podle daných kritérií (délka...) a jejich parametrů (tolerance...).

- 1) Provedte rozbor úlohy. Stanovte parametry a tolerance pro hodnocení kvality součástky včetně přesnosti měření.
- 2) Navrhněte vhodné měřicí pracoviště pro daný typ součástek (jehly).
- 3) Stanovte detekovatelné vady součástek a vytvořte databázi správných a chybných součástek. Navrhněte metody kalibrace nebo učení.
- 4) Navrhněte algoritmy pro nalezení součástky a pro zjištění jejích parametrů.
- 5) Napište aplikaci, která na základě nastavených parametrů stanoví, zda součástka patří do kategorie správných nebo vadných.
- 6) Zhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

Žára J., Beneš B., Sochor J., Felkel P.: Moderní počítačová grafika, Computer Press, 1998, ISBN 80-251-0454-0

Hlaváč V., Šonka M.: Počítačové vidění, Grada, Praha 1992, ISBN 80-85424-67-3

Faugeras O.: Three-Dimensional Computer Vision, The MIT Press 1993

Kraus K.: Photogrammetrie 1 und 2, Ummeler / Bonn, 1996

Termín zadání: 9.2.2015

Termín odevzdání: 25.5.2015

Vedoucí práce: Ing. Miloslav Richter, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Táto bakalárska práca sa zaoberá problematikou vizuálnej kontroly. Ako kontrolované objekty sú zvolené ihly. V práci sú popísané základné poznatky k téme počítačového videnia. V prvej časti sú rozpísané metódy spracovania obrazu ako sú vyhladzovacie filtre a hranové filtre. Zvolené programovacie prostredie je Matlab. Pomocou tohto programu sa dajú prezentovať výsledky jednotlivých metód a výsledky sa ľahko môžu porovnávať. V ďalšej časti sú uvedené príklady výsledných použitých metód na vzorovom obrázku.

Kľúčové slová

Zorné pole, ohnisková vzdialenosť, histogram, prahovanie, šum v obraze, Matlab

Abstract

This bachelor thesis deals with the issue of visual inspection. As a group of controlled objects we have chosen needles. In this work basic knowledge on the topic of computer vision is described. In the first part, methods of image processing such as smoothing filters and edge filters are described. The selected programming environment is Matlab. This environment can easily present and compare the obtained result. Following part presents the examples of the used resultant methods on the sample picture.

Keywords

Field of view, focal length, histogram, thresholding, image noise, Matlab

Bibliografická citace:

LANCZ, M. *Vizuální kontrola*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 52s. Vedoucí bakalářské práce byl Ing. Miloslav Richter, Ph.D.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Vizuální kontrola jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne:

.....
podpis autora

Pod'akovanie

Ďakujem vedúcemu bakalárskej práce Ing. Miloslavovi Richterovi, Ph.D. za účinnú metodickú, pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovaní mojej bakalárskej práce.

V Brně dne:

.....
podpis autora

Obsah

1	Úvod	10
2	ZÁKLADNÉ POZNATKY K SPRACOVANIU OBRAZU	11
2.1	Reprezentácia obrazových dát.....	11
2.2	Digitalizácia	11
2.3	Kvantovanie	11
2.4	Vzorkovanie	12
2.5	Detekcia hrán v obraze	12
2.6	Šum v obraze.....	13
2.7	Zorné pole a ohnisková vzdialenosť	14
2.8	Histogram obrazu	15
2.9	Matlab	15
2.10	Konvolúcia	16
3	Cieľ práce	18
4	Metódy a postupy riešenia algoritmov	19
4.1	Rozbor úlohy a stanovenie chýb	19
4.2	Snímacie zariadenia	19
4.3	Vyhladzovacie filtre	20
4.3.1	Priemerovacie filtre	20
4.3.2	Gaussov filter	21
4.4	Hranové filtre	21
4.4.1	Globálne prahovanie	21
4.4.2	Adaptívne prahovanie	22
4.4.3	Robertsov operátor	23
4.4.4	Sobelov operátor	24
4.4.5	Prewittov operátor.....	25
4.4.6	Laplaceov operátor.....	25
4.5	Návrh pracoviska na snímanie objektu	26
4.5.1	Zrkadlové osvetlenie	27
4.5.2	Rozptýlené osvetlenie	28
4.5.3	Snímanie skenerom	29
4.5.4	Snímanie podsvieteným osvetlením.....	30
4.6	Riešenie stanovených chýb	32
4.6.1	Štandardná poloha ihly a osa ihly	33

4.6.2	Meranie parametrov ihly	33
4.6.3	Hlavná funkcia	34
4.6.4	Funkcia main.m.....	37
4.6.5	Funkcia „search2“	39
4.6.6	Riešenie chyby: ohnutá ihla	39
4.6.7	Riešenie chyby: opotrebovaný hrot.....	41
4.6.8	Riešenie chyby: diera v tele ihly	42
4.6.9	Riešenie chyby: zlomená ihla.....	44
4.7	Meranie dĺžky ihly v programe	46
5	Záver.....	48
5.1	Súčasný stav	48
5.2	Návrh ďalšieho postupu	49

Zoznam obrázkov

Obrázok 2-1 Príklad ideálnej hrany [11] prevzatá a prerobená.....	13
Obrázok 2-2 Histogram obrazu.....	15
Obrázok 2-3 Programové prostredie Matlab.....	16
Obrázok 4-1Prahovanie jedným prahom	22
Obrázok 4-2 Princíp Robertsovho operátora [4]	24
Obrázok 4-3 Porovnanie Sobelovho a Prewittovho operátora	25
Obrázok 4-4 Vyhotovené pracovisko.....	27
Obrázok 4-5 Priame osvetlenie z viacerých zdrojov.....	28
Obrázok 4-6 Snímka skenerom.....	29
Obrázok 4-7 Konštrukcia	31
Obrázok 4-8 Porovnanie svetelných zdrojov	32
Obrázok 4-9 Popis ihly	32
Obrázok 4-10 Ihla a kalibračné znaky	34
Obrázok 4-11 Oknová aplikácia.....	35
Obrázok 4-12 vývojový diagram súčasného programu.....	37
Obrázok 4-13 Najväčšie objekty	38
Obrázok 4-14 Ohnutá ihla.....	39
Obrázok 4-15 Stredná línia ihly	40
Obrázok 4-16 Aplikácia po vyhodnotení chyby: ohnutá ihla.....	40
Obrázok 4-17 Opatrebovaný hrot	41
Obrázok 4-18 Aplikácia po vyhodnotení chyby: opotrebovaný hrot.....	42
Obrázok 4-19 Diera v tele ihly.....	42
Obrázok 4-20 Aplikácia po vyhodnotení chyby: diera v tele ihly	43
Obrázok 4-21 Zlomená ihla	44
Obrázok 4-22 Aplikácia po vyhodnotení chyby: zlomená ihla.....	45

Zoznam tabuliek

Tabuľka 1 Namerané hodnoty s programom	46
--	----

1 ÚVOD

Na snímanie objektov slúžia viaceré obory ako sú: spracovanie obrazu, počítačové videnie, analýza scén alebo analýza digitálneho obrazu a iné. Tieto obory, podobne ako ľudské oko a mozog, slúžia vlastne na to, aby nasníмали obrázok a potom ho vyhodnotili. Snímanie objektov sa využíva v oblastiach, kde je nutná dvoj- alebo viacdimenzová spätná väzba obrazu. Často sa nachádza pri rutinných kontrolách automatizovanej alebo poloautomatizovanej linkovej výroby. Robot, ktorý sa pohybuje v priestore a zároveň vykonáva samostatné úlohy, sa len ťažko orientuje bez počítačového videnia.

Potenciálnou úlohou, ktorá využíva počítačové videnie a spracovanie obrazu môže byť: nasnímanie súčiastok, vyhodnotenie obrazových informácií (či má daná súčiastka správny tvar) a poprípadе detekovanie typu vady na súčiastke (ak sa jedná o kategóriu chybnéj súčiastky). V mojej práci sa budem zaoberať práve touto problematikou a za skúmané súčiastky som si zvolil ihly. V dnešnom modernom svete ihlu považujeme za samozrejmú a považujeme ju ako jednoduchý nástroj na vykonávanie šijacích úkonov.

Predmetom mojej bakalárskej práce je vytvoriť program, ktorý dokáže pomocou nasnímaných obrázkov a algoritmov rozoznať poškodenia ihliel. V tejto práci sú uvedené základné znalosti, ktoré sú potrebné k riešeniu danej úlohy. Popísané sú najrozšírenejšie princípy a metódy spracovania obrazu. Ďalej sú ujasnené základné vlastnosti úlohy a parametre. Zvolená je vhodná kategória objektov na snímanie a sú prezentované testovacie snímky, ktoré budú pri ďalšej práci využívané. Sú stanovené chyby, metódy a procesy pre rozdelenie detekovaných objektov do skupín (vadná alebo vyhovujúca súčiastka). Ďalšou úlohou bude skategorizovať chybu na súčiastke, ktorá bola priradená ako nevyhovujúca súčiastka. V závere je uvedené vyhodnotenie celej práce ako aj výsledky použitých metód a cieľov, ktoré boli dosiahnuté. Táto bakalárska práca je rozšírenou verziou mojej vlašnej bakalárskej práce.

2 ZÁKLADNÉ POZNATKY K SPRACOVANIU OBRAZU

2.1 Reprezentácia obrazových dát

Ako reprezentujeme obraz?

Existuje viacero definícií a jedna z najpoužívanějších je [1]:

„Obraz je možné reprezentovať vo všeobecnosti viacerými spôsobmi. V počítačovej grafike, konkrétne v našom prípade budeme používať dvojrozmerný obraz bodov s obmedzenými rozmermi, matematicky definovaný ako karteziánsky súčin dvoch spojitých intervalov, ktoré vymedzujú rozsah obrazu. Formálne použijeme matematický model, ktorým je spojitá funkcia dvoch premenných – obrazová funkcia:”

$$z = f(x, y) \quad (2.1)$$

ktorú môžeme napísať ako karteziánsky súčin:

$$f: (< x_{min}, x_{max} > \times < y_{min}, y_{max} >) \rightarrow H \quad (2.2)$$

Kde f je reálna, spojitá funkcia, ktorá obsahuje argumenty reálne. Ďalej x a y reprezentujú súradnice bodu v dvojrozmernom obraze, ktorých funkcia nadobúda nejaké hodnoty z oboru hodnôt H . V mojom prípade budem potrebovať jednu hodnotu, lebo program počas spracovania obrazu používa šedotónový obraz.

Pri fotografovaní sa často používa snímka s troma zložkami, ktorými sú RGB (červená, zelená a modrá). Fotografia so zložkami RGB sa dá v prostredí Matlab previesť na šedotónový obraz pomocou funkcie „rgb2gray“. Táto funkcia pretransformuje zložky RGB pomocou odstránením informácie o sýtosti a odtieni pri zachovaní jasů.

2.2 Digitalizácia

Proces, ktorým dostaneme diskretný obraz $I_{i,j}$ zo spojitého obrazu $f(x, y)$ sa nazýva digitalizácia. V počítačovej grafike pracujeme s diskretným obrazom a diskretnými hodnotami. Obraz sa reprezentuje v matici a každá hodnota v matici reprezentuje obrazový bod čiže pixel. Hodnota pixelu udáva jasovú hodnotu obrazového bodu. Digitalizáciu môžeme rozdeliť na dve časti: kvantovanie a vzorkovanie.

2.3 Kvantovanie

Princípom kvantovania je, že obor hodnôt obrazovej funkcie diskretizujeme. Obor hodnôt sa tak rozdelí na intervaly, ku ktorým je pridelená jedna hodnota. Podľa spôsobu

rozdelenia intervalov môže byť rovnomerný a nerovnomerný. Rovnomerné rozdelenie intervalov sa využíva častejšie ako nerovnomerné. Rozdiel medzi rovnomerným a nerovnomerným rozdelením intervalom spočíva v používaní dĺžky intervalu. Kým rovnomerné rozdelenie intervalu používa konštantnú dĺžku intervalu, tak nerovnomerné rozdelenie využíva premenlivú dĺžku intervalu. Rovnomerné rozdelenie intervalu je jednoduchšie realizovateľné a je rýchlejšie pre výpočet, ale presnosť rozloženia hodnôt meranej veličiny je horšia.

2.4 Vzorkovanie

Vzorkovanie spojitaj funkcie znamená, že vo vopred určených intervaloch zaznamenávame hodnoty – vzorky. Takúto funkciu môžeme označiť I_i s intervalom medzi vzorkami označený ako Δx . Vzorky sa dajú nadefinovať nasledujúcim spôsobom:

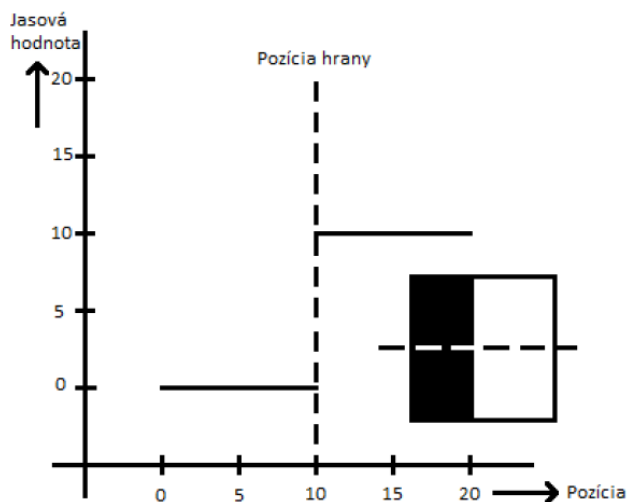
$$I_i = f(x_0 + i\Delta x), i = 0, 1, \dots \quad (2.3)$$

Ak sa berie do úvahy, že iba spojitá funkcia je najpresnejšia, tak môžeme predpokladať, že vzorkovanie spôsobí stratu informácií. Zo vzorca vyplýva, že čím hustejší je počet bodov, teda sú menšie prírastky, tým je presnejší popis obrazu. Cieľom vzorkovania je, aby navzorkovaný obraz bol čo najviac podobný originálnemu spojitému obrazu.

2.5 Detekcia hrán v obraze

V skutočnosti sa hrana dá definovať viacerými spôsobmi. Jedna z najpoužívanějších a najjednoduchších definícií je ideálna hrana. V jednodimenzovom prostredí to znamená, že hrana je jednoznačná zmena v jasovej úrovni v šedotónovom obraze, ktorá nastane v jednom špecifickom mieste. Čím väčšia je zmena v jasovej úrovni, tým jednoznačnejšie sa dá hrana detekovať.

Príklad ideálnej hrany je na obrázku 2-1.



Obrázok 2-1 Príklad ideálnej hrany [11] prevzatá a prerobená

„Každá hrana v obraze predstavuje jasovú nespojitost v obrazovej funkcii.“ [12]. Keď v šedotónovom obraze nájdeme výrazný skok medzi úrovňami, znamená to, že je hrana identifikovaná. Metóda pre hľadanie hrán, ako napríklad derivácia digitálneho obrazu môže zvýrazniť jasové zmeny v obraze.

Kroková hrana prezentovaná v obr. 2-1 je ideálna z dôvodu ľahkej rozpoznateľnosti. Obtiažnosť pri rozpoznateľnosti hrany má v praxi viaceré príčiny.

Jedným z nich je skutočnosť, keď objekt nemá ideálny obrys. Môže totiž nastať situácia, keď je ihla na matnicu položená tak, že ucho ihly nie je viditeľné. V tom prípade na obrázku z fotoaparátu bude rozpoznateľné telo ihly a jej hrot, ale ucho ihly nebude viditeľné. Tento prípad je veľmi nepravdepodobný, preto som sa s týmto prípadom nezaoberal.

Pri snímaní obrazu sa hrana objektu nachádza v rozpätí viacerých pixelov. Nasnímaný obraz obsahuje pixele, ktoré reprezentujú jasové hodnoty spojitého obrazu. Podľa ideálneho obrazu by obrys objektu mal mať nulovú jasovú hodnotu v každom pixely a ostatné body obrazu by mali mať maximálnou jasovou hodnotou. V skutočnosti sa ostatné body v obraze sa približujú k hodnote maximálnej a hrana obrysu objektu s pozadím sa rozprestiera na viacerých pixeloch.

Ďalším dôvodom nerozpoznateľnosti hrany je šum v obraze.

2.6 Šum v obraze

Pri snímaní obrazu je možné, že sa obraz znečistí šumom nejakého typu. Tento šum pri ďalších prácach treba čo najefektívnejšie odstrániť.

Šum v obraze znamená, že signál obrazu má skreslený charakter alebo obsahuje nežiaduci signál, ktorý zakrýva informácie obrazu. V šedotónovom digitálnom obraze je šum vidno ako šedotónové body, ktoré rušia spojitost' obrazu napríklad: zrnitosť.

„V digitálnom obraze rozeznávame najčastejši dva typy šumu:

- náhodný šum, také nezávislý šum, je spôsoben napríklad vadnými CCD elementy

(příkladem tohoto typu šumu je šum typu „sůl a pepř“)

- Gaussův šum, také závislý šum, kde je každý pixel obrazu mírně pozmeněn“ [13]

Náhodný šum může nastat, keď premiestňujeme digitálny obraz z jedného miesta na druhé. Keď nasnímaný obraz je A a šum pri premiestnení je N , tak výsledný obraz B je:

$$B = A + N \quad (2.4)$$

Šum N je nezávislý na obraze A .

Aditívny Gaussov šum je pripočítaná náhodná hodnota ku každému pixelu s Gaussovým rozdelením s nulovou strednou hodnotou. To znamená, že s malou pravdepodobnosťou bude k bodu v obraze pripočítaná vyššia jasová hodnota.

$$G(r, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}} \quad (2.5)$$

[7]

2.7 Zorné pole a ohnisková vzdialenosť

Zorné pole (FOV) ľudského oka je priestor, ktorý je viditeľný bez akéhokoľvek pohybu oka. Zorný uhol ľudského oka je uhol medzi dvoma najvzdialenejšími horizontálnymi alebo vertikálnymi krajnými bodmi, v ktorom dokáže oko vidieť. Tieto dva body v horizontálnom smere sú krajné body vľavo od ľavého oka a vpravo od pravého oka. Veľkosť zorného ľudského oka je približne 90 stupňov od horizontálnej osi hlavy a 50 stupňov od vertikálnej osi hlavy. Na zorný uhol majú vplyv vlastnosti pozorovaného objektu, ako napríklad: intenzita osvetlenia, farba objektu, veľkosť objektu a.i.. [10]

Zorné pole kamery je rozsah pozorovateľného sveta v danom okamžiku. Zorný uhol kamery je napevno daný, nemení sa starnutím ako ľudské oko. V prípade kamery je to uhol medzi krajnými bodmi (vo vodorovnom a zvislom smere), kedy je ešte senzor citlivý na elektromagnetické vlny.

Senzory, ktoré snímajú elektromagnetické vlny sú elektronické snímače CCD a CMOS. Senzor prevedie viditeľné svetlo na digitálne hodnoty. Digitalizácia signálu v čase sa nazýva vzorkovanie a digitalizácia signálu v aplimtúde sa nazýva kvantovanie. Vzorkovanie ovplyvňuje rozlíšenie obrázku. Čím viac vzoriek dokážeme snímať senzorom, tým je rozlíšenie väčšie.

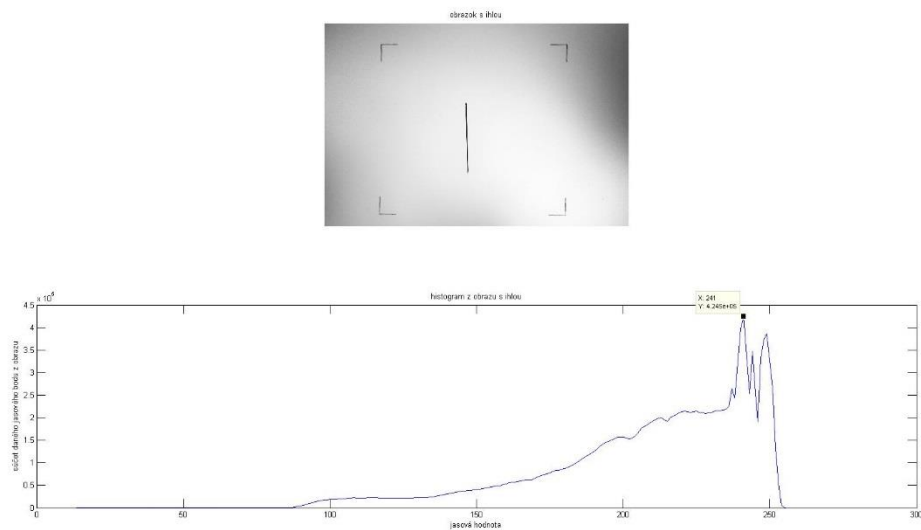
Pri snímaní obrazu je rozhodujúca ohnisková vzdialenosť. Ohnisková vzdialenosť je vzdialenosť snímaného objektu od objektívu. Čím je ohnisková vzdialenosť kratšia, tým väčšie sa reprezentuje snímaný objekt na obrázku.

2.8 Histogram obrazu

Niektoré metódy spracovania obrazu využívajú histogram. Tieto metódy vytvárajú z originálneho obrazu čiastočný alebo celkovo spracovaný obraz.

Histogram jasu obrazu je funkcia $H(p)$ obrazu $f(i,j)$ a udáva počet pixelov jasových úrovní na obrázku. „Môžeme ho chápať aj ako kvantovanú hustotu pravdepodobnosti.“ [9]

To znamená, že u šedotónového 8-bitového obrazu udáva histogram výskyt daného jasu od 0 až do 255.



Obrázok 2-2 Histogram obrazu

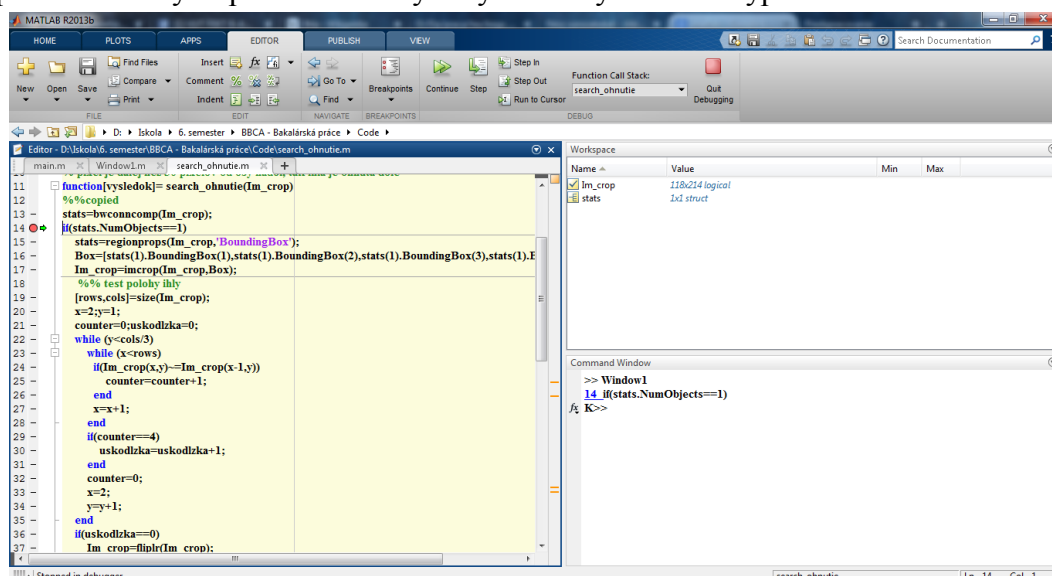
Na obrázku 2-2 vidno na hornej polovici obrázku je originál obrazu a na dolnej polovici obrázku 2-2 je jeho histogram. Na vodorovnej osi grafu sú hodnoty 8-bitového obrazu od 0 do 255. Na zvislej osi grafu je súčet jednotlivých hodnôt obrazu. Z grafu vidíme, že väčšina jasových hodnôt je okolo hodnoty 250. Najväčšia hodnota v grafe je 241 a na obraze je 42 450 pixelov s jasovou hodnotou 241. Diagram vyjadruje skutočnosť, že väčšina pixelov obrázku zodpovedá jasovej úrovni, ktorá sa približuje k bielej farbe (jasová hodnota 255).

2.9 Matlab

Matlab je vysokoúrovňový programovací jazyk a programovacie prostredie. Využíva sa najmä na zložité výpočty, vizualizáciu a programovanie. S Matlabom sa dajú analyzovať údaje, modelovať systémy, vytvárať aplikácie a.i. S jazykom Matlab, matlabovskými prostriedkami a zabudovanými funkciami sa dá problematika riešiť viacerými spôsobmi.

Matlab sa dá použiť pre riešenie rôznorodých problémov ako sú napríklad: spracovanie signálu, spracovanie obrazu alebo videa, riešenie regulačných obvodov, testovanie a meranie systémov a.i. [8]

Ako študent fakulty VUT FEKT som mal dostupnú licencovanú verziu Matlabu. Pri riešení problematiky spracovania obrazu som použil uvedený program. Program Matlab je ideálnym prostriedkom na spracovanie obrazu. Okrem efektívnosti maticových výpočtov obsahuje aj nástroj na spracovanie obrazu tzv. „Image Processing Toolbox“. Tento toolbox ponúka veľké množstvo základných funkcií a algoritmov na spracovanie obrazu. Jednotlivé algoritmy dokážu využívať súbežne viac vlákien v reálnom čase pre výpočet s viacerými procesormi a tým výrazne zrýchliť čas výpočtu.



Obrázok 2-3 Programové prostredie Matlab

2.10 Konvolúcia

Keďže pracujeme s diskretnými funkciami, ktoré používame pre reprezentáciu číslcového signálu, tak pre spracovanie týchto signálov je potrebná konvolúcia v diskretnom tvare. Nasnímaný obraz v Matlabe je spracovaný a konvolúciou je z originálneho obrazu vytvorený nový obraz, ktorý je čiastočne alebo celkovo spracovaný.

„Konvolúcia je metóda, ktorá systematicky prechádza celý obraz a na výpočet novej hodnoty bodu využíva malé okolie O reprezentatívneho bodu. Táto hodnota je zapísaná do nového obrazu. Diskrétna konvolúcia má tvar:

$$g(x, y) = \sum_{(m,n) \in O} h(x - m, y - n) f(m, n) \quad (2.6)$$

kde f je obrazová funkcia pôvodného obrazu, g je obrazová funkcia nového obrazu, h sa nazýva konvolučná maska alebo konvolučné jadro, h udáva koeficienty jednotlivých bodov v okolí O . Najčastejšie sa používajú obdĺžnikové masky s nepárnym počtom riadkov a stĺpcov, pretože v tom prípade môže reprezentatívny bod ležať v strede masky.“ [7]

V lokálnom okolí určitého bodu sa transformácia dá rozdeliť na vyhladzovanie a zaostrenie. Lokálne okolie určitého bodu závisí na matici určitého filtra.

Vyhladzovacie transformácie sú metódy, ktoré sa snažia potlačiť šum v obraze, ale nevýhodou je, že rozostrejujú hrany objektu.

Transformácie typu „rozostrenie“ sú výhodné pri detekcii hrán, ale nevýhodou týchto metód je, že zosilňujú šum v obraze.

Metódy predspracovania je možné rozdeliť aj podľa matematických vlastností. Tieto metódy môžu byť lineárne a nelineárne.

Lineárne metódy, ako napríklad priemerovací filter, počítajú novú jasovú hodnotu daného bodu ako lineárnu kombináciu vstupných bodov. Počet vstupných bodov závisí na veľkosti matice priemerovacieho filtra.

Nelineárne metódy, ako napríklad mediánový filter, berú do úvahy len body s určitými vlastnosťami. Úlohou tohto filtra je, aby eliminoval veľké jasové rozdiely v okolí daného bodu. Táto metóda je vhodná na potláčanie impulzného šumu, ale jeho nevýhodou je poškodzovanie tenkých hrán a ostrých rohov.

3 CIEĽ PRÁCE

Cieľom tejto práce je vytvoriť aplikáciu na vyhodnotenie kvality valcových súčiastok. Ako kontrolované súčiastky som si zvolil ručné šijacie ihly, z dôvodu, že majú valcovitý tvar, sú ľahko rozpoznateľné jej časti a sú súčasťou každonenného použitia v domácnosti. Práca sa člení na :

- 1) rozbor úlohy, stanovenie parametrov a tolerancie
- 2) návrh vhodného pracoviska pre snímanie ihl
- 3) stanovenie detekovateľných chýb ihl, vytvorenie databázy správnych a chybných ihl a návrh kalibrácie
- 4) návrh algoritmov, ktoré zdetekujú chybu ihly podľa bodu číslo 3.
- 5) vytvorenie aplikácie, ktorá vyhodnotí nasnímané ihly

Ihly sú často používané v bežnom priemysle. Ihly na šitie textilu môžeme rozdeliť na ručné a na strojové. Moja aplikácia sa zaoberá klasickými ručnými ihlami. Príklad je na obrázku 4-5.

4 METÓDY A POSTUPY RIEŠENIA ALGORITMOV

Ihly majú valcovitý tvar a jednoznačne rozoznatel'né konce (dierku na jednej strane a hrot na druhej strane). Algoritmy by mali rozhodnúť o tom, či je tvar ihli správny alebo chybný a v prípade chyby by mal algoritmus detekovať vadu.

4.1 Rozbor úlohy a stanovenie chýb

Mojou úlohou je vyhodnotiť kvalitu ihly a vytvoriť na tento účel vyhodnocovacie algoritmy, ktoré budú plniť zadanú úlohu. V prípade, keď algoritmus nájde chybu ihly, tak rozhodne, že ihla nevyhovujúca. Vady alebo nesprávnosti ihly sú rôznorodé. Po dlhšom vyhľadávaní noriem som nenašiel presnú definíciu chýb, preto som stanovil nasledujúce chyby:

- 1) Ohnutá ihla
- 2) Opotrebovaný hrot ihly
- 3) Diera v tele ihly
- 4) Zlom v tele ihly

4.2 Snímacie zariadenia

Pri práci som použil viacero snímacích zariadení. Boli to tri fotoaparáty a jeden skener.

Fotoaparáty som upevnil na konštrukciu špeciálnou skrutkou, ktorú mi vyrobili na objednávku práve pre tento účel. Fotoaparát sa upevní na hornej časti konštrukcie, na matnicu je umiestnená ihla a na dolnej časti konštrukcie svieti svetelný zdroj smerom k fotoaparátu.

Fotoaparát Canon EOS 1000D

1. Snímací senzor: typ CMOS (22,2 x 14,8 mm), všetky pixle 10,5M , efektívne pixle 10,1M, pomer strán 3:2
2. Procesor : DIGIC III
3. Objektív: ekvivalent 1,6-násobku ohniskovej vzdialenosti objektívu
4. Zaostreň: typ TTL-CT-SIR a CMOS senzorom, automaticky zaostruje do 10m, 7 af bodov
5. Výstupný formát obrazu: RAW a JPEG
6. ISO senzitivita: od 100 do 1600

Fotoaparát Canon EOS 100D

1. Snímací senzor: 22.3 x 14.9mm CMOS, všetky pixle 18,5M , efektívne pixle 18,0M, pomer strán 3:2
2. Procesor: Digic 5
3. Objektív: ekvivalent 1,6-násobku ohniskovej vzdialenosti objektívu

4. Zaostrenie: typ TTL-CT-SIR s CMOS senzorom
5. Výstupný formát obrazu: RAW a JPEG
6. ISO senzitivita: od 100 do 1600

multifunkčná tlačiareň Canon MP160

1. Rozlíšenie: do 600x1200 dpi
2. Hĺbka farieb: farebné (48/24 bit), šedotónové (16/8 bit)
3. Efektívna šírka skenovania: A4/LTR

[15]

Fotoaparát Toshiba Camileo P10

1. Snímací senzor: typ CMOS, všetky pixle 5M
2. Výstupný formát obrazu: JPEG
3. Objektív: Pevne zaostrený objektív ($f=7,0$, $F3,2$)
4. Ohnisková vzdialenosť: 20cm (Marco mode); 90cm~200cm (Portrait mode); 200cm ~ nekonečno (Normal mode)
5. ISO senzitivita: 400

[16]

Pri fotení snímok som použil tieto nastavenia:

Canon EOS 1000D: $f=5,6$, $M=1/40$, $ISO=400$, JPEG

Canon EOS 100D: $f=5,6$, $M=1/40$, $ISO=400$, JPEG

Toshiba Camileo P10: Macro mode

4.3 Vyhladzovacie filtre

Vyhladzovanie obrazu filtrom sa používa v prípade, keď šum skreslí nasnímaný obraz. V prípade, ak vieme jednoznačne určiť druh pôsobiaceho šumu, potom ho vieme efektívne odstrániť použitím konkrétneho vyhladzovacieho filtra.

4.3.1 Priemerovacie filtre

Priemerovací filter vytvorí zo vstupného obrazu $f(i,j)$ priemerovaný výstupný obraz $g(i,j)$ konvolúciou. Najpoužívanější je konvolučná maska 3×3 . Táto maska je optimálna pre výpočet a efektívnosť odstránenia šumu. Nevýhodou je, že mierne rozostří výsledný obraz oproti pôvodnému obrazu.

Príklad priemerovacej masky:

$$h_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, h_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, h_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \dots \quad (4.1)$$

,

Maska v okolí každého pixelu vstupného obrazu $f(i,j)$ je vynásobená priemerovacím filtrom. Vo výstupnom obraze je zapísaná hodnota priemerovacieho filtra h .

Priemerovacie filtre môžu mať rôzne masky. Často sa používa v strede masky väčší koeficient, aby počítaný pixel mal dominantnejšiu hodnotu. Tým je dosiahnuté, aby priemerovací filter menej rozostřil výstupný obraz.

4.3.2 Gaussov filter

Na vyhladenie obrazu, ktorý obsahuje Gaussov šum, je optimálny Gaussov filter. Pracuje na podobnom princípe ako priemerovací filter. Gaussov filter má koeficienty dané Gaussovým rozdelením. To znamená, že hodnoty bližšie sa k strede masky majú väčšiu hodnotu ako hodnoty na kraji masky.

4.4 Hranové filtre

Jednoduchým spôsobom vyhľadávania hrán je prahovanie. Prahovanie môže byť globálne alebo adaptívne. Predpokladá sa, že snímaný objekt sa výrazne odlišuje v jasovej úrovni od pozadia.

4.4.1 Globálne prahovanie

Pri práci som vyskúšal metódu globálneho prahovania. V Matlabe bol použitý na obraze s ihlami a výsledok je uvedený na obrázku 4-1.

Globálne prahovanie znamená, že zo vstupného šedotónového obrázku $g(i,j)$ sa vytvorí binárny obraz $f(i,j)$ pomocou jedného alebo dvoch prahov. Pri 8-bitovom obraze môže pixel naberať hodnotu od 0 až do 255, kde 0 znamená najtmavšiu jasovú úroveň a 255 znamená najjasnejšiu jasovú úroveň. Globálny prah bude potom v rozmedzí 0 až 255. 8-bitový obraz znamená, že v obrázku môžu byť hodnoty jasu od 0 až do 2^8 .

Globálny prah je možné zistiť:

1. experimentálne
2. z histogramu
3. percentuálne
4. z globálnych znalostí

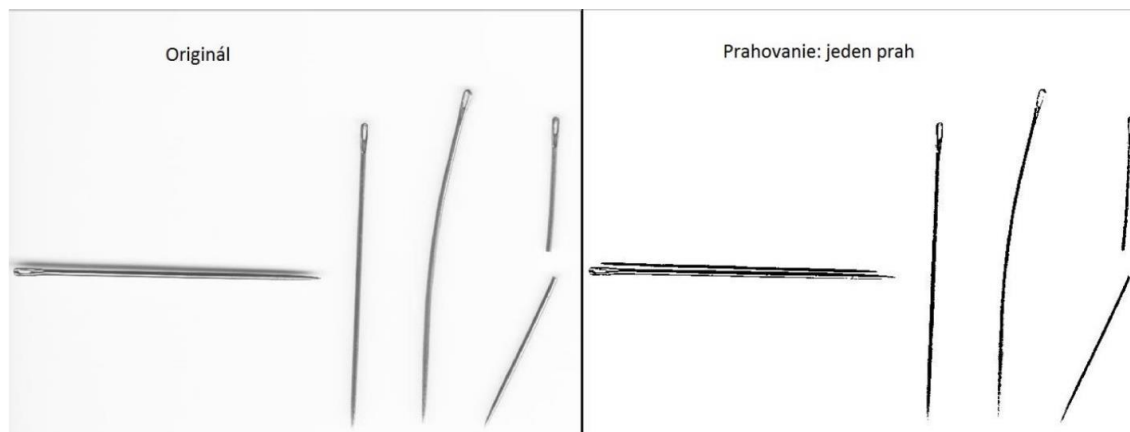
Používanie jedného prahu znamená, že keď pixel má hodnotu väčšiu alebo rovnú prahu T , teda vo výslednom obrázku bude mať hodnotu 1. Keď bod v obrázku je pod prahom, bude mať hodnotu 0.

$$f(i,j) = \begin{cases} 1 & \text{pre } g(i,j) \geq T \\ 0 & \text{pre } g(i,j) < T \end{cases} \quad (4.2)$$

[3]

Po prahovaní dostaneme binárny (čiernobiely) obraz. Na ďalšom obrázku je znázornené prahovanie jedným prahom podľa histogramu. Metóda je aplikovaná v matlabovskom prostredí.

Príklad prahovanie jedným prahom:



Obrázok 4-1 Prahovanie jedným prahom

Na ľavej strane obrázku 4-1 je originál obrázku v šedotónovom, 8-bitovom prevedení a na pravej strane je obraz čiernobiely. Čiernobiely obraz sme dostali prahovaním v prostredí Matlab. Prah bol nastavený na hodnotu 150. To znamená, že všetky pixele, ktoré boli menšie než 150, vo výslednom obrázku majú hodnotu 0 a pixele ktoré majú hodnotu 150 alebo viac až do 255 majú hodnotu 1. Ako na obrázku vidíme, globálne prahovanie vytriedi pixely podľa jedného kritéria. Dôsledkom toho môže byť skreslený obrys ihly pri použití priameho osvetlenia na objekt. Hrana ihly alebo telo ihly odrazí svetlo a tak na obraze bude vidno svetlejší odlesk.

Prahovanie s dvoma alebo viacerými prahmi je podobné ako prahovanie s jedným prahom. Má vstup $f(i,j)$, ale na výstupe $g(i,j)$ už nebude obraz binárny. Výstupný obraz bude mať $n+1$ jasových úrovní, kde n znamená počet prahov.

$$f(i,j) = \begin{cases} 1 & \text{pre } g(i,j) \in A_1 \\ 2 & \text{pre } g(i,j) \in A_2 \\ \dots & \\ n & \text{pre } g(i,j) \geq A_n \\ 0 & \text{inak} \end{cases} \quad (4.3)$$

[3]

Globálne prahovanie som použil pri spracovaní obrázku pre jeho jednoduchosť. Výsledky touto metódou boli postačujúce.

4.4.2 Adaptívne prahovanie

V prípade, keď je na obrázku nasnímaných viacero objektov, ale majú rozdielne jasové úrovne alebo keď je na obrázku objekt nerovnomerne osvetlený, tak je možné používať adaptívne prahovanie. Pri metóde adaptívneho prahovania sa dá určiť bod prahu pre každý bod v obraze zvlášť. Tým sa zvýši úspešnosť vyhľadávania hrán, ale rýchlosť výpočtu sa zníži.

Chow a Kanenho metóda je jedným zo spôsobov adaptívneho prahovania. Chow a Kanenho metódou rozdelíme obrázok na viac častí. Po rozdelení vypočítame osobitne pre

každú časť optimálny prah z histogramu. Pre daný bod vypočítame prahovú hodnotu interpoláciou prahových hodnôt podobrazov. Pre real-time je jeho používanie menej vhodné z dôvodu náročného výpočtu.

V mojom programe používam adaptívny priemerovací filter pre zvýraznenie vlastností obrazu, na segmentáciu a generovanie binárneho obrazu som použil Otsuho metóda. Po zvolení malej časti okolia daného bodu v obraze (masku) a pomocou tejto masky sa dá vypočítať prahová hodnota. Prahovú hodnotu je možné vypočítať priemerovaním jasových hodnôt okolitých pixelov, mediánom alebo strednou hodnotou najväčšej a najmenšej hodnoty okolitých bodov v obraze. Čas výpočtu sa predĺži zväčšením veľkosti masky. [6]

Ďalším spôsobom hranových filtrov je Robertsov operátor. Hranu v diskretnom obraze vnímame ako výraznú zmenu v susedných pixeloch. Hrana je určená gradientom, t.j. veľkosťou a smerom. Smer sa dá opísať vektorovým operátorom nabla ∇ :

$$\nabla f(x, y) = \left(\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right) \quad (4.4)$$

a veľkosť gradientu môžeme vypočítať ako dĺžku vektora:

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x} \right)^2 + \left(\frac{\partial f(x, y)}{\partial y} \right)^2} \quad (4.5)$$

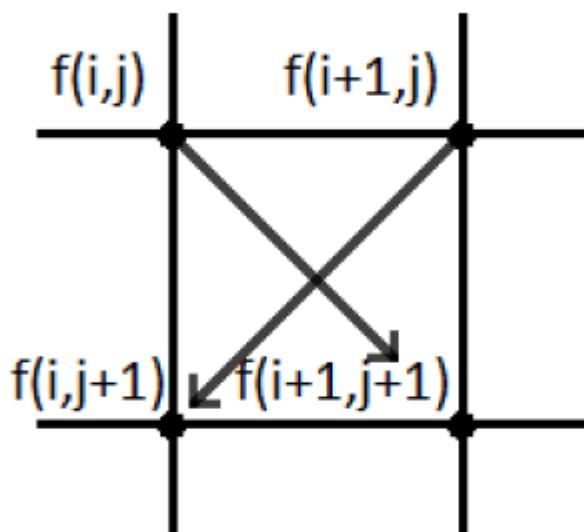
[4]

4.4.3 Robertsov operátor

Tento operátor nepoužíva konvolúciu, ale používa pre výpočet gradientu daný pixel a 3 okolné pixely. Robertsov operátor má nasledovný tvar:

$$|\nabla f(i, j)| = |f(i, j) - f(i + 1, j + 1)| + |f(i, j + 1) - f(i + 1, j)| \quad (4.6)$$

Robertsov operátor detekuje hrany na vstupnom obraze $f(i, j)$ kolmo po hlavnej a po vedľajšej diagonále. Súčtom absolútnych hodnôt zmien v smere hlavnej a vedľajšej diagonály vypočítame gradient.



Obrázok 4-2 Princíp Robertsovho operátora [4]

V maticovom prevedení to vyzerá nasledovne:

$$h_1 = \begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix}, h_2 = \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix} \quad (4.7)$$

V rovnici 4.6 je reprezentovaná maska v smere diagonály a rovnica 4.7 je ukážkou masky v smere vedľajšej diagonály. Vo výstupnom obraze $g(i,j)$ sú zapísané hodnoty súčtov oboch rovníc. Takto určíme hrany použitím Robertsovho operátora.

4.4.4 Sobelov operátor

Podobne ako to bolo pri Robertsovom operátore, pri Sobelovom operátore je výpočet gradientu smerovo závislý. Maska Sobelovho operátora má 8 variácií. Každá maska je vhodná pre výpočet gradientu iného smeru (vertikálne smery zhora nadol a opačne, horizontálne zľava doprava a opačne, na oboch diagonálach v oboch smeroch).

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}, h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \dots \quad (4.8)$$

Sobelov operátor slúži pre výpočet gradientu vo vertikálnom smere zdola nahor, po vedľajšej diagonále opačným smerom, atď..

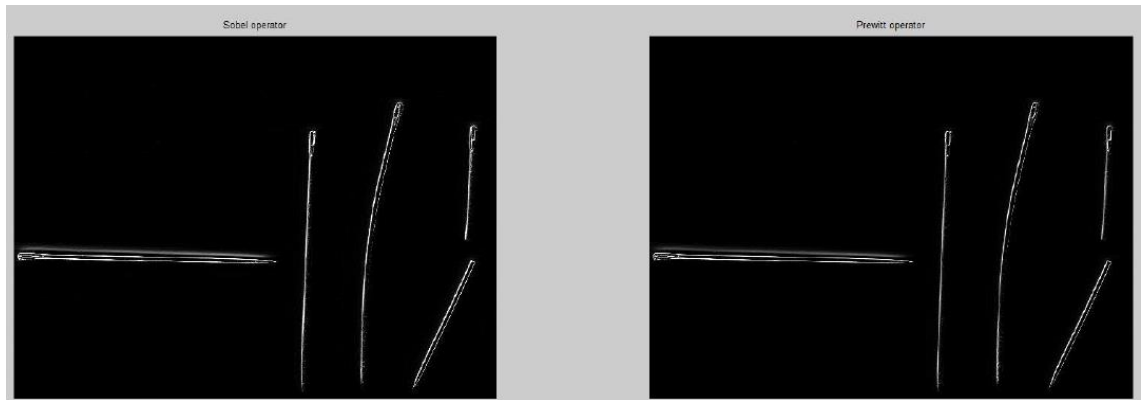
Robertsov a Sobelov operátor aproximujú výpočet prvej derivácie. Laplaceov operátor aproximuje výpočet druhej derivácie a je invariantný na rotáciu. To znamená, že výpočet bude zhodný na ktorýkoľvek smer výpočtu gradientu.

4.4.5 Prewittov operátor

Prewittov operátor pracuje podobným princípom ako Sobelov operátor. Rozdiel medzi operátormi Sobel a Prewitt sú koeficienty konvolučnej masky.

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}, h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \dots \quad (4.9)$$

Výsledky Sobelovho a Prewittovho operátora sa veľmi podobajú.



Obrázok 4-3 Porovnanie Sobelovho a Prewittovho operátora

Na ľavej strane obrázku je znázornený výsledok súčtu vertikálneho a horizontálneho smeru Sobelovho operátora. Na pravej strane je výsledok súčtu vertikálneho a horizontálneho smeru Prewittovho operátora. V praxi sa používa často Sobelov operátor. Pre experimentálne pokusy som tiež použil Sobelov operátor.

4.4.6 Laplaceov operátor

„Laplaceov operátor (alebo len Laplace) je diferenciálny operátor vo vektorovej analýze, definovaný ako divergencia gradientu daného skalárneho, alebo vo všeobecnosti tenzorového poľa. Ak je aplikovaný na skalárne pole, výsledkom je opäť skalárne pole, ak je aplikovaný na tenzorové pole, výsledkom je tenzorové pole rovnakého stupňa. Označuje sa symbolom Δ “ [5]

Laplaceov operátor je známy ako operátor priechodu nulou (zero crossing operator). Matematický opis Laplaceovho operátora na dvojdimenzový obraz vyzerá nasledovne:

$$\Delta f(x, y) = \frac{\partial^2 f(x, y)}{\partial^2 x} + \frac{\partial^2 f(x, y)}{\partial^2 y} \quad (4.10)$$

Tento operátor reaguje na hrany dva krát. Prvý krát, keď hrana je vzostupná a druhý krát, keď hrana je zostupná. V prvom prípade, keď operátor reaguje na vzostupnú hranu,

má hodnotu kladnú a pri zostupnej hrane má hodnotu zápornú. V praxi používame buď len kladnú hodnotu, to znamená, že vzostupná hrana sa neberie do úvahy, alebo počítame s absolútnou hodnotou, a tak zo záporných hodnôt dostaneme kladné hodnoty.

Laplaceov operátor je použiteľný s dvoma konvolučnými maskami:

a) výpočet s operátorom v horizontálnych a vertikálnych smeroch:

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.11)$$

b) výpočet s operátorom v horizontálnych, vertikálnych a diagonálnych smeroch:

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.12)$$

Rozdiel medzi konvolučnými maskami 4.11 a 4.12 spočíva v tom, že v prípade masky 4.11 zanedbáme smery diagonálne a tak hrany v týchto smeroch sa vo výslednom obraze $g(i,j)$ nezobrazia.

Nevýhodou tohto operátora je, že zvýrazní všetky vysokofrekvenčné hrany, teda aj šum. Pre menšie zvýraznenie šumu je vhodné zvoliť konvolučnú masku väčšiu ako 3×3 .

4.5 Návrh pracoviska na snímanie objektu

Snímanie objektu nie je vždy jednoduchá záležitosť. Nedeštruktívna vizuálna kontrola obvykle funguje takým spôsobom, že svetelný zdroj nasmerujeme na snímaný objekt a následný obraz (po dopade svetla na objekt) je snímaná vhodnou snímacou technikou napr. kamerou.

Vlastnosti objektov pre vizuálnu kontrolu ovplyvňujú výber metódy snímania. Najdôležitejšie z týchto vlastností objektov sú:

1. Objekt je transparentný (priesvitný) alebo polotransparentný
2. Objekt má lesklý povrch
3. Objekt má matný povrch
4. Objekt odrazí svetelné lúče iným spôsobom, než to je žiadúce

Keďže som si ako snímaný objekt vybral ihly, bolo potrebné vyhotoviť vhodnú konštrukciu, pomocou ktorej budem vedieť ihly nasnímať. Skelet konštrukcie sa skladá z troch častí a to:

1. podstavec
2. snímacia plocha
3. držiak snímacieho zariadenia.

Všetky tri časti sú vyhotovené zo železných L-profilov , ktoré som zváraním spojil do jedného celku tak, ako je to vidno na obrázku 4-4. Stredná časť konštrukcie mi slúžila ako snímacia plocha a vytvoril som ju jednoduchým položením mliečneho skla vhodných rozmerov na konštrukciu. Vrchná časť kostry konštrukcie slúži na ukotvenie snímacieho zariadenia. Na tento účel som si dal vyrobiť špeciálny skrutkový spoj, ktorý slúžil na pevné ukotvenie kamery.

Pomocou tejto konštrukcie som vyhotovil vhodné snímky na ďalšie spracovanie obrazu. Pri návrhu pracovnej scény som postupoval nasledovným spôsobom. Na stôl som položil zhotovený stojan. Na snímacu plochu som položil matnicu so zakreslenými kalibračnými znakmi vo vzdialenosti 7 cm od seba. Fotoaparát som pripevnil na vrchnú časť stojana – držiak tak, aby bol objektív nasmerovaný kolmo na kalibračné znaky na matnici. Do priestoru medzi kalibračné znaky som po jednom položil skúmané ihly a zhotovil som za každým príslušný snímok.

Použil som pri tom metódu tzv. podsvieteného osvetlenia tak, aby svetelné lúče dopadali na snímacu plochu zospodu kolmo. Preto som uložil pod matnicu textíliu z fotografického dáždika na plošinu v 45 stupňovom uhle. Textília má postriebný povrch a slúži na rovnomerné rozptýlenie svetla na matnicu. Cieľom bolo získať čo najkontrastnejšie snímky.

Výsledky získané po vyhodnotení programu som zhrnul v kapitole „Meranie dĺžky ihly v programe“.



Obrázok 4-4 Vyhotovené pracovisko

Nasledujúce podkapitoly popisujú metódy snímania objektov.

4.5.1 Zrkadlové osvetlenie

Pri snímaní objektov je najjednoduchším spôsobom použiť zrkadlové osvetlenie. Znamená to, že medzi zdrojom svetla a kamerou od snímaného objektu je malý uhol a tak kamera sníma odrazené lúče od objektu a jeho okolia.

Tento spôsob snímania na meranie objektu sa v praxi nepoužíva často. Veľkou nevýhodou tejto metódy je, že snímaný objekt vytvára tieň na pozadí. Po spracovaní obrazu môže tieň pôsobiť ako falošná hrana objektu.

Okrem tieňa je problém s objektmi, ktoré majú lesklý povrch a odrážajú lúč zdroja svetla. To znamená, že keď sú snímané poniklované ihly touto metódou, odráža sa priame biele svetlo zdroja v určitých oblastiach. To môže pôsobiť ako falošná hrana pri spracovaní obrazu.

Táto metóda sa je vhodná na kontrolu kvality niklovania ihl. Keď na lesklý poniklovaný povrch ihly svietime rovnobežným svetlom, tak telo ihly odráža zdroj svetla rovnomerne smerom ku kamere. Podľa stupňa odrážaného lesku môže uvedený algoritmus vyhodnotiť správnosť poniklovania ihly.

4.5.2 Rozptýlené osvetlenie

Rozptýlené osvetlenie je podobné ako zrkadlové osvetlenie. Rozdiel medzi týmito dvoma metódami môže byť v pridaní ďalšieho svetelného zdroja alebo vytvorení rozptylu svetelných lúčov (napr. prostredníctvom matnice) medzi zdrojom svetla a snímaným objektom.

Pridanie ďalšieho svetelného zdroja spočíva v použití viacerých bodových svetelných zdrojov s rovnakými parametrami. Namiesto bodových zdrojov svetla je možné použiť zdroj typu reflektor. Cieľom použitia takýchto zdrojov svetla je, aby lúče nesmerovali len z jedného bodu na snímaný objekt, ale smerovali z viacerých bodov. Takéto osvetlenie pôsobuje, že na pozadí sa tieň objektu rozdelí na tieň a polotiene.



Obrázok 4-5 Priame osvetlenie z viacerých zdrojov

Druhým spôsobom rozšírenia svetelného zdroja je použitie zdroja rovnobežného svetla. Rovnobežný svetelný zdroj vyžaruje lúče rovnomerne na objekt z viacerých strán a tak sa vytvárajú na pozadí rovnomerné polotiene a tieň.

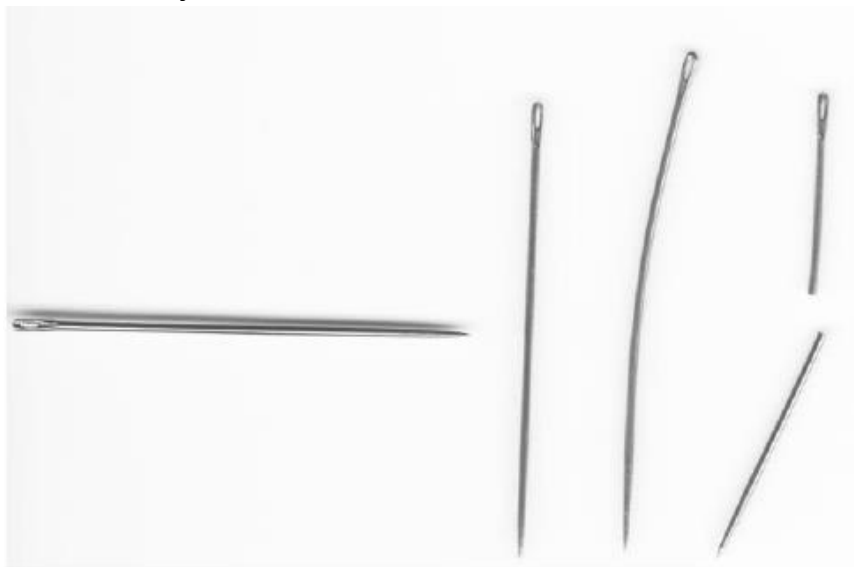
4.5.3 Snímanie skenerom

Snímky boli snímané multifunkcionálnym skenerom MP160. Tento skener má za hlavný účel snímať dokumenty a fotky. Pri snímaní ihlíc týmto skenerom som vyskúšal viaceré nastavenia snímania, ktoré prístroj umožnil. Pri spracovaní som vybral z naskenovaných snímok ihlíc ten, ktorý má najlepšie vlastnosti, čo sa týka rozlíšenia a kontrastu obrysu oproti pozadiu. Výsledok je na obrázku 4-6. Skener Canon MP160 má snímacie rozlíšenie 600 dpi.

Nedostatky metódy, ktoré ovplyvňujú snímanie:

1. Obraz musí byť „statický“, pretože skener musí snímať celú plochu bez pohybu (rádovo sekundy)
2. Mechanické vady alebo nečistoty na priesvitnom skle alebo na pozadí skenera vytvárajú falošný obrys na obrázku
3. Obraz sa nedá zväčšiť, pretože pomer pixelov na milimeter je pevne daný
4. Podsvietenie skeneru priamo odráža odlesk z ihly a tak vytvára falošný kontrastný obrys

Tieto nedostatky sa dajú kompenzovať zvýšením kvality použitých materiálov na snímanie objektov. Zorné pole skenera je nastavené výrobcom. Falošné kontrastné obrysy, ktoré sú spôsobené svetlom dopadajúcim priamo na objekt sa dajú redukovať znížením intenzity osvetlenia.



Obrázok 4-6 Snímka skenerom

Na obraze 4-6 som uviedol príklad snímania skenerom. Na snímke vidno, že na pozadí sa vytvára tieň. Na tele ihly sa odráža svetlo zdroja od lesklého povrchu. Pri

vyhľadávání hrán z tohto obrázku by som algoritmom našiel viac falošných hrán. Práve z toho dôvodu je pre moju prácu tento typ snímania nevhodný.

4.5.4 Snímanie podsvieteným osvetlením

Pri štandardnom osvetlení môže snímka obsahovať nejednotné osvetlenie, odzrkadlenie, falošné tieň a iné chyby, ktoré by spôsobili nepresnosť merania. Preto som použil osvetlenie podsvietením ako vidno na ilustračnom obrázku 4-7. Takýto spôsob osvetlenia zaisťuje plynulosť svetla samozrejme okrem objektu, ktorý chceme kontrolovať. Na snímke tak bude vidno obrys snímaného objektu bez rušivých odleskov. Cieľová oblasť bude mať nízku jasovú hodnotu, kým pozadie bude mať vysokú jasovú hodnotu.

Takýmto spôsobom sa v priemyselnej technike dá kontrolovať úroveň kvapaliny v priesvitnej fľaši. V našom prípade na obrázku bude ľahko rozpoznateľné ucho, telo a hrot ihly.



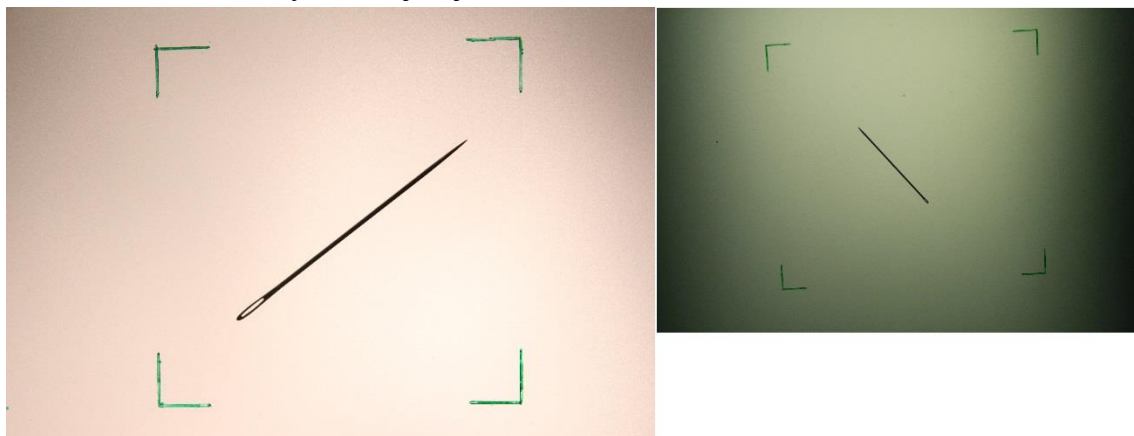
Obrázok 4-7 Konštrukcia

Na meranie ihl som vyhotovil konštrukciu zo železa, ktorá je znázornená na obrázku 4-7. Táto konštrukcia obsahuje základný rám a stojan. Stojan je opretý cez železné nosníky. Na stojane sú držiaky pre polo transparentné sklo (matnica). Na hornej časti stojana je držiak na kameru. Svetelný zdroj som umiestnil tak, aby jeho lúče sa rozptýlili na striebornej textílii a odtiaľ sa odrazili na matnicu smerom zo spodu.

Konštrukciu som vyhotovil tak, aby sa pre ďalšie práce vizuálnej kontroly mohli rozšíriť ďalšie možnosti snímania. Na hornej časti je možné pridať ďalšie zdroje svetla. Tieto zdroje môžu slúžiť na snímame metódou zrkadlového osvetlenia.

Ihlu (resp. ihly) som položil na matnicu. Ako zdroj svetla som vyskúšal svetelný zdroj s volfrámovou žiarovkou a žiarivku. Žiarivka, ktorú som použil je 11 wattová a volfrámové žiarovky boli 40 wattové.

Rozdiel medzi svetelnými zdrojmi je vidno na obrázku 4-8.



Obrázok 4-8 Porovnanie svetelných zdrojov

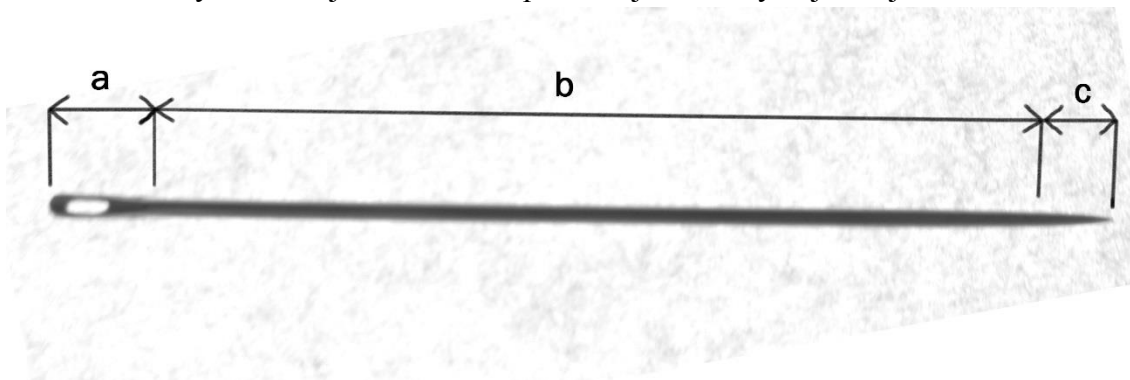
Na ľavej strane obrázku 4-8 je snímok s dvomi 40 wattovými volfrámovými svetelnými zdrojmi a na pravej strane je obraz s jednou 11 wattovou žiarivkou. Počas spracovania obrazu je dôležité, aby na obrázku boli čo najjednoduchšie rozpoznateľné všetky hrany.

Na hornú časť stojanu som umiestnil kameru takým spôsobom, že objektív bol nasmerovaný kolmo na matnicu smerom dole. Vyhotovený obraz som uložil do počítača do adresára programu. Z adresára programu sa obraz načíta do prostredia Matlab a pomocou vytvorených algoritmov vyhodnotí nasnímaný objekt.

4.6 Riešenie stanovených chýb

V tejto kapitole sú opísané postupy riešenia, ktoré som použil pri tvorbe algoritmov a prostredia snímania.

Po dlhšom vyhľadávaní noriem som nenašiel presnú definíciu parametrov skúmaného objektu. Pre ďalší popis objektu snímania som si stanovil vlastné parametre a vlastné tolerance častí ihly. Nasledujúci obrázok reprezentuje snímaný objekt a jeho časti.



Obrázok 4-9 Popis ihly

Šijacie ihly som rozdelil na tri hlavné časti:

1. Ucho (uško) ihly
2. Telo ihly

3. Hrot (špica) ihly

Celkovú dĺžku ihly tvorí súčet dĺžok ucha ihly, tela ihly a hrotu ihly dohromady.

4.6.1 Štandardná poloha ihly a osa ihly

Pre ďalšie spracovanie obrazu som stanovil štandardnú polohu ihly. To znamená, že ihla na obrázku je orientovaná vodorovne (paralelne k ose x). Ucho ihly je na obraze na ľavej strane a hrot ihly je na pravej strane. Z tejto úvahy vyplýva, že vstupným parametrom algoritmov pre vyhodnotenie ihly je obraz v štandardnej polohe.

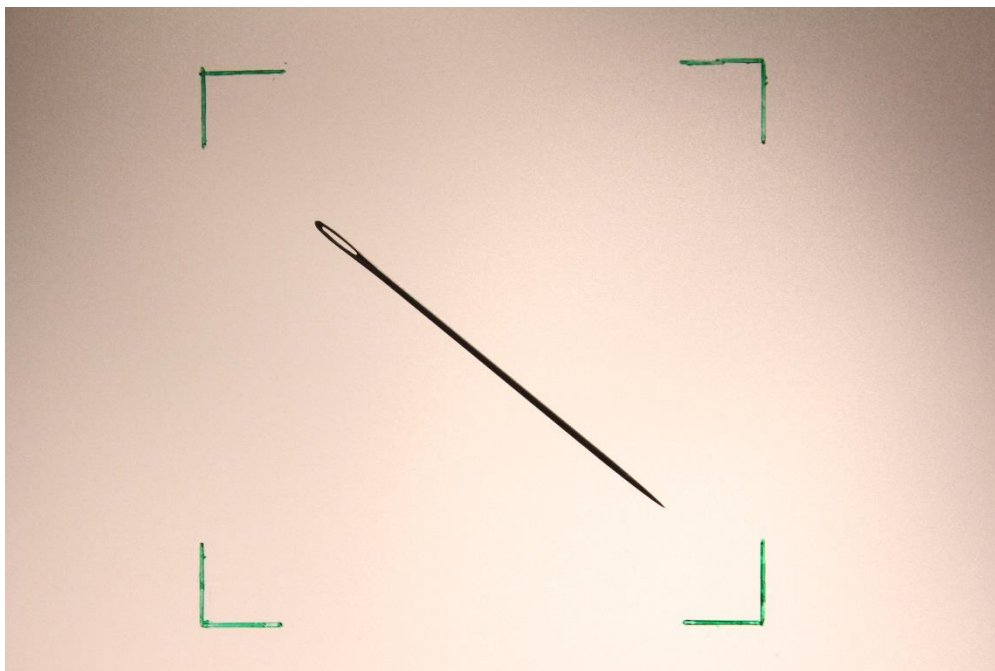
Aby som získal ihlu v štandardnej polohe, prevediem nasnímaný obraz na šedotónový obraz a potom s adaptívnym prahovaním na čiernobiely obraz. S funkciou „regionprops“ viem nájsť najmenší štvorec okolo každého objektu, plochu, dĺžku, výšku a orientáciu vzhľadom k vodorovnej osi každého objektu. Tieto informácie uloží do premennej „stats“. Každý objekt v premennej stats má samostatný index. Predpokladám, že na obrázku nebudú žiadne nečistoty väčšie ako kalibračný znak. Vyberiem index najväčšej plochy a predpokladám že je to index ihly. Podľa najmenšieho štvorca okolo najväčšej plochy „stats.BoundingBox“ orežem obrázok s funkciou „imcrop“. Tento obraz otočím podľa orientácie, ktorú som získal do premennej „stats“. Takto získaný obraz ihly je teraz vo vodorovnej polohe, ale je potrebné ešte určiť či hrot ihly je na ľavej alebo na pravej strane obrazu. Keď hrot ihly je na ľavej strane obrazu, je potrebné otočiť celý obraz podľa zvislej osi.

V programe hľadám otvor (rozšírenie) v ľavej polovici obrázku. S „for“ cyklom prejdem každý stĺpec obrazu. Keď po stĺpcoch prejde cyklus štyrikrát z minimálnej hodnoty jasú na maximálnu hodnotu jasú, znamená to, že v hľadanom stĺpci sa nachádza časť ucha ihly. Súčet takýchto stĺpcov reprezentuje dĺžku ucha ihly na ľavej strane obrazu. Keď cyklus na ľavej strane obrazu nenašiel ani jeden taký stĺpec, znamená to, že je ihla orientovaná v opačnom smere. Maticu obrázku v takom prípade invertujem podľa osi y. Takým spôsobom dostaneme ihlu do štandardnej polohy.

4.6.2 Meranie parametrov ihly

Pre meranie parametrov ihly je potrebné nasnímať spolu s ihlou kalibračné znaky. Na matnicu som nakreslil kalibračné znaky ako vidno na obrázku 4-10. Tieto znaky sú na seba kolmo a od seba vzdialené na dĺžku 7 centimetrov. Kalibračné znaky sú vlastne dve jedno centimetrové úsečky, ktoré sú umiestnené na seba kolmo a vytvárajú pravý uhol.

Podľa týchto parametrov ďalej môžem prepočítať koľko centimetrov zodpovedá počtu pixelov.



Obrázok 4-10 Ihla a kalibračné znaky

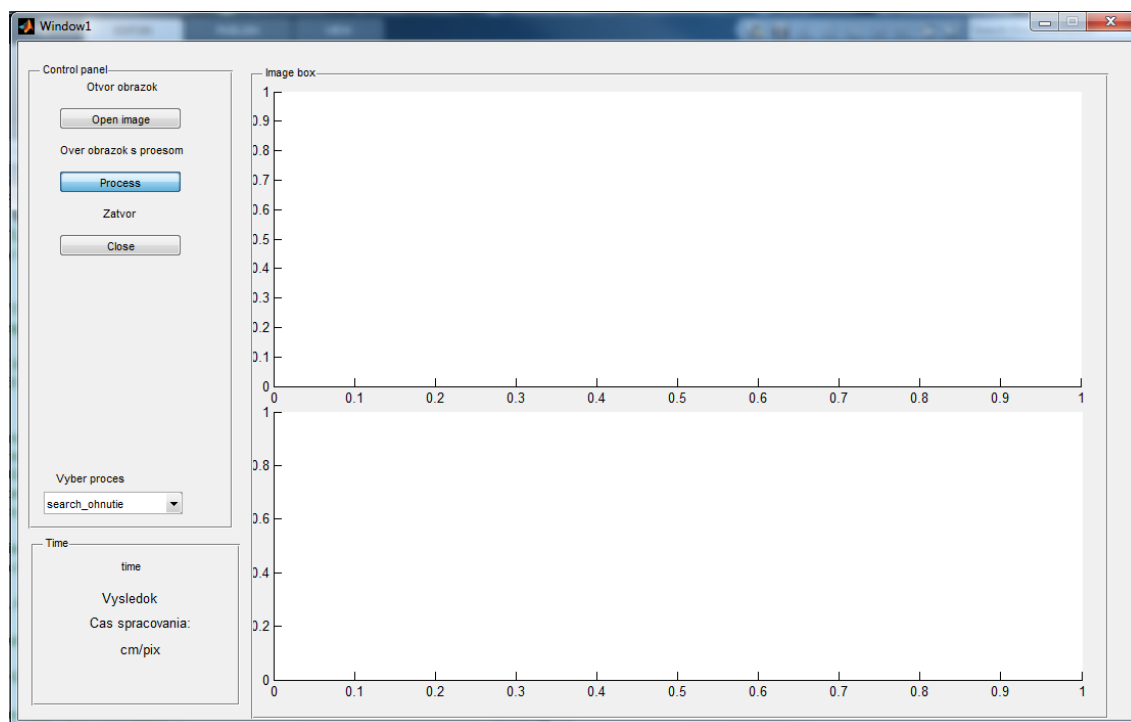
Najprv obraz prevediem na šedotónový a potom s adaptívnym prahovaním na čiernobiely. S funkciou „regionprops“ viem nájsť plochu najväčšieho objektu. Predpokladám že tento objekt je ihla.

„MajorAxisLength“ a „MinorAxisLength“ vráti skalár, ktorý určuje dĺžku (v pixeloch) hlavnej a vedľajšej osi.

S vlastnosťami „regionprops“ ako sú „MajorAxisLength“ a „MinorAxisLength“ viem akú dĺžku a výšku má ihla. Dĺžku ihly prepočítam v pomere pixel na milimeter.

4.6.3 Hlavná funkcia

Pre lepšiu orientáciu užívateľovi pri kontrole objektu som vytvoril v Matlabu oknovú aplikáciu pomocou funkcie „guide“. Táto funkcia otvorí dialógové okno, ktoré umožňuje vybrať medzi možnosťami vytvoriť oknovú aplikáciu podľa vzoru, otvoriť uloženú oknovú aplikáciu alebo vytvoriť novú prázdnu aplikáciu. Moja aplikácia je znázornená na obrázku 4-11.



Obrázok 4-11 Oknová aplikácia

V pravej časti aplikácie je panel „Image box“. Tento panel sa delí na dve časti hornú a dolnú časť. Po otvorení aplikácie každá časť obsahuje osi pre grafický objekt. Po výbere obrázku v hornej časti panela je načítaný obraz z adresára do Matlabu. V dolnej časti panela je obraz po spracovaní a vyhodnotení ihly.

V ľavej hornej časti aplikácie je „Control panel“. V tomto paneli sú tlačítka „Open image“, „Process“, „Close“ a pop-up menu pre výber procesu.

S tlačítkom „Open image“ sa otvorí dialógové okno pre výber obrázku typu „.jpg“, „.tif“, „.png“ a „.gif“. Po výbere obrázku sa načíta obraz do Matlabovského adresára a vykreslí sa načítaný obraz do pravej hornej časti aplikácie.

Prvým krokom po načítaní obrázku je výber typu algoritmu a druhým krokom je spustenie vybraného algoritmu. Typ algoritmu je uložený do Matlabu ako „proces“. Ako prvý v poradí je nastavený algoritmus „search_ohnutie“. Z toho dôvodu, keď užívateľ po štarte programu a otvorení obrázku neuskutoční sám výber procesu, následne sa spustí algoritmus typu „ohnutie ihly“, lebo je nastavený ako prvý v poradí. Po výbere procesu a stlačení tlačítka „Process“ podľa daného algoritmu vyhodnotí program či je ihla vyhovujúca alebo je ihla vadná. Posledný použitý obraz počas vyhodnotenia sa vykreslí na pravú dolnú stranu aplikácie. Do informačného panela je vypísaný výsledok vyhodnotenia ihly.

V prípade, keď používateľ otvorí dialógové okno pre výber obrázku ale nezvolí obrázok, tak sa nenačíta žiadny obrázok do Matlabovského adresára. Takisto nevykreslí obraz do panela „Image box“ a v prípade, keď používateľ stlačí tlačítko „Process“, tak výsledok bude „No image“.

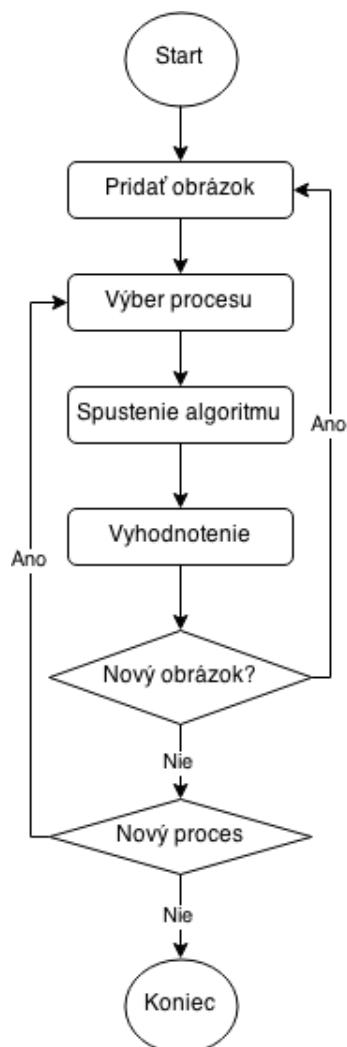
V ľavej dolnej strane aplikácie je panel „information“. V tomto paneli sú ďalšie informácie ohľadne spracovania. Panel obsahuje čas spracovania obrazu, výsledok vyhodnotenia a vypočítaný pomer pixelu na milimeter.

Čas spracovania obrazu meriam pomocou funkcií „tic“ a „toc“. Funkcia „tic“ spustí stopky podľa centrálnej procesorovej jednotky a funkcia „toc“ načíta uplynulý čas od momentu spustenia stopiek. Tieto dve funkcie navzájom spolupracujú. Ako prvá funkcia je „tic“ a uplynulý čas uloží do premennej „proc_time“ pomocou „toc“. Konečný čas spracovania reprezentuje uplynulý čas od začiatku funkcie „main.m“ až do konca tejto funkcie. Typ tejto premennej je double. Do textového pola „processtime“ v informačnom paneli je vypísaný ako string.

```
str1=strcat('Cas spracovania: ',num2str(proc_time));  
str1=strcat(str1,' sekund(a/y));  
set(handles.processtime,'String',str1);
```

Pomocou funkcie „num2str“ prevedie program typ double na typ string. Funkcia „strcat“ zret’azí premenné typu string.

Na ďalšom obrázku je znázornený priebeh programu prostredníctvom vývojového diagramu. Podrobnejší popis algoritmov je uvedený v ďalších kapitolách.



Obrázok 4-12 vývojový diagram súčasného programu

4.6.4 Funkcia main.m

Cieľom funkcie main.m je, aby načítaný obrázok z adresáru matlab predal do funkcie, ktorú vybral užívateľ na riešenie chyby ihly. Vstupnými parametrami funkcie „main.m“ sú vybraný proces a načítaný obraz. Na získanie procesu z pop-up menu a obrázku z hlavnej funkcie mi slúžia nasledovné príkazy

```

popupmenu4value=get(handles.setprocess, 'String');
process = popupmenu4value{get(handles.setprocess,'Value')};

```

Prvý príkazový riadok mi umožní získať všetky možnosti z rozbaľovacieho menu a uložiť ich do stĺpcovej matice do premennej „popupmenu4value“.

Druhý príkazový riadok získa aktuálnu vybranú hodnotu z rozbaľovacieho menu a uloží túto hodnotu do premennej „proces“.

```

cla(handles.processedimage);
original = handles.orig_im;

```

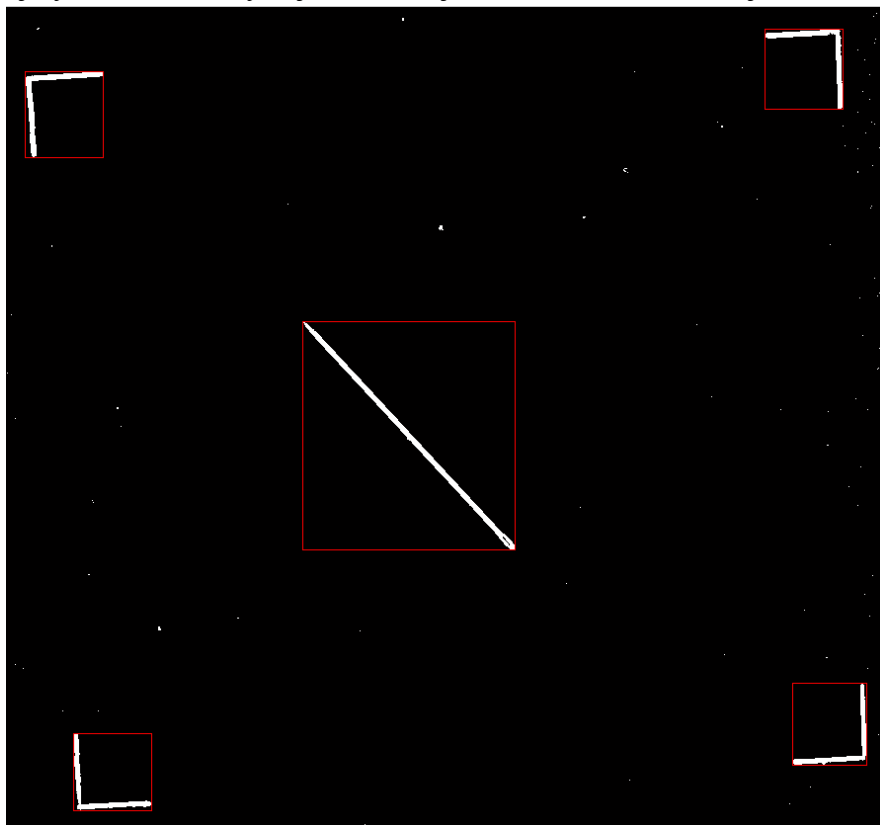
Prvý príkazový riadok slúži na to, aby osi pre grafické objekty vyprázdnil. Toto je potrebné v prípade, keď používateľ už vyhodnotil jeden obrázok a chce vyhodnotiť ďalší obraz. Aby konečné obrazy neboli vo vzájomnom rozpore je potrebné osi vyprázdniť.

Druhý príkazový riadok získa obraz z osi s menom „originalimage“, kde je načítaný pôvodný obraz a uloží do premennej original ako maticu.

Môže sa stať, že matrica alebo fotoaparát nebude presne v symetrii a preto by nebolo vyhodnotenie dostatočne presné. Tento problém riešim tak, že načítaný obrázok otočím vodorovne k ose x pomocou kalibračných znakov. Pomocou funkcie „regionprops“ vyhl'adám všetky objekty na obrázku podľa veľkosti plochy objektu, najmenšieho štvorca okolo objektu a extrémnych bodov.

```
stats=regionprops(BW,'area','BoundingBox','Extrema');
```

Do premennej „stats“ uloží tieto hodnoty. Pomocou „stats.Area“ získam veľkosti objektu podľa pixelu a uloží do premennej „Areas“ ako maticu. V premennej „Areas“ sú uložené veľkosti objektu do jedného riadku, kde stĺpec znamená index daného objektu. Ďalej vyhl'adám indexy najväčších objektov. Pre znázornenie je uvedené na obrázku 4-13.



Obrázok 4-13 Najväčšie objekty

Okolo najväčších objektov som vykreslil štvorec pre lepšie znázornenie.

Ďalej som vybral ľavý horný a pravý horný kalibračný znak a podľa pozície týchto znakov som otočil obrázok vodorovne k ose x. Keď sú dva kalibračné znaky na jednej vodorovnej úrovni, vtedy môžeme namerať vzdialenosť medzi koncovými bodmi pravého a ľavého kalibračného znaku v pixeloch. Premennú „pompixnamm“ získam ako

podiel nameranej vzdialenosti a sedemdesiatimi. Premenná „pompixnamm“ bude vyjadrovať pomer pixele na milimeter.

4.6.5 Funkcia „search2“

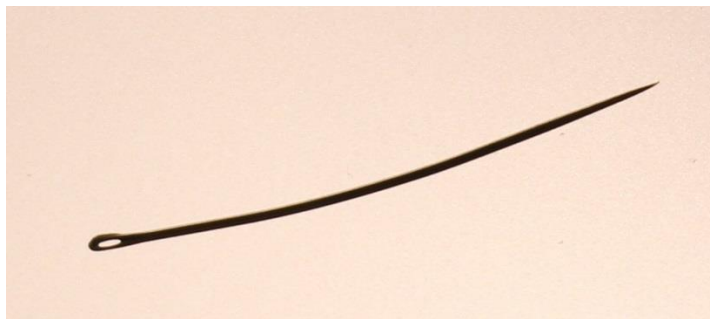
Funkcia search2 vyhľadá z čiernobieleho obrazu krajné body ihly. Vrátené hodnoty sú koordináty krajných bodov. Prvé dve hodnoty sú hodnoty pre krajný bod na ľavej strane a druhé dve hodnoty sú koordináty krajného bodu na pravej strane.

V search2 som použil funkciu „size“, ktorý mi vráti parametre vstupného obrazu. Do premennej „rows“ uloží počet riadkov a do premennej „cols“ uloží počet stĺpcov obrazu.

Opakovane prejde cyklus po riadkoch každého stĺpca. Do premennej x a y vloží koordináty krajného bodu vyhovujúce podmienkam pre ľavú časť ihly. Pre krajný bod na druhej strane je princíp podobný.

4.6.6 Riešenie chyby: ohnutá ihla

Chyba: ohnutá ihla znamená, že telo ihly je ohnuté vzhľadom ku svojej osi. Os ihly vznikne spojením dvoch bodov a to hrotu ihly a ucha ihly. Ohnutie ihly môže nastať buď na jednu stranu od osi ihly alebo na druhú stranu. Ohyb od smeru osi sa pohybuje v určitých toleranciách. Tieto hodnoty som nastavil podľa vlastných skúseností. Príkladom ohnutia reprezentuje obrázok 4-14.



Obrázok 4-14 Ohnutá ihla

Chybu rieši funkcia „search_ohnutie“. Výstupom je hodnota, ktorá informuje o výsledku a ten môže predstavovať tri stavy:

1. ihla nie je ohnutá
2. ihla je ohnutá
3. na obrázku nie je iba jedna ihla.

Štandardný stav "ihla nie je ohnutá" znamená, že vzhľadom k svojej osi ihla nevybočuje z tolerancie ohnutia.

Stav "ihla je ohnutá" znamená, že ihla vzhľadom k svojej osi vybočuje z tolerancie ohnutia a tým pádom je nevyhovujúca.

Posledný výstupný stav "na obrázku nie je iba jedna ihla" sa vyskytuje zriedkavo, lebo je veľmi nepravdepodobné, že sa na matnicu dostane viac ihli súčasne alebo je na nej nejaká nečistota. Ohnutie ihly som testoval dvoma spôsobmi.

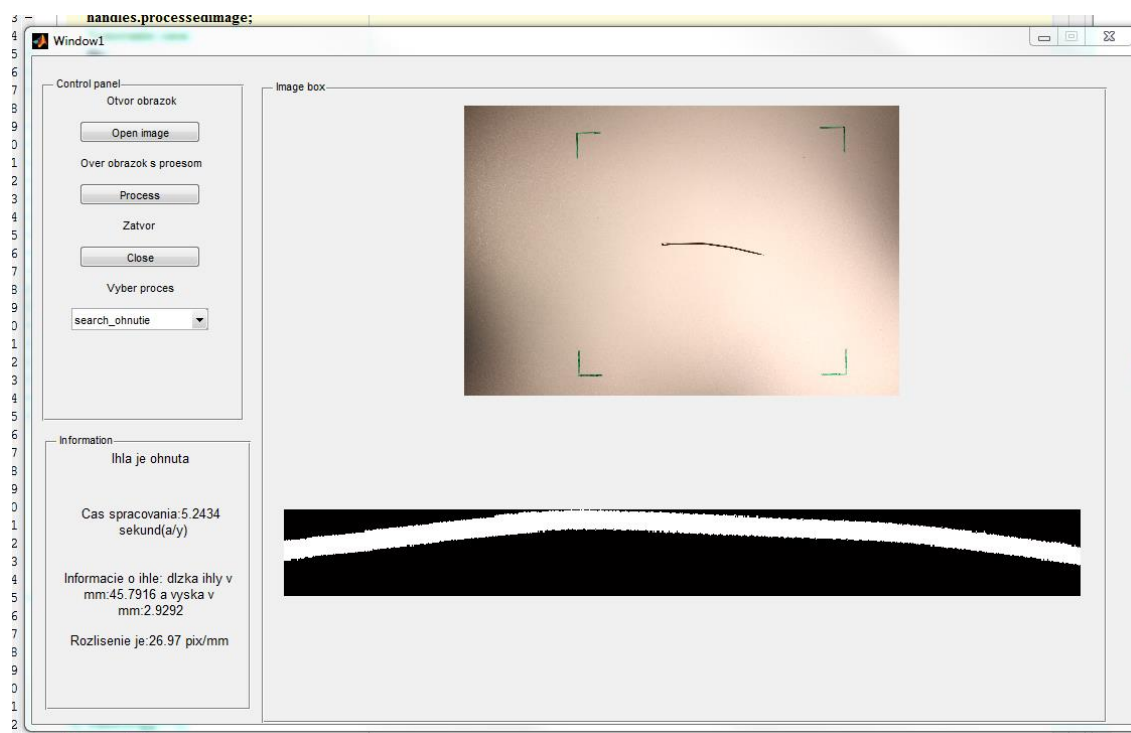
Prvým spôsobom bolo vyhľadanie krajných bodov ihly pomocou funkciou „search2“. Krajné body ihly predstavujú ucho a hrot ihly, ktoré je potrebné spojiť priamkou. Túto priamku som nazval „strednalínia“. V ďalšom postupe je možné vynechať zo skúmania hrot a ucho ihly a skúmať iba jej telo. Algoritmus prešiel cyklicky celý obraz zľava doprava a zhora nadol. Spočítal som všetky body obrazu nad a pod priamkou „strednalínia“. V prípade, keď je ihla rovná, tak sa počty pixelov relatívne zhodujú nad a pod priamkou a algoritmus vyhodnotí ihlu ako vhodnú. V opačnom prípade, keď je ihla ohnutá, tak sa jeden počet líši od druhého a algoritmus vyhodnotí ihly ako nevhodnú.



Obrázok 4-15 Stredná línia ihly

Na obrázku 4-15 je znázornená rovná ihla počas spracovania obrazu. Priamka od ucha k hrotu ihly je reprezentovaná modrou čiarou v strede ihly.

Druhý spôsob testovania chyby som použil v aplikácii. Tento spôsob spočíva v tom, že vyhľadám horný okraj tela ihly a uložíam do matice „obluk“. Po zderivovaní matice „obluk“ som dostal maticu, ktorá opisuje zmeny horného okraja ihly v ose y podľa osi x. Túto maticu som nazval „prvader“. V prípade, keď ihla nie je ohnutá, tak v oboch polovičkách matice „prvader“ je súčet hodnôt približujúci sa k nule. V tomto prípade algoritmus vyhodnotí ihlu ako vyhovujúcu. V opačnom prípade sa súčet hodnôt v ľavej polovici matice „prvader“ líši od nuly. V pravej polovici tejto matice je súčet hodnôt s opačným znamienkom. Znamienka súčtov sa nezhodovali kvôli tomu, že hranica ihly na ľavej strane ihly smeruje hore a na pravej strane smeruje dole alebo opačne.

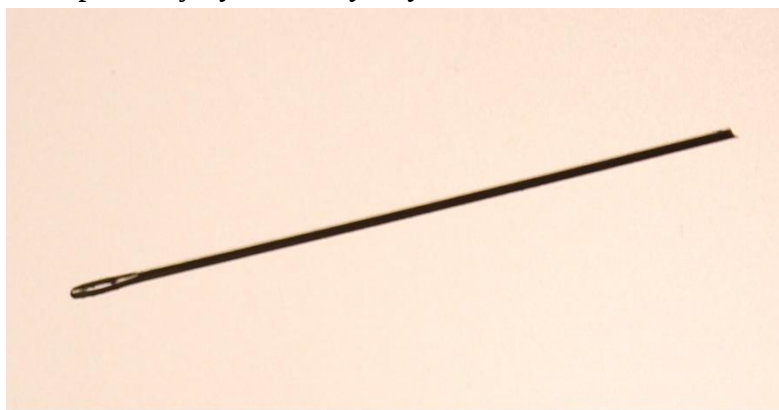


Obrázok 4-16 Aplikácia po vyhodnotení chyby: ohnutá ihla

Na obrázku 4-16 je znázornená aplikácia po vyhodnotení chyby „ohnutie ihly“. Čas spracovania bol 5,24 sekúnd, dĺžka ihly bola 45,79 mm a rozlíšenie je 26,97 pixelov na milimeter.

4.6.7 Riešenie chyby: opotrebovaný hrot

Chyba: opotrebovaný hrot ihly znamená, že hrot ihly je otupený alebo zaoblený a ihla je nevhodná na používanie. Hrot ihly je ostrý, aby mohol preniknúť materiálom. Pri používaní ihly je hrot ihly vystavený trecej sile a to môže spôsobiť opotrebovanie hrotu ihly. V iných prípadoch môže chyba nastať pri automatizovanej výrobe, keď prístroj nevhodne vybrúsi hrot ihly. Túto chybu je treba detekovať pri vizuálnej kontrole, aby sa defektné produkty vyradili z výroby.



Obrázok 4-17 Opotrebovaný hrot

Algoritmus som napísal do funkcie menom „search_hrot“. Vstupným parametrom je čiernobiely obraz skúmanej ihly. Výstupným parametrom je premenná s hodnotou stavu vyhodnotenia. Výstupná hodnota môže mať dva stavy a to:

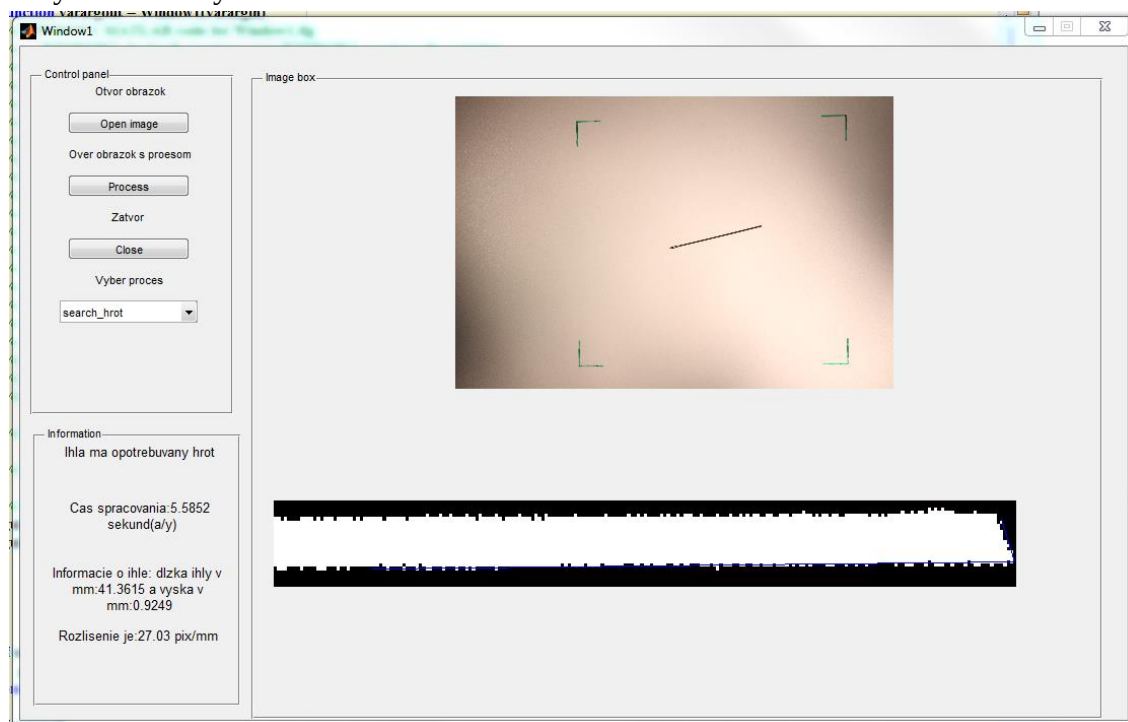
1. ihla má opotrebovaný hrot - to znamená, že ihla je nevyhovujúca
2. ihla má dobrý hrot - znamená, že je ihla vyhovujúca.

Výstupná hodnota je znázornená v hornej časti informačného panela.

Vychádzajúc zo vstupného obrazu ihly, ktorá je v štandardnej polohe, je potrebné preskúmať jej pravú časť s hrotom na konci. Vyhľadám priamku k hornej a dolnej hranici tela ihly. Ďalej je potrebné nájsť body, kde sa telo ihly a hrot ihly stretávajú. Jeden z týchto bodov sa nachádza na hornej hranici tela ihly a druhý bod na dolnej hranici tela ihly. Prepojením koncového bodu hrotu ihly s dvoma bodmi, kde sa hrot pripája na telo ihly, tak vytvoríme dve priamky, ktoré sa spolu stretávajú v koncovom bode hrotu ihly. Tieto priamky nazývam „rampa hrotu“. Uhol medzi rampou hrotu a hranicou tela ihly vyjadruje, či ihla má opotrebovaný hrot, alebo má vyhovujúci hrot. Nenašiel som dostupnú dokumentáciu takého kritéria vizuálnej kontroly ihly, preto som toto kritérium nastavil podľa vlastných skúseností.

Hranice ihly som stanovil z viacerých meraní spriemerovaním hodnôt. Bod, kde sa stretne hranica tela ihly s rampou je určený ďalšou časťou algoritmu. Hornú a dolnú hranicu ihly ošetrujem zvlášť cyklicky z ľavej strany na pravú stranu. Hrot ihly sa pripája

k telu ihly v bode, kde priemerná jasová hodnota okolia klesne pod kritérium jasovej hodnoty hranice ihly.

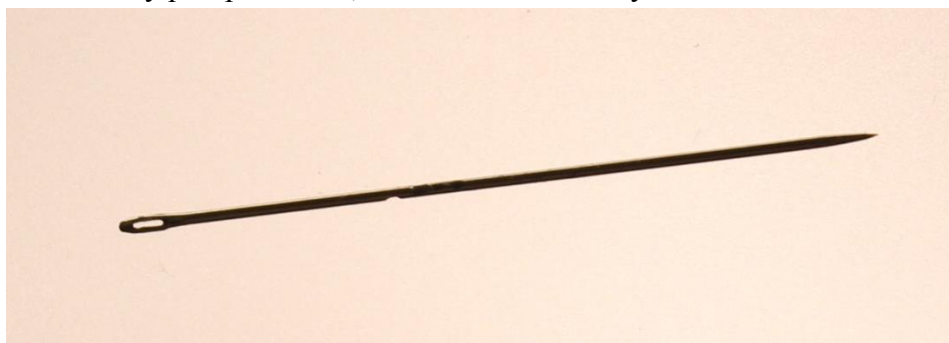


Obrázok 4-18 Aplikácia po vyhodnotení chyby: opotrebovaný hrot

Na obrázku 4-18 je reprezentovaná aplikácia po vyhodnotení ihly s opotrebovaným hrotom. Vstupný obrázok je v hornej časti „Image Box“ panela a posledný použitý obraz je vykreslený v dolnej časti panela. Vyhodnotenie tejto chyby trvalo viac ako 5,59 sekúnd.

4.6.8 Riešenie chyby: diera v tele ihly

Chyba: diera v tele ihly znamená, že vo valcovitom tvare ihly je nerovnosť. Táto chyba sa dá rozpoznať tak, že horná a dolná hranica tela ihly nie je symetrická. Táto chyba ihly môže spôsobiť poškodenie textilu pri použití ihly počas šitia. Diera v tele ihly nemusí presahovať do celej hĺbky ihly a preto nie je vždy spozorovateľná voľným okom. Pri práci s obrazom predpokladám, že ihla je vyhovujúca alebo má jednu chybu. Ak riešim chybu diera v tele ihly predpokladám, že ihla nemá iné vady.



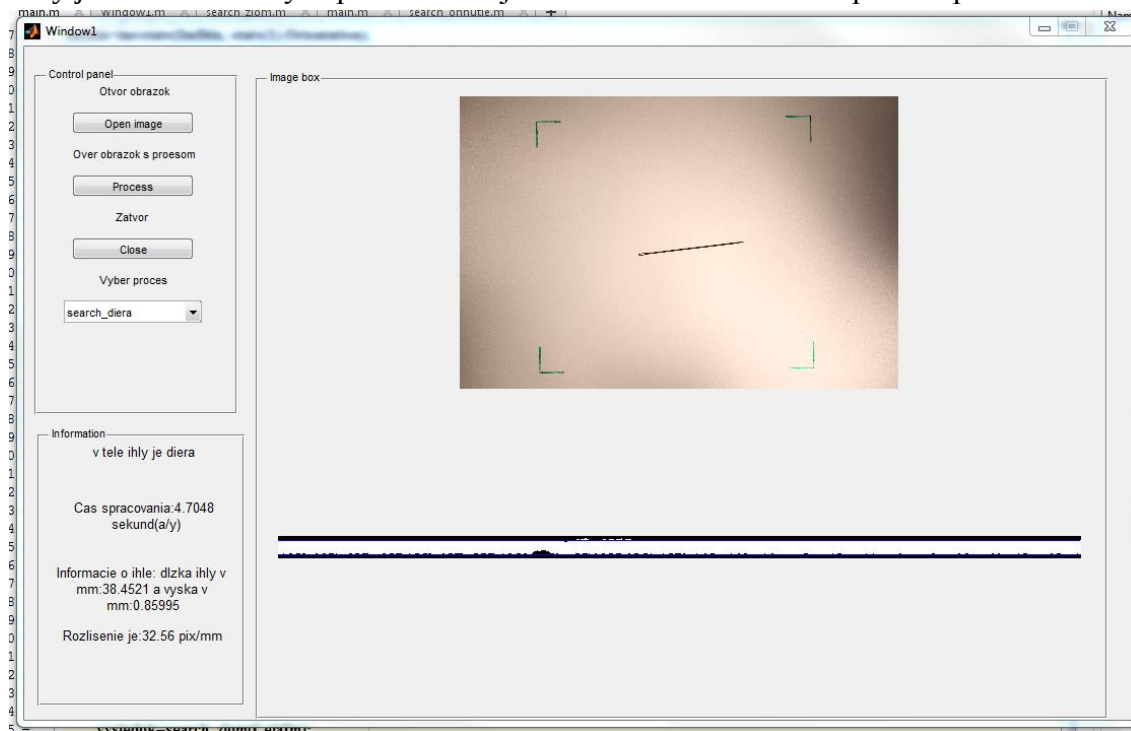
Obrázok 4-19 Diera v tele ihly

Vychádzal som zo skutočnosti, že hlavnou časťou ihly je jej telo, a preto je dôležité kontrolovať symetriu ihly. Diera na čiernobiely obraz je rozpoznateľná tak, že kontinuita hranice ihly je narušená.

Do funkcie „search_diera“ som vytvoril algoritmus, ktorý vyhľadá túto chybu v tele ihly. Vstupným parametrom funkcie je čiernobiely obraz ihly. Skúmaná časť ihly je telo, preto v tejto funkcii oddelím hrot a ucho ihly pomocou funkcie „imcrop“ a tak získam telo ihly. Na výslednom obraze vyhľadám hornú a dolnú hranicu ihly. V každom stĺpci sú dve zmeny jasovej úrovne smerom zhora nadol. Prvá zmena v jasovej úrovni je horná hranica ihly a druhá zmena je dolná hranica ihly. V prípade keď je ihla rovná a bez chyby „diera v tele“, tak horná hranica a dolná hranica je kontinuálna a neprerušená. V opačnom prípade, keď v tele ihly je diera, tak dolná alebo horná hranica ihly nie je súvislá.

V programe cyklicky ošetrím symetriu okolia každého bodu na hranici smerom zľava doprava. V ošetrovanom bode skúmam okolie bodu, tak získam okno jasových hodnôt. V prípade, keď ihla má rovnú hranicu tela ihly, tak po hornej hranici ihly má toto okno v hornej polovici nulové jasové hodnoty a v dolnej polovici má jasové hodnoty rovné jasovej hodnote 1. Keď ihla má v tele dieru, tak skúmané okolie bodu má všetky hodnoty rovné nule alebo má len veľmi málo hodnôt rovné 1. Týmto spôsobom sa dá vyhodnotiť, či má ihla v tele dieru.

Výstupná premenná môže mať hodnotu buď „Nie je diera v tele“ alebo hodnotu „v tele ihly je diera“. Táto výstupná hodnota je uvedená v informačnom paneli aplikácie.

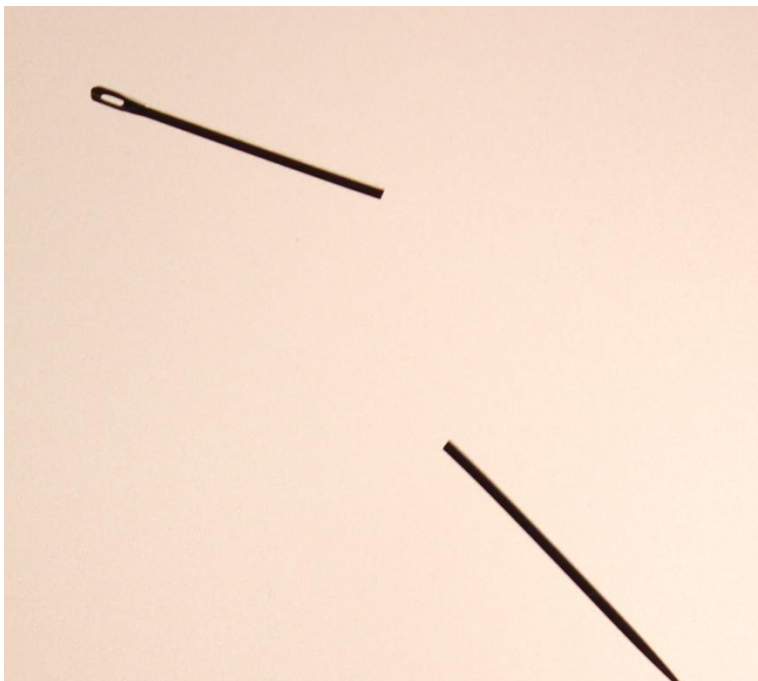


Obrázok 4-20 Aplikácia po vyhodnotení chyby: diera v tele ihly

Na obrázku 4-20 je aplikácia po vyhodnotení chyby „diera v tele ihly“. V dolnej časti panela „Image Box“ je posledný použitý obrázok počas algoritmu. S modrými priamkami je znázornená horná a dolná hranica ihly.

4.6.9 Riešenie chyby: zlomená ihla

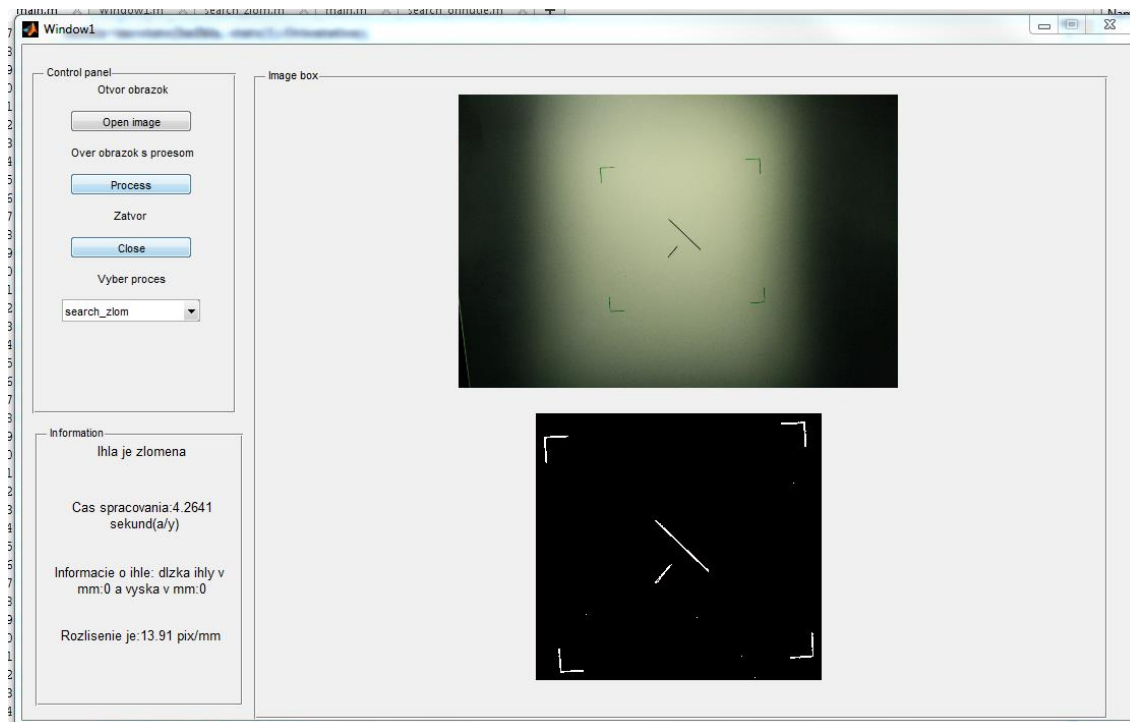
Chyba: zlom v tele ihly spočíva v tom, že ihla je zlomená na dve alebo viac častí. Pri fyzickej námahe je možné, že sa ihla zlomí a tak sa stane nepoužiteľnou. Najčastejšie sa tak môže stať napr. pri použití nesprávnej veľkosti ihly pri šití hrubej tkaniny. Vytvorený algoritmus vyhľadá zlom zo vstupného čiernobieleho obrazu. Výstupným parametrom je hodnota, ktorá reprezentuje stav ihly, či je zlomená alebo je celistvá.



Obrázok 4-21 Zlomená ihla

Chybovosť ihly rieši algoritmus prostredníctvom funkcie „search_zlom“. Predpokladám, že na vstupnom obraze je ihla zlomená a má svoje diely na obraze. Algoritmus vyhľadá všetky časti ihly na obrázku a podľa počtu dielov rozhodne, či je ihla celistvá alebo je zlomená.

Výstupná hodnota môže mať stavy: 'Ihla je zlomena' a 'Ihla nie je zlomena'. Táto hodnota je vypísaná v aplikácii informačného panela.



Obrázok 4-22 Aplikácia po vyhodnotení chyby: zlomená ihla

V prípade, keď je ihla celistvá, ale na matnici sa nachádza nečistota podobná ihle, tak program vyhodnotí ihlu ako nevhodnú. V prípade že algoritmus vyhodnotí ihlu ako zlomenú, hodnoty dĺžky a výšky ihly sú nulové.

4.7 Meranie dĺžky ihly v programe

Tabuľka 1 Namerané hodnoty s programom

	1000D		100D		P10	
Xr[mm]	Xm [mm]	t [s]	Xm [mm]	t [s]	Xm [mm]	t [s]
57,00	57,17	18,14	57,28	4,62	57,47	5,47
48,00	50,00	4,92	48,33	4,70	48,80	5,33
42,00	41,99	16,91	42,26	3,75	42,64	6,16
42,00	42,20	2,71	42,01	5,74	42,32	5,82
42,00	42,38	16,30	42,33	4,82	42,70	5,50
39,00	39,11	21,96	39,12	4,30	39,36	5,96
35,00	34,80	7,99	34,83	4,53	35,33	5,60
32,00	32,00	4,84	32,23	5,13	30,88	5,61

	Δ 1000D		Δ 100D		Δ P10	
Xr[mm]	Δ [mm]	δ [%]	Δ [mm]	δ [%]	Δ [mm]	δ [%]
57,00	0,17	0,30	0,28	0,49	0,47	0,82
48,00	2,00	4,00	0,33	0,68	0,80	1,64
42,00	0,01	0,02	0,26	0,62	0,64	1,50
42,00	0,20	0,47	0,01	0,02	0,32	0,76
42,00	0,38	0,90	0,33	0,78	0,70	1,64
39,00	0,11	0,28	0,12	0,31	0,36	0,91
35,00	0,20	0,57	0,17	0,49	0,33	0,93
32,00	0,00	0,00	0,23	0,71	1,12	3,63

Priemerná relatívna chyba s 1000D v [%]

0,82

Priemerná relatívna chyba s 100D v [%]

0,51

Priemerná relatívna chyba s P10 v [%]

1,48

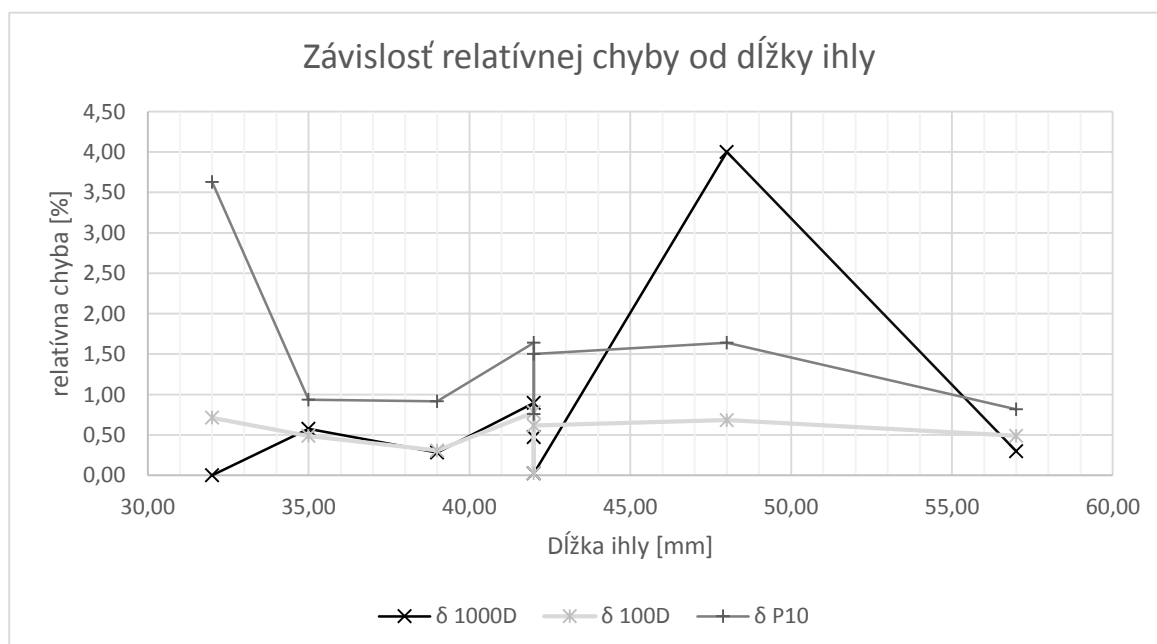


Figure 1 Diagram 1 závislosť relatívnej chyby od dĺžky ihly

S programom som vyhodnotil osem vyhovujúcich ihli s tromi rôznymi snímacími zariadeniami a namerané hodnoty som zadal do tabuľky č. 1. Za správnu hodnotu dĺžky ihly som považoval hodnotu nameranú pravítkom. Vypočítal som relatívnu chybu každej ihly. V grafe som zobrazil závislosť relatívnej chyby od dĺžky ihly. Snímky snímané kamerou Toshiba Camileo P10 mali priemernú relatívnu chybu 1,48%, snímky s kamerou Canon EOS 1000D mali priemernú relatívnu chybu 0,82% a snímky s kamerou Canon EOS 100D mali priemernú relatívnu chybu 0,51%. Ako najlepšie snímacie zariadenie sa osvedčila kamera Canon EOS 100D s rozlíšením obrazu 18,5 M pixelov.

5 ZÁVER

Na začiatku práce sú opísané základné poznatky k spracovaniu obrazu a počítačového videnia. Tieto poznatky ujasňujú definície a pojmy použité v spomínaných témach. Ďalej sú opísané metódy na spracovanie obrazu a následné porovnanie metód. Jednotlivé metódy sú znázornené na príkladoch, ktoré sú priložené v kapitolách danej metódy.

V kapitole „Metódy a postupy riešenia algoritmov“ je rozoberaná problematika snímania obrazu. Podľa môjho názoru je najvhodnejšou metódou pre túto úlohu snímanie podsvieteným osvetlením. Pre túto metódu som vyhotovil pracovisko vhodné pre snímanie obrazu uvedenou metódou. Kovový stojan som zvolil z dôvodu jeho stability. Aby som získal čo najlepšie snímky pre ďalšie spracovanie obrazu, bolo nevyhnutné, aby bola kamera pevne prikotvená k vrchnej časti stojana

V kapitole „Riešenie stanovených chýb“ som opísal funkcie a algoritmy, ktoré som vytvoril v programovacom prostredí Matlab. Aby bolo možné spustiť aplikáciu je potrebné mať k dispozícii program Matlab a pri programovaní som používal jeho verziu R2013b. Funkcie vyhodnotia kvalitu snímaných súčiastok. Každá funkcia rieši chybu určitého typu (napr. funkcia „search_ohnutie“ vyhodnotí ohnutie ihly). Chyby som stanovil v kapitole 4.1. „Rozbor úlohy a stanovenie chýb“

Funkcia „search2“ je dôležitá pri usporiadaní obrazu ihly do štandardnej polohy predtým ako dôjde k predaniu obrazu ďalším funkciám. Táto funkcia vyhladá krajné body ihly, podľa ktorých obraz usporiada ihly do štandardnej polohy. Na vyhotovenie vhodného pracoviska na meranie súčiastok som potreboval návrhnúť konštrukciu samotného stojana a manuálnu zručnosť pri jeho zvaraní. Pri vyhotovení konštrukcie stojana som získal nové skúsenosti z oblasti kovoobrábania. Počas písania programu som získal nové poznatky pri riešení praktických problémov programovania v jazyku Matlab.

5.1 Súčasný stav

Vyhotovené je pracovisko pre snímanie ihliel. Položením ihly na matnicu na stojane sú umiestnené ihly do zorného pola kamery. Na stojane je možné nastaviť viacero polôh pre snímanie kamerou, v ktorých je kamera v stabilnom bode. Po nasnímaní obrazu je treba snímok preniesť do počítača, do priečinku zdrojového kódu. Do Matlabovského prostredia je načítaný obraz pomocou funkcie „imread“, kde vstupnými parametrami je názov a typ snímku.

Pri snímaní sa predpokladá ideálny stav, keď ihla leží takým spôsobom, aby dierka bola vidno a bola na ľavej strane obrázku.

Je nasnímaná databáza vyhovujúcich a nevyhovujúcich súčiastok. V databáze sú snímky ihliel s rôznymi vadami. Jedna vadná ihla obsahuje len jednu chybu.

V súčasnom stave práca obsahuje aplikáciu, ktorá vyhodnotí kvalitu súčiastky podľa

zadaných kritérií. Parametre a tolerancie sú nastaviteľné v skripte. Predspracovanie obrazu adaptívnym prahovaním je podľa môjho názoru pre tieto účely postačujúce. Nenašiel som dostupnú dokumentáciu ohľadne kontroly ihliel alebo možných chýbách pri výrobe ihly. Typy chýb som vydedukoval z vlastných skúseností pri používaní ihly a zo skúsenosti profesionálnych krajčírov.

5.2 Návrh ďalšieho postupu

Pri ďalšom postupe plánujem rozšíriť algoritmy a databázu snímok pre ďalšie typy ihliel ako napr.: ihly do šijacích strojov, ihly používané v zdravotníctve, obuvníctve, brašniarstve a iné. Ďalšou možnosťou rozšírenia programu je vytvorenie nových algoritmov, ktoré riešia iné chyby ako napríklad: nesprávne poniklovanie ihly alebo chybné ucho ihly. Na skúmanie správnosti poniklovania ihly by mohla slúžiť metóda - snímania zrkadlovým osvetlením.

V budúcnosti vyskúšam pri tvorbe čiernobieleho vstupného obrazu pre funkcie ďalšie metódy hranových detektorov ako napr. Laplaceov operátor. Ďalšou možnosťou vylepšenia algoritmu je optimalizácia rotácie ihly do štandardnej polohy.

Keďže nič nie je úplne dokonalé, aj tento program má svoje nedostatky. Preto plánujem rozšíriť a ochrániť súčasné funkcie a algoritmy od ostatných chybových možností pri snímaní. Po optimalizácii by program mal vyfiltrovať obrázok od rušivých objektov a nečistôt, ktoré sa môžu dostať na matnicu počas snímania.

Funkcie som použil a napísal v Matlabe. Na tento účel je možné použiť aj iné programovacie jazyky ako napr.: jazyk C alebo C++. V budúcnosti chcem rozšíriť môj program ako samostatnú spustiteľnú aplikáciu písanú v programovacom jazyku C. Následne by sa program dal využiť pre real-time operáciu. Mám v umysle vyhotoviť maketu pásovej výroby, ktorá by mala simulovať deje pri výrobe a selektovaní súčiastok.

Literatura

- [1] *Pocitacova grafika* [online]. *Reprezentacia obrazu* [online]., aktualizácia: 2014 [cit. 2015-5-22]. Dostupné na internete:
<http://www.martintimko.net/pocitacova-grafika/reprezentacia-obrazu/>
- [2] *Digital Image Processing* [online]. Posledná aktualizácia: 2006 [cit. 2013-12-31]. Dostupné na internete:
<http://dip.sccg.sk/>
- [3] *Multimediální interaktivní didaktický systém* [online]. Posledná aktualizácia: 2010 [cit. 2013-12-30]. Dostupné na internete:
http://midas.uamt.feec.vutbr.cz/ZVS/lectures-pdf/09_Segmentace_obrazu.pdf
- [4] JIŘÍ, ŽÁRA A JIŘÍ ŽÁRA. MODERNÍ POČÍTAČOVÁ GRAFIKA. VYD 1. BRNO: COMPUTER PRESS, 2004, 609 s. ISBN 80-251-0454-0.
- [5] *Wikipedia: Laplaceov operátor* [online]. Posledná aktualizácia: 13:18 10.3.2013 [cit. 2014-1-3]. Dostupné na internete:
http://sk.wikipedia.org/wiki/Laplaceov_oper%C3%A1tor
- [6] *mathworks* [online]. *regionprops* aktualizácia: 2014 [cit. 2015-5-22]. Dostupné na internete:
<http://www.mathworks.com/help/images/ref/regionprops.html>
- [7] *dip sccg* [online]. aktualizácia: 2014 [cit. 2015-5-22]. Dostupné na internete:
<http://dip.sccg.sk/predspra/predspra.htm>
- [8] *MathWorks MatLab* [online]. Posledná aktualizácia: 2015 [cit. 2014-1-2]. Dostupné na internete:
<http://www.mathworks.com/products/matlab/index.html>
- [9] *Digital Image Processing* [online]. Posledná aktualizácia: 2006 [cit. 2013-12-31]. Dostupné na internete:
<http://dip.sccg.sk/>
- [10] *Wikipedia: Zorné pole* [online]. Posledná aktualizácia: 22:25 13. 7. 2013 [cit. 2013-12-30]. Dostupné na internete:
http://cs.wikipedia.org/wiki/Zorn%C3%A9_pole
- [11] PARKER, J. ALGORITHMS FOR IMAGE PROCESSING AND COMPUTER VISION. NEW YORK: WILEY COMPUTER PUBLISHING, 1997, XIII, 417 S. ISBN 04-711-4056-2.
- [12] *vutbr* [online]. *Uvod_do_pocitacoveho_videni* aktualizácia: 2014 [cit. 2015-5-22]. Dostupné na internete:
http://midas.uamt.feec.vutbr.cz/POV/lectures-pdf/01_Uvod_do_pocitacoveho_videni.pdf
- [13] *Informačný systém Masarykovy Univerzity* [online]. Posledná aktualizácia: 25. 2. 2012 [cit. 2013-12-30]. Dostupné na internete:
http://is.muni.cz/th/387035/fi_m/Diplomova_praca.pdf
- [14] *Fashion-history* [online]. *history-needles-sewing* aktualizácia: 2014 [cit. 2015-5-22]. Dostupné na internete:
<http://fashion-history.lovetoknow.com/fashion-history-eras/history-needles-sewing>
- [15] *canon-europe* [online]. *eos_1000d* aktualizácia: 2015 [cit. 2015-5-22]. Dostupné na internete:
http://www.canon-europe.com/for_home/product_finder/cameras/digital_slr/

[16] *toshiba* [online]. aktualizácia: 2015 [cit. 2015-5-22]. Dostupné na internete:
<http://www.toshiba.eu/>

Seznam příloh

Příloha 2. Zdrojové texty a obrázky

Příloha 3. CD/DVD