

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

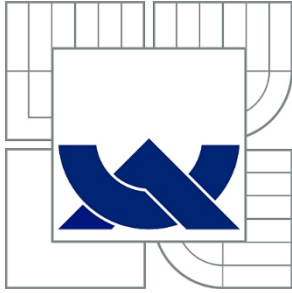
VIZUÁLNÍ KONTROLA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

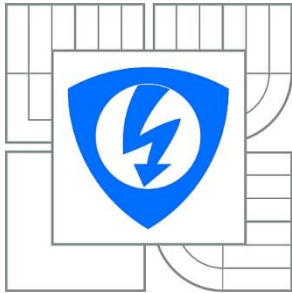
MICHAL LANCZ

BRNO 2014



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 2014



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

ID: 146885

Ročník: 3

Akademický rok: 2013/2014

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) Proveďte rozbor úlohy. Stanovte parametry a tolerance pro hodnocení kvality součástky včetně přesnosti měření.
- 2) Navrhněte vhodně 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: 10.2.2014

Termín odevzdání: 26.5.2014

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.

Bibliografická citace práce:

LANCZ, M. VIZUÁLNÍ KONTROLA. BRNO: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ, 2014. 34 s. VEDOUCÍ BAKALÁŘSKÉ PRÁCE, ING. MILOSLAV RICHTER, PH.D..

Jako autor uvedené diplomové (bakalářské) práce dále prohlašuji, že v souvislosti s vytvořením této diplomové (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.

POĎAKOVANIE

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

V Brne dne:.....

podpis autora:.....

ABSTRAKT

Táto bakalárská 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.

KEÚČOVÉ 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.

KEY WORDS: Field of view, focal length, histogram, thresholding, image noise, Matlab

OBSAH

ZOZNAM OBRÁZKOV.....	10
ZOZNAM TABULIEK.....	11
ZOZNAM SYMBOLOV A SKRATIEK.....	12
1 ÚVOD.....	13
2 ZÁKLADNÉ POZNATKY K SPRACOVANIU OBRAZU.....	14
2.1 REPREZENTÁCIA OBRAZOVÝCH DÁT.....	14
2.2 DETEKCIA HRÁN V OBRAZE.....	15
2.3 ŠUM V OBRAZE.....	15
2.4 ZORNÉ POLE A OHNISKOVÁ VZDIALENOSŤ.....	16
2.5 HISTOGRAM OBRAZU.....	16
2.6 MATLAB.....	17
2.7 KONVOLÚCIA.....	17
3 CIEĽ PRÁCE.....	18
4 METÓDY A POSTUPY RIEŠENIA ALGORITMOV.....	18
4.1 ROZBOR ÚLOHY A STANOVENIE CHÝB.....	18
4.2 VYHLADZOVACIE FILTRE.....	19
4.2.1 PRIEMEROVACIE FILTRE.....	19
4.2.2 GAUSSOV FILTER.....	19
4.3 HRANOVÉ FILTRE.....	19
4.3.1 GLOBÁLNE PRAHOVANIE.....	19
4.3.2 ADAPTÍVNE PRAHOVANIE.....	20
4.3.3 ROBERTSOV OPERÁTOR.....	21
4.3.4 SOBELOV OPERÁTOR.....	22
4.3.5 PREWITTIV OPERÁTOR.....	22
4.3.6 LAPLACEOV OPERÁTOR.....	23
4.3.7 CANNYHO DETEKTOR.....	24
4.4 NÁVRH PRACOVISKA NA SNÍMANIE OBJEKTU.....	25
4.4.1 ZRKADLOVÉ OSVETLENIE.....	25
4.4.2 ROZPTÝLENÉ OSVETLENIE.....	25
4.4.3 ZADNÉ OSVETLENIE.....	26
4.4.4 SNÍMANIE SKENEROM.....	26
4.4.5 VYHOTOVENIE PRACOVISKA NA MERANIE IHIEL.....	27
4.5 RIEŠENIA STANOVENÝCH CHÝB.....	28
4.5.1 FUNKCIA SEARCH2.....	29
4.5.2 RIEŠENIE CHYBY: ZLOM V TELE.....	29
4.5.3 RIEŠENIE CHYBY: OHNUTÁ IHLA.....	30
4.5.4 RIEŠENIE CHYBY: CHÝBAJÚCE UCHO IHLY.....	31
4.5.5 RIEŠENIE CHYBY: OPOTREBOVANÝ HROT IHLY.....	32
4.5.6 RIEŠENIE CHYBY: DIERA V TELE IHLY.....	32
4.5.7 RIEŠENIE CHYBY: NESÚHLAS V PARAMETROCH IHLY.....	33

5 ZÁVER	33
5.1 SÚČASNÝ STAV	34
5.2 NÁVRH ĎALŠIEHO POSTUPU	34
POUŽITÁ LITERATÚRA	35

ZOZNAM OBRÁZKOV

<i>Obr. 2-1 Ideálna hrana [2] prevzatá a prerobená.....</i>	<i>15</i>
<i>Obr. 2-2 Histogram obrazu.....</i>	<i>17</i>
<i>Obr. 4-2 Priemerovacie masky.....</i>	<i>19</i>
<i>Obr. 4-4 Prahovanie jedným prahom.....</i>	<i>20</i>
<i>Obr. 4-5 Princíp Robertsovho operátora [9].....</i>	<i>21</i>
<i>Obr. 4-6 Porovnanie Sobelovho a Prewittovho operátora</i>	<i>23</i>
<i>Obr. 4-6 Zdroje svetla [1].....</i>	<i>26</i>
<i>Obr. 4-7 Snímka skenerom</i>	<i>27</i>
<i>Obr. 4-8 Konštrukcia.....</i>	<i>27</i>
<i>Obr. 4-9 Popis ihly.....</i>	<i>28</i>
<i>Obr. 4-10 Výpočet uhlu</i>	<i>29</i>
<i>Obr. 4-10 Rozdelenie častí zlomenej ihly.....</i>	<i>30</i>
<i>Obr. 4-11 Ukážka osi ihly po algoritme</i>	<i>31</i>
<i>Obr. 4-12 Ukážka osi ihly po algoritme s ohnutím</i>	<i>31</i>
<i>Obr. 4-13 Ukážka hrotu po algoritmu.....</i>	<i>32</i>
<i>Obr. 4-14 Ukážka vyhovujúcej ihly po algoritme.....</i>	<i>33</i>

ZOZNAM TABULIEK

Tab. 2-1 Zoznam dôvodov nerozpoznatel'nosti krokovej hrany.....15

Tab. 4-1 Zoznam Cennyho pravidiel24

ZOZNAM SYMBOLOV A SKRATIEK

a.i. – a iné

FOV – Field Of View

VUT FEKT – Vysoké Učení Technické v Brně – Fakulta elektrotechniky a komunikačních
technológií

tzv. – takzvaný

t.j. – to jest

atď. – a tak ďalej

resp. – respektíve

1 ÚVOD

Na snímanie objektov slúžia viaceré obory ako sú: spracovanie obrazu, počítačové videnie, analýza scén alebo analýza digitálneho obrazu a.i.. Tieto obory slúžia vlastne preto, podobne ako ľudské oko a mozog, aby nasnímali obrázok a potom ho vyhodnotili. Snímanie objektov sa využíva v oblastiach, kde je nutná dvoj- alebo viacdimeznová spätná väzba obrazu. Často sa nachádza pri rutinných kontrolách automatizovanej alebo poloautomatizovanej linkovej výroby. Robot vykonávajúci samostatné úlohy, ktorý sa pohybuje v priestore, sa 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 popríklad detekovanie typu vady na súčiastke (ak sa jedná o kategóriu chybné súčiastky).

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.

2 ZÁKLADNÉ POZNATKY K SPRACOVANIU OBRAZU

2.1 Reprezentácia obrazových dát

Ako reprezentujeme obraz ?

Na túto otázku sa dá z viacerých zdrojov nájsť viac definícií. Jedna z najpoužívanejších definícií je:

„Obraz (obrazová data) lze reprezentovat jako spojité rozložení jasu (intenzity, optické hustoty) v rovině. Rozložení hodnot jasu je definováno obrazovou funkcí: ” [1]

$$f(x, y) \tag{2.1}$$

$$f(x, y, t) \tag{2.2}$$

Kde f je reálna, spojitá funkcia, ktorá obsahuje argumenty reálne. Ďalej x a y znamenajú koordináty v priestore a t hodnotu v čase. Hodnoty t môžu nadobúdať hodnoty od 0 až nekonečno. Pre praktické použitie je aplikovaný diskretizovaný obraz podľa rovnice 2.1.

„Definiční obor D hodnot obrazové funkce:

$$x \in (x_{min}, x_{max}), y \in (y_{min}, y_{max}) \tag{2.3}$$

Obor hodnot H obrazové funkce:

$$0 \leq f(x, y) \leq f_{max} \tag{2.4}$$

” [1]

Táto definícia nám opisuje šedotónový obraz. Takýto obraz je potrebný k jednoduchšiemu spracovaniu obrazu v počítačovej grafike. Pre digitálne spracovanie je najčastejšie využívaný obraz kvantovaný do rovnakých veľkostí intervalov K . V počítačovej grafike to znamená, že to je prevod do binárnej podoby.

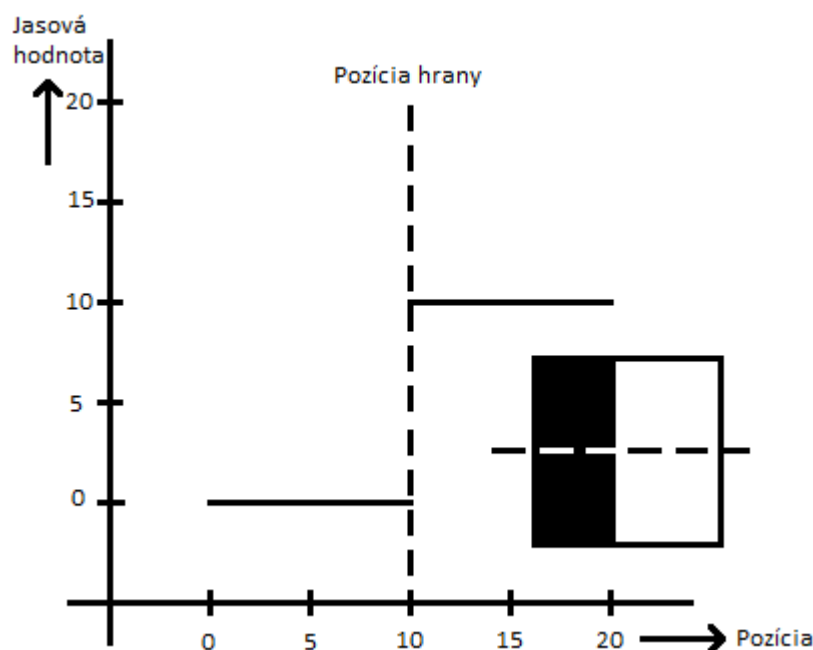
$$K = 2^b \tag{2.5}$$

Kde b je počet bitov. Pre ľudské oko je postačujúce približne 64 jasových úrovní a viac, pri ktorých už nevznikajú rušivé artefakty. Pre spracovanie obrazu je najčastejšie používaný 8-bitový obraz. Každý pixel môže mať iba jednu hodnotu od 0 až do 255, kde 255 je biela farba (najintenzívnejšia jasová úroveň) a 0 je čierna farba (najmenej intenzívna jasová úroveň).

2.2 Detekcia hrán v obraze

V skutočnosti sa hrana dá definovať viacerými spôsobmi. Jedna z najpoužívanejších 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



Obr. 2-1 Ideálna hrana [2] prevzatá a prerobená

„Každá hrana v obraze predstavuje jasovú nespojitost v obrazovej funkcii.“ [1]. 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 preto, lebo je ľahko rozpoznateľná. V skutočnosti sa kroková hrana nedá rozpoznať z viacerých dôvodov.

Tab. 2-1 Zoznam dôvodov nerozpoznatelnosti krokovej hrany

- | |
|---|
| <ol style="list-style-type: none"> 1) nasnímaný objekt nemá ideálny obrys 2) pri snímaní sa hrana objektu nachádza v rozpätí viacerých pixelov 3) z dôvodu šumu v obraze |
|---|

2.3 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 spojitosť obrazu napríklad: zrnitosť.

„V digitálnom obraze rozeznávame najčastejši dva typy šumu:

- náhodný šum, také nezávislý šum, je spôsoben napríklad vadnými CCD elementy (príkladom tohoto typu šumu je šum typu „sól a pepř“)
- Gaussův šum, také závislý šum, kde je každý pixel obrazu mírně pozmeněn“ [3]

Náhodný šum môže nastať, 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:

$$B = A + N \quad (2.6)$$

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.

$$G(r, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}} \quad (2.7)$$

„kde z je náhodná hodnota, σ je smerodajná odchylka“ [6]

2.4 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. Tento uhol je okolo 90 stupňov od osi hlavy. Vo vertikálnom smere je zorný uhol 50 stupňov od osi hlavy. Na zorný uhol majú vplyv vlastnosti pozorovaného objektu, ako napríklad: intenzita osvetlenia, farba objektu, veľkosť objektu a.i.. [4]

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.

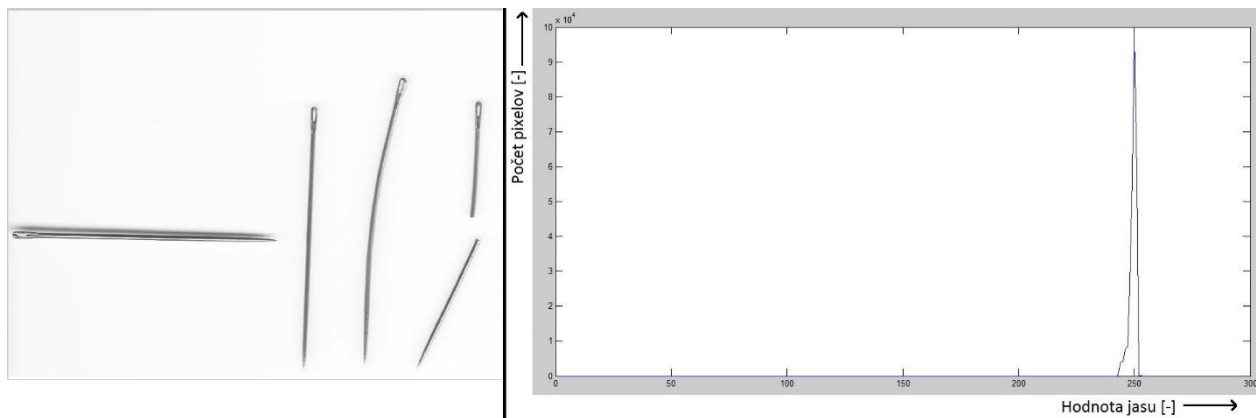
2.5 Histogram obrazu

Histogram využívajú niektoré ďalšie metódy, ktoré podľa histogramu 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.“ [6]

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

Príklad histogramu s obrazom



Obr. 2-2 Histogram obrazu

Na obrázku 2-2 vidno na ľavej strane originál obrazu a na pravej strane jeho histogram. Na vodorovnej ose grafu sú hodnoty 8 bitového obrazu od 0 do 255. Na zvislej ose grafu je súčet jednotlivých hodnôt obrazu. Z grafu vidíme, že väčšina jasových hodnôt je okolo hodnoty 255. Graf vyjadruje skutočnosť, že väčšina pixelov obrázku zodpovedá jasovej úrovni, ktorá sa približuje k bielej farbe (jasová hodnota 255).

2.6 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 viac vlákien pre výpočet s viacerými procesormi a tým výrazne zrýchliť čas výpočtu.

2.7 Konvolúcia

Konvolúciu využívame pri ďalších metódach spracovania obrazu. 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:

$$g(x, y) = \sum_{(m,n) \in O} h(x - m, y - n) f(m, n) \quad (2.8)$$

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 [6]

Maska má obyčajne nepárny počet stĺpcov a riadkov v pravidelnom v štvorcovom prevedení, aby počítaný bod ležal v strede masky (pre lepší výpočet).

3 CIEĽ PRÁCE

Cieľom tejto práce je vytvoriť aplikáciu na vyhodnotenie kvality valcových súčiastok. Kontrolovanými súčiastkami som si zvolil ihly.

Práca sa člení na :

- 1) rozbor úlohy, stanovenie parametrov a tolerancie
- 2) návrh vhodného pracoviska pre snímanie ihliel
- 3) stanovenie detekovateľných chýb ihliel, vytvorenie databázy správnych a chybných ihliel 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 každodennom priemysle. Ihly na šitie textilu môžeme rozdeliť na ručné a na strojové. Moja aplikácia sa zaoberá klasickými ručnými ihlami. Príklad je na obrázku 4-9.

4 METÓDY A POSTUPY RIEŠENIA ALGORITMOV

Ihly majú valcovitý tvar, jednoznačne rozoznatelné konce (dierku na jednej strane a hrot na druhej strane). Algoritmy by mali rozhodnúť o tom, či je tvar ihliel správny alebo nie.

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 ihla je nevyhovujúca. Vady alebo nesprávnosti ihly sú rôznorodé. Po dlhšom vyhľadávaní noriem som nenašiel presnú definíciu chýb, preto som stanovil nasledujúce chyby podľa vlastného uváženia:

- 1) zlom v tele ihly
- 2) ohnutá ihla
- 3) chýbajúce ucho ihly
- 4) opotrebovaný hrot ihly
- 5) diera v tele ihly
- 6) nesúhlas v parametroch ihly

4.2 Vyhľadzovalie filtre

Vyhľadzovanie obrazu filtrom používame v prípade, keď šum skreslí nasnímaný obraz. V prípade, keď vieme aký šum pôsobil na obraz, tak vieme efektívnejšie odstrániť šum filtrom, ktorý vyhladzuje obraz s daným šumom.

4.2.1 Priemerovacie filtre

Priemerovací filter vytvorí zo vstupného obrazu $f(i,j)$ priemerovaný výstupný obraz $g(i,j)$ konvolúciou. Najpoužívanejšia je konvolučná maska 3x3. Táto maska je optimálna pre výpočet a efektívnosť odstránenia šumu. Nevýhodou je, že mierne rozostří výsledný obraz oproti pôvodnému obrazu.

Príklad priemerovacej masky:

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

Obr. 4-2 Priemerovacie masky

Maska v okolí každého pixelu vstupného obrazu $f(i,j)$ je vynásobená priemerovacím filtrom. Vo výstupnom obraze je zapísaná hodnota priemerovacieho filtru h . Priemerovacie filtre môžu mať rôzne masky. Často sa používa v strede masky väčší koeficient, aby počítaný pixel mal dominantnejšiu hodnotu. Tým dosiahneme, aby priemerovací filter menej rozostřil výstupný obraz.

4.2.2 Gaussov filter

Na vyhladenie obrazu, ktorý obsahuje Gaussov šum, je optimálny Gaussov filter. Pracuje podobným princípom 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.

Gaussova krivka v tomto prípade znamená pravdepodobnosť, s akou sa vyskytuje a aký veľký je rozdiel medzi pixelom v originálnom obraze a zašumenom obraze. Čím je krivka „užšia“, tým je menšia pravdepodobnosť výskytu zvýšeného šumu.

4.3 Hranové filtre

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

4.3.1 Globálne prahovanie

Pri práci bola vyskúšaná metóda globálneho prahovania. V Matlabe bol použitý na obraze s ihlami a výsledok je uvedený na obrázku 4-4.

Globálne prahovanie znamená, že zo vstupného šedotónového obrázku $g(i,j)$ urobíme binárny obraz $f(i,j)$ pomocou jedného alebo dvoch prahov. Pri 8-bitovom obraze môže mať pixel

hodnotu od 0 až do 255. 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 .

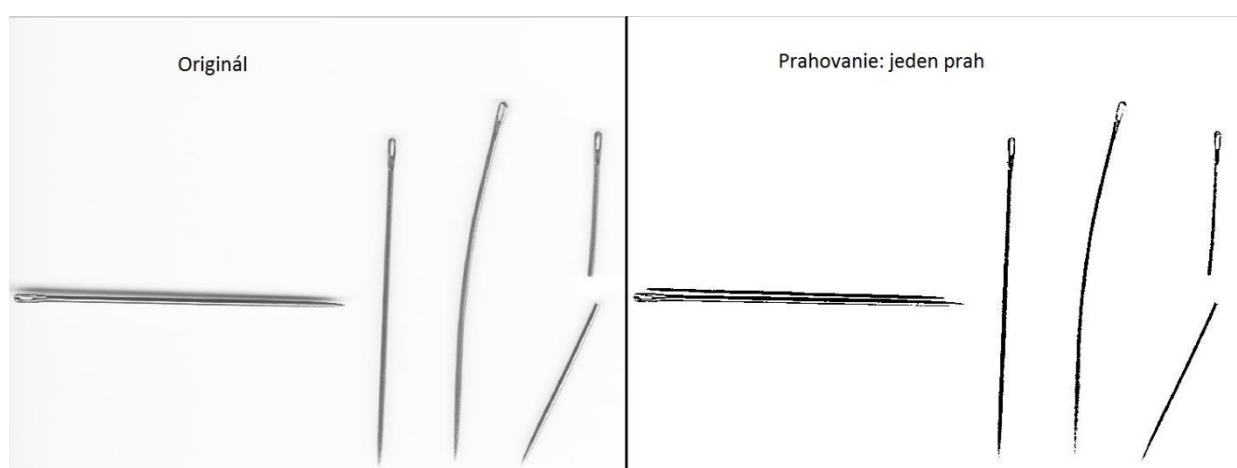
Pri používaní jedného prahu to znamená, že keď pixel má hodnotu väčšiu alebo rovnú prahu T , vo výslednom obrázku bude mať hodnotu 1. Keď bod v obrázku je pod prahom, bude mať hodnotu 0.

$$f(i,j) = \begin{cases} 1 & \text{pre } g(i,j) \geq T \\ 0 & \text{pre } g(i,j) < T \end{cases} \quad (4.1)$$

[5]

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

Príklad prahovania jedným prahom



Obr. 4-4 Príklad prahovania jedným prahom

Na obrázku 4-1 na ľavej strane je originál obrazu 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 pixle, ktoré boli menšie než 150, vo výslednom obrázku majú hodnotu 0 a pixle ktoré majú hodnotu 150 alebo viac až do 255 majú hodnotu 255.

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. Vo výstupnom obrázku bude toľko jasových úrovní, koľko sme si zvolili prahov.

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

4.3.2 Adaptívne prahovanie

V prípade, keď je na obrázku nasnímaných viacero objektov, ale majú rozdielne jasové úrovne, môžeme používať adaptívne prahovanie. Pri metóde adaptívneho prahovania vypočítame bod prahu pre každý bod v obraze zvlášť. Tým zvýšime úspešnosť vyhľadávania hrán, ale rýchlosť výpočtu sa zníži.

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.

Druhá možnosť je používanie lokálneho prahovania. Zvolíme si malú časť okolia daného bodu v obraze (masku) a pomocou tejto masky vypočítame prahovú hodnotu. Prahovú hodnotu môžeme vypočítať priemerovaním jasových hodnôt okolných pixelov, mediánom alebo strednou hodnotou najväčšej a najmenšej hodnoty okolných bodov v obraze. Výpočet sa zvýši zvýšením 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 nabra ∇ :

$$\nabla f(x, y) = \left(\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right) \quad (4.3)$$

a veľkosť gradientu môžeme vypočítať ako dĺžku vektora:

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x} \right)^2 + \left(\frac{\partial f(x, y)}{\partial y} \right)^2} \quad (4.4)$$

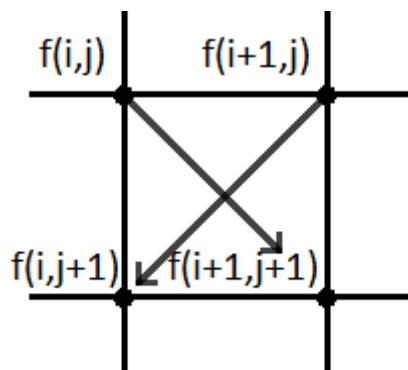
[9]

4.3.3 Robertsov operátor

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

$$|\nabla f(i, j)| = |f(i, j) - f(i + 1, j + 1)| + |f(i, j + 1) - f(i + 1, j)| \quad (4.5)$$

Robertsov operátor detekuje hrany na vstupnom obraze $f(i, j)$ kolmo po hlavnej a po vedľajšej diagonále. Výpočet určí gradient ako súčet absolútnych hodnôt zmien v smere hlavnej a vedľajšej diagonáli.



Obr. 4-5 Princíp Robertsovho operátora [9]

V maticovom prevedení to vyzerá nasledovne:

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (4.6),(4.7)$$

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

4.3.4 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).

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad (4.8),(4.9)(4.10)$$

$$h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \dots$$

V tomto prípade h_1 slúži pre výpočet gradientu vo vertikálnom smere zdola nahor, h_2 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.

4.3.5 Prewittiv operátor

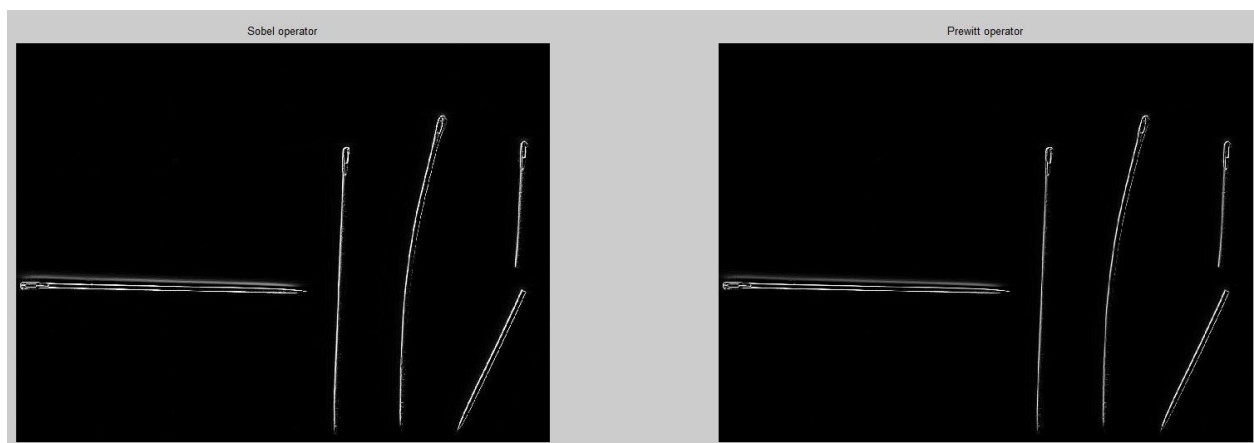
Tento operátor som použil pri práci v Matlabe. Je užitočný pri ďalších prácach.

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

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad (4.11),(4.12)(4.13)$$

$$h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \dots$$

Výsledky Sobelovho a Prewittovho operátora sa veľmi podobajú.



Obr. 4-6 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.

4.3.6 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 “ [11]

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

$$\Delta f(x, y) = \frac{\partial^2 f(x, y)}{\partial^2 x} + \frac{\partial^2 f(x, y)}{\partial^2 y} \quad (4.14)$$

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 negatívnu. V praxi používame buď len pozitívnu hodnotu, to znamená, že vzostupná hrana sa neberie do úvahy, alebo počítame s absolútnou hodnotou, a tak z negatívnych hodnôt urobíme pozitívne hodnoty.

Laplaceov operátor je použiteľný s dvoma konvolučnými maskami:

a) výpočet s operátorom v horizontálnych a vertikálnych smeroch:

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.15)$$

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

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.16)$$

Rozdiel medzi konvolučnými maskami 4.15 a 4.16 je , že v prípade 4.15 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 než 3×3 .

4.3.7 Cannyho detektor

John F. Canny (1953) v roku 1986 definoval základné pravidlá, ktoré má splniť prahový detektor. Podľa týchto pravidiel stanovil metódu, s ktorou sa dá hrana najoptimálnejšie vyhľadať.

Tab. 4-1 Zoznam Cennyho pravidiel

- 1) Miera chýb: Hranový detektor má reagovať len na hrany a musí nájsť všetky hrany. Nesmie vynechať ani jednu.
- 2) Poloha hrany: Poloha hrany v skutočnosti a poloha hrany nájdená hranovým detektorom má byť taká malá ako sa len dá.
- 3) Reakcia: Hranový detektor nesmie nájsť viac hranový pixel, kde je len jeden hranový pixel.

[2]

Cannyho detektor je závislý na šume a preto treba obraz vyfiltrovať. Obvykle používame na filtráciu Gaussov filter.

Príklad Gaussovej masky 5×5 :

$$h = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (4.17)$$

Princíp Cannyho detektora:

- 1) Načítame vstupný obraz I .
- 2) Vytvoríme jednodimenzovú Gaussovú masku G a skonvolujeme s obrazom I . Štandardná odchýlka s z Gaussianu je parametrom pre hranový detektor.
- 3) Vytvoríme jednodimenzové masky pre prvú deriváciu Gaussianu vo smere x a y a nazveme ich G_x a G_y . Používame ten istý parameter s , ako v prípade 2
- 4) Použijeme konvolúciu masky G po obraze I po riadkoch (aby sme dostali x -ové prvky I_x) a po stĺpcoch (aby sme dostali y -ové prvky I_y).
- 5) Konvolúciu použijeme na obraz I_x s G_x aby sme dostali derivovaný obraz I_x' vo smere x a konvolujeme obraz I_y s G_y pre derivovaný obraz I_y' .
- 6) Výsledný obraz dostaneme spojením dvoch obrazov x a y nasledovnou rovnicou:

$$M(x, y) = \sqrt{I_x'(x, y)^2 + I_y'(x, y)^2} \quad (4.18)$$

[2]

Tento detektor hrán som nepoužil, kvôli jeho pracnosti.

4.4 Návrh pracoviska na snímanie objektu

Snímanie objektu nie je vždy jednoduchá záležitosť. Vizuálna kontrola obvykle funguje tak, že so svetelným zdrojom posvietime na snímaný objekt a následný obraz po dopade svetla na objekt snímame kamerou.

Pre rôzne typy objektov a pre rôzne ciele vizuálnej kontroly rozlišujeme metódy snímania, ktoré sú ovplyvnené nasledovnými parametrami:

- objekt je transparentný (priesvitný) alebo polo transparentný
- objekt má lesklý povrch
- objekt má matný povrch
- objekt odrazí svetlo iným spôsobom, než to je žiaduce.

Nasledujúce podkapitoly popisujú metódy snímania objektov.

4.4.1 Zrkadlové osvetlenie

Zrkadlové osvetlenie je najjednoduchším spôsobom snímania objektov. 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 sa v praxi nepoužíva často na meranie objektu. Veľkou nevýhodou tejto metódy je, že snímaný objekt vytvára tieň na pozadí. Tieň na pozadí môže pôsobiť po spracovaní obrazu ako falošná hrana objektu.

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

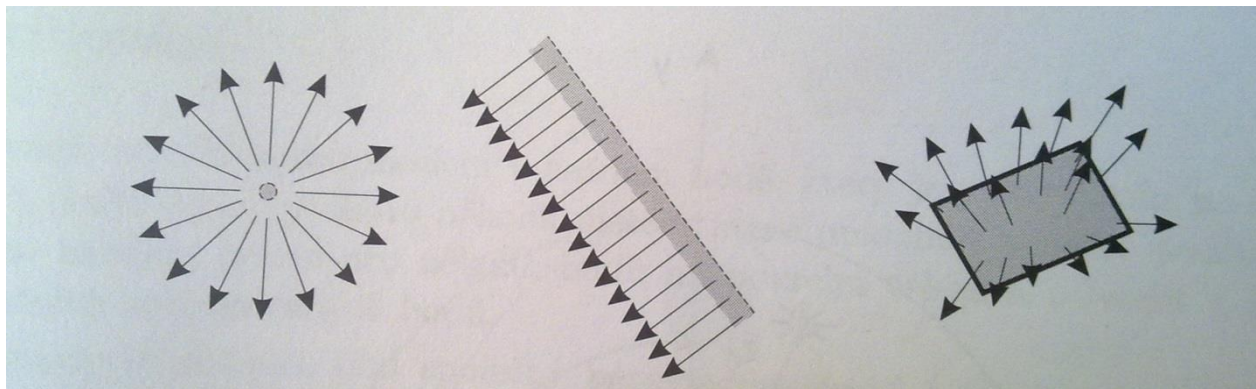
Táto metóda sa môže použiť na kontrolu kvality niklovania ihiel. Keď na lesklý poniklovaný povrch ihly svietime rovnobežným svetlom, tak telo ihly odráža zdroj svetla rovnomerne smerom ku kamere. Podľa stupňa odrážaného lesku môže uvedený algoritmus vyhodnotiť správnosť poniklovania ihly.

4.4.2 Rozptýlené osvetlenie

Rozptýlené osvetlenie je podobné ako zrkadlové osvetlenie. Rozdiel medzi týmito dvoma metódami môže byť rozšírenie svetelného zdroja alebo rozptýlenie svetelných lúčov svetelného zdroja medzi zdrojom svetla a snímaným objektom.

Rozšírenie svetelného zdroja spočíva v použití viac bodových svetelných zdrojov s podobnými 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 spôsobuje, že na pozadí sa tieň objektu rozdelí na tieň a polotiene.

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.



Obr. 4-6 Zdroje svetla [1]

Zľava: bodový zdroj svetla, zdroj rovnobežného svetla a plošný zdroj.

4.4.3 Zadné osvetlenie

Zadné osvetlenie objektu je najlepším spôsobom snímania rovných súčiastok. Nepriesvitné objekty sa dajú snímať bez odrazu lesklého povrchu a bez tieňa na pozadí. Pri spracovaní obrazu je tak minimálna pravdepodobnosť detekovania falošných hrán.

Metóda spočíva v tom, že zdroj svetla (plošný, rovnobežný alebo bodový zdroj) vyžaruje lúče v smere snímaného objektu. Objekt vytvára rovnomerný tieň. Ďalej je v smere svetla kamera (obr. 4-7). Takto snímaný obraz obsahuje obrys snímaného objektu a okolité body obrazu reprezentujú svetelný zdroj. Tým pádom, kde na obrázku je snímaný svetelný zdroj má hodnota pixlu vysokú hodnotu a kde dopadá tieň objektu má hodnota pixlu malú hodnotu. Pre spracovanie obrazu je táto metóda najvhodnejšia a preto som ju pri snímaní obrazu použil.

Pre lepšie rozloženie lúčov a aby boli lúče kolmé na snímaný objekt, som použil matnicu. Matnica je polo transparentný objekt s matným povrchom.

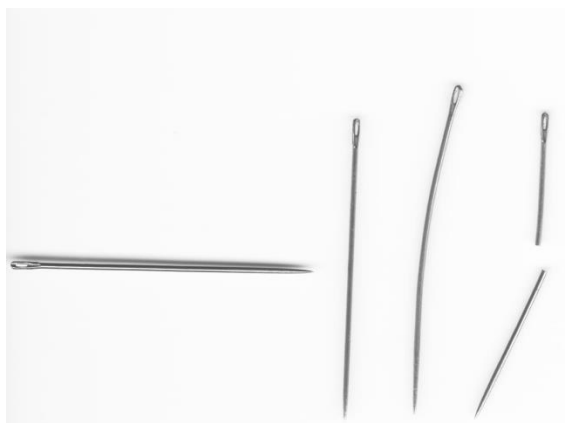
4.4.4 Snímanie skenerom

Snímky boli snímané multifunkčným skenerom MP160. Tento skener má za hlavný účel snímať dokumenty a fotky. Pri snímaní ihlami týmto skenerom boli vyskúšané viaceré nastavenia snímania, ktoré mi prístroj umožnil. Zvolený najoptimálnejší obrázok som použil pri spracovaní.

Nedostatky metódy, ktoré ovplyvňujú snímanie:

- Ihly po ich položení nie sú vždy umiestnené tak, aby bolo vidno dierku, lebo nie je otočená ihla správnym smerom
- Mechanické vady alebo nečistoty na priesvitnom skle alebo na pozadí vytvárajú falošný obrys na obrázku
- Zorné pole je obmedzené na dané parametre skeneru
- 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 sa u takéhoto skenera bez mechanického poškodenia nedá zväčšiť. Falošné kontrastné obrisy, ktoré sú spôsobené svetlom dopadajúcim priamo na objekt sa dajú redukovať zmenšením intenzity osvetlenia.



Obr. 4-7 Snímka skenerom

Na obraze 4-7 som uviedol príklad snímania skenerom. Na príklade vidno, že na pozadí sa vytvára tieň. Na tele ihly sa odráža svetlo zdroja od lesklého povrchu. Pri vyhľadávaní hrán z tohto obrázku by som algoritmom našiel viac falošných hrán. Tento spôsob snímania obrazu pre moju prácu je nevhodný.

4.4.5 Vyhotovenie pracoviska na meranie ihiel

Na meranie ihiel som vyhotovil konštrukciu zo železa, ktorá je znázornená na ďalšom obraze. 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. Na spodnej časti stojana sa dá umiestniť svetelný zdroj.



Obr. 4-8 Konštrukcia

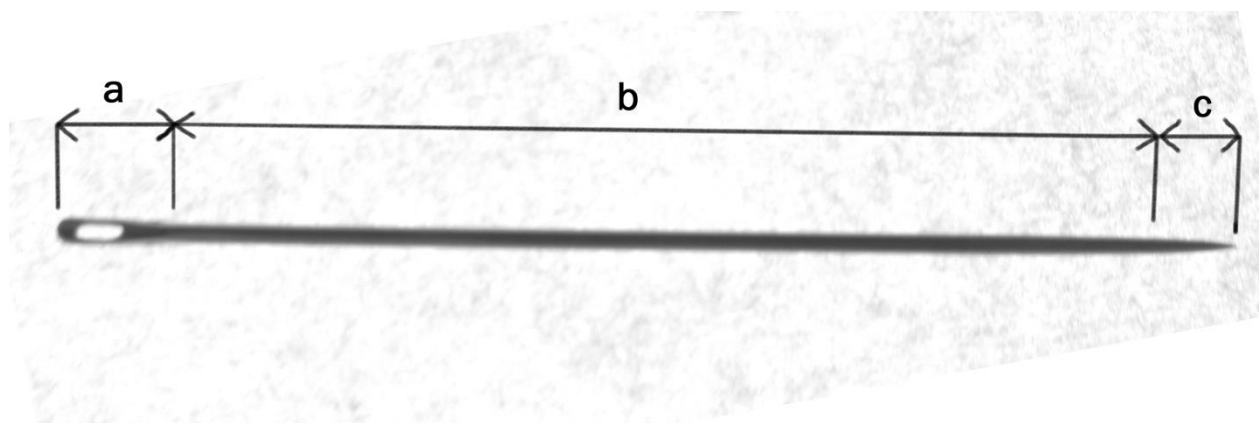
Konštrukciu som vyhotovil tak, aby pre ďalšie práce vizuálnej kontroly sa mohli rozšíriť možnosti snímania. Na hornej časti je možné pridať ďalšie zdroje svetla. Tieto zdroje môžu slúžiť na snímanie metódou zrkadlového osvetlenia.

Ihla (resp. ihly) som položil na matnicu. Pod stojan som umiestnil zdroj svetla typu reflektor. Na hornej časti stojanu som umiestnil kameru a takým spôsobom som nasnímal obraz. Obraz som previedol do počítača a uložil som to do adresára hlavnej funkcie. V hlavnej funkcii sa obraz volá do prostredia Matlab a pomocou algoritmov vyhodnotí ihlu.

4.5 Riešenia 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 tolerancie častí ihly. Nasledujúci obrázok reprezentuje snímaný objekt a jeho časti.



Obr. 4-9 Popis ihly

Časti ihly sú:

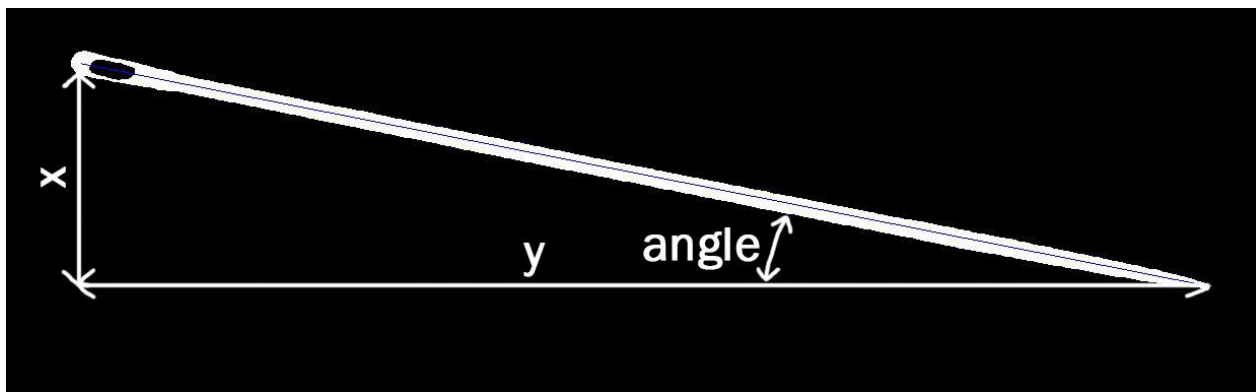
- a- ucho (uško) ihly
- b- telo ihly
- c- hrot ihly

Časti a, b a c z obrazu 4-9 tvoria spoločne dĺžku ihly. Bod na obraze, ktorý je na konci hrotu sa dá považovať za krajný bod ihly. Bod na obraze, ktorý je na opačnej strane predošlého bodu, to je krajný bod ucha a dá sa považovať za druhý krajný bod ihly.

Zo snímaného objektu sa dajú detekovať rôzne vady, ktoré definujem a riešim v nižšie uvedených kapitolách.

Pri ďalších úvahách som si zvolil štandardnú polohu ihly tak, že ucho ihly je na ľavej strane a hrot ihly je na pravej strane. Ihla je na obraze vodorovne orientovaná. Z tejto štandardnej polohy následne operujú ďalšie algoritmy.

Aby som nasnímaný obraz previedol na čiernobiely obraz, tak som použil metódu globálneho prahovania. Pre rotáciu ihly do štandardnej polohy som vytvoril algoritmus „search2“ a následne použil funkciu „imrotate“. Zabudovaná funkcia „imrotate“ otočí vstupný obraz „A“ o uhol „angle“. Vstupnými parametrami sú „A“ a „angle“. „Angle“ som vypočítal z výstupných hodnôt funkcie „search2“. Tangent uhlu rotácie sa rovná podielu vzdialeností krajných bodov ihly po horizontálnej a vertikálnej osi.



Obr. 4-10 Výpočet uhlu

Z matematiky som odvodil pre uvedený, prípad na obrázku 4-10:

$$\tan(\text{angle}) = \frac{x}{y} \quad (4.19)$$

$$\text{angle} = \tan^{-1}\left(\frac{x}{y}\right) \quad (4.20)$$

Podľa tejto rovnice som vypočítal uhol pre rotáciu do štandardného tvaru.

4.5.1 Funkcia search2

Funkcia search2 vyhledá z čiernobieleho obrazu krajné body ihly. Vrátene 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 hodnoty vyhovujúce podmienkam. Vypočíta priemer v blízkosti hľadaného bodu a následne prepíše do premenných x a y. Pre krajný bod na druhej strane je princíp podobný.

4.5.2 Riešenie chyby: zlom v tele

Chyba: zlom v tele ihly spočíva v tom, že ihla je zlomená na dve alebo viac častí. Môj algoritmus vyhledá zlom zo vstupného čiernobieleho obrazu. Výstupným parametrom je hodnota, ktorá je počas algoritmu nastavená na určitú hodnotu. Hodnota reprezentuje stav ihly, či je zlomená alebo je celistvá.

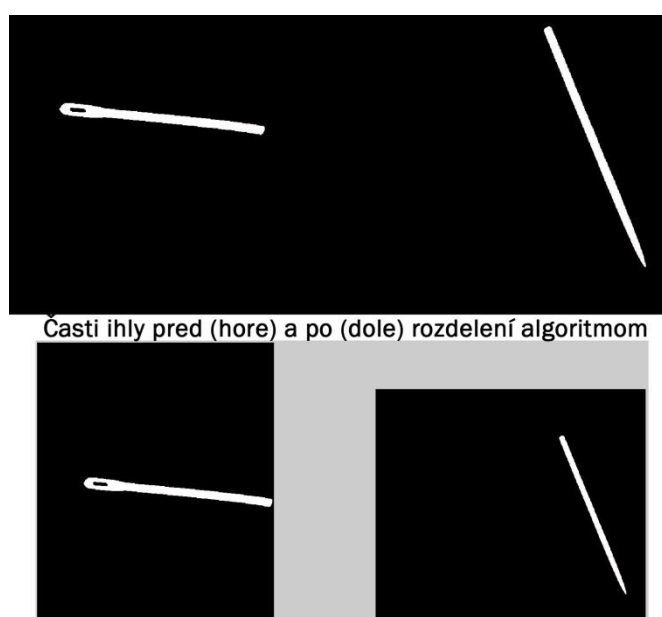
Chybu riešim vo funkcii „search_zlom“. Predpokladám, že na vstupnom obraze je ihla zlomená a má obidva diely na obraze. Ďalej predpokladám ideálny prípad, keď obdĺžnik okolo jednej časti ihly nepresahuje do obdĺžnika okolo druhej časti ihly. Obdĺžnik okolo ihly znamená najmenší obdĺžnikový výrez z pôvodného obrazu, do ktorého sa zmestí zlomená časť ihly. Algoritmus rieši zlom v prípade, keď na obrázku sú dva diely zlomenej ihly.

Rozlišujem dve možnosti polohy častí ihliel. Prvou možnosťou je, že štvorce okolo zlomenej ihly sú umiestnené vedľa seba. Umiestnenie štvorcov vo vertikálnom smere nemá vplyv na

rozpoznanie zlomu. Druhou možnosťou je, že štvorce zlomenej ihly sú umiestnené na obrázku pod sebou. Na rozpoznanie zlomu v tomto prípade nemá vplyv horizontálne rozmiestnenie.

V prvej polovici algoritmu skúmam zlom zo vstupného obrazu, keď obdĺžniky sú umiestnené vedľa seba. Program postupuje zľava doprava po riadkoch. Algoritmus vyhľadá pravý krajný bod zlomenej časti ihly na ľavej strane obrazu a ak následne nájde druhú časť ihly, tak sa výstupná premenná nastaví na prvú chybovú hodnotu. Táto hodnota vyjadruje, že v ihle je zlom a ďalej vyjadruje to, že zlomené časti sa umiestňujú vedľa seba na vstupnom obraze.

Druhá polovica algoritmu je podobná prvej. Rozdiel medzi časťami algoritmu spočíva v tom, že cyklus prechádza v druhej časti zhora nadol, teda opačným smerom. Keď identifikuje zlomené časti ihly umiestnené pod sebou, tak sa nastaví výstupná premenná na druhú chybovú hodnotu. Táto chybová hodnota napovedá o tom, že ihla je zlomená a zlomené časti sú umiestnené pod sebou.



Obr. 4-10 Rozdelenie častí zlomenej ihly

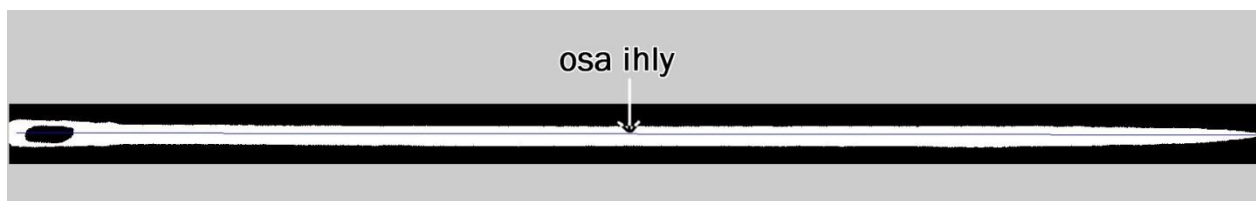
Na hornom obrázku je vstupný čiernobiely obraz do funkcie. Počas funkcie som názorne ukázal dva diely ihliel na dolnej časti obrazu. V ľavom spodnom obrázku je znázornená časť zlomenej ihly orezaná po stĺpci, kde algoritmus našiel bod rezu (premenná „rez“). V pravom spodnom obrázku je ihla orezaná od stĺpca, kde algoritmus našiel bod rezu.

4.5.3 Riešenie chyby: ohnutá ihla

Chyba: ohnutá ihla znamená, že ihla je ohnutá ku svojej ose, od hrotu k uchu ihly. Ohnutie môže nastať smerom dole k osi a smerom hore k osi. Ohyb od smeru osi sa pohybuje v určitých toleranciách. Tieto hodnoty som nastavil podľa svojich skúseností.

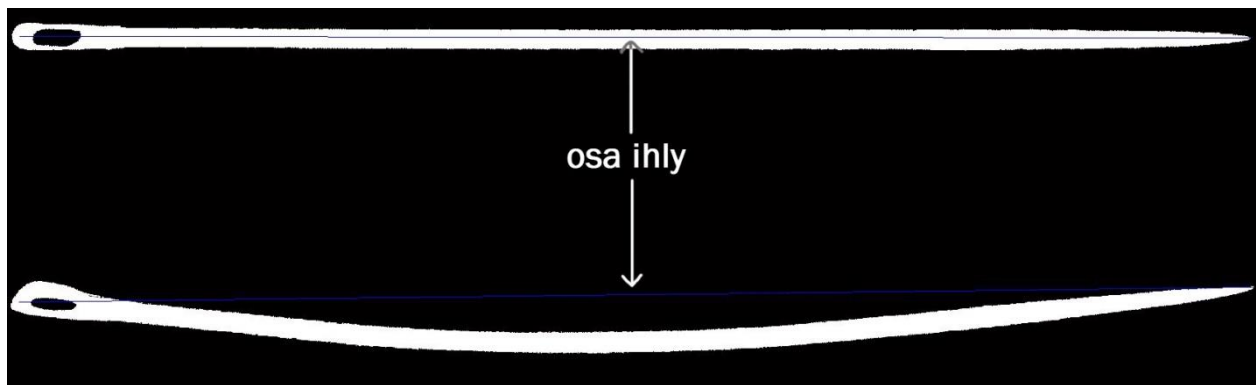
Chybu rieši funkcia „search_ohnutie“. Vstupnými parametrami funkcie sú: čiernobiely obraz a hodnota výšky stĺpca ktorý reprezentuje osu ihly. Túto hodnotu som vyhľadal pomocou funkcie „search2“. Výstupom je hodnota, ktorá môže mať tri stavy. Štandardná hodnota je 1, ktorá znamená, že ihla nie je ohnutá podľa tolerancie. Hodnota 2 znamená ohnutie smerom hore od osi a hodnota 3 znamená ohnutie smerom nadol od osi.

Algoritmus vyhľadá všetky body ihly a keď bod spĺňa podmienku stanovenú toleranciou rozdielu od bodu k ose ihly, tak do výstupnej premennej uloží stav, ktorý určí smer ohnutia.



Obr. 4-11 Ukážka osi ihly po algoritme

Na obrázku je vidno osu ihly po určení algoritmu. Podľa diferencie od body ihly a bodu osi vyhodnotí algoritmus, či je ihla ohnutá alebo nie. Nasledujúci obrázok znázorní rozdiel medzi ihlou ohnutou a rovnou.



Obr. 4-12 Ukážka osi ihly po algoritme s ohnutím

Výstupom algoritmu so vstupným obrazom správnej ihly (ihla z hornej časti) bola hodnota stavu vyhovujúca. Naopak algoritmus vyhodnotil stav ihly ohnutej (ihla z dolnej časti) ako nevyhovujúca a výsledkom bolo, že ihla je ohnutá smerom dole od osi ihly.

4.5.4 Riešenie chyby: chýbajúce ucho ihly

Chyba: chýbajúce ucho ihly môže nastať vtedy, ak sa vyskytne chyba vo výrobe a ihla nemá dierku ani na jednom konci ihly. Táto chyba znehodnotí ihlu a preto je dôležité detekovať túto chybu. Chybu môže zapríčiniť slabé lisovanie ucha čo možnou príčinou celej vadnej série.

Pre riešenie tejto chyby som vytvoril funkciu „search_usko2“. Vstupom funkcie je čiernobiely orezaný obraz, hodnota vzdialenosti ucha v pixloch a tolerancia udávaná v percentách. Výstupom je premenná, ktorá môže mať štyri stavové hodnoty.

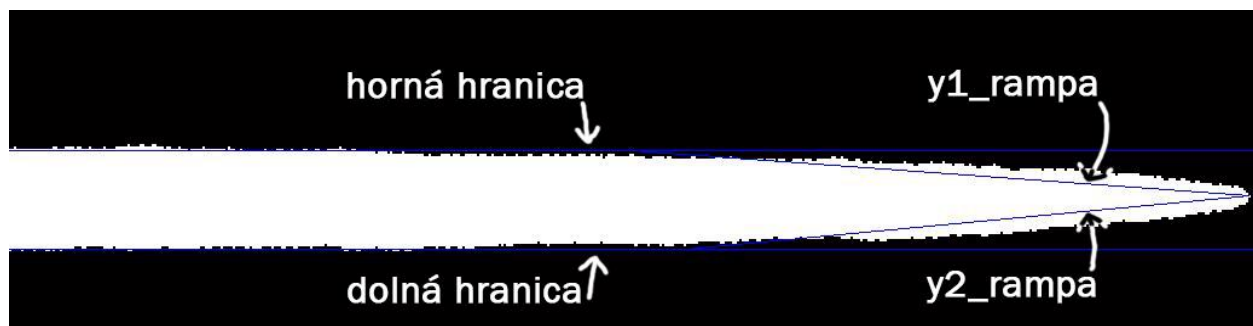
Prvá stavová hodnota znamená, že ihla má ucho a je v zadanej tolerancii. Druhá stavová hodnota znamená, že ihla má ucho, ale to je väčšie než zadaná tolerancia. Tretia stavová hodnota znamená, že ihla má uško, ale vzdialenosť jeho krajných bodov je menšia než povolená tolerancia. Štvrtá stavová hodnota znamená, že ihla nemá žiadne ucho.

Algoritmus prechádza vstupný obraz po stĺpcoch zľava doprava. Ucho ihly je z čiernobieleho obrazu rozpoznať tak, že pixle smerom zhora nadol menia hodnotu štyrikrát. Algoritmom spočítam počet derivácií v jednom stĺpci a následne zväčšujem premennú „prem2“. Prem2 je vzdialenosť od krajných bodov uška. Keď táto vzdialenosť je menšia než tolerancia, tak sa výstupná premenná nastaví na tretiu stavovú hodnotu. V opačnom prípade, keď je vzdialenosť väčšia než tolerancia uška, tak sa nastaví premenná na druhú stavovú hodnotu. Ak žiadna derivácia nenastala počas algoritmu, výstupná premenná sa nastaví na štvrtú stavovú hodnotu. Ak vzdialenosť krajných bodov uška je v rozsahu tolerancie, vyhodnotí sa ihla ako vyhovujúca.

4.5.5 Riešenie chyby: opotrebovaný hrot ihly

Chyba: opotrebovaný hrot ihly znamená, že pri nadmernom použití ihly sa hrot opotrebuje. V iných prípadoch chyba môže nastať pri automatizovanej výrobe, keď prístroj nevhodne vybrúsi hrot ihly. Túto chybu je treba detekovať pri vizuálnej kontrole.

Algoritmus som napísal do funkcie „search_hrot“. Vstupným parametrom je čiernobiely obraz. Výstupom parametrom je premenná s hodnotou stavu vyhodnotenia. Vo funkcii som použil premennú „konst“, ktorá mi slúži ako konštanta pri prepočtoch z radiánov na stupne. Premenná „percento“ vyjadruje percento výrezu zo vstupného obrazu. Tento výrez obsahuje pre mňa zaujímavú časť snímaného objektu.



Obr. 4-13 Ukážka hrotu po algoritmu

Horná a dolná hranica vyjadrujú rovnobežky s telom ihly. Rovnobežku od krajného bodu hrotu k hornej alebo dolnej hranici som stanovil ako „y1_rampa“ a „y2_rampa“.

Uhol medzi hornou resp. dolnou hranicou a rovnobežkami y1_rampa resp. y2_rampa určí stav hrotu ihly a sú pomenované ako „angle1“ a „angle2“. Keď angle1 alebo angle2 dosiahne väčšiu hodnotu než určitá tolerancia, tak výstupná hodnota sa nastaví do stavu, ktorý vyjadruje, že hrot ihly je opotrebovaný.

Hranice ihly som stanovil z viacerých meraní spriemerovanou hodnotou. Bod, kde sa stretne hranica s rampou je určený ďalšou časťou algoritmu. V cykle sa hľadajú body, kde hraničný pixel obrazu ihly prekročí toleranciu hranice o určitú hodnotu. Tieto body sú stredobodom rámp s hranicami.

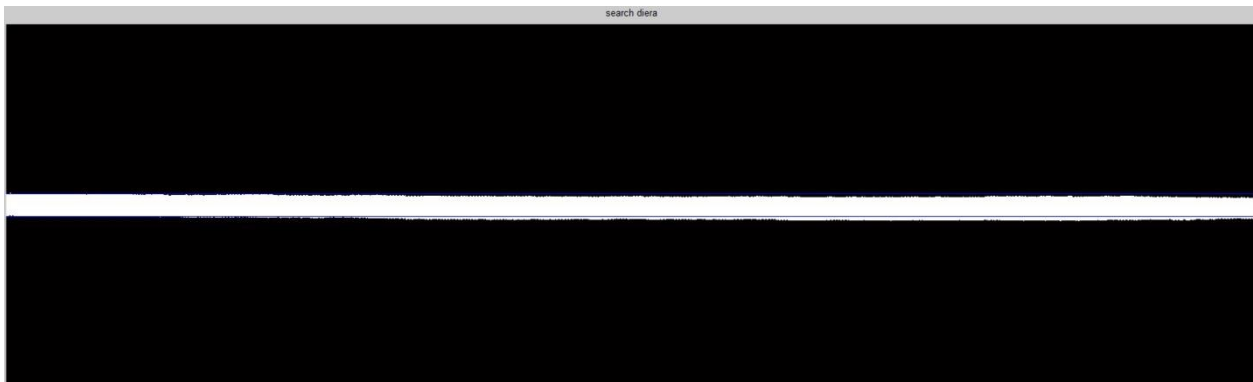
4.5.6 Riešenie chyby: diera v tele ihly

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 ne je vždy spozorovateľná voľným okom.

Vychádzal som zo skutočnosti, že dôležitou časťou ihly je jej telo, a preto je dôležité kontrolovať symetriu ihly. Diera na čiernobielym obraze 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. Vo funkcii odrežem hrot a ucho ihly pomocou funkcie „imcrop“. Na výslednom obraze vyhľadám hornú a dolnú hranicu ihly. V programe určujem smerom sprava doľava bod, ktorý je pod toleranciou hornej hranice alebo nad toleranciou dolnej hranice. Tolerancie sa nastavujú v programovej časti funkcie. Keď program nájde bod, ktorý je vyhovujúci podmienkam, tak nastaví výstupnú premennú na

chybovú hodnotu. To znamená, že v tele ihly je diera alebo nerovnosť. Štandardná hodnota výstupnej premennej je stav „1“. Tento stav znamená, že ihla je vyhovujúca.



Obr. 4-14 Ukážka vyhovujúcej ihly po algoritme

Na obrázku je orezaný vstupný obrázok bez hrotu a ucha. Je na ňom vyznačená horná a dolná hranica, podľa ktorých algoritmus vyhodnotí stav ihly.

4.5.7 Riešenie chyby: nesúhlas v parametroch ihly

Chyba: nesúhlas v parametroch ihly znamená, že nameraná hodnota určitého parametru ihly je mimo rozsah tolerancie. Táto hodnota sa dá merať pomocou vizuálnej kontroly. Pri vizuálnej kontrole je potrebné si určiť metrickú hodnotu vzdialenosti. Podľa tejto vzdialenosti sa dajú určiť parametre ihly.

Meranie vzdialenosti je treba určiť kalibráciou. Kamera umiestnená v pevnej polohe na stojane nasníma kalibračnú snímku. Podľa kalibračnej snímky sa dá určiť vzdialenosť jedného pixlu od druhého. Touto vzdialenosťou sa už dajú namerať parametre ihly.

Podľa noriem je možné zadať parametre a tolerancie ihly. Následne algoritmom vyhodnotiť tieto parametre ihly (ako napríklad dĺžka ihly, parametre a ich tolerancie uška ihly, uhol hrotu ihly a.i.).

5 ZÁVER

Na začiatku práce sú opísané základné poznatky k spracovaniu obrazu a počítačového videnia. Tieto poznatky ujasňujú frázy 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 ďalšej kapitole je rozoberaná problematika snímania obrazu. Najvhodnejšou metódou pre túto úlohu je zadné osvetlenie. Pre túto metódu som vyhotovil pracovisko vhodné pre snímanie obrazu touto metódou. Kovový stojan som zvolil z dôvodu jeho stability. Pri manipulácii kamerou na stojane som potreboval, aby sa pozícia kamery nemenila a tak som mohol nasnímať vhodné snímky pre ďalšie spracovanie obrazu.

V kapitole „Riešenie stanovených chýb“ som opísal funkcie a algoritmy, ktoré som vytvoril v programovacom prostredí Matlab. Pre vyskúšanie aplikácie je potrebný program Matlab. Funkcie vyhodnotia kvalitu snímaných súčiastok. Jednotlivé funkcie riešia chyby určitého typu. Chyby som stanovil v kapitole 4.1. „Rozbor úlohy a stanovenie chýb“

Funkcia „search2“ je dôležitá pri usporiadaní obrazu ihly do štandardného tvaru pred predaním funkciám. Táto funkcia vyhledá krajné body ihly, podľa ktorých obraz môže rotovať.

Najviac času som potreboval na vyhotovenie vhodného pracoviska na meranie súčiastok. Potreboval som k tomu manuálnu zručnosť zvárania a navrhnutia konštrukcie stojana. Počas vyhotovenia konštrukcie som získal nové skúsenosti v oblasti kovoobrábania a počas písania programu som získal skúsenosti s riešením praktických problémov programovania v jazyku Matlab.

5.1 Súčasný stav

Vyhotovené je pracovisko pre snímanie ihl. Položením ihly na matnicu na stojane sú umiestnené ihly do vzorného pola kamery. Na stojane je viac polôh pre kameru. Držiak kamery dovoľuje pohyb kamerou. Po nasnímaní obrazu je treba snímok preniesť do počítača do adresára zdrojového kódu. Do Matlabovského prostredia je načítaný obraz pomocou funkcie „imread“, kde vstupným parametrom je názov 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 ihl 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 prahovaním je podľa môjho názoru pre tieto účely postačujúce. Algoritmy neobsahujú všetky chybové kontroly, preto je obmedzené množstvo snímok v databáze o správnych a chybných ihlách.

5.2 Návrh ďalšieho postupu

Pri ďalšom postupe plánujem rozšíriť databázu snímok. Ako ďalšou možnosťou rozšírenia programu chcem vytvoriť ďalšie algoritmy, ktoré riešia iné chyby ako som riešil v tejto práci ako napríklad: kontrola kvality poniklovania ihly. Na túto chybu je vhodná metóda snímania so zrkadlovým osvetlením.

V budúcnosti vyskúšam ďalšie metódy pri tvorbe čiernobieleho obrazu. Ďalej plánujem spresniť programovú časť, ktorá sa zaoberá umiestnením nasnímaného objektu do štandardného tvaru.

Plánujem optimalizovať súčasné funkcie a ochrániť algoritmy od väčšiny chybových stavov. Pre lepšiu orientáciu užívateľa chcem vytvoriť grafické rozhranie, kde bude možné vidieť nasnímaný obraz po výpočte algoritmom, prípadne chyby ihly.

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. Pre ďalšiu prácu by som rád vyhotovil maketu pásovej výroby, ktorá by mala simulovať deje pri výrobe a selektovaní súčiastok.

POUŽITÁ LITERATÚRA

- [1] JÄHNE, Bernd. PRACTICAL HANDBOOK ON IMAGE PROCESSING FOR SCIENTIFIC APPLICATIONS. Boca Raton: CRC Press, 1997, x, 589 s. ISBN 08-493-8906-2.
http://cs.wikipedia.org/wiki/Zorn%C3%A9_pole
- [2] PARKER, J. ALGORITHMS FOR IMAGE PROCESSING AND COMPUTER VISION. NEW YORK: WILEY COMPUTER PUBLISHING, 1997, XIII, 417 s. ISBN 04-711-4056-2.
- [3] *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
- [4] *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
- [5] *Multimediálny interaktívny 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
- [6] *Digital Image Processing* [online]. Posledná aktualizácia: 2006 [cit. 2013-12-31]. Dostupné na internete:
<http://dip.sccg.sk/>
- [7] *Wikipedia: Gaussian filter* [online]. Posledná aktualizácia: 14:06 26. 9. 2013 [cit. 2014-1-2]. Dostupné na internete:
http://en.wikipedia.org/wiki/File:Gaussian_Filter.svg
- [8] *MathWorks MatLab* [online]. Posledná aktualizácia: 2014 [cit. 2014-1-2]. Dostupné na internete:
<http://www.mathworks.com/products/matlab/index.html>
- [9] JIŘÍ, ŽÁRA A JIŘÍ ŽÁRA. MODERNÍ POČÍTAČOVÁ GRAFIKA. VYD 1. BRNO: COMPUTER PRESS, 2004, 609 s. ISBN 80-251-0454-0.
- [10] *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