





Sem vložte zadání Vaší práce.





ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA TEORETICKÉ INFORMATIKY



Bakalářská práce

## Zpracování obrazu pro měření mikrotvrdoti

*Aleš Saska*

Vedoucí práce: Ing. Martin Šlapák

5. května 2015



---

## Poděkování

Děkuji vedoucímu práce Ing. Martinu Šlapákovi za cenné rady, věcné připomínky, vstřícnost a možnost častých konzultací.



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 5. května 2015

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2015 Aleš Saska. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Saska, Aleš. *Zpracování obrazu pro měření mikrotvrdosti*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

---

## Abstrakt

Tato práce se zabývá problematikou rozpoznávání obrazu výstupních dat z mikrotvrdoměru. V úvodní části jsou rozebrány současné metody měření mikrotvrdomosti. Následuje popis navržených algoritmů pro detekci vpichu a určení jeho rozměru. První pracuje na principu analýzy histogramu a druhý využívá metody hranové detekce. Navržené metody jsou implementovány v jazyce C++ za použití knihovny OpenCV. V závěru práce jsou porovnány výsledky navržených metod s původním software dodávaným s mikrotvrdoměrem.

**Klíčová slova** zpracování obrazu, OpenCV, počítačové vidění, mikrotvrdomost

---

## Abstract

This thesis deals with an image recognition of data from microhardness. In the beginning there is a review of present methods for measuring microhardness followed by description of two designed algorithms. First is based on a histogram analysis and the second works with an edge detection. All algorithms are implemented in the C++ language with an OpenCV library. Finally there is a chapter about comparing results with output from original software.

**Keywords** image processing, OpenCV, computer vision, microhardness





---

# Obsah

Úvod	1
<b>1 Analýza a návrh</b>	<b>3</b>
1.1 Mikrotvrдость . . . . .	3
1.2 Způsob měření . . . . .	3
1.3 Definice problému . . . . .	5
1.4 Možné postupy pro vyřešení . . . . .	5
<b>2 Realizace</b>	<b>7</b>
2.1 Hledání prahů . . . . .	7
2.2 Hledání hran . . . . .	11
2.3 Sloučení výsledků z obou metod . . . . .	16
2.4 Výpočet mikrotvrđosti . . . . .	16
2.5 Dávkové zpracování dat . . . . .	16
<b>3 Měření kvality výstupu</b>	<b>17</b>
3.1 Porovnání dodávaného SW a implementovaného algoritmu . . . . .	17
<b>Závěr</b>	<b>23</b>
<b>A Seznam použitých zkratek</b>	<b>25</b>
<b>Literatura</b>	<b>27</b>
<b>B Obsah přiloženého CD</b>	<b>29</b>



---

## Seznam obrázků

1.1	Ukázka měřicího hrotu (převzato z [2]) . . . . .	4
1.2	Přístroj Qness Q10 (převzato z [3]) . . . . .	5
2.1	Celkový koncept programu . . . . .	7
2.2	Hodnoty jasu pro součet řádků předzpracovaného obrazu (k vidění lehce rozpoznatelný vpich) . . . . .	9
2.3	Hodnoty jasu pro součet sloupců předzpracovaného obrazu (k vidění lehce rozpoznatelný vpich) . . . . .	9
2.4	Hodnoty jasu pro součet sloupců předzpracovaného obrazu společně s obrazem vpichu . . . . .	10
2.5	Ukázka šíření algoritmu pro vyhledání středu vpichu (čáry ohraničují oblasti vybrané podle součtu jasů, zelená barva označuje prohledanou oblast a červený kříž ukazuje na nalezený střed vpichu)	11
2.6	Porovnání nepředzpracovaného a předzpracovaného obrazu . . . . .	12
2.7	Originální obrázek, obrázek po aplikaci dilatace a obrázek po aplikaci eroze (převzato z [8]) . . . . .	13
2.8	Dekompozice složení jádra Prewittova operátoru . . . . .	13
2.9	Parametrizace přímky pro Houghovu transformaci . . . . .	14
2.10	Počet průsečíků (křížků) před aplikací shlukování. Zdánlivě roztržené křížky jsou důsledkem mnoha průsečíků téměř v jednom bodě. . . . .	15
2.11	Počet průsečíků (křížků) po aplikaci shlukování. Ostře ohraničené křížky značí přítomnost pouze jednoho bodu. . . . .	15
3.1	Špatně rozpoznatelný vpich (neúspěch) – dodávaný software . . . . .	20
3.2	Špatně rozpoznatelný vpich (úspěch) – implementovaný algoritmus . . . . .	20
3.3	Špatně rozpoznatelný vpich 2 (neúspěch) – dodávaný software . . . . .	21
3.4	Špatně rozpoznatelný vpich 2 (úspěch) – implementovaný algoritmus . . . . .	21
3.5	Dobře rozpoznatelný vpich 2 (úspěch) – dodávaný software . . . . .	22

3.6	Dobře rozpoznatelný vpich 2 (úspěch, ale mírně nepřesný) – implementovaný algoritmus . . . . .	22
-----	--	----

---

# Úvod

Jednou ze základních mechanických vlastností konstrukčních materiálů je tvrdost. Z tohoto důvodu je velmi často měřena v technické praxi. Jelikož v současné době dochází k neustálému zmenšování součástí a nelze použít klasické metody měření, je stále častěji používána metoda měření mikrotvrdosti.

Znalost tvrdosti materiálu je potřebná pro výzkum nových materiálů. Uplatňuje se i u měření úspěšnosti kalení nebo legování oceli. Velkou důležitost má například v letectví, kde umožňuje konstrukci lehčích letadel, čímž významnou mírou přispívá k vyšší ekologičnosti celého odvětví.

Vývoj nových materiálů probíhá v komerční a z velké části i v akademické sféře (vysoké školy, výzkumné ústavy). Jednou z vysokých škol v naší republice, zabývajících se o tuto problematiku, je Univerzita Karlova. Na její katedře fyziky materiálů je přístroj Qness Q10 pro měření mikrotvrdosti materiálů. Její zjištění probíhá na základě vtlačení měřicího hrotu do materiálu, vyfocení deformace a její následné změření. Úspěšnost vyhodnocování obrázků u dodávaného SW je nedostatečná, vznikl tedy návrh pro vytvoření alternativního programu.

Tato práce se věnuje problematice rozpoznávání obrázků za použití grafické knihovny OpenCV. V první fázi je nejprve nadefinován pojem mikrotvrdost a popsány metody pro její měření. V další části je navržena implementace rozpoznání obrázku vyfoceného přístrojem následovaná popisem navržených algoritmů. V závěru je provedeno testování navržených algoritmů a zhodnocení jejich úspěšnosti.

**Cílem práce** je tedy vytvořit program, který na základě konfigurace experimentu a výstupních obrazových dat z mikrotvrdoměru určí mikrotvrdost materiálu v různých místech. U každého obrázku měření mikrotvrdosti budou změřeny rozměry vpichu v pixelech a následně přepočítány na reálné rozměry v mikrometrech. Z nich bude poté vypočtena mikrotvrdost měření.



---

# Analýza a návrh

## 1.1 Mikrotvrдость

Mikrotvrдость definujeme jako odolnost materiálu k deformaci. Existuje mnoho testů pro její měření. Tyto testy mohou být prováděny v mikroskopickém nebo makroskopickém měřítku.

**Makroskopické testy** jsou prováděny s větším zatížením (typicky 1 kg nebo více). Testy se liší podle tvaru použitého hrotu a podle zaměření (materiálu).

1. Test podle Vickerse (HV, viz. [1]), který má nejširší možnosti použití
2. Test podle Brinella (HB)
3. Test podle Knoop (HK), používá se na měření malých oblastí
4. Test podle Janka (HB), pro měření mikrotvrlosti dřeva
5. Test podle Meyera (HB), pro měření mikrotvrlosti dřeva

**Mikroskopické testy** jsou prováděny s malým celkovým zatížením (obvykle 0.2–1 kg) a produkují zářezy o velikosti 50  $\mu m$ .

1. Test podle Vickerse (HV)
2. Test podle Brinella (HB)

## 1.2 Způsob měření

Měření mikrotvrlosti spočívá v zatlačení hrotu do materiálu a změření velikosti stopy vpichu. K vytvoření vrypu se používá diamantový hrot (viz obr. 1.1) ve tvaru hranolu.

### 1.2.1 Vickers hardness test

Test podle Vickerse byl vynalezen v roce 1921 Robertem Smithem a Georgem Sandlandem ve firmě Vickers jako alternativa k testu podle Brinella. Je založen na principu zjištění odolnosti k tvarové deformaci. Tento test má nejširší možnosti použití a může být použit k měření všech kovů. Změřená jednotka mikrotvrdomosti je známá jako Vickers Pyramid Number (HV).



Obrázek 1.1: Ukázka měřícího hrotu (převzato z [2])

### 1.2.2 Přístroj Qness Q10

K analýze byly použity vzorky z přístroje Qness Q10. Přístroj je možno osadit více typy hrotů pro provedení různých typů testů. K dispozici byl pouze hrot pro test podle Vickerse. Proto se dále v této práci budeme zabývat pouze tímto testem.

Přístroj navíc podporuje více druhů testů:

1. Test podle Vickerse (HV)
2. Test podle Brinella (HB)
3. Test podle Knoop (HK)





Obrázek 1.2: Přístroj Qness Q10 (převzato z [3])

### 1.3 Definice problému

Vyzkoušením dodávaného programu byla zjištěna nedostatečná úspěšnost rozpoznání rozměrů vpichu. K naměřeným obrázkům z přístroje je tedy vytvořen alternativní SW, který je analyzuje a změří velikost deformované části. Následuje dosazení do vzorce a vypočítání mikrotvrlosti.

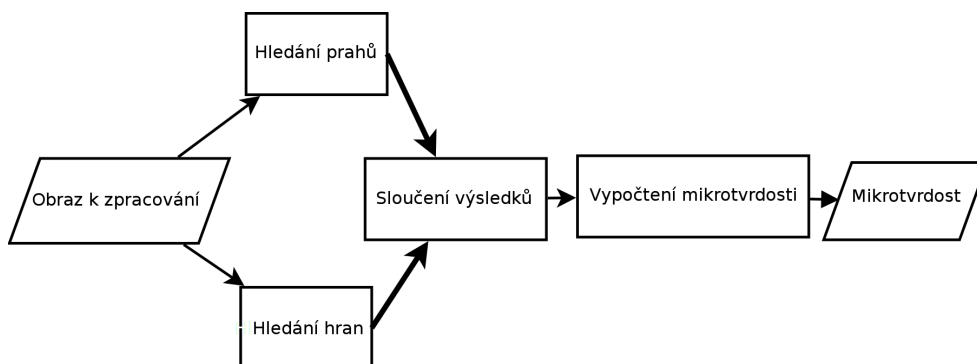
### 1.4 Možné postupy pro vyřešení

Před samotným zpracováním obrazu se ukázalo (viz kapitola 2.6) jako nejvhodnější provést odstranění nežádoucího pozadí (v této práci nazváno předzpracování). Poté následuje fáze zjištění velikosti vpichu. Nejprve byla vyzkoušena naivní metoda spočívající v možnosti zjištění velikosti hrotu zakládající se na analýze histogramu součtů hodnot jasů pro řádky a sloupce (viz obr. 2.4). Tento postup je nazván „Hledání prahů“ (viz kapitola 2.1). Jelikož hrot má tvar kosočtverce s ostře ohraničenými hranami, jako další možný způsob identifikace vpichu byla vyzkoušena hranová detekce (nazvána „Hledání hran“ 2.2).



## Realizace

Zpracování obrazu lze rozdělit do 4 fází (viz obr. 2.1). V úvodní se provede analýza obrazu pomocí „Hranové detekce“ (viz kapitola 2.2) a „Detekce prahů“ (viz kapitola 2.1). Návratovou hodnotou těchto funkcí je seznam možných výsledků, které jsou dále statisticky zpracovány v části „Sloučení výsledků“ (viz kapitola 2.3). V poslední fázi proběhne vypočítání (viz kapitola 2.4) mikrotvrlosti podle změřených rozměrů vpichu.



Obrázek 2.1: Celkový koncept programu

### 2.1 Hledání prahů

#### 2.1.1 Předzpracování obrazu

Tato část je realizována ve funkci *preprocess()*. Základní myšlenka je, že největší část obrázku zabírá pozadí. Z tohoto důvodu lze aplikovat selekci pozadí pomocí histogramu.

Nejdříve je vypočítán histogram barev (odstínů šedi) pomocí procházení všech všech pixelů. Dále je tento histogram vyhlazen tak, že je hodnota nahrazena průměrem z okolních 4 hodnot. Ve výsledném vyhlazeném histogramu je

následně nalezeno maximum (odstín šedi s největším zastoupením v obrázku) a tento odstín je označen za pozadí. V následném kroku jsou z obrazu vyběleny všechny pixely s barvou větší (hodnota jasu u pixelu je větší = světlejší) než tato barva pozadí.

Jelikož tato funkce nedokázala vyhladit všechny šum pozadí a v obrazu zanechávala malé fragmenty, bylo nutno následně přidat odstranění malých souvislých oblastí. V funkci *removeSmallContinuousAreas()* se tedy pomocí BFS vyhledávání <sup>1</sup> najdou všechny souvislé oblasti a následně se odstraní ty, které jsou menší než parametr *maxareasize*.

**Algoritmus vyhledávání souvislých oblastí** pracuje na principu prohledávání obrazu. Nejprve si vytvoří pomocné pole o stejné velikosti, jako má obraz, a všechny hodnoty nastaví na hodnotu nenavštíveno. Poté postupně prochází všechny body obrazu a pokud má v pomocném poli hodnotu nenavštíveno a v obrazu je jiná barva než bílá (barva pozadí), spustí prohledávání do šířky (BFS). Toto prohledávání prochází všechny body, které nejsou bílé a jsou nenavštívené. Na jeho konci jsou všechny hodnoty, které navštívilo, označeny za souvislou oblast. Pokud hlavní cyklus algoritmu narazí na nenavštívený pixel obrazu s barvou pozadí, pouze změní jeho status na hodnotu navštíveno. Takto algoritmus pokračuje přes všechny pixely obrazu.

### 2.1.2 Vyhledání vpichu

Tato část je provedena ve funkci *parse()*. Skládá se z několika dílčích částí:

**Vyhledání prahů** Jako první je potřeba provést vyhledání prahů ve funkci *findXYLimits()*. Nejdříve jsou vypočítány prahové hodnoty jasu pro řádky a sloupce. Výpočet je založen na myšlence, že největší část obrázku je pozadí. Rovnice pro výpočet prahu pro pozadí je

$$BGthreshold(y) = median(SumH(y)) + \frac{\sum_{x=1}^x \sum_{y=1}^y i(x, y)}{x * y} + thresholdoffset.$$

První část vzorce

$$median(SumH(y))$$

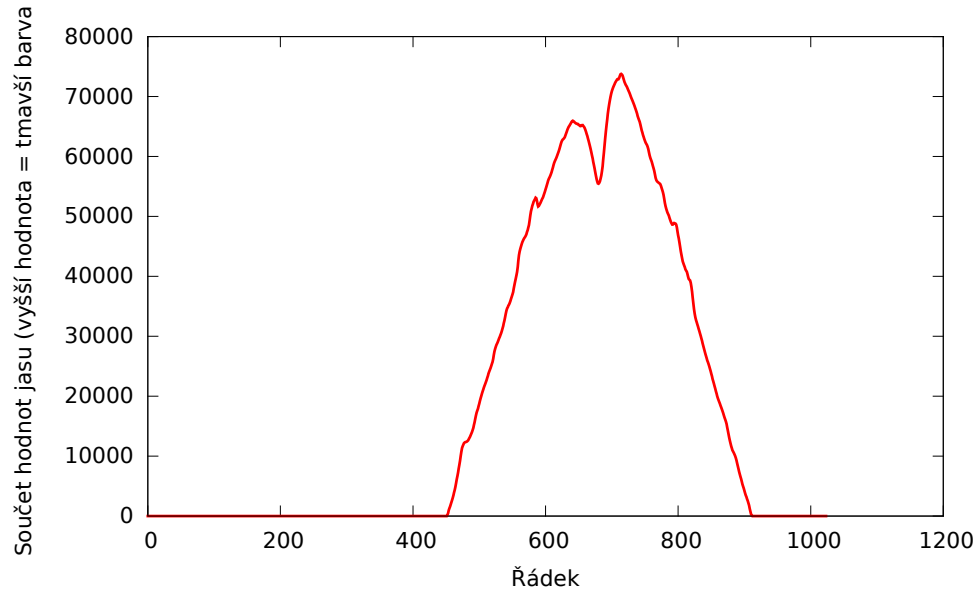
vyjadřuje medián ze součtu jasů pro sloupce, druhá celkový aritmetický průměr jasu všech hodnot pixelů a třetí je parametr *thresholdoffset*. Poté jsou proiterovány všechny sloupce a oblasti s jasnem vyšším než tento jsou přidány do výsledku pro sloupce. Výpočet jasu pro součet pro řádky je proveden podle vzorce:

$$SumH(y) = \sum_{x=1}^x i(x, y)$$

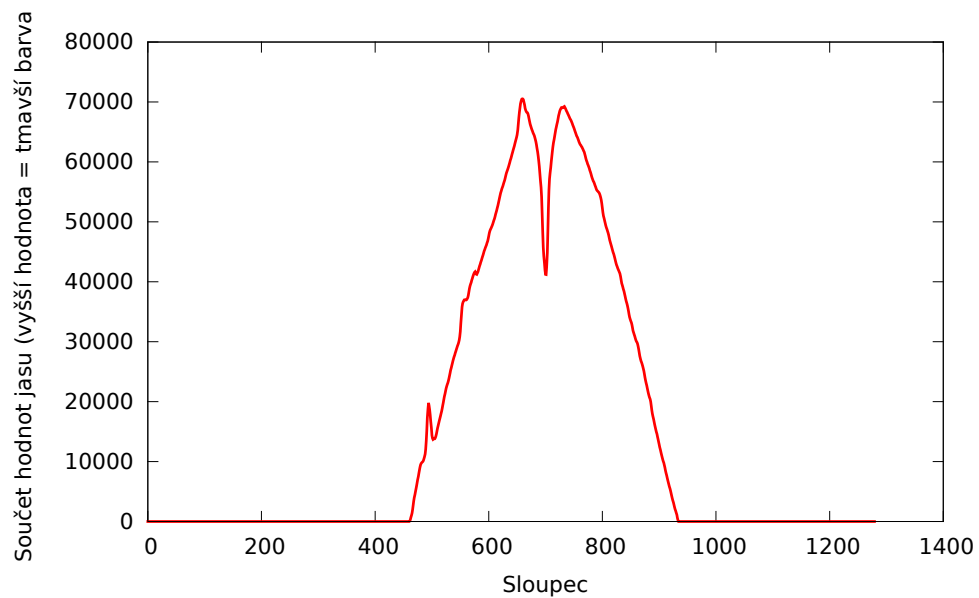
---

<sup>1</sup>Breadth-first search [4] (prohledávání do šířky) je algoritmus prohledávání stavového prostoru.

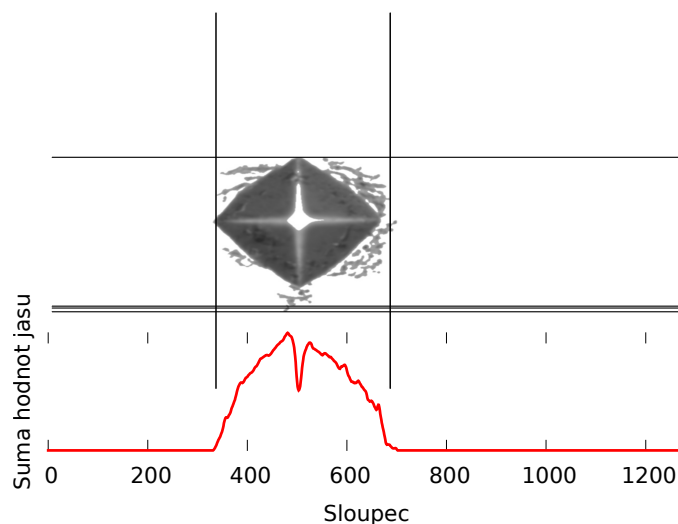
$SumH(y)$  je vektor se součty jednotlivých řádků,  $X$  je počet sloupců v obraze a  $i(x, y)$  je obraz. Analogicky je proveden celý postup  $i$  na řádky.



Obrázek 2.2: Hodnoty jasu pro součet řádků předzpracovaného obrazu (k vidění lehce rozpoznatelný vpich)



Obrázek 2.3: Hodnoty jasu pro součet sloupců předzpracovaného obrazu (k vidění lehce rozpoznatelný vpich)

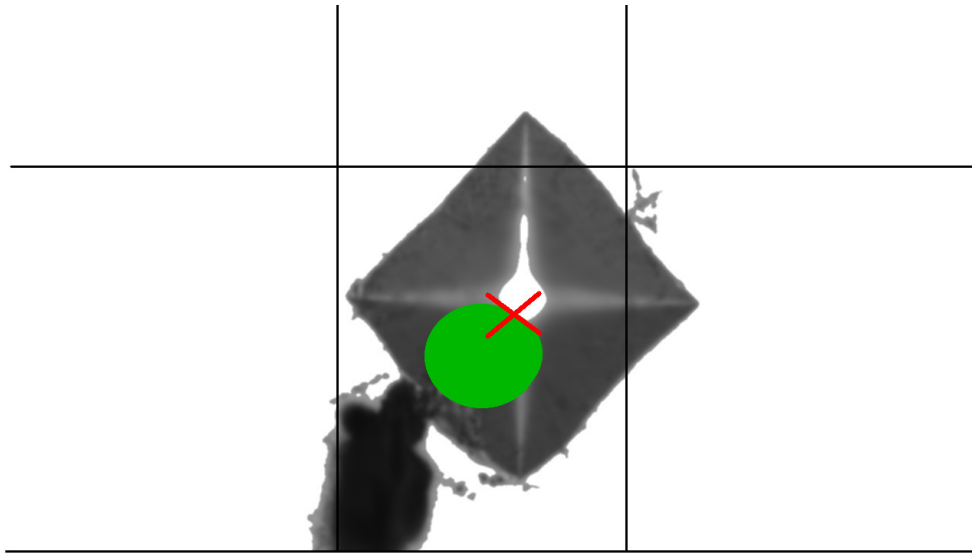


Obrázek 2.4: Hodnoty jasu pro součet sloupců předzpracovaného obrazu společně s obrazem vpichu

**Vyhledání středu vpichu** Výstupem předchozí funkce je seznam oblastí pro sloupce a pro řádky. Kombinací všech oblastí pro sloupce se všemi oblastmi pro řádky jsou možné čtverce reprezentující oblast vpichu, pro něž je zkoušeno nalezení vpichu. Z výsledku jsou nejprve odstraněny malé (oblasti s plochou menší než  $30 \times 30$  px) a nečtvercové záznamy (poměr výšky a šířky je větší než 1,2). V dalším kroku se provádí samotné vyhledání vpichu. Toto vyhledávání je založeno na BFS prohledávání. Začne se ve středu (geometrický střed čtverce) plochy ohraničené vypočítanými prahy. Zde je spuštěno prohledávání, které se zastaví na prvním bílém bodě a ten označí jako střed vpichu (viz obr. 2.5). Poté se pomocí druhého vyhledávání označí celá oblast s bílou barvou. Následně je zde spuštěno třetí BFS vyhledávání, které postupuje od krajů této bílé oblasti a zjistí její potenciální středy.

**Zjištění rozměrů vpichu** Na všechny tyto potenciální středy je aplikována funkce *findDimensionsForCenter()*, která zjistí velikost vpichu od těchto středů. Zjištění velikosti je provedeno na všechny 4 strany (nahoru, dolů, doleva, doprava) pomocí vyhledání prvního přechodu tmavá/bílá. Parametry (šířka, výška, pozice středu) každého takto vypočteného výsledku jsou následně uloženy pro pozdější zpracování.

**Vyhledání nejlepšího výsledku** je realizováno funkcí *findBestResultIdx()*. Uvnitř se na každý možný výsledek zavolá funkce *compareWithRec()* vrace-



Obrázek 2.5: Ukázka šíření algoritmu pro vyhledání středu vpichu (čáry ohraničují oblasti vybrané podle součtu jasů, zelená barva označuje prohledanou oblast a červený kříž ukazuje na nalezený střed vpichu)

jící SAD <sup>2</sup> s šablonou vpichu. Jako návratová hodnota celé funkce je index výsledku s nejmenším SAD. Návratová hodnota této funkce je pozice nejlepšího výsledku v poli všech možných kandidátů na výsledek.

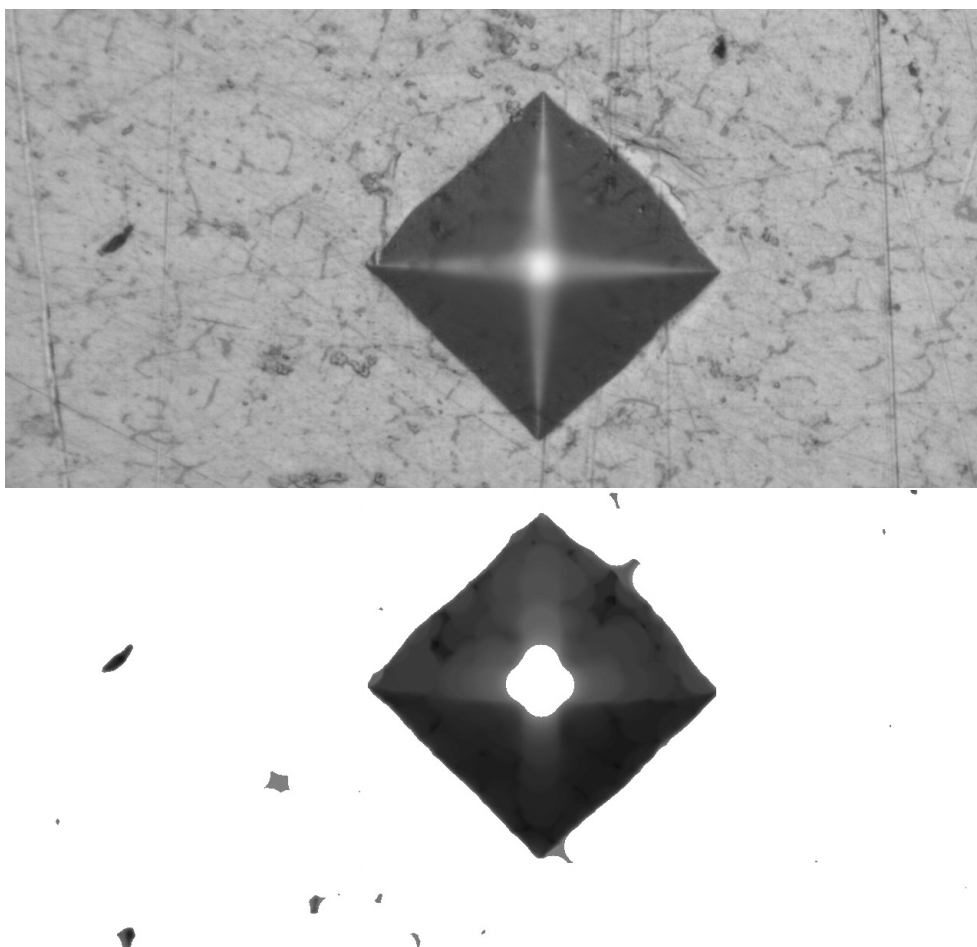
## 2.2 Hledání hran

Pro druhý algoritmus bylo použito hledání hran v obrazu u něhož se využívá kosočtvercového tvaru vpichu. Tato metoda byla rozdělena na tři části: předzpracování, vyhledání hran a následné vyhledání vpichu.

### 2.2.1 Předzpracování obrazu

Na rozdíl od předzpracování v metodě Hledání prahů jsou zde využity pouze funkce OpenCV knihovny. Tato část kódu je prováděna ve funkci *preprocessCV()*. V první fázi je provedeno rozostření obrazu pomocí funkce *blur*. Následuje použití morfologické operace *opening*. V ní se odstraní malé objekty z obrazu. Poté je zavoláním *normalize()* modifikován histogram obrazu (nejsvětlejšímu místu je přiřazena bílá a nejtmavějšímu černá). Následuje už pouze oříznutí nejsvětlejší části na bílou pomocí funkce *threshold*. Tímto sledem operací zajistíme, že nám zůstane tmavý vpich na bílém pozadí.

<sup>2</sup>SAD — sum of absolute differences (suma absolutních rozdílů) je algoritmus pro měření podobnosti dvou obrazů.



Obrázek 2.6: Porovnání nepředzpracovaného a předzpracovaného obrazu

**Opening transformace** [5] je druh morfologické operace [6] používané pro odstranění malých objektů (světlých na tmavém pozadí). Skládá se z eroze následované dilatací obrazu. Vzorec pro opening transformaci tedy je:

$$dst = open(image) = dilate(erode(image))$$

1. Dilatace [10] se skládá z konvoluce obrazu s jádrem matice, v níž mají nenulové prvky obvykle tvar čtverce nebo kruhu. V průběhu konvoluce obrazu s maticí je vypočítána maximální hodnota pixelu a pixel obrazu nahrazen touto maticí. Dilatace tedy způsobí zvětšení světlých oblastí.
2. Naproti tomu eroze [9] je používaná k vypočtení lokálního minima skrz oblast jádra matice operátoru. Výsledek této operace je tedy zvětšení tmavých oblastí v obrazu.





Obrázek 2.7: Originální obrázek, obrázek po aplikaci dilatace a obrázek po aplikaci eroze (převzato z [8])

### 2.2.2 Vyhledání vpichu

Po odstranění nežádoucího pozadí nastupuje fáze samotného zjištění velikosti vpichu.

**Hranová detekce** Pro detekci hran byla použita aplikace X složky Prewittova operátoru (viz [11]). Matematicky tento operátor používá dvě  $3 \times 3$  jádra, která jsou aplikována na původní obrázek pomocí konvoluce. Výsledek poté vyjadřuje změny obrazu v horizontálním či vertikálním směru.

$$\begin{vmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{vmatrix} = \begin{vmatrix} +1 \\ +1 \\ +1 \end{vmatrix} \begin{vmatrix} -1 & 0 & +1 \end{vmatrix}$$

Obrázek 2.8: Dekompozice složení jádra Prewittova operátoru

Obraz s aplikovanou detekcí je poté naškálován na stupnici šedi od 0 do 255.

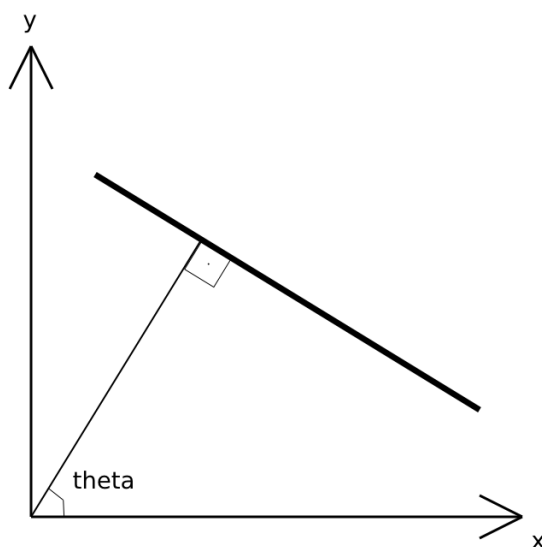
**Houghova transformace [12]** Po zjištění hran v obrazu je potřeba provést samotnou hranovou detekci. Proto byla zvolena Houghova transformace.

Výsledkem této transformace je seznam přímek, z něhož se vypočítají průsečíky všech navzájem kolmých přímek (tento výpočet je proveden ve funkci *getOrthogonalIntersections()*). Jelikož hrany ve zdetekovaném obrázku nejsou ideálně rovné, Houghova transformace obvykle nalezne více téměř rovnoběžných přímek. Po vypočítání jejich průsečíků získáváme shluky bodů, které jsou od sebe minimálně vzdálené. Velké množství bodů by se negativně projevilo na časové složitosti další části algoritmu. Proto se na seznam bodů aplikuje shluková analýza pomocí funkce *removeGroups()*.

**Houghova transformace** se používá pro detekci parametricky popsateľných elementů (přímek, kružnic atd.) v obrazu. Je založena na principu převodu obrazu z kartézských na polární souřadnice. Přímku v rovině můžeme popsat několika způsoby. Jako model přímky lze použít rovnici

$$x \cos(\phi) + y \sin(\phi) = r$$

pro kterou platí:  $r$  je délka normály od přímky k počátku souřadnic,  $theta$  je úhel mezi normálou a osou  $x$  (viz obr. 2.9). Jestliže se do předchozí rovnice dosadí souřadnice boxu  $[x,y]$ , pak množina všech řešení  $[r,theta]$  vytvoří v Houghově prostoru spojitou křivku. Pokud se tímto způsobem promítnou všechny body ležící na nějaké přímce, ukáže se, že se křivky odpovídající jednotlivým bodům přímky protnou v bodě  $[r_{max}, theta_{max}]$  (dvojice těchto parametrů jsou hledané body přímky).

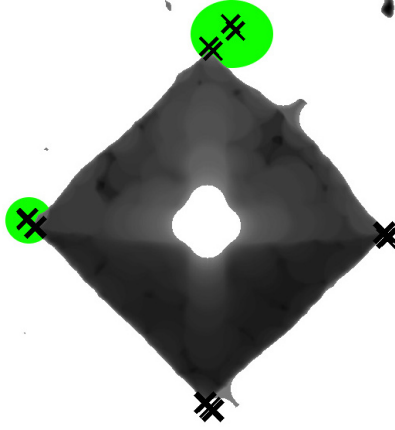


Obrázek 2.9: Parametrizace přímky pro Houghovu transformaci

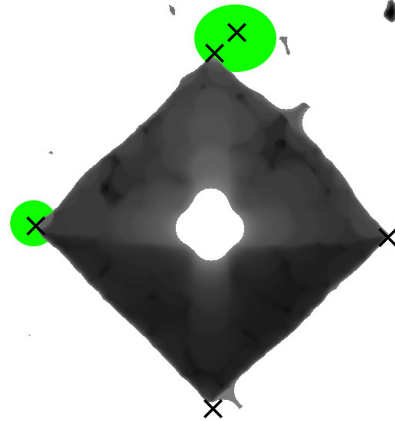
**Shlukování blízkých průsečíků** Smyslem algoritmu je nahradit blízké body jedním reprezentantem. Z počátku označíme všechny body jako neshlukované. V algoritmu se poté vezme jeden neshlukovaný bod a vytvoří z něj shluk o počtu 1. Poté se postupně prochází ostatní neshlukované body. Pokud je procházený bod vzdálen méně než maximální vzdálenost pro shlukování (parametr funkce), je přidán do shluku. Následně se upraví střed shluku: souřadnice  $x$  a  $y$  středu jsou vypočítány jako vážený průměr:

$$\frac{\text{souřadniceShluku} * \text{početBodůVeShluku} + \text{souřadniceBodu}}{\text{početBodůVeShluku} + 1}.$$

Jelikož se nám posunul střed, musí se celý shluk zkusit analogicky spojit s ostatními. Takto se pokračuje dokud nejsou všechny body ve shluku. Složitost uvedeného algoritmu je  $O(n^2)$ . Viz ukázka obrázků před 2.10 a po 2.11 zpracování touto funkcí.



Obrázek 2.10: Počet průsečíků (křížků) před aplikací shlukování. Zdánlivě roztržené křížky jsou důsledkem mnoha průsečíků téměř v jednom bodě.



Obrázek 2.11: Počet průsečíků (křížků) po aplikaci shlukování. Ostře ohraničené křížky značí přítomnost pouze jednoho bodu.

**Vyhledání rohů** je potřebné pro určení finální velikosti vpichu. Je zde provedeno otestování všech možných kombinací pomocí 4krát do sebe vnořeného procházení průsečíků. Teoretická časová složitost tohoto algoritmu je sice  $O(n^4)$ , avšak z důvodu předchozího shlukování a implementované filtrace nevyhovujících bodů je reálná složitost akceptovatelná. Každý takto zjištěný výsledek je přidán do seznamu možných výsledků.

**Vyhledání nejlepšího výsledku** Tato část je realizovaná pomocí stejné funkce jako v metodě Hledání podle prahů v kapitole 2.1.2.

### 2.3 Sloučení výsledků z obou metod

Z množiny všech výsledků je následně statisticky vypočítán výsledný rozměr vpichu v souboru `stat.cpp` pomocí funkce `findAndDecideBestStat()`. Nejprve jsou ze seznamu odstraněny všechny málo pravděpodobné výsledky. To je provedeno ve dvou fázích. Nejdříve se v `filterResultsByDistanceFromCenter()` změří aritmetický průměr všech středů a u každého záznamu se vypočítá vzdálenost od tohoto průměrného středu. Následně se ze seznamu odstraní nejvzdálenějších 25 % záznamů. Poté se v druhé fázi odstraní 12,5 % nejmenších a 12,5 % největších výsledků. Výsledek této fáze je aritmetický průměr všech profiltrovaných výsledků.

### 2.4 Výpočet mikrotvrlosti

V souboru `hardness.cpp` v funkci `computeHardness()` je poté provedeno vypočtení mikrotvrlosti metodou podle Vickerse:

$$HV \approx \frac{0.1891F}{d^2}$$

Pro dosažení do tohoto vzorce je potřeba vypočítat rozměry vpichu v jednotkách SI. To zajistíme změřením velikosti pixelu u všech typů objektivů (objektiv 40× a 65×). V případě používaného přístroje je rozlišení při objektivu 40×  $0.1277 \mu\text{m}/\text{px}$  a při objektivu 65×  $0.0840 \mu\text{m}/\text{px}$ .

### 2.5 Dávkové zpracování dat

Pro účely zpracování testovacích dat byl vytvořen program `test`. Na vstupu program načte a rozparsuje výstup z originálního SW, provede zavolání implementovaného algoritmu (program `main`) a porovná oba výsledky. Nakonec na výstup vypíše mikrotvrdot vypočítanou originálním SW, algoritmem z této práce a odchylku výsledků od sebe.

## Měření kvality výstupu

Na testovacích datech přiložených na CD bylo následně provedeno testování kvality výstupu implementovaného algoritmu. Pomocí programu test se ukázalo, že pravděpodobnost identifikace vpichu je lepší než u dodávaného SW.

### 3.1 Porovnání dodávaného SW a implementovaného algoritmu

Následuje ukázka ze sady testovacích dat. Výsledná data jsou záznam z více měření jednoho materiálu.

Tabulka 3.1: Porovnání výsledné mikrotvrlosti na testovacích datech DR6-TD s ručně otagovanými hodnotami a s výsledky z dodávaného SW

Obrázek	Ručně	Tato práce	Orig. SW
2_2_2015__11_10__42_64.JPG	93.7	102.7 (109.61 %)	89.7 (95.73 %)
2_2_2015__11_7__3_648.JPG	89.3	64.2 (71.89 %)	—
2_2_2015__11_7__20_137.JPG	94.6	68.9 (72.83 %)	—
2_2_2015__11_7__45_627.JPG	92.8	90.6 (97.63 %)	—
2_2_2015__11_8__1_945.JPG	99	102.7 (103.74 %)	—
2_2_2015__10_51__42_446.JPG	95.5	75.7 (79.27 %)	11.4 (11.94 %)
2_2_2015__11_8__31_663.JPG	96.6	90 (93.17 %)	—
2_2_2015__11_9__1_615.JPG	96.1	118.7 (123.52 %)	—
2_2_2015__11_10__59_957.JPG	96.5	91.6 (94.92 %)	—
2_2_2015__11_11__15_339.JPG	96.5	175.4 (181.76 %)	94.1 (97.51 %)
2_2_2015__11_11__38_551.JPG	97	79.2 (81.65 %)	—
2_2_2015__11_11__57_240.JPG	94.6	92.8 (98.10 %)	—
30_1_2015__15_50__10_759.JPG	96.9	98.7 (101.86 %)	96.9 (100.00 %)
2_2_2015__11_13__20_420.JPG	92.8	93 (100.22 %)	—
2_2_2015__11_13__29_998.JPG	96.1	89.3 (92.92 %)	94.6 (98.44 %)

### 3. MĚŘENÍ KVALITY VÝSTUPU

2_2_2015__11_13__44_54.JPG	91	125 (137.36 %)	—
2_2_2015__11_13__53_554.JPG	95.5	93.6 (98.01 %)	95 (99.48 %)
2_2_2015__11_14__23_803.JPG	94.4	79.6 (84.32 %)	94.4 (100.00 %)
30_1_2015__15_52__15_341.JPG	94.4	95.8 (101.48 %)	—
2_2_2015__11_14__44_285.JPG	97.5	111.5 (114.36 %)	—
2_2_2015__11_15__38_620.JPG	99.4	101.6 (102.21 %)	—
2_2_2015__11_16__31_754.JPG	102	102.1 (100.10 %)	100 (98.04 %)
30_1_2015__15_53__27_164.JPG	91.5	322.4 (352.35 %)	91.5 (100.00 %)
2_2_2015__11_16__55_92.JPG	96.5	96.3 (99.79 %)	—
2_2_2015__11_17__14_685.JPG	97.5	97.6 (100.10 %)	—
2_2_2015__11_18__17_803.JPG	98.5	92.5 (93.91 %)	—
2_2_2015__11_18__42_170.JPG	93.2	93.2 (100.00 %)	—
2_2_2015__11_19__12_575.JPG	96.1	76.1 (79.19 %)	—
30_1_2015__15_55__8_2.JPG	95.7	100.9 (105.43 %)	95.7 (100.00 %)
2_2_2015__11_19__52_916.JPG	99.4	77.7 (78.17 %)	—
2_2_2015__11_20__14_569.JPG	97.5	95.8 (98.26 %)	—
2_2_2015__11_20__41_354.JPG	94.6	275 (290.70 %)	—
2_2_2015__11_21__11_478.JPG	97.5	118 (121.03 %)	—
2_2_2015__11_21__34_410.JPG	97.9	96.9 (98.98 %)	96.9 (98.98 %)
30_1_2015__15_57__32_178.JPG	95	309 (325.26 %)	95 (100.00 %)
2_2_2015__11_22__1_258.JPG	94.2	78.7 (83.55 %)	—
30_1_2015__15_58__5_905.JPG	96	96.4 (100.42 %)	96 (100.00 %)
2_2_2015__11_22__32_645.JPG	97.9	102.2 (104.39 %)	—
2_2_2015__11_22__54_563.JPG	94.2	68.3 (72.51 %)	—
2_2_2015__11_23__19_570.JPG	98.5	102.1 (103.65 %)	—
30_1_2015__15_59__18_788.JPG	98.9	93.7 (94.74 %)	98.9 (100.00 %)
2_2_2015__11_24__52_577.JPG	96.1	78.4 (81.58 %)	—
2_2_2015__11_25__10_923.JPG	96.5	82 (84.97 %)	—
30_1_2015__16_1__4_416.JPG	92.7	94.3 (101.73 %)	92.7 (100.00 %)
2_2_2015__11_25__50_812.JPG	96.5	348 (360.62 %)	—

Z výše uvedené tabulky je vidět nespolehlivost originálního SW, ale jeho přesnost (pokud rozpozná výsledek, rozpozná ho s vysokou přesností). Originální SW měl problémy s rozpoznáním, zatímco algoritmus navržený v této práci rozpoznal vpich ve většině případů.

Tabulka 3.2: Souhrnné výsledky z testování na datasetu DR6-TD

Název	Tato práce	Orig. SW
průměrné % hodnoty etalonu	119.38 %	93.34 %
standartní odchylka (% hodnoty etalonu)	61.11 %	23.33 %
úspěšnost měření	45/45	15/45

### 3.1. Porovnání dodávaného SW a implementovaného algoritmu

---

Tabulka 3.3: Souhrnné výsledky z testování na datasetu DR6-RD

Název	Tato práce	Orig. SW
průměrné % hodnoty etalonu	83.56 %	98.65 %
standartní odchylka (% hodnoty etalonu)	18.92 %	2.64 %
úspěšnost měření	45/45	19/45

Tabulka 3.4: Souhrnné výsledky z testování na datasetu DAR6-TD

Název	Tato práce	Orig. SW
průměrné % hodnoty etalonu	83.24 %	98.67 %
standartní odchylka (% hodnoty etalonu)	13.02 %	3.98 %
úspěšnost měření	43/43	23/43

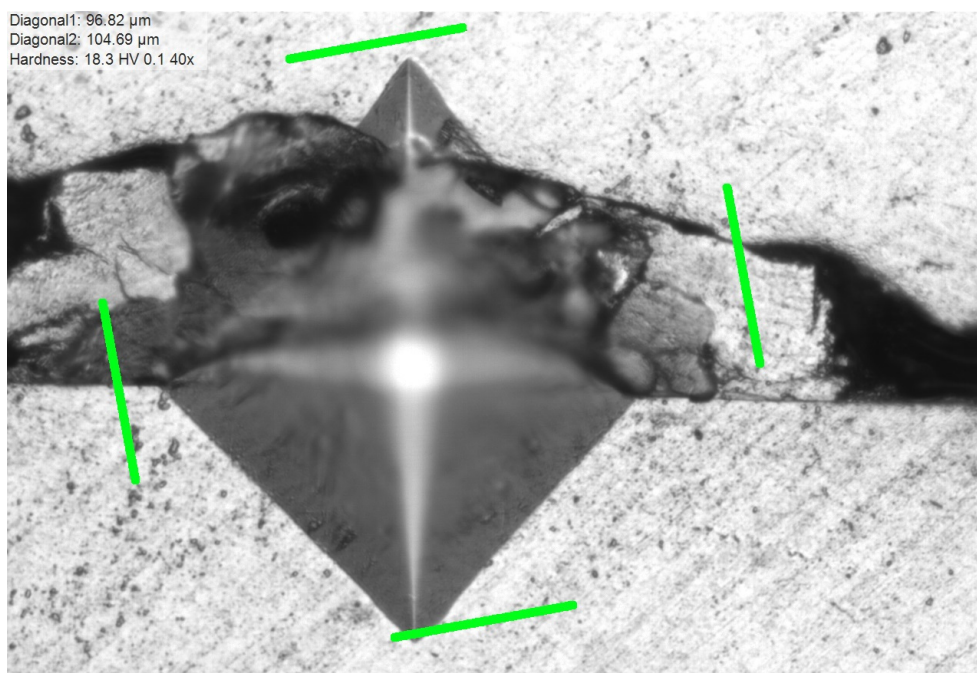
Tabulka 3.5: Souhrnné výsledky z testování na datasetu DAR6-RD

Název	Tato práce	Orig. SW
průměrné % hodnoty etalonu	85.78 %	99.82 %
standartní odchylka (% hodnoty etalonu)	9.45 %	0.62 %
úspěšnost měření	44/44	21/45

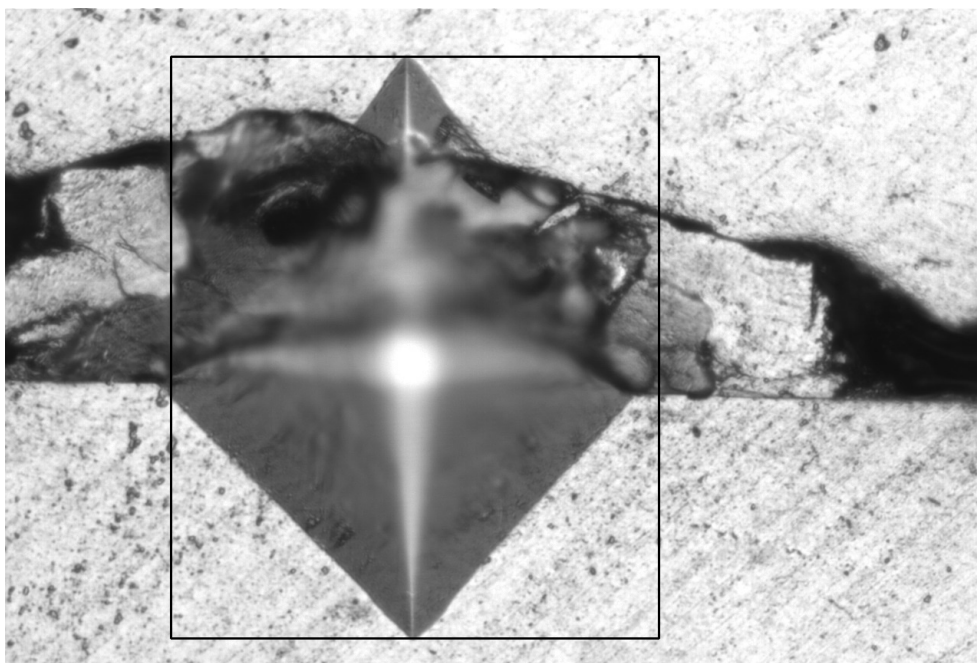
Testování na zbývajících datasetech ověřilo předchozí zjištění, že implementovaný algoritmus má velkou pravděpodobnost změření velikosti, ale jeho přesnost je oproti dodávanému sw menší.

### 3. MĚŘENÍ KVALITY VÝSTUPU

---



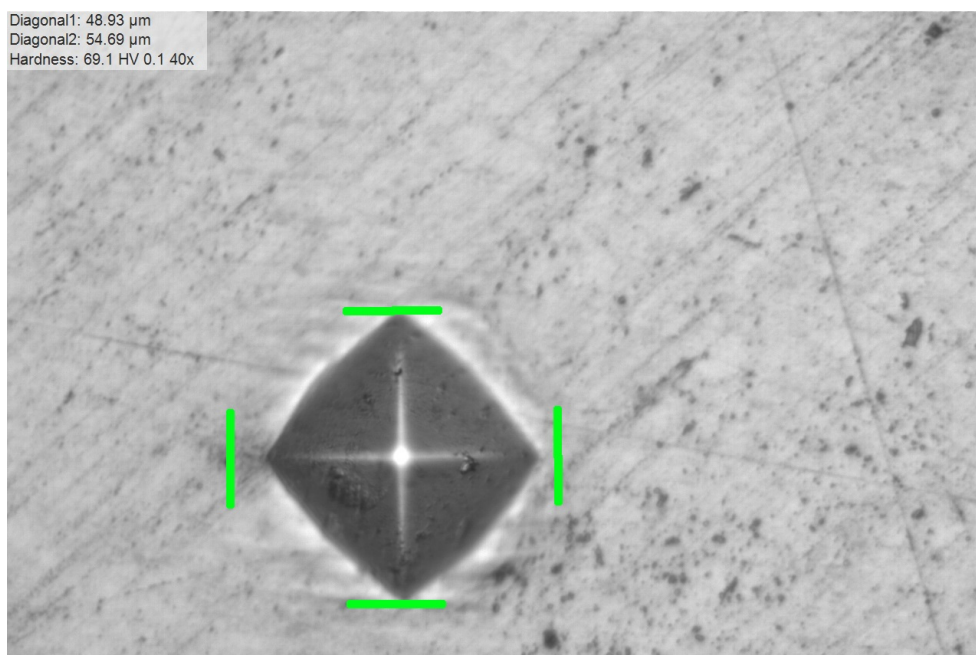
Obrázek 3.1: Špatně rozpoznatelný vpich (neúspěch) – dodávaný software



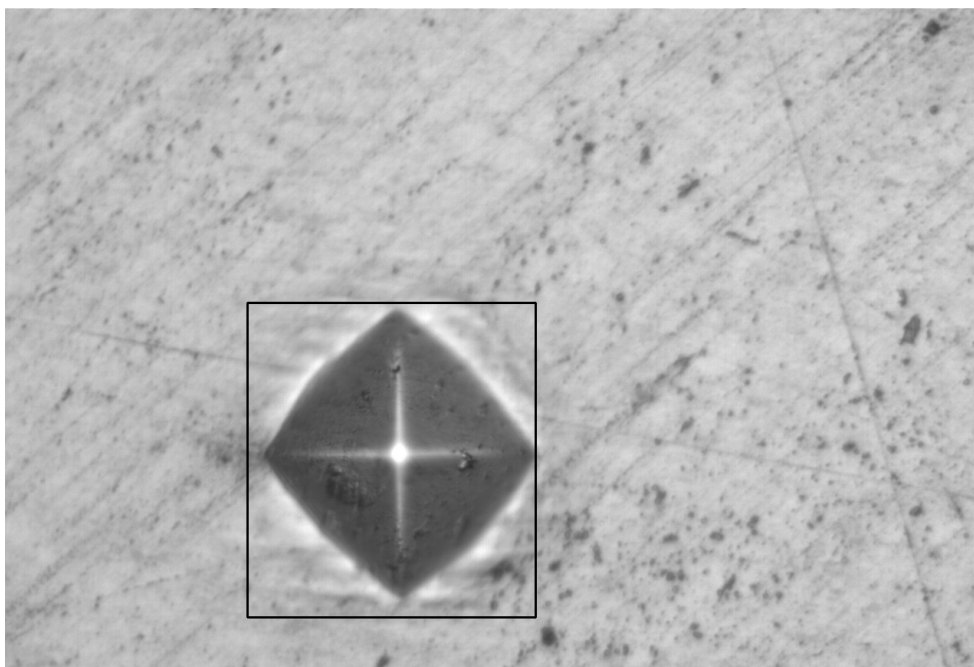
Obrázek 3.2: Špatně rozpoznatelný vpich (úspěch) – implementovaný algoritmus



### 3.1. Porovnání dodávaného SW a implementovaného algoritmu



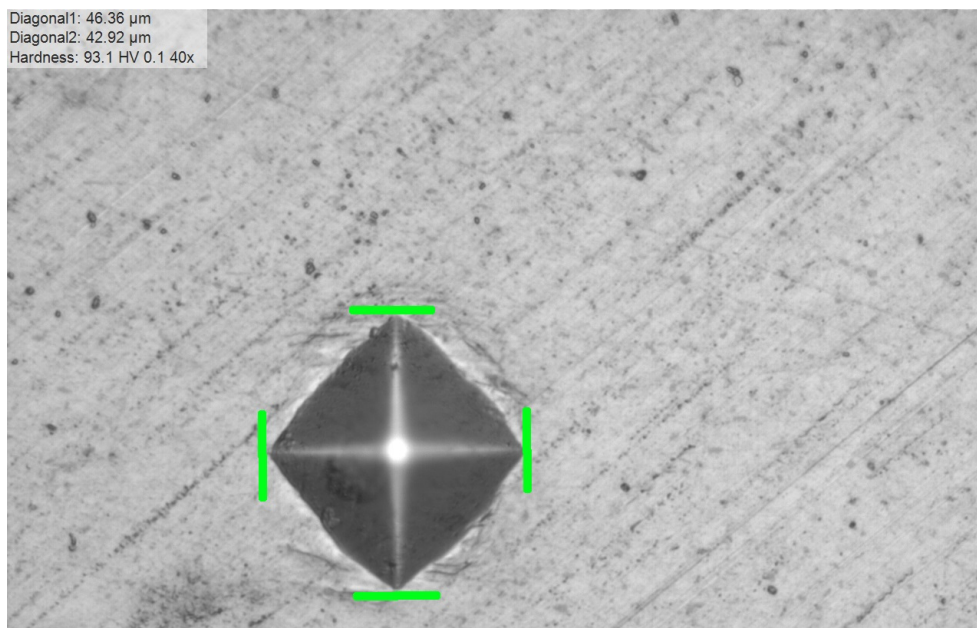
Obrázek 3.3: Špatně rozpoznatelný vpich 2 (neúspěch) – dodávaný software



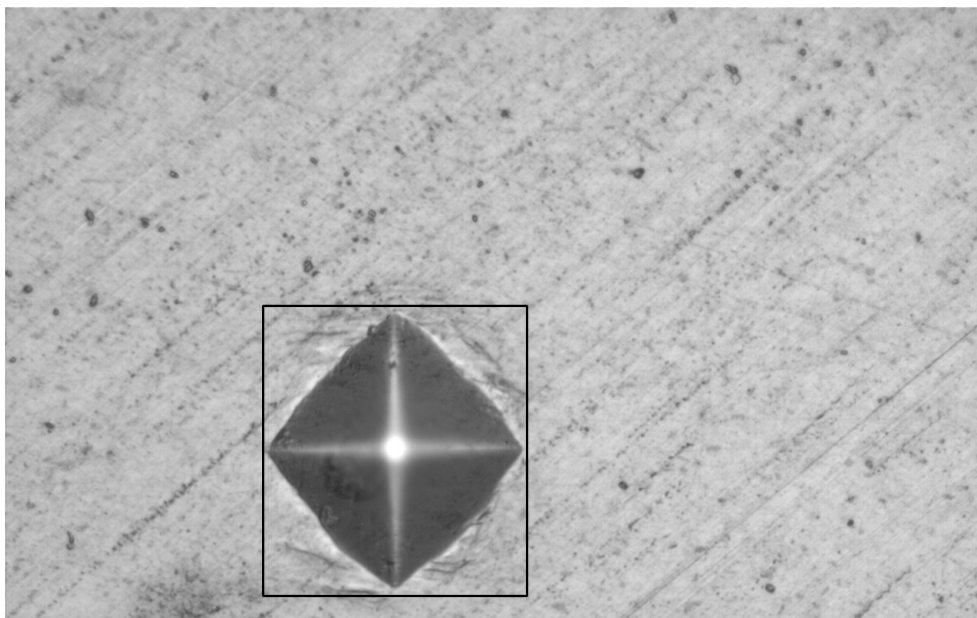
Obrázek 3.4: Špatně rozpoznatelný vpich 2 (úspěch) – implementovaný algoritmus

### 3. MĚŘENÍ KVALITY VÝSTUPU

---



Obrázek 3.5: Dobře rozpoznatelný vpich 2 (úspěch) – dodávaný software



Obrázek 3.6: Dobře rozpoznatelný vpich 2 (úspěch, ale mírně nepřesný) – implementovaný algoritmus

---

## Závěr

Předložená práce se zabývá problematikou rozpoznávání obrazu. V úvodu práce je provedeno definování problému měření mikrotvrlosti a vytyčení cílů.

V práci se podařilo vytvořit dva funkční algoritmy pro rozpoznání vpichu, jeho změření, a následně bylo provedeno vypočítání mikrotvrlosti. Vyvinuté algoritmy jsou založeny na deterministickém přístupu a pro rozpoznávání využívají charakteristických vlastností vpichu. Tyto algoritmy byly úspěšně implementovány v jazyce C++ za použití knihovny OpenCV.

Navržené algoritmy byly podrobeny testování na vzorku vstupních dat. Vytyčený cíl úspěšnosti měření byl splněn a v mnoha ohledech překročen. Podařilo se dosáhnout velké úspěšnosti změření (na testovacích datech 100 %) u špatně rozpoznatelných obrázků, zatímco úspěšnost originálního sw je pouze 44 %. Možný potenciál ve zlepšení vidím u přesnosti vypočítané mikrotvrlosti lehce rozpoznatelných obrázků. Zde se standardní odchylka liší o cca 30 % (40 % vs 10 %).

Všechny cíle vytyčené v zadání práce se podařilo naplnit. Přesto je v navržených algoritmech stále značný potenciál pro optimalizace s ohledem na přesnost a robustnost detekce. Věřím, že tato práce poslouží všem případným zájemcům o tuto problematiku a v jejím lepším pochopení.



## Seznam použitých zkratk

**BFS** Breadth-first search

**SAD** Sum of absolute differences

**XML** Extensible markup language

**SW** Software



---

# Literatura

- [1] R.L. Smith & G.E. Sandland, *An Accurate Method of Determining the Hardness of Metals, with Particular Reference to Those of a High Degree of Hardness* 1922
- [2] Vickers anvil diamonds.jpg strana naposledy edit. 2012-03-24 [cit. 2015-04-01] Dostupný z WWW: [http://en.wikipedia.org/wiki/File:Vickers\\_anvil\\_diamonds.jpg](http://en.wikipedia.org/wiki/File:Vickers_anvil_diamonds.jpg)
- [3] Qness Gmbh, Qness Q10 [cit. 2015-04-01] Dostupný z WWW: <http://www.qness.at/en/products/q10-micro-hardness-tester/detail.html>
- [4] Donald E. Knuth: *The Art Of Computer Programming*, Boston: Addison-Wesley,
- [5] Opening. In More Morphology Transformations [online], strana naposledy edit. 2014-09-21 [cit. 2015-02-28]. Anglická verze. Dostupný z WWW: [http://docs.opencv.org/doc/tutorials/imgproc/opening\\_closing\\_hats/opening\\_closing\\_hats.html](http://docs.opencv.org/doc/tutorials/imgproc/opening_closing_hats/opening_closing_hats.html)
- [6] More Morphology Transformations [online], strana naposledy edit. 2015-03-25 [cit. 2015-04-01]. Anglická verze. Dostupný z WWW: [http://docs.opencv.org/doc/tutorials/imgproc/opening\\_closing\\_hats/opening\\_closing\\_hats.html#theory](http://docs.opencv.org/doc/tutorials/imgproc/opening_closing_hats/opening_closing_hats.html#theory)
- [7] Closing. In More Morphology Transformations [online], strana naposledy edit. 2014-09-21 [cit. 2015-02-28]. Anglická verze. Dostupný z WWW: [http://docs.opencv.org/doc/tutorials/imgproc/opening\\_closing\\_hats/opening\\_closing\\_hats.html](http://docs.opencv.org/doc/tutorials/imgproc/opening_closing_hats/opening_closing_hats.html)
- [8] Eroding and Dilating. [online] strana naposledy edit. 2015-03-25 [cit. 2015-04-01] Anglická verze. Dostupný z WWW: [http://docs.opencv.org/doc/tutorials/imgproc/erosion\\_dilatation/erosion\\_dilatation.html](http://docs.opencv.org/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html)

## LITERATURA

---

- [9] erode. Image filtering [online], strana naposledy edit. 2014-09-21 [cit. 2015-02-28]. Anglická verze. Dostupný z WWW: <http://docs.opencv.org/modules/imgproc/doc/filtering.html?highlight=erode#erode>
- [10] dilate. Image filtering [online], strana naposledy edit. 2014-09-21 [cit. 2015-02-28]. Anglická verze. Dostupný z WWW: <http://docs.opencv.org/modules/imgproc/doc/filtering.html?highlight=dilate#dilate>
- [11] J.M.S. Prewitt „Object Enhancement and Extraction“ in „Picture processing and Psychopictorics“, Academic Press,1970
- [12] Hough Line transform [online], strana naposledy edit. 2014-09-21 [cit. 2015-02-28]. Anglická verze. Dostupný z WWW: [http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough\\_lines/hough\\_lines.html](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html)



---

## Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	bin .....	adresář se spustitelnou formou implementace
	src .....	adresář s zdrojovými daty
	impl .....	adresář se zdrojovými kódy implementace
	thesis .....	zdrojová forma práce ve formátu $\text{\LaTeX}$
	testdata.....	adresář se testovacími daty
	text .....	text práce
	thesis.pdf .....	text práce ve formátu PDF
	thesis.ps .....	text práce ve formátu PS
	desky.pdf .....	Obal na desky ve formátu PDF